



Application Server Configuration Guide

© 1997–2003 edocs® Inc. All rights reserved.

edocs, Inc., One Apple Hill Dr., Natick, MA 01760

The information contained in this document is the confidential and proprietary information of edocs, Inc. and is subject to change without notice.

This material is protected by U.S. and international copyright laws. edocs and eaPost are registered in the U.S. Patent and Trademark Office.

No part of this publication may be reproduced or transmitted in any form or by any means without the prior written permission of edocs, Inc.

eaSuite, eaDirect, eaPay, eaCare, eaAssist, eaMarket, and eaXchange are trademarks of edocs, Inc.

All other trademark, company, and product names used herein are trademarks of their respective companies.

Printed in the USA.

Table of Contents

Preface	7
About Customer Self-Service and eaSuite™	7
About This Guide.....	9
Application Server Administration Requirements.....	9
Related Documentation	10
Obtaining edocs Software and Documentation	10
If You Need Help	11
1 Upgrading from WebLogic 6.1 to 7.0	13
Upgrading Startup Scripts for WebLogic 7.0.....	14
About Example Domains in This Guide	14
About WebLogic Server Domains.....	15
WebLogic 7.0 Directory Structure	15
7.0 Components Installed with the Configuration Wizard	15
Migrating eaDirect J2EE Applications.....	16
2 Configuring WebLogic Server for UNIX	17
Overview	17
About Example UNIX Domains in This Guide	18
UNIX Permissions for WebLogic Server	18
Starting and Stopping WebLogic Server.....	19
About Sourcing Your Configuration.....	19
Providing Usernames and Passwords to Start a Server.....	19
Starting and Stopping an Active Application Server.....	19
Starting and Stopping WebLogic Server	20
Capturing Your UNIX Environment for eaDirect.....	20
Using <i>edx_config</i> to capture environment data	21
Using WebLogic 7.0 Configuration Wizard Scripts to Start an Application Server.....	22
Passing UNIX Environment Data to WebLogic	22
Using <i>edx.config</i> to source your configuration	23
Sourcing Your Configuration	23
3 Configuring WebLogic Server for Windows	27
Overview	27
About Example Windows Domains in This Guide	28
Starting and Stopping WebLogic Server.....	28
Starting and Stopping an Active Application Server.....	28
Starting and Stopping WebLogic Server	29
Capturing Your Windows Environment for eaDirect.....	30
Using <i>edx_env.bat</i> and <i>edx_load.config.bat</i> to Store Environment Data.....	30

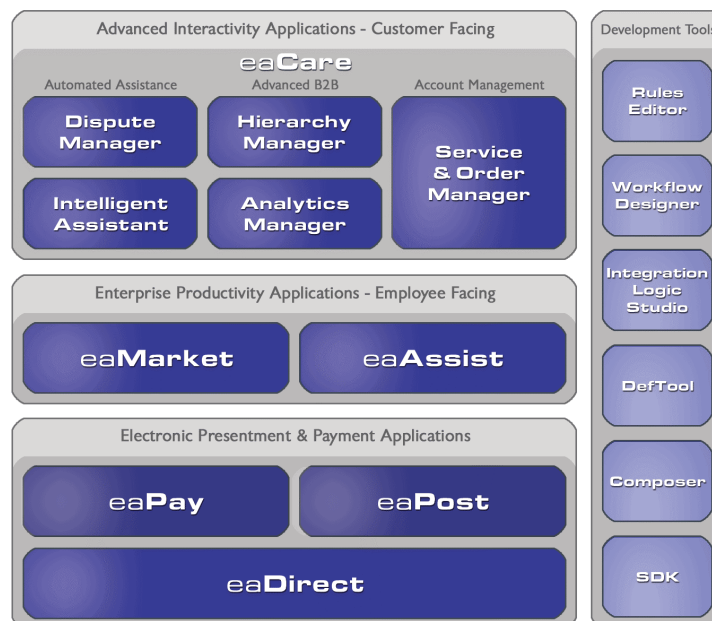
Passing Windows Environment Data to WebLogic 6.1	32
Sourcing Your Configuration as a Windows Service for WebLogic 6.1	32
Sourcing Your Configuration in a Startup Script for WebLogic 6.1	34
Passing Windows Environment Data to WebLogic 7.0	35
Passing Your Configuration to WebLogic 7.0 Running as a Windows Service	35
Passing Your Configuration in a Startup Script for WebLogic 7.0.....	38
Windows Services for eaDirect.....	39
Setting Up a WebLogic Server Instance as a Windows Service.....	39
Setting Up the eaDirect Scheduler as a Windows Service	40
Troubleshooting Tips for WebLogic on Windows	41
4 Configuring Java Resources for WebLogic	43
Overview	43
Configuring Java Database Connectivity (JDBC) for eaDirect	43
About JDBC Connections for eaDirect	44
Configuring JDBC Connections for WebLogic	44
Configuring Java Messaging Services (JMS) for eaDirect	45
About JMS Resources for eaDirect	45
Configuring JMS Resources for WebLogic	46
Starting the Scheduler	47
Where to Go From Here	49
5 Configuring WebSphere Server for Unix	51
Overview	51
UNIX Permissions for WebSphere Server	51
Starting and Stopping WebSphere	52
Starting and Stopping an Active Application Server.....	52
UNIX Environment Variables for WebSphere	53
About UNIX Config Files for Your eaDirect Environment.....	53
Setting UNIX Environment Data with <i>edx_config</i>	53
Passing UNIX Environment Data to WebSphere	54
Preparing WebSphere Server for Unix.....	56
Creating An Application Server Instance For WebSphere	56
Java Virtual Machine (JVM) Settings for WebSphere 5	56
Java Virtual Machine (JVM) Settings for WebSphere 4	61
6 Configuring Java Resources for WebSphere for UNIX.....	65
Overview	65
Configuring Java Database Connectivity (JDBC) for eaDirect	65
About JDBC Connections for eaDirect	65
Configuring the JDBC Driver for Websphere 4	66
Configuring JDBC Connections for WebSphere 4 and 5	67
Configuring Java Messaging Services (JMS) for eaDirect	67
About JMS Resources for eaDirect	68
Configuring JMS for WebSphere 5 for UNIX.....	68
Configuring JMS for WebSphere 4 for UNIX.....	69
Restarting the Application Server	72
Where to Go From Here	73
Deploying the eaDirect Application	73

Deployment Using WebSphere 5	73
Deployment Using WebSphere 4	74
Starting the Scheduler and Logger	77
Starting the eaDirect Logger.....	77
Starting the eaDirect Scheduler.....	78
7 Configuring WebSphere Server for Windows	81
Windows Environment Variables for WebSphere	81
About eaDirect Config Files for Your Windows Environment.....	81
Setting Windows Environment Data.....	81
Passing Windows Environment Data to WebSphere	83
Preparing WebSphere Server for Windows	83
Creating An Application Server Instance For WebSphere	83
Java Virtual Machine (JVM) Settings	84
8 Configuring Java Resources for WebSphere for Windows.....	87
Overview	87
Configuring Java Database Connectivity (JDBC) for eaDirect	87
About JDBC Connections for eaDirect.....	87
Configuring the JDBC Driver for Websphere 4	88
Configuring JDBC Connections for WebSphere 4 and 5	89
Configuring Java Messaging Services (JMS) for eaDirect	89
About JMS Resources for eaDirect.....	90
Configuring JMS for WebSphere 4 for Windows.....	90
Restarting the Application Server.....	93
Where to Go From Here.....	94
Deploying the eaDirect Application	94
Deployment with WebSphere 5.....	94
Deployment with WebSphere 4.....	95
Starting the Scheduler and Logger	98
Starting the eaDirect Logger.....	98
Starting the eaDirect Scheduler.....	100
Appendix A: WebLogic Reference	103
Java DataBase Connectivity (JDBC)	103
WebLogic Environment Variables.....	103
WebLogic 6.1 and 7.0.....	103
Java DataBase Connectivity (JDBC) for WebLogic.....	104
WebLogic 6.1 and 7.0 for SQL Server (Windows)	104
WebLogic 6.1 and 7.0 for Oracle (Solaris and HP-UX).....	107
Java Messaging Services for WebLogic 6.1 and 7.0.....	108
JMS Connection Factories	109
JMS (JDBC) Stores	109
JMS Servers	110
JMS Topics.....	110
JMS Session Pools and Consumers for Annotation and Dispute	110
JMS Session Pools and Consumers for Logging	111
Appendix B: WebSphere Reference.....	115

WebSphere Environment Variables	115
WebSphere 5 for Oracle (Unix).....	116
JDBC Connection Pools	116
Data Sources	116
JMS Resources (WebSphere 5, Oracle, Unix).....	117
WebSphere Topic Destinations (WebSphere 5, Oracle, Unix)	117
WebSphere 4 for Oracle (Unix).....	118
JDBC Connection Pools (WebSphere 4, Oracle, Unix)	118
JDBC Tx Data Sources (WebSphere 4, Oracle, Unix)	119
WebSphere 5 for DB2 (Unix)	119
JDBC Connection Pools	119
JMS Resources (WebSphere 5, DB2, Unix)	120
Topic Destinations (WebSphere 5, DB2, Unix)	121
WebSphere 4 for DB2 (Unix)	121
JDBC Connection Pools	121
JDBC Tx Data Sources (WebSphere 4, DB2, Unix).....	122
WebSphere 4 for SQL Server (Windows)	123
JDBC Connection Pools	123
JDBC Tx Data Source (WebSphere 4, MSSQL, Windows)	123
Index	125

