



# BEA WebLogic Collaborate

A Component of BEA WebLogic Integration

## Programming BEA WebLogic Collaborate Management Applications

BEA WebLogic Collaborate Release 2.0  
Document Edition 2.0  
July 2001

## Copyright

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, Operating System for the Internet, Liquid Data, 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, BEA WebLogic Server, BEA WebLogic Integration, E-Business Control Center, BEA Campaign Manager for WebLogic, and Portal FrameWork are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

### **Programming BEA WebLogic Collaborate Management Applications**

<b>Document Edition</b>	<b>Date</b>	<b>Software Version</b>
2.0	July 2001	2.0

---

# Contents

## About This Document

What You Need to Know ..... vi  
e-docs Web Site ..... vi  
How to Print this Document ..... vii  
Related Information ..... vii  
Contact Us! ..... vii  
Documentation Conventions ..... viii

## 1. Introduction

WebLogic Collaborate Applications ..... 1-1  
Management Applications ..... 1-2

## 2. Developing Management Applications

Standard and Custom Management Tools ..... 2-1  
About Management Applications ..... 2-2  
    MBeans and the MBean Server ..... 2-2  
        MBean Packages ..... 2-3  
        MBean Server Implementation ..... 2-3  
    MBeans ..... 2-3  
Programming Steps for Management Applications ..... 2-5  
    Step 1: Import the Necessary Packages ..... 2-6  
    Step 2: Get a Reference to the MBeanServer Object ..... 2-7  
    Step 3: Construct an ObjectName Object ..... 2-8  
    Step 4: Query the MBean Server ..... 2-8  
    Step 5: Read the Attributes of the MBean ..... 2-9  
    Step 6: Navigate Across MBeans ..... 2-10  
    Step 7: Handle Exceptions ..... 2-11

---

## Index

---

# About This Document

This document describes how to develop applications to exchange business messages and monitor run-time activities through the BEA WebLogic Collaborate™ system.

This document is organized as follows:

- Chapter 1, “Introduction,” provides an overview to developing applications for the BEA WebLogic Collaborate environment.
- Chapter 2, “Developing Management Applications,” describes how to create applications that monitor run-time activities in WebLogic Collaborate using the BEA WebLogic Collaborate Managed Beans (MBeans).

---

# What You Need to Know

This document is intended primarily for:

- Business process designers who use the WebLogic Process Integrator Studio to design workflows that can be used with a BEA WebLogic Collaborate environment.
- Application developers who write Java applications that manage the exchange of business messages or monitor run-time statistics in a BEA WebLogic Collaborate environment.
- System administrators who set up and administer BEA WebLogic Collaborate applications.

For an overview of the BEA WebLogic Collaborate architecture, see “[Overview](#)” in *Introducing BEA WebLogic Collaborate*.

## e-docs Web Site

BEA product documentation is available at the following location:

<http://e-docs.bea.com>

# How to Print this Document

You are reading the PDF version of this document, either online or a printout. You can print the entire document or any portion of the document from Adobe Acrobat Reader. If you do not have the Adobe Acrobat Reader, you can get it for free from the Adobe Web site at the following location:

<http://www.adobe.com>

Alternatively, you can print a copy of the HTML version of this document from a Web browser, one file at a time, by using the File—>Print option on your Web browser.

## Related Information

For more information about Java 2 Enterprise Edition (J2EE), Extended Markup Language (XML), and Java programming, see the Javasoft Web site at the following URL:

<http://java.sun.com>

## Contact Us!

Your feedback about the WebLogic Collaborate documentation is important to us. Send us e-mail at [docsupport@bea.com](mailto:docsupport@bea.com) if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the WebLogic Collaborate documentation.

In your e-mail message, please indicate that you are using the documentation for Release 2.0 of WebLogic Collaborate.

---

If you have any questions about this version of WebLogic Collaborate, or if you have problems installing and running WebLogic Collaborate, contact BEA Customer Support through BEA WebSupport at the following location:

<http://www.bea.com>

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

## Documentation Conventions

The following documentation conventions are used throughout this document.

<b>Convention</b>	<b>Item</b>
<b>boldface text</b>	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
<i>italics</i>	Indicates emphasis or book titles.



