



BEA WebLogic Java Adapter for Mainframe™

Introduction

Copyright

Copyright © 2002 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, Jolt, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Manager, BEA WebLogic Commerce Server, BEA WebLogic E-Business Platform, BEA WebLogic Enterprise, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Personalization Server, BEA WebLogic Portal, BEA WebLogic Process Integrator, BEA WebLogic Server and How Business Becomes E-Business are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

BEA WebLogic Java Adapter for Mainframe Introduction

Part Number	Date	Software Version
N/A	January 2002	5.0

Contents

1. BEA WebLogic Java Adapter for Mainframe Introduction

Business Benefits.....	1-2
WebLogic Java Adapter for Mainframe Components.....	1-4
The Communication Resource Manager.....	1-5
The Gateway	1-6
WebLogic Java Adapter for Mainframe Features	1-6
WebLogic Server Support.....	1-7
Communication Protocol Support.....	1-7
Distributed Transaction Support	1-8
Scalability.....	1-8
Administration Console.....	1-11
eGen Application Generator.....	1-13
Samples	1-14
CrossPlex Support.....	1-14

2. Using WebLogic JAM

Getting Started.....	2-1
----------------------	-----

Index



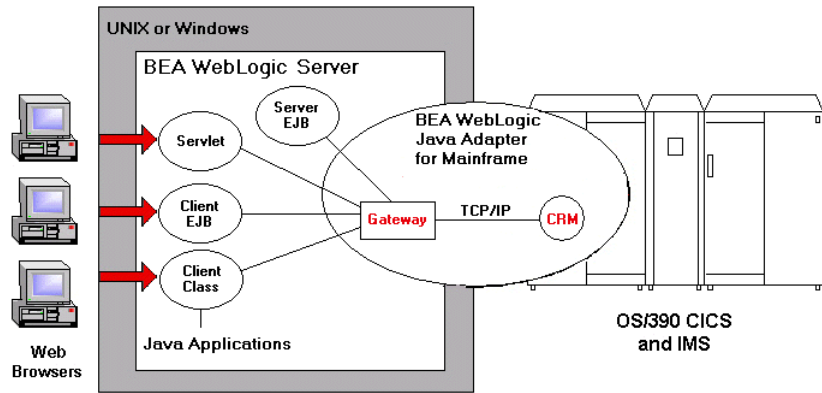
1 BEA WebLogic Java Adapter for Mainframe Introduction

To succeed in the new economy, your e-business depends on application integration. Specifically, integration of back-end systems with Web-based e-business applications. As the complexity of these Web-based applications evolve, the business transactions supported by these applications often require direct integration with existing mainframe legacy applications. In order to meet the rigorous demands for fast and flawless request execution, e-business transactions must integrate with mainframe applications in real time.

BEA WebLogic Java Adapter for Mainframe (hereafter referred to as WebLogic JAM) fulfills this requirement by providing bi-directional, request-response integration between Java applications and mainframe applications.

With WebLogic JAM, you can extend the life of existing mainframe applications by integrating them with J2EE applications running on WebLogic Server. Once configured and deployed, WebLogic JAM allows WebLogic Server applications to invoke or be invoked by mainframe applications. The following figure provides an overview of WebLogic JAM components and connectivity.

Figure 1-1 Typical WebLogic JAM Distributed Configuration



BEA WebLogic JAM enables applications running on BEA WebLogic Server and mainframe systems to interact using two main components: the Gateway and the Communications Resource Manager (CRM). The Gateway is a Java application that runs in an instance of WebLogic Server. The CRM typically runs on the mainframe. The Gateway and the CRM communicate using TCP/IP. These two WebLogic JAM components work together to allow applications to communicate even though the applications are running in disparate environments.

Business Benefits

BEA WebLogic JAM delivers the following business benefits to help your e-business grow and succeed:

- Leverage existing IT investments with fast, easy integration of mainframe application functionality into Java/Web-based environments. BEA WebLogic JAM enables mainframe applications to be invoked from the WebLogic environment without coding changes. This frees your developers to focus on building new applications that focus on your competitive edge, rather than creating and maintaining a custom mainframe integration solution.
- An easy-to-use Web-based Administration Console to configure and administer WebLogic JAM. The console is an extension of the WebLogic Server console

and provides a localized administration point for the configuration of the WebLogic JAM product.