About Customer Self-Service and eaSuite™

edocs has developed the industry's most comprehensive software and services for deploying Customer Self-Service solutions. **eaSuite™** combines electronic presentment and payment (EPP), order management, knowledge management, personalization and application integration technologies to create an integrated, natural starting point for all customer service issues. eaSuite's unique architecture leverages and preserves existing infrastructure and data, and offers unparalleled scalability for the most demanding applications. With deployments across the healthcare, financial services, energy, retail, and communications industries, and the public sector, eaSuite powers some of the world's largest and most demanding customer self-service applications. eaSuite is a standards-based, feature rich, and highly scalable platform, that delivers the lowest total cost of ownership of any self-service solution available.



eaSuite is comprised of four product families:

- Electronic Presentment and Payment (EPP) Applications

- Advanced Interactivity Applications
- Enterprise Productivity Applications
- Development Tools

Electronic Presentment and Payment (EPP) Applications are the foundation of edocs' Customer Self-Service solution. They provide the core integration infrastructure between organizations' backend transactional systems and end users, as well as rich e-billing, e-invoicing and e-statement functionality. Designed to meet the rigorous demands of the most technologically advanced organizations, these applications power Customer Self-Service by managing transactional data and by enabling payments and account distribution.

eaDirect™ is the core infrastructure of enterprise Customer Self-Service solutions for organizations large and small with special emphasis on meeting the needs of organizations with large numbers of customers, high data volumes and extensive integration with systems and business processes across the enterprise. Organizations use eaDirect with its data access layer, composition engine, and security, enrollment and logging framework to power complex Customer Self-Service applications.

eaPay™ is the electronic payment solution that decreases payment processing costs, accelerates receivables and improves operational efficiency. eaPay is a complete payment scheduling and warehousing system with real-time and batch connections to payment gateways for Automated Clearing House (ACH) and credit card payments, and payments via various payment processing service providers.

eaPost® is the account content distribution system that handles all the complexities of enrollment, authentication and secure distribution of summary account information to any endpoint, while also bringing customers back the organization's Website to manage and control their self-service experience.

Advanced Interactivity Applications are a comprehensive set of advanced customer-facing self-service capabilities that enable the full range of business and consumer customer service activities. These sophisticated modules have the flexibility to completely customize the Customer Self-Service solution to meet vertical industry and specific company requirements.

eaCare™ consists of a rich set of sophisticated self-service modules – Dispute Manager, Intelligent Assistant, Hierarchy Manager, Analytics Manager, and Service and Order Manager - for automated assistance, advanced business-to-business applications and account management. These capabilities come together to create a web self-service dashboard for customers to access all service offerings from a single, easy-to-use interface. eaCare's modularity accelerates time to market with components that can be deployed incrementally in a phased approach.

Enterprise Productivity Applications are employee-facing solutions that empower customer service representatives, sales agents, account managers, marketing managers, broker-dealers and channel partners within an organization and external partner organizations to facilitate self-service and to support assisted service. Employees leverage edocs' Customer Self-Service solution to deliver customer service, access information, create and deploy marketing and customer service content, and perform activities for the benefit of customers.

eaAssist™ reduces interaction costs and increases customer satisfaction by enabling enterprise agents – customer service representatives (CSRs), sales agents, broker-dealers and others – to efficiently access critical account data and service-related information to effectively service customers. Through its browser interface designed especially for the enterprise agent, eaAssist enables agents to take advantage of customer-facing online capabilities to provide better service by more efficiently resolving customer account inquiries at the point of customer contact.

eaMarket™ is the personalization, campaign and content management solution that enables organizations to increase revenue and improve customer satisfaction by weaving personalized marketing and customer service messages throughout the Customer Self-Service experience. The transactional account data that provides the foundation for a Customer Self-Service solution – such as transaction activity, service or usage charges, current task and prior service history – bring valuable insight into customers and can help optimize personalized marketing and customer service campaigns. eaMarket leverages that data to present relevant marketing and customer service messages to customers.

edocs' **Development Tools** are visual development environments for designing and configuring edocs' Customer Self-Service solutions. The Configuration Tools encompass data and rules management, workflow authoring, systems integration, and a software development kit that makes it easy to create customer and employee-facing self-service applications leveraging eaSuite.

About This Guide

This guide describes how to install eaDirect and configure the third-party applications that make up the eaDirect production environment. This guide is intended for system administrators and other technical personnel responsible for installing, configuring, and maintaining eaDirect. It assumes in-depth understanding of and practical experience with system administrator responsibilities, including:

Application Server Administration Requirements

- Install and configure your application server
- Start and stop your application server
- Use administrative tools
- Manage users, privileges, and resources
- Configure Java resources
- Package and deploy web applications
- Monitor application server performance

If you are unfamiliar with any of these tasks, please consult the related documentation for your system requirements.

Related Documentation

A PDF version of this guide is also available.

Online	How to Access
A PDF of this guide	A PDF of this guide is available on the product CD-ROM.

This guide is part of the eaDirect documentation set. For more information about implementing your eaDirect application, see one of the following guides:

Print Document	Description
eaDirect Installation and Configuration Guides	How to install and configure eaDirect in a distributed environment.
<i>Deploying and Customizing J2EE Applications</i>	How to customize J2EE web applications for deployment with the eaSuite.
<i>Data Definition Guide</i>	How to create data extraction and definition rules for an eaDirect application with the DefTool.
<i>Presentation Design Guide</i>	How to design data presentment for an eaDirect application with the Composer.
<i>Data Presentation Production Guide</i>	How to set up and run a live eaDirect application in a J2EE environment.

Obtaining edocs Software and Documentation

You can download edocs software and documentation directly from Customer Central at <https://support.edocs.com>. After you log in, click on the Downloads button on the left. When the next page appears, you will see a table displaying all of the available downloads. To search for specific items, select the Version and/or Category and click the Search Downloads button. If you download software, an email from edocs Technical Support will automatically be sent to you (the registered owner) with your license key information.

If you received an edocs product installation CD, load it on your system and navigate from its root directory to the folder where the software installer resides for your operating system. You can run the installer from that location, or you can copy it to your file system and run it from there. The product documentation included with your CD is in the Documentation folder located in the root directory. The license key information for the products on the CD is included with the package materials shipped with the CD.

If You Need Help

Technical support is available to customers who have valid maintenance and support contracts with edocs. Technical support engineers can help you install, configure, and maintain your edocs application.

edocs provides global Technical Support services from the following Support Centers:

US Support Center

Natick, MA
Mon-Fri 8:30am – 8:00pm US EST
Telephone: 508-652-8400

Europe Support Center

London, United Kingdom
Mon-Fri 9:00am – 5:00 GMT
Telephone: +44 20 8956 2673

Asia Pac Rim Support Center

Melbourne, Australia
Mon-Fri 9:00am – 5:00pm AU
Telephone: +61 3 9909 7301

Customer Central

<https://support.edocs.com>

Email Support

<mailto:support@edocs.com>

When you report a problem, please be prepared to provide us the following information:

- What is your name and role in your organization?
- What is your company's name?
- What is your phone number and best times to call you?
- What is your e-mail address?
- In which edocs product did a problem occur?
- What is your Operating System version?
- What were you doing when the problem occurred?
- How did the system respond to the error?
- If the system generated a screen message, please send us that screen message.
- If the system wrote information to a log file, please send us that log file.

If the system crashed or hung, please tell us.



Upgrading from WebLogic 6.1 to 7.0

If you are upgrading from WebLogic Server 6.x to WebLogic Server 7.0, **you must consult your WebLogic documentation**. This guide offers only an overview of the required steps. For detailed instructions, see the *WebLogic Server 7.0 Upgrade Guide* at <http://e-docs.bea.com/wls/docs70/upgrade/index.html>.

To upgrade your server from WebLogic Server 6.1 to 7.0:

1. Make a backup copy of your 6.x domain before you begin the upgrade procedure. After you start the server using WebLogic Server 7.0 classes, you cannot downgrade to 6.x.
2. Install WebLogic Server 7.0. See the BEA topic for the Installation Guide.

**Caution**

The installer will prevent you from installing the new version directly over the old version. You must select a new directory location.

The Installer will create a new BEA Home directory. From the BEA Quick Start, you may then run the Configuration Wizard to create a server domain.

3. Modify your 6.x startup scripts to work with WebLogic Server 7.0. See the BEA topic *Modifying Startup Scripts*.
4. Ensure that you have considered differences in the WebLogic Server 7.0 directory structure that may require you to make file location changes before startup. See the BEA topic *Understanding the WebLogic Server 7.0 Directory Structure*.
5. Port your applications to WebLogic Server 7.0. See the BEA topic *Porting an Application from WebLogic Server 6.x to WebLogic Server 7.0*.
6. If necessary, perform other upgrade procedures as described in your WebLogic documentation.
7. To upgrade a cluster of servers, follow the above steps for each server and then follow the steps outlined in Setting Up WebLogic Clusters in Using WebLogic Server Clusters. In cases where you invoke an application by using RMI/T3 or RMI/IIOP, WebLogic Server 6.1 and 7.0 are interoperable. Within a domain, however, all servers must be of the same version.

8. For information on upgrading WebLogic Server license files, see the BEA topic [Upgrading Licenses from WebLogic Server 6.x in the Installation Guide](#).