Convention	Item
monospace text	<p>Indicates code samples, commands and their options, data structures and their members, data types, directories, and filenames and their extensions. Monospace text also indicates text that you must enter from the keyboard.</p> <p><i>Examples:</i></p> <pre>#include &lt;iostream.h&gt; void main ( ) the pointer psz chmod u+w * \tux\data\ap .doc tux.doc BITMAP float</pre>
<b>monospace boldface text</b>	<p>Identifies significant words in code.</p> <p><i>Example:</i></p> <pre>void <b>commit</b> ( )</pre>
<i>monospace italic text</i>	<p>Identifies variables in code.</p> <p><i>Example:</i></p> <pre>String <i>expr</i></pre>
UPPERCASE TEXT	<p>Indicates device names, environment variables, and logical operators.</p> <p><i>Examples:</i></p> <pre>LPT1 SIGNON OR</pre>
{ }	Indicates a set of choices in a syntax line. The braces themselves should never be typed.
[ ]	<p>Indicates optional items in a syntax line. The brackets themselves should never be typed.</p> <p><i>Example:</i></p> <pre>buildobjclient [-v] [-o name] [-f <i>file-list</i>]... [-l <i>file-list</i>]...</pre>
	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.

---

Convention	Item
...	<p>Indicates one of the following in a command line:</p> <ul style="list-style-type: none"> <li>■ That an argument can be repeated several times in a command line</li> <li>■ That the statement omits additional optional arguments</li> <li>■ That you can enter additional parameters, values, or other information</li> </ul> <p>The ellipsis itself should never be typed.</p> <p><i>Example:</i></p> <pre>buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...</pre>
.	<p>Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.</p>

---

# 1 Introduction

This section describes the following types of applications:

- WebLogic Collaborate Applications
- Management Applications

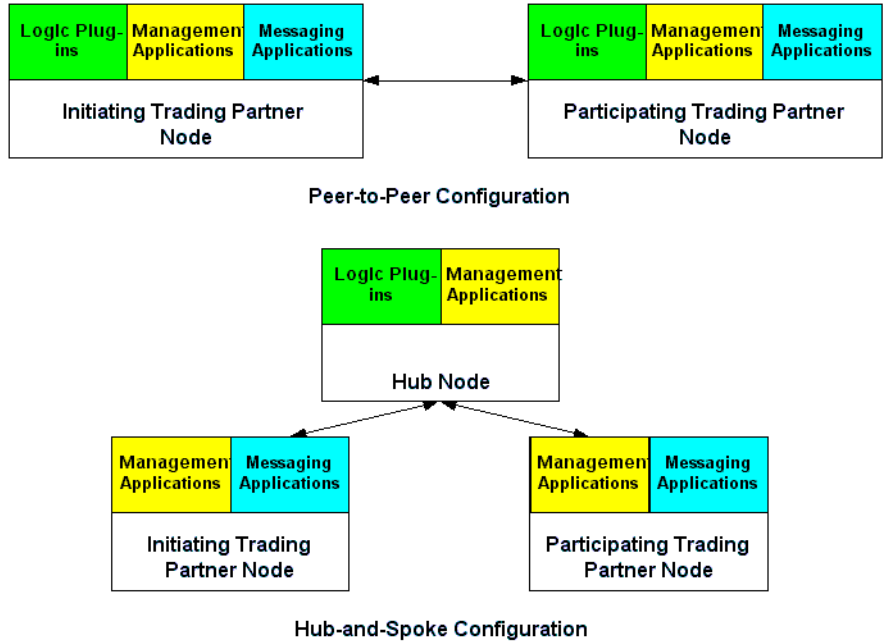
## WebLogic Collaborate Applications

This document introduces WebLogic Collaborate MBeans and management applications. MBeans are one of three types of component applications available within WebLogic Collaborate. Management applications use MBeans to monitor WebLogic Collaborate. In addition to MBeans, WebLogic Collaborate allows you to use the following kinds of applications:

- Logic plug-ins, for customized routing, filtering, and information processing, as described in *Programming BEA WebLogic Collaborate Logic Plug-Ins*
- Messaging applications, as described in *Programming BEA WebLogic Collaborate Messaging Applications*

The following figure shows where these types of applications reside in the WebLogic Collaborate system.