- Support for distributed transactions using the two-phase commit protocol. This protocol guarantees data integrity by ensuring that transactional updates are committed by all participating transaction managers, or are fully rolled back by all transaction managers. This support enables you to coordinate mainframe resources and resources in your WebLogic Server environment atomically.
- Generate J2EE applications from your mainframe application source code. The eGen Application Generator development tool generates Java applications from COBOL data structure definitions. The eGen utility can generate EJBs, servlets, or stand-alone clients from data structures defined in a COBOL copybook. These generated applications contain the data translation code needed to translate data between Java and COBOL data types. This enables your development team to avoid having to modify your existing mainframe applications, and focus on developing new applications.
- Data translation between Java, XML, and COBOL mainframe data types (packed decimal, binary, arrays, redefines and occurs) and full EBCDIC conversion is supported.
- Support for using mainframe security credentials. WebLogic JAM can be configured so that mainframe security credentials (user id/password) have to be supplied by client applications. These security credentials are then sent to the mainframe system for authentication, prior to invoking the specified mainframe application.
- Clustering with BEA WebLogic Server provides the highest levels of scalability and availability. Multiple instances of WebLogic JAM can be configured in a cluster, providing load balancing and failover of the integration services provided by WebLogic JAM.
- BEA WebLogic JAM seamlessly plugs into BEA WebLogic Integration, a powerful process integration and workflow engine, to provide an advanced e-business infrastructure that incorporates process rules, workflow, and application components. Within the WebLogic Integration framework, this software combination enables you to create business processes, process rules, and workflows that access mainframe applications.
- Interoperability with the proven BEA WebLogic E-Business Platform provides a reliable infrastructure for B2C and B2B e-business applications. BEA also provides the convenience and support of a global, 24x7 support organization.

The following sections provide an overview of the WebLogic Java Adapter for Mainframe product:

- WebLogic Java Adapter for Mainframe Components
- WebLogic Java Adapter for Mainframe Features

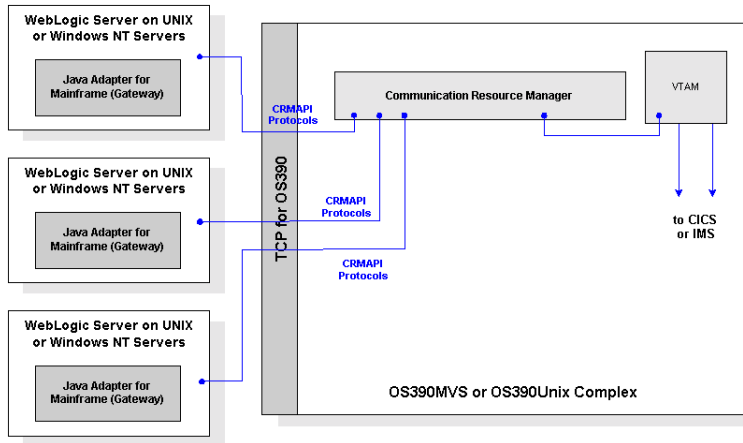
WebLogic Java Adapter for Mainframe Components

WebLogic JAM enables applications running on WebLogic Server and mainframe systems to interact using two main components:

- The Communication Resource Manager
- The Gateway

These two WebLogic JAM Components work together to enable applications to communicate even though the applications are running in disparate environments. The following figure provides an overview of WebLogic JAM components and connectivity.

Figure 1-2 WebLogic JAM Environment



The Communication Resource Manager

The Communications Resource Manager (CRM) is the WebLogic JAM component that manages communications resources. The CRM coordinates the flow of data between Java applications running on a WebLogic Server platform and applications running on a mainframe.

The CRM uses SNA and TCP/IP communication protocols to communicate with the mainframe and WebLogic JAM Gateway respectively. TCP/IP protocols always flow between the WebLogic JAM Gateway and the CRM. SNA protocols always flow from the CRM to the mainframe and from VTAM on the mainframe to the CRM depending upon the configuration option.

For a complete list of supported operating systems, refer to the *BEA WebLogic Java Adapter for Mainframe Release Notes*.