Upgrading Startup Scripts for WebLogic 7.0

If you used WebLogic Server startup scripts with a previous version of the product, modify them to work with WebLogic Server 7.0. For details, see your WebLogic documentation.

To edit your startup script for WebLogic 7.0:

1. Modify `bea.home` property to point to your BEA home directory containing the `license.bea` file for WebLogic Server 7.0. For example:

```
-Dbea.home=C:\bea700
```
2. Modify `WL_HOME` to point to your WebLogic Server 7.0 installation directory. For example:

```
WL_HOME=c:\bea700\weblogic700
```
3. Modify `PATH` so that it includes your `%WL_HOME%` 7.0 home. For example:

```
PATH=%WL_HOME%\bin;%PATH%
```
4. Modify `CLASSPATH` so that it points to the WebLogic Server 7.0 classes. For example:

```
CLASSPATH=%WL_HOME%\lib\weblogic_sp.jar;%WL_HOME%\lib\weblogic.jar
```
5. Modify or eliminate any WebLogic Server 6.x startup script directory structure tests. For example, if your script tries to verify a relative path, either fix the directory structure test or remove it.



WebLogic Server 7.0 installs the JVM, `JDK 1.3.1_02`, with the server installation. The `setenv.sh` scripts provided with the server all point to the JVM.

About Example Domains in This Guide

This guide uses the example UNIX domain `$WL_HOME/config/mydomain`, and the example Windows domain `%WL_HOME%\config\mydomain`.

WebLogic 7 users may use the Domain Configuration Wizard to create `/config/mydomain`, or replace these pathnames with a custom domain created by your system administrator. For more information, see [About WebLogic Server Domains](#).



If you use a custom domain, the examples in this guide must be changed accordingly or they may not work. edocs does not recommend that you accept the default path of `/user_projects`.

About WebLogic Server Domains

Creation of application server domains changed from WebLogic 6.1 to WebLogic 7. WebLogic 6.1 creates a default domain during install, `$WL_HOME/config/mydomain`. WebLogic 7 creates only a default domain directory, the parent directory of `$WL_HOME/user_projects`, and each user must run the Domain Configuration Wizard to create a domain.

WebLogic 7.0 Directory Structure

This directory . . .	Contains . . .
<code>weblogic700</code>	Folders for files shared by all WebLogic Platform components, as well as individual folders for each installed WebLogic Platform component.
<code>common</code>	Files shared by WebLogic Platform components, including template jar files used by the Configuration Wizard when creating domains. The Configuration Wizard reads the contents of this folder and displays template options based on the jar files in the directory. See BEA topic Configuring Domains, Servers, and Clusters with the Configuration Wizard for more information about using the Configuration Wizard.
<code>samples</code>	Sample code, resources, and preconfigured domains designed to help you learn how to develop your own applications using WebLogic Platform. Samples are organized by component installed on the system. Each WebLogic Platform component has its own folder within the samples folder. The server folder in the samples folder contains the examples and petstore domains, source code for the examples and petstore sample applications, and PointBase evaluation software.
<code>server</code>	WebLogic Server program files.
<code>uninstall</code>	Code required to uninstall the WebLogic Server software.
<code>workshop</code>	WebLogic Workshop application and documentation files and files to support the Workshop sample domain.

7.0 Components Installed with the Configuration Wizard

Component	Function
<code>config.xml</code>	The <code>config.xml</code> file is an XML document that describes the configuration of a WebLogic Server domain. To learn more about <code>config.xml</code> , see BEA WebLogic Server Configuration Reference.

Component	Function
<code>/applications</code>	Stores the default Web Application for servers in the domain. If you select a template that includes WebLogic Server examples, then this directory also stores JAR and EAR files for individual examples.
<code>/logs</code>	Stores log files for servers in the domain.
<code>setEnv.cmd,</code> <code>setEnv.sh</code>	Set environment variables for domain servers.
<code>startWebLogic.cmd,</code> <code>startWebLogic.sh</code>	Start the Administration Server for a custom domain.
Windows Start Menu Items	Provide Start menu support for starting domain servers in a Windows environment.

Migrating eaDirect J2EE Applications

When migrating to a new version of eaDirect **or** upgrading to a new version of WebLogic, you must first undeploy J2EE applications running on your application server. Once you have upgraded your software and/or migrated your database, you should then redeploy the new versions of each eaSuite web application, including eaDirect and samples. These sample applications provide examples of how to use the current features of eaDirect. You will need to:

- See the *Release Notes* for your version for important information about specific migration requirements and settings, new features, and how to add support for those features to your application.
- See *Deploying and Customizing J2EE Applications* for information about the components that make up the J2EE and Web applications, and the procedures for recreating EAR and WAR files.
- Merge your custom code into the sample application, and rename it. For more information, see *Deploying and Customizing J2EE Applications*.
- Redeploy your custom application.

For more information on deployment, see the eaDirect guide to *Deploying and Customizing J2EE Applications*.

2

Configuring WebLogic Server for UNIX

Overview

This chapter assumes in-depth understanding of and practical experience with [Application Server Administration](#). Consult WebLogic Server documentation at <http://bea.com> as necessary.

You must start your WebLogic Server instance and bring up the Administrative Console before you begin this chapter.

Caution If you cannot bring up the WebLogic Console, you will be unable to proceed with configuring your application server for eaDirect.

edocs recommends that you install and configure eaDirect in the same top-level directory structure, first on the database server, then the application server.

If you have not already installed database server components and configured the database server for eaDirect, do so now.

Tip For distributed environments, ensure that you have any required **database client software** installed on WebLogic Server and any other client machines of your database server. See the [Quick Reference Appendix](#) for software requirements for your platform.

This chapter provides instructions for configuring WebLogic Server to support eaDirect. It includes:

- [UNIX Permissions for WebLogic Server](#)
- [Starting and Stopping WebLogic Server](#)
- [UNIX Environment Variables for WebLogic Server](#)
- [Where to Go From Here](#)

Caution The installation and configuration examples shown in this guide use default eaDirect pathnames, privileges, and permissions. If you choose not to accept the default values, make sure your values are consistent on all servers across your installation of eaDirect.

About Example UNIX Domains in This Guide

This guide uses the example UNIX domain `$WL_HOME/config/mydomain`. WebLogic 7 users may use the Domain Configuration Wizard to create the UNIX domain `$WL_HOME/config/mydomain`, or replace these pathnames with a custom domain created by your system administrator. For more information, see [About WebLogic Server Domains](#).



If you use a custom domain, the examples in this guide must be changed accordingly or they may not work. edocs does not recommend that you accept the default path of `/user_projects`.

UNIX Permissions for WebLogic Server



This section applies to all UNIX platforms.

Application servers running eaDirect will not function correctly without access to eaDirect config files, storage directories, and related resources. When installing eaDirect on WebLogic Server, you were asked to specify the owner and group permissions (`userid:groupid`) of the WebSphere application server. If you made a mistake during installation, you must change the owner and group permissions for the directory, including all subfolders, to the application server owner.

This guide uses the example username and password `edxadmin:edxadmin` as the application server owner and group for the eaSuite.

DEFAULT	EXAMPLE	CUSTOM
Specified during installation.	<code>edxadmin:edxadmin</code>	

For details on owner and group permissions for WebLogic Server, please consult the application server documentation for your platform.

To reset user and group permissions:

1. Switch user to the default owner of WebLogic Server home directory, for example `root`.

```
su - root
```
2. Recursively change the user and group permissions of the application server installation directory and all subdirectories to the application server owner.

```
chown -R edxadmin:edxadmin /export/home/bea
```
3. Switch user to WebLogic Server owner and configure WebLogic Server with your new owner.

```
su - edxadmin
```



You should also verify the owner information in any profile files used by the database server owner and application server owner. See your server documentation for details.

Starting and Stopping WebLogic Server

Developers and system administrators will need to be familiar with how to stop and start WebLogic Server and any active web applications for your platform.

- eaDirect for **WebLogic** is supported on Windows, Solaris, and HP-UX.

See the [Quick Reference Appendix](#) for a detailed list of system requirements and settings for each platform.

About Sourcing Your Configuration

Before you start your server instance, you must edit its WebLogic Server startup script to **source** your customized version of the configuration file `edx.config`, thus passing your eaDirect environment to WebLogic Server at startup. For details, see [Passing UNIX Environment Data to WebLogic](#).

Providing Usernames and Passwords to Start a Server

By default, a WebLogic Server instance prompts you to enter a username and password in the command shell that runs the server process. For details on bypassing this process, see BEA documentation at:

<http://e-docs.bea.com/wls/docs70/adminguide/startstop.html#1090559>

Starting and Stopping an Active Application Server

Improperly starting or stopping an application server in an active eaDirect production environment can produce unexpected and unintended results. You can create custom startup and shutdown scripts that include all your command parameters, as well as the command used to start or stop the Scheduler, to schedule and run jobs in the eaDirect Command Center.

The default command-line startup shell scripts are fine for an inactive production environment where there are no running jobs. However, the startup process will stop immediately if you enter a `Ctrl+C` (often used to force a hard shutdown of the server) in the startup directory, or if you close the terminal session. This can damage your configuration file. edocs recommends using the web console and/or the SHUTDOWN command to ensure a graceful shutdown.

To start WebLogic in an active eaDirect production environment, edocs recommends that you use the `nohup` command to ignore hang-ups. This will leave the server running in the background even if you end your terminal session or try to force a hard shutdown, providing a more stable production environment.

Starting and Stopping WebLogic Server

You can start or stop the administrative server:

- from the web console
- from the command line

For more details on starting and stopping WebLogic Server and for verifying startup, please consult the server documentation for your platform.