**Figure 1-1 Types of WebLogic Collaborate Applications**



For an introduction to the WebLogic Collaborate system, see [Introducing BEA WebLogic Collaborate](#).

## Management Applications

WebLogic Collaborate management applications monitor run-time activities, such as message traffic and conversation statistics, on WebLogic Collaborate. WebLogic Collaborate provides a unified administrative tool, the WebLogic Collaborate Administration Console, that monitors the server at run time. In addition to the system tools provided by BEA, developers can create custom management applications that provide comparable monitoring functionality.

Developers can implement a variety of management applications. These applications can:

- Monitor activities on a WebLogic Server node
- Provide run-time statistics for server nodes, delivery channels, business transaction definitions, trading partners, and business messages.

For all management applications, WebLogic Collaborate provides a set of Managed Beans, or *MBeans*, which are special JavaBeans with attributes and methods for management operations. These MBeans are BEA implementations of the Java Management Extensions (JMX) Managed Beans API, which is defined in the Java Management Extensions Specification published by Sun Microsystems, Inc.

For more information about WebLogic Collaborate management applications, see Chapter 2, “Developing Management Applications.”

# 1 *Introduction*

---

# 2 Developing Management Applications

The following sections describe how to create WebLogic Collaborate management applications that monitor run-time activity on WebLogic Collaborate nodes:

- Standard and Custom Management Tools
- About Management Applications
- Programming Steps for Management Applications

## Standard and Custom Management Tools

The WebLogic Collaborate Administration Console provides run-time monitoring of WebLogic Collaborate activities. Developers who want additional tools can create custom management applications that provide the same monitoring information displayed by the WebLogic Collaborate Administration Console.

Custom management applications may be desirable, for example, if you want read-only access to real-time statistics, such as the number of messages exchanged in a particular conversation or the number of messages received by WebLogic Collaborate. In addition, custom applications can perform certain administrative tasks programmatically, such as shutting down a particular delivery channel or WebLogic Server, or leaving or terminating a particular conversation.

**Note:** To configure the repository, use one of the following tools. Note that custom management applications cannot configure the repository.

- WebLogic Collaborate Administration Console, as described in the [BEA WebLogic Collaborate Administration Console Online Help](#)
- Bulk Loader, as described in “[Working with the Bulk Loader](#)” in *Administering BEA WebLogic Collaborate*

# About Management Applications

The following sections describe management applications in WebLogic Collaborate:

- MBeans and the MBean Server
- MBeans

## MBeans and the MBean Server

WebLogic Collaborate provides the application programming interfaces (APIs) needed to create custom management applications that monitor run-time activity on WebLogic Collaborate nodes. The WebLogic Collaborate Administration Console tools also use these APIs to provide real-time monitoring information.

These APIs consist of sets of Java Management Extensions (JMX) Managed Beans, or *MBeans*, which are special JavaBeans with attributes and methods for management operations. For more information about JMX, particularly the use of the JMX API (including the MBeanServer and MBeans), see the Java Management Extensions Specification published by Sun Microsystems, Inc., at the following URL:

<http://www.java.sun.com/products/JavaManagement/index.html>



## MBean Packages

WebLogic Collaborate provides the `com.bea.b2b.management` package for creating custom management applications. This package provides WebLogic Collaborate MBeans used for creating management applications that monitor run-time activity on server nodes. It also provides the `ManagementException` class for handling errors that occur in a run-time management application. For detailed information about this package, see the [BEA WebLogic Collaborate Javadoc](#).

In this release, all MBeans are implemented as Standard MBeans, which make up a class that implements its own MBean interface. WebLogic Collaborate MBeans are not implemented as remote MBeans. This means that all management applications must reside on the server being monitored.

## MBean Server Implementation

WebLogic Collaborate registers the MBean with the MBeanServer instance that is created when a WebLogic Server instance starts. For more information on the WebLogic Server MBeanServer, see the [BEA WebLogic Server Administration Guide](#) at the following URL:

<http://e-docs.bea.com/wls/docs61/adminguide/index.html>

## MBeans

The `com.bea.b2b.management` package contains all of the WebLogic Collaborate MBeans, which are described in the following table.