The Gateway

The WebLogic JAM Gateway component is a server that runs within WebLogic Server and communicates with the CRM using the TCP/IP protocol. The WebLogic JAM Gateway acts as a gateway to route requests and responses between WebLogic Server (Java) and mainframe systems, such as CICS and IMS. The Gateway component also forwards configuration information to the CRM at start-up.

A WebLogic JAM Gateway used in conjunction with the CRM enables applications running on an instance of WebLogic Server to connect to back-end applications. Each WebLogic Server instance that needs to access back-end applications should have a WebLogic JAM Gateway defined for it. This WebLogic JAM Gateway routes requests received from Java client applications running on this instance of WebLogic Server to the CRM, which then will forward them to the back-end application. The Gateway also routes requests made by mainframe client applications that have come through the CRM to the necessary WebLogic Server applications.

WebLogic Java Adapter for Mainframe Features

The following sections describe features in WebLogic JAM:

- WebLogic Server Support
- Communication Protocol Support
- Distributed Transaction Support
- Scalability
- WebLogic Administration Console
- eGen Application Generator
- Samples
- CrossPlex Support

WebLogic Server Support

BEA WebLogic JAM is used in conjunction with WebLogic Server to integrate existing mainframe applications with applications developed for WebLogic Server.

BEA WebLogic Server provides a tightly integrated, comprehensive infrastructure that delivers commerce, personalization, campaign management, enterprise integration, workflow management, and business-to-business collaboration.

BEA WebLogic Server supports a broad variety of clients, including Web browsers, wireless devices, and programmatic clients. On the server, BEA WebLogic Server supports the leading Unix, Linux, Windows, and mainframe operating systems. On the back-end, BEA WebLogic Server integrates with relational databases, message queues, and legacy systems.

BEA WebLogic Server implements the Java 2 Enterprise Edition (J2EE) platform specification that includes Servlets, Java Server Pages (JSPs), Enterprise JavaBeans (EJBs), Java Messaging Service (JMS), and other platform services. J2EE services provide access to standard network protocols, database, and messaging systems. When developing applications, developers can create, assemble, and deploy components that use these services.

WebLogic JAM enables Servlets, JSPs and EJBs to invoke CICS, IMS or batch applications that exist on your mainframe systems. WebLogic JAM also enables CICS, IMS, or batch applications to invoke EJBs or JMS services.

Communication Protocol Support

As previously mentioned, WebLogic JAM provides bi-directional, online request-response between Java applications and mainframe applications. WebLogic JAM facilitates Java applications making requests of mainframe services as well as mainframe clients making requests of Java services.

WebLogic JAM is especially useful as it enables Java applications to integrate with CICS programs using DPL and integrates with IMS programs using implicit APPC. Java client applications can link with CICS programs, and CICS clients can link with Java applications with the same ease as DPL between CICS programs on different

regions. Similarly, Java client applications can communicate with IMS programs with the same ease as implicit APPC between different IMS programs. WebLogic JAM supports the following communications protocols:

- CICS DPL
- CICS DTP using CICS or CPI-C programming APIs
- IMS Implicit APPC
- IMS Explicit APPC using APPC/MVS or CPI-C programming APIs
- Batch APPC using APPC/MVS or CPI-C programming APIs

Distributed Transaction Support

BEA WebLogic JAM supports distributed transactions for enterprise applications using the two-phase commit protocol. This support enables WebLogic JAM to coordinate operations against mainframe resources and resources in your WebLogic Server environment atomically.

Scalability

BEA WebLogic JAM can scale as your mainframe integration needs to grow. WebLogic JAM takes advantage of WebLogic Server clusters to provide load balancing and failover.

The following sections provide descriptions of the various WebLogic JAM scalability features.

- Gateway Cluster Support
- Multiple-Gateway Support
- Multiple Region Support

Gateway Cluster Support

BEA WebLogic JAM enables clients within WebLogic Server to request mainframe services. Clients can run on any machine with network access to the WebLogic Server cluster hosting one or more WebLogic JAM Gateways. When clients request a mainframe service, this request will be transparently processed by one of the WebLogic JAM Gateways in the WebLogic Server cluster. This redundancy provides failover for the service request if one of the WebLogic JAM Gateways becomes unavailable. Thus, the WebLogic JAM Gateway cluster support enables reliable access to mainframe systems.