To stop WebLogic Server from the web console:

1. In your browser, enter the console URL `http://localhost:7001/console`.
2. Expand your domain until `myserver` appears.
3. Right-click `myserver` and select **Stop this server**.

To start WebLogic Server from the command line:

1. `su - edxadmin`
2. `cd $WL_HOME/config/mydomain`
3. `nohup ./startWebLogic.sh &`
4. `WLS_PW=edxadmin`

To stop WebLogic Server from the command line:

1. `su - edxadmin`
2. `cd $WL_HOME/config/mydomain`
3. `./setEnv.sh`
4. `java weblogic.Admin -url localhost:7001 SHUTDOWN -username system -password edocs`

Capturing Your UNIX Environment for eaDirect



This section applies to all UNIX platforms.

eaDirect installs several configuration files that you use to define your eaDirect environment. These configuration scripts are required **only on the application server**.

<code>\$EDX_HOME/bin/edx_config</code>	executable shell script prompts you to define environment variables required by your application server
<code>\$EDX_HOME/config/edx_env</code>	non-editable configuration file stores the environment variables you specify in <code>edx_config</code>
<code>\$EDX_HOME/config/edx.config</code>	shell script passes the environment data in <code>edx_env</code> to your application server when sourced in your startup script

This section describes how to use `edx_config` to capture your environment variables and store them in `edx_env`. For information on `edx.config`, see [Passing UNIX Environment Data to WebLogic](#).

Using `edx_config` to capture environment data

`edx_config` prompts you to enter values for your Java and database installation, including absolute directory pathnames or user identification information. It stores these values in the configuration file `edx_env`.

You should run `edx_config` any time you need to modify your eaDirect environment. Do not modify `edx_env` directly.

Caution Do not confuse `edx_config` (underscore) with `edx.config` (dot), which sources your configuration at server startup. For details, see [Using `edx.config` to source your configuration](#).

Caution Be sure the time zone (TZ) for your server is set to your system time zone. eaDirect jobs can fail if Java system time does not match actual system time.

Use the [Quick Reference Appendix](#) to enter values for your platform for each of the specified parameters. You may want to print the *Environment Variables* sections for easy reference. You can accept the default values, if appropriate, or enter your own.

To capture environment data with `edx_config` for WebLogic 6.1 and 7.0:

1. Switch user to the `$EDX_HOME` owner, in this example `edxadmin`.

```
su - edxadmin
```
2. Navigate to the `bin` directory for eaDirect on your application server, for example

```
cd $EDX_HOME/bin
```
3. Run the script `edx_config`.

```
./edx_config
```

4. Enter values as prompted by the script for your database home, database username and password, application server, Java home, and application server home. Use the [Quick Reference Appendix](#) for your version and platform.

**Caution**

Make sure that the database values you enter in this session are the same values specified during database configuration. Consult your DBA for any custom settings specific to your platform.

Using WebLogic 7.0 Configuration Wizard Scripts to Start an Application Server

When you use the Configuration Wizard to create a domain, the wizard also creates a script that you can use to start an Application Server for the domain. To use the script, enter one of the following commands at a command prompt:

- `mydomain\startWebLogic.cmd` (Windows)
- `mydomain/startWebLogic.sh` (UNIX and Windows. On Windows, this script supports the MKS and Cygnus BASH UNIX shell emulators.)

where `mydomain` is the directory in which you located your domain.

The script sets values for some domain-specific variables and then calls the master startup script, `$WL_HOME\server\bin\startWLS.sh`, where `$WL_HOME` is the location where you installed WebLogic Server. The master startup script sets environment variables, such as the location of the JVM, and then starts the JVM with WebLogic Server arguments.

This guide uses the example domain `$WL_HOME/config/mydomain`. WebLogic 7 users may use the Domain Configuration Wizard to create the domain `$WL_HOME/config/mydomain`, or replace these pathnames with a custom domain created by your system administrator. For more information, see [About WebLogic Server Domains](#).

**Caution**

If you use a custom domain, the examples in this guide must be changed accordingly or they may not work.

Passing UNIX Environment Data to WebLogic

**Tip**

This section applies to all UNIX platforms supported by WebLogic.

This section describes how to use `edx.config` to pass your environment data to WebLogic at server startup. For information on `edx.config`, see [Capturing Your UNIX Environment for eaDirect](#).

Using *edx.config* to source your configuration

edx.config is a shell script that you call and process in your application server startup script to pass your eaDirect environment (stored in **edx_env**) to WebLogic. This procedure is also called **sourcing your configuration**.



Do not confuse **edx.config** (dot) with **edx_config** (underscore), which prompts you to enter the environment data stored in **edx_env**. For details, see [Using edx_config to capture environment data](#).

To source your configuration for WebLogic (overview):

1. Edit `$WL_HOME/config/mydomain/startWebLogic.sh` to set the server name to be started (WebLogic 7.0 only), set the home directory for eaDirect, set Java options for your JVM, and call and process (source) the configuration script **edx.config**.
2. (optional for WebLogic 7.0 only) Edit `$WL_HOME\server\bin\startWLS.sh` to set the location and Java options for your JVM and set other environment variables.

Sourcing Your Configuration

For WebLogic 6, the **startWebLogic.sh** script for your domain sets values for both domain and environment variables such as the JVM. You source your eaDirect configuration directly in **startWebLogic.sh**.

For WebLogic 7, the **startWebLogic.sh** script for your domain sets values for some domain-specific variables and then calls the master startup script, `$WL_HOME\server\bin\startWLS.sh`. The master startup script sets environment variables, such as the location of the JVM, and then starts the JVM with WebLogic Server arguments.

edocs recommends that WebLogic 7 users source your eaDirect configuration directly in **startWebLogic.sh** for your domain, for consistency with this new feature of WebLogic 7.

WebLogic 7 users with clustered installations or with custom domain names will need to study the new features of domain configuration in your WebLogic Server documentation at <http://e-docs.bea.com/wls/docs70/adminguide/startstop.html>.

This eaDirect guide documents the default WebLogic 6.1 domain `config/mydomain` and recommends that WebLogic 7 users use the Domain Configuration Wizard to create this domain for configuration. For more information, see [About WebLogic Server Domains](#).



WebLogic and Java directory paths are slightly different for WebLogic 6.1 and WebLogic 7.0. Be sure to use the correct paths for your version.

To edit `startWebLogic.sh` for WebLogic 6.1 or 7.0:

1. Switch user to your application server owner, for example `edxadmin`.

```
su - edxadmin
```

2. Stop WebLogic Server and all application server instances. For details, see [Starting and Stopping WebLogic Server](#).

3. Navigate to the application server startup script **for your domain** and open the file for editing. For example:

```
cd $WL_HOME/config/mydomain/startWebLogic.sh
vi startWebLogic.sh
```

4. (WebLogic 7 only) Set the server name variable for the server instance you wish to start. For example:

```
SERVER_NAME=myserver
```

5. Define and export the environment variable for your eaDirect home directory. For example:

```
EDX_HOME=/opt/EDCSbd
export EDX_HOME
```

6. **Optimize JVM Memory** by increasing the memory arguments allocated to the Java Virtual Machine (JVM) on the application server. For example (quotes are optional):

```
set MEM_ARGS="-ms128m -mx128m -Xss1m -noclassgc"
```

Parameter	Description	Example
<code>-ms</code>	Sets the initial Java heap size	<code>-ms128m</code>
<code>-mx</code>	Sets the maximum Java heap size	<code>-mx128m</code>
<code>-Xss</code>	Sets the maximum native stack size for any thread	<code>-Xss1m</code>
<code>-noclassgc</code>	Disables the Java class garbage collection	<code>-noclassgc</code>

For WebLogic 7, you may set this variable either here or in `startWLS.sh`, which `startWebLogic.sh` calls. Values in the master `startWLS.sh` will overwrite those in `startWebLogic.sh`. See your WebLogic 7.0 documentation for more information.

**Tip**

Be careful not to overallocate memory to the JVM. System processes like the garbage collector consume available memory, and performance can degrade when the application server is not able to respond to other requests.

7. **Password:** (optional) Set `WLS_USER` equal to your system username and `WLS_PW` equal to your system password for no username and password prompt during server startup. Both are required to bypass the startup prompt.

**Caution**

Instead of using a cleartext password variable, BEA recommends that you use a boot identity file. See WebLogic Server documentation at <http://bea.com>.

8. Source `edx.config` just after the `STARTMODE` command to set production mode (and immediately before the `$JAVACMD` to start the JVM). The dot and space preceding the pathname are a required part of the syntax. For example:

```
STARTMODE=false
```

```
. /opt/EDCSbd/config/edx.config
```

9. Save and close `startweblogic.sh`.

3

Configuring WebLogic Server for Windows

Overview

This chapter assumes in-depth understanding of and practical experience with [Application Server Administration](#). Consult WebLogic Server documentation at <http://bea.com> as necessary.

edocs recommends that you install and configure eaDirect in the same top-level directory structure, first on the database server, then the application server.

If you have not already installed database server components and configured the database server for eaDirect, do so now.

Tip

For distributed environments, ensure that you have any required **database client software** installed on your application server and any other client machines of your database server. See the [Quick Reference Appendix](#) for software requirements for your platform.