**Table 2-1 WebLogic Collaborate MBeans**

Label	Description
WLCMBean	Represents a WebLogic Collaborate instance. Used for monitoring a WebLogic Collaborate instance at run time.
DeliveryChannelMBean	Represents a delivery channel. Used for monitoring delivery channels on WebLogic Collaborate at run time.

**Table 2-1 WebLogic Collaborate MBeans (Continued)**

<b>Label</b>	<b>Description</b>
ConversationMBean	Represents a business conversation managed by the Transaction Manager on the WebLogic Collaborate instance. Used for monitoring active transactions within a delivery channel.
TradingPartnerSessionMBean	Represents a session with a trading partner. Used for monitoring trading partners.
MessageMBean	Represents a message in a conversation. Used for monitoring messages.
CollaborationAgreementMBean	Represents a collaboration agreement. A collaboration agreement represents a technical agreement between two parties on how they plan to communicate with each other using a specific protocol.

**Note:** For WebLogic Collaborate Release 2.0, all MBeans are centralized in this package. In previous releases, these MBeans were split between the c-hub and c-enabler packages. If you are upgrading from WebLogic Collaborate 1.0 or 1.0.1, you must modify any management applications you have written to make use of the new MBeans. The following table outlines changes between Release 1.0 MBeans and Release 2.0 MBeans.

**Table 2-2 WebLogic Collaborate 1.0 and 2.0 MBean Labels**

<b>WLC 1.0 Label</b>	<b>WLC 2.0 Label</b>	<b>Server Location</b>
EnablerMBean	WLCMBean	Enabler
HubMBean	WLCMBean	Hub
CSPACEMBean	DeliveryChannelMBean	Hub
ConversationMBean	ConversationMBean	Enabler
GlobalConversationMBean	ConversationMBean	Hub
EnablerSessionMBean	TradingPartnerSessionMBean	Enabler
CollaboratorMBean	TradingPartnerSessionMBean	Hub
MessageMBean	MessageMBean	Hub and Enabler

# Programming Steps for Management Applications

To access WebLogic Collaborate MBeans using the JMX API, a Java application must perform the following steps:

- Step 1: Import the Necessary Packages
- Step 2: Get a Reference to the MBeanServer Object
- Step 3: Construct an ObjectName Object
- Step 4: Query the MBean Server
- Step 5: Read the Attributes of the MBean
- Step 6: Navigate Across MBeans
- Step 7: Handle Exceptions

The WebLogic Collaborate Administration Console uses the JMX API and WebLogic Collaborate MBeans to monitor running WebLogic Collaborate instances.

# Step 1: Import the Necessary Packages

To work with MBeans, a management application must import the necessary packages. At a minimum, the application must import the packages described in the following table.

**Table 2-3 Packages that Must Be Imported**

<b>Label</b>	<b>Description</b>
<code>javax.management.*;</code>	Required for JMX MBeans, as mandated in the Java Management Extensions Specification published by Sun Microsystems, Inc.
<code>javax.naming.*;</code>	Required for retrieving the MBean server object using JNDI lookup. Only the following are required: <ul style="list-style-type: none"><li>■ <code>javax.naming.Context</code></li><li>■ <code>javax.naming.InitialContext</code></li></ul>
<code>com.bea.b2b.management.*</code>	Required for all management applications.
<code>weblogic.management.*</code>	Required for all management implementations. Gets MBeanServer and MBeanHome.

The code in the following listing imports the necessary packages for a management application.

**Listing 2-1 Importing Packages for a Management Application**

```
import javax.management.*;
import javax.naming.Context;
import javax.naming.InitialContext;
import com.bea.b2b.management.*;
import weblogic.management.*;
```

---

## Step 2: Get a Reference to the MBeanServer Object

WebLogic Collaborate uses the MBeanServer that is instantiated when an instance of WebLogic Server is started. To get a reference to the MBeanServer, the MBeanHome of that server is required. The MBeanHome of the MBeanServer is available from the server's JNDI tree at:

```
weblogic.management.MBeanHome.JNDI_NAME.serverName
```

An administration server publishes an MBeanHome for each server in the domain on its JNDI tree. The administration MBeanHome is available only from the JNDI tree of the administration server at:

```
weblogic.management.MBeanHome.ADMIN_JNDI_NAME
```

The underlying MBeanServer for any MBeanHome can be obtained by invoking the `getMBeanServer()` method on that MBeanHome.