In terms of making your systems more scalable, WebLogic JAM takes advantage of the sophisticated load balancing features of WebLogic clusters for WebLogic Server to mainframe requests so that all load-balancing in WebLogic JAM occurs among a group of servers arranged in a WebLogic cluster. When requests are made to remote WebLogic JAM services, load balancing occurs among all WebLogic JAM Gateways in the cluster that offer the service. Mainframe to WebLogic Server request load balancing is accomplished by the WebLogic JAM CRM. The CRM load balances between all connected gateways without regard to their cluster arrangement.

Multiple-Gateway Support

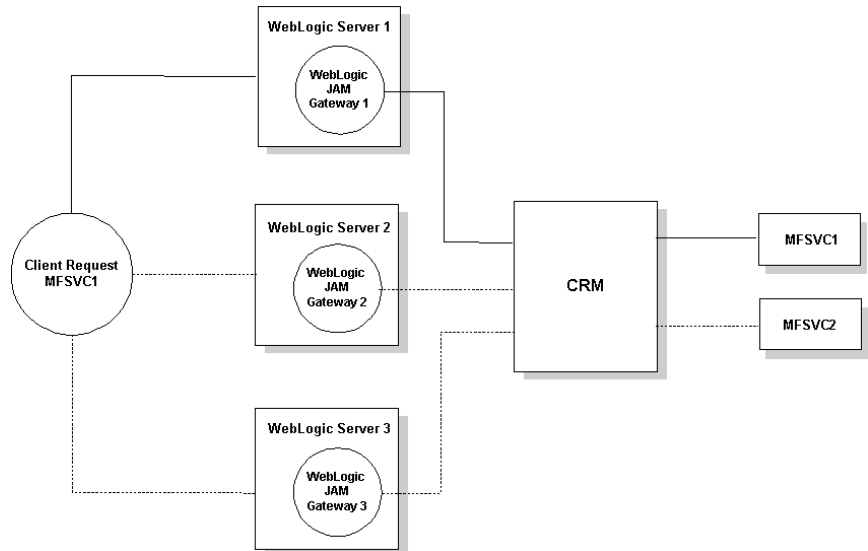
BEA WebLogic JAM multiple gateway support provides a feature that enables a single instance of the CRM to service multiple instances of WebLogic JAM Gateways. As a result, multiple WebLogic Servers can share a single instance of the CRM as a common access point to mainframe systems.

The resulting configuration, administration, and operational requirements are similar to those for a single CRM, and resource requirements are substantially less than those for an equivalent multiple CRM configuration. This feature provides the following advantages:

- Overall, a single CRM consumes less system resources than multiple CRMs.
- The CRM has a single configuration which is common to all WebLogic Servers.
- Transactional requests are tightly integrated when distributed across multiple gateways to a single CRM.

- The resulting configuration is much easier to understand, configure, administer, and operate.
- The CRM can load balance mainframe requests for services provided by the connected WebLogic Servers.

Figure 1-3 Sample Multi-Gateway Non-Clustered Configuration



Multiple Region Support

Services provided to the WebLogic Server JAM application can be physically located on different CICS or IMS regions. The calling application does not have to know the physical location of the mainframe application. If a service is advertised on multiple regions, then WebLogic JAM provides load balancing and failover of the service across the multiple regions.

WebLogic Administration Console

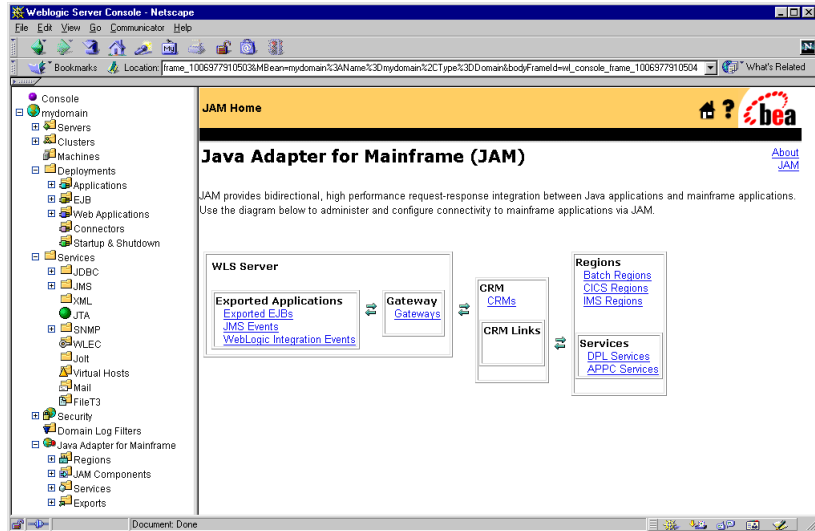
BEA WebLogic JAM provides an easy-to-use interface for configuring and administering connectivity between WebLogic Server, CICS, and IMS application servers. WebLogic JAM provides an extension of the WebLogic Server Administration Console and presents a set of objects that can be manipulated in order to create or modify a configuration. These objects consist of Regions, Gateways, CRMs, Services, and Exports.