This chapter provides instructions for configuring WebLogic Server to support eaDirect. It includes:

- [Starting and Stopping WebLogic Server](#)
- [Windows Services for eaDirect](#)
- [Windows Environment Variables for WebLogic](#)
- [Where to Go From Here](#)

Caution

The installation and configuration examples shown in this guide use default eaDirect pathnames, privileges, and permissions. If you choose not to accept the default values, make sure your values are consistent on all servers across your installation of eaDirect.

About Example Windows Domains in This Guide

This guide uses the example Windows domain `%WL_HOME%\config\mydomain`. WebLogic 7 users may use the Domain Configuration Wizard to create the Windows domain `%WL_HOME%\config\mydomain`, or replace these pathnames with a custom domain created by your system administrator. For more information, see [About WebLogic Server Domains](#).



If you use a custom domain, the examples in this guide must be changed accordingly or they may not work. edocs does not recommend that you accept the default path of `\user_projects`.

Starting and Stopping WebLogic Server

Developers and system administrators will need to be familiar with how to stop and start WebLogic Server and any active web applications for your platform.

- eaDirect for **WebLogic** is supported on Windows, Solaris, and HP-UX.

WebLogic Server startup script must be edited to **source** your customized version of the configuration file `edx.config`, thus passing your eaDirect environment to WebLogic Server at startup. For details, see [Passing Environment Data to WebLogic Server](#) for your platform.

See the [Quick Reference Appendix](#) for a detailed list of system requirements and settings for each platform.

You can start or stop the administrative server:

- from the web console
- from the command line
- as a Windows Service

For details on starting and stopping WebLogic Server and for verifying startup, please consult the server documentation for your platform.

Starting and Stopping an Active Application Server

Improperly starting or stopping an application server in an active eaDirect production environment can produce unexpected and unintended results. You can **and should** create custom startup and shutdown scripts that include all your command parameters, as well as the command used to start or stop the eaDirect Scheduler.

The default command-line startup shell scripts are fine for an inactive production environment where there are no running jobs. However, the startup process will stop immediately if you enter a `Ctrl+C` (often used to force a hard shutdown of the server) in the startup directory, or if you close the terminal session.

By default, if you use the Windows Control Panel to stop a server instance, the Windows Service Control Manager (SCM) kills the server's Java Virtual Machine (JVM). If you kill the JVM, the server immediately stops all processing. Any session data is lost. If you kill the JVM for an Administration Server while the server is writing to the config.xml file, you can corrupt the config.xml file. See BEA Documentation for Enabling Graceful Shutdowns from the Control Panel at:

<http://e-docs.bea.com/wls/docs70/adminguide/startstop.html#1107060>

Starting and Stopping WebLogic Server

To start WebLogic in an active eaDirect production environment, edocs recommends that you use the `nohup` command to ignore hang-ups. This will leave the server running in the background even if you end your terminal session or try to force a hard shutdown, providing a more stable production environment.

This section provides a quick reference for typical installations. For details about starting and stopping WebLogic Server 6.1, see:

<http://e-docs.bea.com/wls/docs61/adminguide/startstop.html#1026476>

For details about starting and stopping WebLogic Server 7.0, see:

<http://e-docs.bea.com/wls/docs70/adminguide/startstop.html#1117614>

To start WebLogic Server from the web console:

1. In your browser, enter the console URL `http://localhost:7001/console`.
2. Expand your domain until `myserver` appears.
3. Right-click `myserver` and select **Start this server**.

To stop WebLogic Server from the web console:

1. In your browser, enter the console URL `http://localhost:7001/console`.
2. Expand your domain until `myserver` appears.
3. Right-click `myserver` and select **Stop this server**.

To start WebLogic Server as a Windows Service:

1. Install the Windows Service for your server instance. For details, see [Setting Up a WebLogic Server Instance as a Windows Service](#).
2. Select **Start Menu>Settings>Control Panel**.
3. Select **Administrative Tools>Services**.
4. Right-click `myserver` and select **Start**.

5. Allow a few minutes for the service to start WebLogic Server.

To stop WebLogic Server as a Windows Service:

1. Select **Start Menu>Settings>Control Panel**.
2. Select **Administrative Tools>Services**.
3. Right-click **myserver** and select **Stop**.
4. Allow a few minutes for the service to stop WebLogic Server.

Capturing Your Windows Environment for eaDirect

eaDirect installs several configuration files that you use to define your eaDirect environment. These configuration scripts are required **only on the application server**.

<code>%EDX_HOME%\config\edx_env.bat</code>	editable configuration file stores environment variables required by your application server
<code>%EDX_HOME%\config\edx_load.config.bat</code>	editable configuration file stores database Java options required by your application server
<code>%EDX_HOME%\config\edx.config.bat</code>	shell script passes the environment data in <code>edx_env.bat</code> and <code>edx_load.config.bat</code> to your application server when called in your startup script

This section describes how to edit `edx_env.bat` and `edx_load.config.bat` to store your environment variables. For information on `edx.config.bat`, see [Passing Windows Environment Data to WebLogic](#).

Using `edx_env.bat` and `edx_load.config.bat` to Store Environment Data

You must edit the configuration file `edx_env.bat` to set values for your eaDirect home, application server home, and Java home directories.

You must also edit the configuration file `edx_load.config.bat` to set values for your database user, password, and server name. This file is called by `edx_env.bat`.

Use the [Quick Reference Appendix](#) to enter application server environment variables for your platform for each of the specified parameters. You may want to print the *Environment Variables* sections for easy reference. You can accept the default values, if appropriate, or enter your own.

To edit Windows environment data with `edx_env.bat`:

1. Navigate to `%EDX_HOME%\config` and open `edx_env.bat`.
2. Modify the default settings to reflect your eaDirect environment. For example:

```
@rem define EDX_HOME
@set EDX_HOME=C:\EDCSbd
@rem define EDX_HOME

@rem define WL_HOME
@set WL_HOME=C:\bea\wlserver\
@rem define WL_HOME

@rem define APP_SERVER
@set APP_SERVER=wl
@rem define APP_SERVER
```



Caution

Make sure you set all paths to the appropriate point releases/patches for WebLogic Server and JDK, if necessary. Check the Release Notes and your system documentation for updated requirements.

3. Save and close the file.

To edit Windows Java Options with `edx_load.config.bat`:



Tip

The following procedure shows example variables for **SQL Server** on **Windows**. Application server administrators on other platforms should consult the [Quick Reference Appendix](#) for appropriate variables.

1. Navigate to `%EDX_HOME%\config` and open `edx_load.config.bat`.
2. Modify the default settings for `com.edocs.tasks.loader` to reflect your database user, password, and server name. Use the settings for your database from the [Quick Reference Appendix](#). For example:

```
@set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.edocs.tasks.loader.user=edx_dba
@set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.edocs.tasks.loader.password=edx
@set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.edocs.tasks.loader.alias=localhost
```

3. Save and close the file.



Caution

Make sure you set all paths to the appropriate point releases/patches for WebLogic Server and JDK, if necessary. Check the Release Notes and your system documentation for updated requirements.

Passing Windows Environment Data to WebLogic 6.1

`edx.config.bat` is a shell script that you call and process in your application server startup script to pass your eaDirect environment (stored in `edx_env.bat` and `edx_load.config.bat`) to WebLogic.



Do not confuse `edx.config.bat` with `edx_env.bat`, in which you enter the environment data to pass to the server. For details, see [Using `edx_env.bat` and `edx_load.config.bat` to Store Environment Data](#).

This section describes how to use `edx.config.bat` to pass your environment data to WebLogic at server startup. For information on `edx_env.bat` and `edx_load.config.bat`, see [Capturing Your Windows Environment for eaDirect](#).

To pass your eaDirect environment to WebLogic 6.1 (overview):

1. Determine whether you wish to start WebLogic as a Windows Service or directly from the startup script. Use the appropriate procedure for your service or startup script.
2. In your domain service or startup script, set your eaDirect home directory, `%EDX_HOME%`.
3. In your domain service or startup script, call and process the configuration script `edx.config.bat`.



`edx_env.bat` calls `edx_load.config.bat`, so you need not source this database config file directly in your service or startup script.

Sourcing Your Configuration as a Windows Service for WebLogic 6.1

edocs recommends installing WebLogic Server as a Windows Service, and modifying the script that calls that service. For WebLogic 6.1, this file is `%WL_HOME%\config\mydomain\InstallNtService.cmd`. Before editing `InstallNtService.cmd`, be sure to save a backup copy in a **different directory**.

Example InstallNtService.cmd for WebLogic 6.1

Bold indicates text that you should add or change from the default.

```
@echo off
SETLOCAL

cd C:\bea\wlserver\config\mydomain

@set EDX_HOME=C:\EDCSbd

@call %EDX_HOME%\config\edx.config.bat

rem *** Set Classpath to load Weblogic Classes

set
CLASSPATH=.;C:\bea\wlserver\lib\weblogic_sp.jar;C:\bea\wlserver\lib\weblogic.jar
```



```

rem *** Set Command Line for service to execute
rem *** %JAVA_HOME%\java will automatically be prepended.

set CMDLINE="%JAVA_OPTIONS% -classpath \"%CLASSPATH%"
-Dweblogic.Domain=mydomain -Dweblogic.Name=myserver
-Djava.security.policy=="C:\bea\wlserver\lib\weblogic.policy\"
-Dbea.home="C:\bea\" weblogic.Server"

rem *** Install the service

"C:\bea\wlserver\bin\beasvc" -install -svcname:myserver
-javahome:"%JAVA_HOME%" -execdir:"C:\bea\wlserver"
-extrapath:"C:\bea\wlserver\bin;C:\EDCSbd\lib" -cmdline:%CMDLINE%
-password:change_on_install

ENDLOCAL

```

To edit InstallNtService.cmd for WebLogic 6.1:

See the example above for default settings. Make sure to change these as needed for your environment.