The following code shows an example of a JNDI lookup for the administration server MBeanHome.

### Listing 2-2 Getting a Reference to the MBeanServer Object

---

```
import javax.naming.Context;
import javax.naming.NamingException;
import javax.naming.AuthenticationException;
import javax.naming.CommunicationException;
import weblogic.jndi.Environment;
import weblogic.management.MBeanHome;

...

MBeanHome home = null;
try {
    Environment env = new Environment();
    ctx = env.getInitialContext();
    home = (MBeanHome) ctx.lookup(MBeanHome.ADMIN_JNDI_NAME);
    RemoteMBeanServer server = home.getMBeanServer();
}
catch (AuthenticationException e) {
    ... //Error handling
} catch (CommunicationException e) {
    ... //Error handling
} catch (NamingException e) {
```

```
        ... //Error handling  
    }
```

---

### Step 3: Construct an ObjectName Object

MBeans are uniquely identified by object names inside the MBeanServer. The `ObjectName` class represents an object name.

For WebLogic Collaborate Release 2.0, only `WLCMBean` is registered with the `MBeanServer`; all other MBeans can be retrieved from `WLCMBean`. `WLCMBean` has a name, a type, and a domain. These attributes are reflected in the MBean's JMX Object Name. The Object Name is the unique identifier for a given MBean across all domains, and has the following structure:

```
domain name:Name=name,Type=type[,attr=value]...
```

The value of `name` is unique for a given domain and a given type. For example:

```
mydomain:Name=WLC,Type=WLC  
ObjectName objectName = new ObjectName("WLC", "WLC", "mydomain");
```

For MBeans, object names can also be used for query operations in which object name expressions are used. The `MBeanServer` uses pattern matching on the object names of the registered MBeans. The matching syntax is consistent with file globbing, which is described in the Java Management Extensions Specification published by Sun Microsystems, Inc.:

- An asterisk (\*) matches any character sequence.
- A question mark (?) matches a single character.

### Step 4: Query the MBean Server

After constructing an object name expression, an application queries the `MBeanServer` by passing the `ObjectName` object corresponding to the expression. To retrieve the set of registered MBeans, the names of which satisfy an object name expression, use the following method:

```
javax.management.MBeanServer.queryNames()
```

The MBeanServer returns a set of objects that satisfy the query criteria. Note that these are `ObjectName` objects that *represent* MBeans; they are *not* direct references to the MBeans themselves.

## Step 5: Read the Attributes of the MBean

Use the `ObjectName` instance, obtained in the previous step, to access other MBeans, provided that the `ObjectName` has one or more attributes of the MBean type. To read the attributes of an MBean, use the following method, passing the `ObjectName` object as a parameter:

```
javax.management.MBeanServer.getAttribute()
```

Once you call the `getAttribute` method by passing the `ObjectName` object for the first MBean, you can get direct references to other MBean instances.

The code in the following listing retrieves a set of attributes associated with `WLCMBean`.

### Listing 2-3 Retrieving Conversation Attributes

---

```
MBeanHome home = Admin.getMBeanHome();
server = home.getMBeanServer();
ObjectName objectName = new ObjectName("WLC", "WLC", "mydomain");
beans = server.queryNames(objectName, null);
Iterator it = beans.iterator();
while (it != null && it.hasNext())
{
    //Should be only one
    obj = (ObjectName)it.next();
    break;
}
if (obj != null)
{
    Date startTime = (Date)server.getAttribute(obj, "ActiveSince");
    Date lastTime = (Date)server.getAttribute(obj, "LastMessageSentTime");
    ConversationMBean[] convs = (Conversation[]) server.getAttribute(obj,
        "ActiveConversations");
    if (convs != null)
    {
        for (int ii=0; ii< convs.length; ii++)
        {
```

```
String protocol = convs[i].getBusinessProtocolName();
    }
}
...