The WebLogic JAM Administration Console makes configuration information available by utilizing the Java Management Extensions (JMX) specification. JMX is an architecture specification for the management of enterprise applications. WebLogic Server provides a JMX implementation that allows applications to register manageable objects.

A start-up class of the WebLogic JAM product reads the XML configuration file and converts the information into JMX-manageable objects. This configuration information can then be remotely administered using the Administration Console and is available throughout the WebLogic Server domain to all instances of the WebLogic JAM product.

The console displays the WebLogic JAM components it administers, including Regions, Gateways, CRMs, Services, and Exports in a graphical tree in the left pane. The right pane displays details about the object selected in the left pane.

Figure 1-4 WebLogic JAM Administration Console



To use the console to configure or administer the WebLogic JAM product, select an item in the left pane, and then choose the desired informational tab in the right pane. The console displays the configurable attributes in the right pane. You can use the online help to find detailed information about the displayed attributes and fields.

The console allows you to configure the connectivity between the WebLogic Server domain and multiple CICS or IMS regions and to define the configuration of multiple instances of WebLogic JAM, distributed across a WebLogic domain.

For administrative purposes, the console allows you to administer the WebLogic JAM components as follows:

- Monitor CRM status
- Control diagnostics tracing within the CRM
- Monitor Gateway status
- Control diagnostic tracing within the Gateway
- Start/Stop Gateway components
- Start/Stop CRM links
- Enable/Disable remote service definitions (DPL and APPC services)

- Enable/Disable local service definitions (export EJB definitions, JMS event, and WebLogic Integration event definitions)

For complete information and instructions for configuring WebLogic JAM, see the *BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide*.

Configuration of a WebLogic JAM Gateway

BEA WebLogic JAM enables you to change configuration information dynamically, that is, while WebLogic JAM is running. In many cases, you do not need to restart WebLogic JAM for your changes to take effect. When a configuration attribute is modified, the new value is immediately reflected in both the current run-time value of the attribute and the persistent value stored in the XML configuration file.

Exceptions do exist. For example, if you change the CRM's listen port, the new port is not used until the next time you restart the CRM and it must be changed in the startup script or JCL for the CRM.

The Administration Console does a validation check on each attribute that is changed. The errors that are supported are out-of-range errors and datatype mismatch errors. In both cases, a message displays telling you that an error has occurred.

WebLogic JAM supports the following dynamic configuration changes:

- Adding and removing WebLogic JAM Gateways
- Enabling and disabling WebLogic JAM Gateways
- Adding and removing Links
- Enabling and disabling Links
- Adding and removing Services
- Enabling and disabling Services

eGen Application Generator

The eGen utility is an application development tool that assists in quickly integrating a mainframe application. This command line utility generates Java source code using a COBOL copybook and a declarative script file as input. The result is a working Java skeleton application that can be used as a basis for new applications or as an addition

to an existing application. The eGen utility can generate Servlets, EJBs or stand-alone clients. All generated skeletons include code that implements connectivity and data translation to mainframe applications.

The four different types of Java application models that can be generated by the eGen utility are described as follows:

- **Client Class** - the client class is a stand-alone Java class that invokes mainframe services. This class may be incorporated into your own EJB or utilized in some other way within your code.
- **Client EJB** - the client EJB is a Stateless Session EJB that invokes mainframe services. It can be called by a servlet or other client programs. This is the normal model for building a production application with access to mainframe services. A servlet that invokes the EJB's methods may be added for testing or demonstration purposes.
- **Servlet Only** - the servlet-only application is a servlet that presents a simple form and invokes mainframe services directly. This model is useful for testing purposes, but is not suitable for production applications.
- **Server EJB** - the server EJB is a Stateless Session EJB that provides a service to mainframe client applications.