1. Stop WebLogic Server and all application server instances. For details, see [Starting and Stopping WebLogic Server](#).
2. Navigate to the %WL_HOME%\config\mydomain subdirectory of your application server home directory.
3. Open `InstallNtService.cmd` by right clicking on its name, and selecting **Edit**.
4. **Source Your Configuration:** Just after the `CLASSPATH` command and immediately **before** the Java command line, set your eaDirect home directory and call your eaDirect environment script, `edx.config.bat`. See the example above.
5. **Java Command Line:** replace the original `JAVA_OPTIONS` settings with the variable `%JAVA_OPTIONS%`. See the example above.
6. **CLASSPATH:** in the `Dweblogic.Name=` setting, change the name of your server if necessary. The default name is `myserver`.
7. **Install the service:** change the default JDK home directory to the variable `%JAVA_HOME%`. See the example above.



Make sure you set all paths to the appropriate point releases/patches for WebLogic Server and JDK, if necessary. Check the Release Notes and your system documentation for updated requirements.

8. **Install the service:** add your eaDirect `\lib` directory to the `-extrapath` setting. See the example above.
9. **Install the service:** after the `%CMDLINE%` setting, set your application server password with the `-password` command. See the example above.
10. Save and close `InstallNtService.cmd`.

Sourcing Your Configuration in a Startup Script for WebLogic 6.1

You can also choose to start WebLogic Server directly by modifying the server startup script to source your configuration. For WebLogic 6.1, this file is `%WL_HOME%\config\mydomain\startWebLogic.cmd`. Before editing `startWebLogic.cmd`, be sure to save a backup copy **in a different directory**.

Example `startWebLogic.cmd` for WebLogic 6.1

Bold indicates text that you should add or change from the default.

```

..
..

@rem set edocs home directory and source your configuration
@set EDX_HOME=C:\EDCSbd
@call %EDX_HOME%\config\edx.config.bat

"%JAVA_HOME%\bin\java" %JAVA_OPTIONS% -classpath "%CLASSPATH%" -
Dweblogic.Domain=%WL_DOMAIN% -Dweblogic.Name=myserver "-
Dbea.home=C:\bea" -Dweblogic.management.password=%WLS_PW%
-Dweblogic.ProductionModeEnabled=%STARTMODE%
-Djava.security.policy=C:\bea\wlserver6.1\lib\weblogic.policy"
weblogic.Server

```

To edit `startWebLogic.cmd` for WebLogic 6.1:

See the example above for default settings. Make sure to change these as needed for your environment.

1. Stop WebLogic Server and all application server instances. For details, see [Starting and Stopping WebLogic Server](#).
2. Navigate to the `%WL_HOME%\config\mydomain` subdirectory of your application server home directory.
3. Open `startWebLogic.cmd` by right clicking on its name, and selecting **Edit**.
4. Immediately **before** the Java command line, set your eaDirect home directory and call your eaDirect environment script, `edx.config.bat`. See the example above.
5. **JAVA:** In the Java command line section, replace the original CLASSPATH with the variable `%CLASSPATH%`. See the example above.
6. **JAVA:** In the Java command line section, replace the original `JAVA_OPTIONS -ms64m -mx64m` with the variable `%JAVA_OPTIONS%`. See the example above.
7. **JAVA:** In the Java command line section, change the name of your server if necessary. The default service name is `myserver`.



Caution

Make sure you set all paths to the appropriate point releases/patches for WebLogic Server and JDK, if necessary. Check the Release Notes and your system documentation for updated requirements.

8. Save and close `startWebLogic.cmd`.

Passing Windows Environment Data to WebLogic 7.0

Tip

This section applies to all Windows platforms.

`edx.config.bat` is a shell script that you call and process in your application server startup script to pass your eaDirect environment (stored in `edx_env.bat` and `edx_load.config.bat`) to WebLogic.

Caution

Do not confuse `edx.config.bat` with `edx_env.bat`, in which you enter the environment data to pass to the server. For details, see [Using `edx_env.bat` and `edx_load.config.bat` to Store Environment Data](#).

This section describes how to use `edx.config.bat` to pass your environment data to WebLogic at server startup. For information on `edx_env.bat` and `edx_load.config.bat`, see [Capturing Your Windows Environment for eaDirect](#).

To pass your eaDirect environment to WebLogic 7.0 (overview):

1. Determine whether you wish to start WebLogic as a Windows Service or directly from the startup script. Use the appropriate procedure for your service or startup scripts.
2. In your **domain** service or startup script, set your eaDirect home directory, `%EDX_HOME%`.
3. In your **domain** service or startup script, call and process the configuration script `edx.config.bat`. This procedure is called **sourcing** your configuration.
4. In the **master** service or startup script, set your CLASSPATH, to use the classpath defined in `edx.config.bat`.

Tip

`edx_env.bat` calls `edx_load.config.bat`, so you need not source this database config file directly in your startup script.

Passing Your Configuration to WebLogic 7.0 Running as a Windows Service

edocs recommends installing WebLogic Server as a Windows Service, and modifying the script that calls that service. For WebLogic 7.0, this file is

`%WL_HOME%\config\mydomain\InstallService.cmd`. This script calls the master startup script `%WL_HOME%\server\bin\InstallSvc.cmd`.

Before editing either of these files, be sure to save a backup copy **in a different directory**.

Example InstallService.cmd for WebLogic 7.0

Bold indicates text that you should add or change from the default.

```

..
..
@set EDX_HOME=C:\EDCSbd
@call %EDX_HOME%\config\edxk.config.bat
@rem Set JAVA_OPTIONS to the java flags you want to pass to the vm.
i.e.:
@rem set JAVA_OPTIONS=-Dweblogic.attribute=value -
Djava.attribute=value
set JAVA_OPTIONS=%JAVA_OPTIONS%
@rem Set JAVA_VM to the java virtual machine you want to run. For
instance:
@rem set JAVA_VM=-server
set JAVA_VM=
@rem Set MEM_ARGS to the memory args you want to pass to java. For
instance:
@rem set MEM_ARGS=-Xms32m -Xmx200m
set MEM_ARGS="-ms128m -mx128m -Xss1m -noclassgc"
@rem Call Weblogic Server service installation script
call "C:\bea\weblogic700\server\bin\installSvc.cmd"
ENDLOCAL

```

To edit InstallService.cmd for WebLogic 7.0:

See the example above for default settings. Make sure to change these as needed for your environment.

1. Stop WebLogic Server and all application server instances. For details, see [Starting and Stopping WebLogic Server](#).
2. Navigate to the %WL_HOME%\config\mydomain subdirectory of your application server home directory.
3. Open `InstallService.cmd` by right clicking on its name, and selecting **Edit**.
4. **USERDOMAIN_HOME**: Set your user domain to the directory of the domain you wish to use, for example %WL_HOME%\config\mydomain. If you used the Domain Configuration Wizard, this variable should be preset correctly.
5. **SERVER_NAME**: Set SERVER_NAME to the name of the server you wish to start up. The default is `myserver`.
6. **DOMAIN_NAME**: Set DOMAIN_NAME to the name of the domain you wish to start up.

7. **Set Production Mode:** When STARTMODE is set to true, the server starts up in production mode. When set to false, the server starts up in development mode. If it is not set, it will default to false.
8. **Set the JVM:** Set `JAVA_VM` to null.
9. **Optimize JVM Memory** by increasing the memory arguments allocated to the Java Virtual Machine (JVM) on the application server. For example (quotes are optional):

```
set MEM_ARGS="-ms128m -mx128m -Xss1m -nothreadgc"
```
10. **Install the service:** Call the master WebLogic Server service installation script in `%WL_HOME%\server\bin`. **This is specific to WebLogic 7.** For more information, see [About WebLogic Server Domains](#).

**Caution**

Make sure you set all paths to the appropriate point releases/patches for WebLogic Server and JDK, if necessary. Check the Release Notes and your system documentation for updated requirements.

11. **Install the service:** add your eaDirect `\lib` directory to the `-extrapath` setting. See the example above.
12. **Install the service:** after the `%CMDLINE%` setting, set your application server password with the `-password` command. See the example above.
13. Save and close `InstallService.cmd`.

To edit `InstallSvc.cmd` for WebLogic 7.0:

**Caution**

This procedure is required by a defect in WebLogic that does not correctly pass classpath settings from the domain to the master when the master script is called. Consult your WebLogic administrator when editing scripts that control multiple domains.

1. Stop WebLogic Server and all application server instances. For details, see [Starting and Stopping WebLogic Server](#).
2. Navigate to the `%WL_HOME%\server\bin` subdirectory of your application server home directory.
3. Open `InstallSvc.cmd` by right clicking on its name, and selecting **Edit**.
4. Set your eaDirect home directory and call your eaDirect environment script. For example:

```
@set EDX_HOME=C:\EDCSbd
@call %EDX_HOME%\config\edx.config.bat
```
5. Save and close `InstallSvc.cmd`.

Passing Your Configuration in a Startup Script for WebLogic 7.0

You can also choose to start WebLogic Server directly by modifying the server startup script to source your configuration. For WebLogic 7.0, there are **two** startup scripts. WebLogic recommends that you start up the server from your domain, using `%WL_HOME%\config\mydomain\startWebLogic.cmd`. This script calls the master startup script `%WL_HOME%\server\bin\startWLS.cmd`.