```

---

All the attributes shown in the preceding code listing can be retrieved by calling `getAttribute`. To invoke a method such as `shutdown` on `WLCMBean`, call the `MBeanServer`. For more information see the JMX specification at the following URL:

<http://java.sun.com>

## Step 6: Navigate Across MBeans

MBeans that are logically related have accessor methods to retrieve references to each other. These methods are strongly typed and return exact MBean types. For example, the `WLCMBean.getActiveDeliveryChannels()` method returns an array of type `DeliveryChannelMBean` that represents all the active delivery channels in the system. Similarly, the `TradingPartnerSessionMBean.getActiveConversations()` method returns an array of type `ConversationMBean` that represents all the active conversations in this session.

For detailed information about these methods, see the *BEA WebLogic Collaborate Javadoc*.



## Step 7: Handle Exceptions

If an error occurs while a WebLogic Collaborate management application is being run, a `com.bea.b2b.management.ManagementException` is thrown. Management applications can catch this exception and process it appropriately, as shown in the following listing.

### **Listing 2-4 Handling ManagementExceptions in Management Applications**

---

```
catch (ManagementException me){
    String msg = "Exception in Management Application: " + me;
    debug(msg);
    throw new Exception(msg);
}
```

---



---

# Index

## A

- applications, introduction 1-1
- attributes
  - examples of retrieving 2-9
  - reading 2-9

## C

- CollaborationAgreementMBean 2-4
- CollaboratorMBean 2-4
- configuring the repository 2-2
- constructing objects 2-8
- contact information vii
- ConversationMBean 2-4
- CSpaceMBean 2-4
- customer support vii

## D

- definition of MBeans 1-3, 2-2
- DeliveryChannelMBean 2-3, 2-4
- documents
  - conventions viii
  - printing vii
  - where to find vi

## E

- EnablerMBean 2-4
- EnablerSessionMBean 2-4

- examples

- JNDI lookup 2-7
  - obtaining a reference 2-7
  - retrieving attributes 2-9
- exceptions, handling 2-11
- expressions for object names 2-8

## G

- GlobalConversationMBean 2-4

## H

- handling exceptions 2-11
- HubMBean 2-4

## I

- implementation of MBean server 2-3
- importing packages 2-6

## J

- Java Management Extensions 2-2
- JMX 2-2
- JNDI lookup, example 2-7

---

## M

- Managed Beans 2-2
- management applications
  - introduction 1-2
  - overview 2-2
- management tools 2-1

### MBean server

- implementation 2-3
- obtaining a reference 2-7
- overview 2-2
- querying 2-8

### MBeans

- CollaborationAgreementMBean 2-4
- CollaboratorMBean 2-4
- ConversationMBean 2-4
- CSPACEMBean 2-4
  - definition 1-3, 2-2
- DeliveryChannelMBean 2-3, 2-4
- EnablerMBean 2-4
- EnablerSessionMBean 2-4
- GlobalConversationMBean 2-4
- HubMBean 2-4
  - navigating across 2-10
  - overview 2-3
  - packages 2-3
  - reading attributes 2-9
- TradingPartnerSessionMBean 2-4
- WLCMBean 2-3, 2-4

- MessageMBean 2-4

- monitoring WebLogic Collaborate 2-1

## N

- navigating across MBeans 2-10

## O

- object name expressions 2-8
- objects, constructing 2-8
- obtaining references to MBean server 2-7

## P

- packages
  - importing 2-6
  - MBeans 2-3
- printing documents vii
- programming steps 2-5

## Q

- querying MBean server 2-8

## R

- reading attributes of MBeans 2-9
- referencing
  - MBean server 2-7
  - object 2-7
- related information vii
- repository, configuring 2-2
- retrieving, example 2-9

## S

- support, customer vii
- support, technical viii

## T

- technical support viii
- TradingPartnerSessionMBean 2-4

## W

- WebLogic Collaborate, monitoring 2-1
- WLCMBean 2-3, 2-4