Samples

BEA WebLogic JAM includes a set of installation verification samples and a variety of other samples that you can use to demonstrate the broad range of WebLogic JAM features. The samples illustrate the interaction between Java applications and both applications running under CICS and applications running under IMS. The samples are also designed to step you through the configuration scenario that meets your specific configuration environment.

CrossPlex Support

BEA WebLogic JAM integrates CrossPlex in a seamless fashion to address customer requirements for reuse of any existing 3270 mainframe applications without the need for mainframe coding changes.

2 Using WebLogic JAM

The following table provides a clear and easy path to productivity using BEA WebLogic JAM. Tasks include installation instructions, running sample applications, configuring and administering JAM, along with topics geared toward providing you with the information you need to get up and running quickly.

Getting Started

This section lists the tasks and documentation available for getting started with WebLogic JAM.

To...	Read...
Learn about the functionality provided by WebLogic Java Adapter for Mainframe.	BEA WebLogic Java Adapter for Mainframe Introduction
Learn about new features, supported platforms, and known limitations.	BEA WebLogic Java Adapter for Mainframe Release Notes
Install WebLogic Java Adapter for Mainframe.	BEA WebLogic Java Adapter for Mainframe Installation Guide
Set up and run WebLogic Java Adapter for Mainframe installation verification samples.	BEA WebLogic Java Adapter for Mainframe Installation Guide
Set up and run WebLogic Java Adapter for Mainframe sample applications.	BEA WebLogic Java Adapter for Mainframe Samples Guide

To...	Read...
Configure BEA WebLogic JAM. <ul style="list-style-type: none">■ Define where the CRM will run.■ Create a logical unit for the CRM.■ Connect the CRM to back-end mainframe systems.■ Enter connectivity information into the WebLogic Administration Console.■ Define a WebLogic JAM Gateway.■ Verify your WebLogic JAM Connectivity Configuration.■ Modify your configuration.	BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide
Generate applications using BEA WebLogic JAM.	BEA WebLogic Java Adapter for Mainframe Programming Guide
Administer BEA WebLogic JAM. <ul style="list-style-type: none">■ Monitor the CRM status.■ Modify CRM trace level settings.■ Monitor CRM links.■ Start and Stop CRM links.■ List and Monitor Gateways.■ Modify Gateway trace level settings.■ Start and Stop Gateways.■ Enable a service.■ Enable an exported WebLogic application.	BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide
Include mainframe applications in WebLogic Integration business processes.	BEA WebLogic Java Adapter for Mainframe Workflow Processing Guide
Perform optional BEA WebLogic JAM Tasks:	
<ul style="list-style-type: none">■ Migrate from previous BEA WebLogic JAM releases.	Migrating From Previous Releases of BEA WebLogic Java Adapter for Mainframe
<ul style="list-style-type: none">■ Integrate applications with BEA WebLogic JAM	BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide

To...	Read...
■ Solve common problems when using BEA WebLogic JAM	Troubleshooting Tips for BEA WebLogic Java Adapter for Mainframe

Index

A

administration console 1-11

B

business benefits 1-2

C

client class 1-14

client EJB 1-14

clustering 1-3

COBOL copybook 1-13

communication protocol support 1-7

Communications Resource Manager 1-5

components 1-4

CrossPlex support 1-14

D

data translation 1-3

distributed transaction support 1-8

dynamic configuration changes 1-13

E

EBCDIC 1-3

eGen application generator 1-13

F

failover 1-9

features 1-6

G

Gateway 1-6

Gateway cluster support 1-9

J

J2EE applications 1-3

Java application models 1-14

Java Management Extensions 1-11

L

load balancing 1-9

M

mainframe security credentials 1-3

multiple region support 1-10

multiple-Gateway support 1-9

P

platform service support 1-7

S

samples 1-14

scalability 1-8

server EJB 1-14

servlet 1-14

SNA protocol 1-5

T

TCP/IP protocol 1-5

two-phase commit protocol 1-3

V

VTAM 1-5

W

WebLogic Integration 1-3

WebLogic Server support 1-7