Before editing either of these files, be sure to save a backup copy **in a different directory**.

Example startWebLogic.cmd for WebLogic 7.0

```
echo off
SETLOCAL
cd C:\bea\weblogic700\config\mydomain

@rem Set SERVER_NAME to the name of the server you wish to start up.
set SERVER_NAME=myserver

....

@rem Set MEM_ARGS to the memory args you want to pass to java. For
instance:
@rem set MEM_ARGS=-Xms32m -Xmx200m
set MEM_ARGS="-ms128m -mx128m -Xss1m -noclassgc"

@rem set edocs home directory and source configuration
@set EDX_HOME=C:\EDCSbd
@call %EDX_HOME%\config\edx.config.bat

@rem Call Weblogic Server
call "C:\bea\weblogic700\server\bin\startWLS.cmd"

ENDLOCAL
```

To edit startWebLogic.cmd for WebLogic 7.0:

See the example above for default settings. Make sure to change these as needed for your environment.

1. Stop WebLogic Server and all application server instances. For details, see Starting and Stopping WebLogic Server.
2. Edit `%WL_HOME%\config\mydomain\startWebLogic.cmd`.
3. **DOMAIN:** Change the default domain to the directory of the domain you wish to use, if required. For example, `%WL_HOME%\config\mydomain`. If you used the Domain Configuration Wizard, this variable should be preset correctly.
4. **SERVER_NAME:** Set `SERVER_NAME` to the name of the server you wish to start up. The default is `myserver`.
5. **Optimize JVM Memory** by increasing the memory arguments allocated to the Java Virtual Machine (JVM) on the application server. For example (quotes are optional):

```
set MEM_ARGS="-ms128m -mx128m -Xss1m -noclassgc"
```

6. Set your edocs home directory `%EDX_HOME%` and call `edx.config.bat` just after the set CLASSPATH statement. For example:

```
@set EDX_HOME=C:\EDCSbd
@call %EDX_HOME%\config\edx_env.bat
```

7. Save and close `startWebLogic.cmd`.

To edit startWLS.cmd for WebLogic 7.0:

Caution

This procedure is required by a defect in WebLogic that does not correctly pass classpath settings from the domain to the master when the master script is called. Consult your WebLogic administrator when editing scripts that control multiple domains.

1. Stop WebLogic Server and all application server instances. For details, see [Starting and Stopping WebLogic Server](#).

2. Edit `%WL_HOME%\server\bin\StartWLS.cmd`.

3. Add the classpath parameter to the java command so it uses the paths to eaDirect classes that are set by `startWebLogic.cmd`. For example:

```
"%JAVA_HOME%\bin\java" %JAVA_OPTIONS% -classpath "%CLASSPATH%" -
Dweblogic.Domain=%WL_DOMAIN% -Dweblogic.Name=myserver "-
Dbea.home=C:\bea" -Dweblogic.management.password=%WLS_PW%
-Dweblogic.ProductionModeEnabled=%STARTMODE%
"-Djava.security.policy=C:\bea\wlserver6.1\lib\weblogic.policy"
weblogic.Server
```

4. Save and close `StartWLS.cmd`.

Windows Services for eaDirect

Setting Up a WebLogic Server Instance as a Windows Service

If you want a WebLogic Server instance to start automatically when you boot a Windows host computer, you can set up the server as a Windows service.

Tip

If you are upgrading eaDirect from a version before 3.x, you will need to uninstall and then reinstall the WebLogic Service to include the new classpath (`edx_common.jar`). You should also delete the Logger service if you had installed it, as eaDirect 3.x and higher does not use a Logger for WebLogic.

For detailed instructions on setting up WebLogic Server 6.1 as a Windows Service, see:

<http://e-docs.bea.com/wls/docs61/adminguide/startstop.html#1043169>

For detailed instructions on setting up WebLogic Server 7.0 as a Windows Service, see:

<http://e-docs.bea.com/wls/docs70/adminguide/startstop.html#1117614>

Setting Up the eaDirect Scheduler as a Windows Service

After all eaSuite EAR files have been deployed to the application server **and** WebLogic is running, you must start the eaDirect Scheduler in order to schedule and run jobs in the eaDirect Command Center. If you attempt to run a new job with the Scheduler not running, the job will not run and you will see ‘Not yet started’ as its status.

To install the Scheduler as a Windows Service, you must modify the Scheduler template file **SCH.txt**, installed to the **bin** directory for eaDirect.

To install the Scheduler as a Windows Service:

1. Navigate to the **bin** directory for eaDirect, or **%EDX_HOME%\bin**.
2. Open the Scheduler template file **SCH.txt** and modify the Java classpath to reflect your active Java environment. For example:

```
classpath=c:\jdk131\lib\tools.jar;c:\bea\wlserver\lib\weblogic.jar;c:\EDCSbd\lib\edx_client.jar;c:\EDCSbd\lib\edx_common.jar
```



Make sure you set all paths to the appropriate point releases/patches for WebLogic Server and JDK, if necessary. Check the Release Notes and your system documentation for updated requirements.

3. Confirm that the following line of code is present in the file for your host and port:

```
-Djava.naming.provider.url=t3://localhost:7001
```
4. If you want the Scheduler to log information to a file rather than to the console, add the following value in **SCH.txt**:

```
-Dcom.edocs.pwc.debug=true <scheduler_logfile>
```
5. Confirm that all the directory references in **SCH.txt** are correct.
6. Save and close **SCH.txt**.



If you have a 2.x or earlier version of the Scheduler installed as a Windows service, you must remove it before installing the new version of the Scheduler.

7. Open a command window, and then change directory to **%EDX_HOME%\bin**. Use the **install** command to install the Scheduler as a Windows Service, for example:

```
C:\> Schedulervc -install C:\EDCSbd\bin\SCH.txt
```
8. If the Scheduler service is installed successfully, a confirmation message appears.

Troubleshooting Tips for WebLogic on Windows

If the WebLogic service is unable to find *edx_load.config.bat* at startup:

Add the path `-extrapath:C:\bea\wlserver\bin;%PATH%` to `installNtService.cmd` (substituting your drive where necessary):

```
"C:\bea\wlserver\bin\beasvc" -install -svcname:myserver
-javahome:"%JAVA_HOME%" -execdir:"C:\bea\wlserver"
-extrapath:"C:\bea\wlserver\bin;%PATH%" -cmdline:%CMDLINE%
-password:lovelyday
```

If you encounter problems when running customized Web pages:

1. From your domain in the WebLogic Server Console, select **Servers** and **myserver**. A tabbed dialog with the name of your server appears in the right pane.
2. On the Configuration/Compilers tab, change the Java Compiler value from the default `javac` to the location of `javac` in the JDK installed **with WebLogic Server**. See the [Quick Reference Appendix](#) for JDK requirements on your platform.
3. Restart WebLogic Server.



Make sure you set all paths to the appropriate point releases/patches for WebLogic Server and JDK, if necessary. Check the Release Notes and your system documentation for updated requirements.

4

Configuring Java Resources for WebLogic for UNIX and Windows

Overview

This chapter assumes in-depth understanding of and practical experience with [Application Server Administration](#). It is designed for experienced WebLogic administrators and primarily presents only the steps and settings specific to eaDirect.

See WebLogic Server documentation at <http://bea.com> for detailed step-by-step instructions on Java resource configuration, performance, and tuning. You must also consult your application server administrator for settings that may be specific to your configuration.

You must start your WebLogic Server instance and bring up the Administrative Console before you begin this chapter.



Caution

If you cannot bring up the WebLogic Console, you will be unable to proceed with configuring your application server for eaDirect.

Configuring Java Database Connectivity (JDBC) for eaDirect



Tip

This section applies to all platforms.

After you have successfully configured the eaDirect database, you must configure Java Database Connectivity (JDBC) resources on the eaDirect application server. JDBC Connections on the application server support data retrieval from relational databases and other data sources.

JDBC properties vary by both application server and database server. Use the [Quick Reference Appendix](#) to select the properties for your platform:

- [WebLogic 6.1 and 7.0 for SQL Server \(Windows\)](#)

- [WebLogic 6.1 and 7.0 for Oracle \(Solaris and HP-UX\)](#)

About JDBC Connections for eaDirect

JDBC connection pools contain named groups of JDBC Connections that are created when the connection pool is registered, usually when starting up WebLogic Server. WebLogic Server opens JDBC Connections to the database during startup and adds these connections to the pool. A J2EE web application borrows a connection from the pool, uses it, and then returns it to the pool by closing it.

JDBC transaction data sources enable JDBC clients to obtain a connection to a Database Management System (DBMS). Each data source points to the value specified for the Name attribute when a JDBC connection pool was configured.

eaDirect requires three sets of **JDBC Connection Pools** and related **JDBC Transaction (Tx) Data Sources**:

- **edxAdmin** supports the Command Center through the eaDirect web application
- **edxLogger** supports eaDirect logging through the eaDirect web application
- **edxUser** supports user data retrieval through custom web applications

For more details on configuring JDBC Connections, please see the JDBC documentation for your application and database servers.



Tip

edxAdmin connection pools support concurrency for scheduling multiple jobs. Tuning **edxAdmin** connection capacity and threads can improve eaDirect email performance.

Configuring JDBC Connections for WebLogic

You will create three sets of JDBC connection pools and three sets of transaction datasources. Their names are specific to edocs across all platforms, but **JDBC properties vary by both application server and database server**. Use the [Quick Reference Appendix](#) to select the properties for your platform:

- [WebLogic 6.1 and 7.0 for SQL Server \(Windows\)](#)
- [WebLogic 6.1 and 7.0 for Oracle \(Solaris and HP-UX\)](#)



Tip

Connection pool size increased in 3.4 and later versions to support concurrency for multiple jobs.

You will be entering the same information six times: one connection pool and one Tx data source each for **Admin**, **User**, and **Logger**. Make sure you have chosen the correct properties for your application server and database server, and that each datasource and its properties maps to the connection pool of the same name.



Caution

Make sure you are using the correct properties for your combination of application server, database, and JDBC resource.

For more details on how to configure JDBC connections, see WebLogic Server documentation at <http://bea.com>.

To configure JDBC Connections for WebLogic:

1. Create a JDBC Connection Pool each for **edxAdmin**, **edxLogger**, and **edxUser**. Use the appropriate JDBC values for your database server.
2. Create a JDBC Transaction (Tx) Data Source each for **edxAdmin**, **edxLogger**, and **edxUser**. Use the appropriate JDBC values for your database server.
3. Remember to move the Targets-Server from **Available** to **Chosen** for each data source as you configure it.
4. Review your connections. Each data source should target the connection pool of the same name (**Admin**, **User**, or **Logger**).
5. When you are finished, proceed to [Configuring Java Messaging Services \(JMS\) for eaDirect](#).

Configuring Java Messaging Services (JMS) for eaDirect

After you have successfully configured JDBC Connections, you will now configure Java Messaging Services (JMS) on the application server for eaDirect. eaDirect requires three sets of JMS resources:

- **edxAnnotation** supports Line Item Dispute and Annotation features
- **edxDispute** supports Line Item Dispute and Annotation features
- **edxLogger** supports eaDirect logging through the eaDirect web application
edxLogger requires **five** JMS consumers and session pools.



Tip

If your web application does not implement Line Item Dispute and Annotation, you need only configure JMS resources for **edxLogger**.

Use the [Quick Reference Appendix JMS Resources](#) settings for WebLogic Server to configure each set of JMS resources.

About JMS Resources for eaDirect

JMS enable web application components to asynchronously send and receive messages.

- **JMS Connection Factories** are data objects that enable Java Messaging Service (JMS) clients to create JMS connections. You define and configure one or more connection factories to create connections with predefined attributes. WebLogic Server adds the connection factories to the JNDI space during startup, and each J2EE web application retrieves a connection factory using the JNDI on the application server.
- **JMS Stores** store persistent messages in a database accessed through a designated JDBC connection pool. The JMS database can be any database that is accessible through a WebLogic-supported JDBC driver. When creating a JMS Store, you must define the name of the **backing store**, and the **JDBC connection pool** and **database table name prefix** for use with multiple instances
- **JMS Servers** manage connections and message requests on behalf of clients.
- **JMS Topics** can be one of two destinations that you can configure for a JMS server. The other destination is a JMS queue. WebLogic Server allows you to configure one or more destinations for the JMS server. You can configure destinations explicitly or with a **destination template** (useful for multiple destinations with similar attribute values).
- **JMS Session Pools** allow a JMS listener (called a **Consumer** in WebLogic) to have multiple threads that will improve performance under heavy load. Each JMS consumer requires its own session pool.

Configuring JMS Resources for WebLogic

You will be entering very similar information many times: one set of JMS resources each for annotation and dispute, and FIVE sets for logging. Make sure you have chosen the correct properties for the resource you are creating, and that each resource maps to others of the **same name**.

Use the JMS settings in the [Quick Reference Appendix](#) to configure JMS settings for WebLogic Server. For general information about configuring Java resources for WebLogic, see WebLogic Server documentation at <http://bea.com>.



Tip

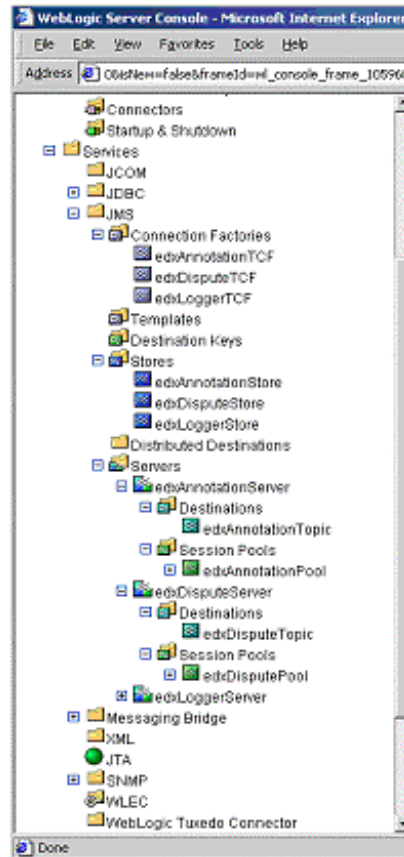
If your web application does not use Line Item Dispute and Annotation, you need only configure JMS for edxLogger.

To configure JMS for WebLogic:

1. Create one JMS Connection Factory each for **edxAnnotation**, **edxDispute**, and **edxLogger**.
2. Create one JMS Store each for each for **edxAnnotation**, **edxDispute**, and **edxLogger**.
3. Create one JMS Server each for each for **edxAnnotation**, **edxDispute**, and **edxLogger**.

4. Create one JMS Topic each for the JMS servers of each for **edxAnnotation**, **edxDispute**, and **edxLogger**.
5. Create one JMS Session Pool each for **edxAnnotation** and **edxDispute**.
6. Create one JMS Consumer each for **edxAnnotation** and **edxDispute**.
7. Create FIVE session pools and consumers for **edxLogger** as listed.

When you have finished, your administrative console should look like this.



Starting the Scheduler



This section applies to all platforms.

You may start the Scheduler from a command line, or as a Windows service. Use the appropriate procedure for your operating system. Procedures are the same for WebLogic 6.1 and 7.0, though your settings may vary from the examples.

To start the eaDirect Scheduler from the UNIX command line:

1. Switch user to the application server owner.

```
su - edxadmin
```
2. Change directory to the `bin` directory of your eaDirect installation, `$EDX_HOME/bin`.
3. Run the Scheduler command for WebLogic Server (`wl_scheduler`), host, and port. This example shows the Scheduler command for WebLogic:

```
./ wl_scheduler -start -url t3://localhost:7001
```



This example uses the default port for the administration server. You can specify another port number if necessary, but it must match the port for the admin server.

4. You can stop the Scheduler by replacing the `-start` parameter with the `-stop` parameter.

To start the Scheduler from a Windows command prompt:

1. Open a command prompt window and change directory to the `bin` directory of your eaDirect installation, `%EDX_HOME%\bin`.
2. Run the Scheduler command for WebLogic Server (`wl_scheduler`), host, and port. This example shows the Scheduler command for WebLogic:

```
C:\> wl_scheduler -start -url t3://localhost:7001
```
3. If the Scheduler starts successfully, a start-up message with the name of the log file appears in the command prompt window. **Do not close this window while eaDirect is running**, as closing it will stop the Scheduler. A log file is created in `%EDX_HOME%\Logs`.
4. You can stop the Scheduler by replacing the `-start` parameter with the `-stop` parameter, or simply by closing the command prompt window.

To start the eaDirect Scheduler as a Windows Service:

1. Install the Scheduler as a [Windows Service](#) if you have not already done so.
2. Start WebLogic Server if it is not already running.
3. From the Start menu, select **Settings>Control Panel**.
4. Double-click **Administrative Tools**, then double-click **Services**.
5. Right-click **Scheduler Service** and select **Start**. You may also click the Start icon.

	Routing and Remot...	Offers routing services to businesses in local area an...		Disabled	LocalSystem
	RunAs Service	Enables starting processes under alternate credentials	Started	Automatic	LocalSystem
	Scheduler Service	eDocs Scheduler service	Started	Manual	LocalSystem

6. To start the Scheduler automatically at startup, right-click the service and select **Properties**. From the Startup Type menu, select **Automatic**.



If you receive error messages when starting up your application server after migration, check to make sure you have undeployed all old versions of edocs J2EE web applications and redeployed the new versions. edocs does not recommend manually editing the `config.xml` files for deployment.

Where to Go From Here

Once WebLogic Server is configured successfully, you may proceed to deploying the eaDirect J2EE applications that power the eaSuite. This requires customizing each web application's deployment code for your environment and platform. For details, see WebLogic Server documentation at <http://bea.com> and *Deploying and Customizing J2EE Applications*.

5

Configuring WebSphere Server for Unix

Overview

This chapter assumes in-depth understanding of and practical experience with WebSphere Enterprise Server. Consult your WebSphere documentation as necessary.

edocs recommends that you install and configure eaDirect in the same top-level directory structure, first on the database server, then the application server.

If you have not already installed database server components and configured the database server for eaDirect, do so now.



Tip

For distributed environments, ensure that you have any required **database client software** installed on your application server and any other client machines of your database server. See the [Quick Reference Appendix](#) for software requirements for your platform.



Caution

The installation and configuration examples shown in this guide use default eaDirect pathnames, privileges, and permissions. If you choose not to accept the default values, make sure your values are consistent on all servers across your installation of eaDirect.

UNIX Permissions for WebSphere Server



Tip

This section applies to all UNIX platforms.

Application servers running eaDirect will not function correctly without access to eaDirect config files, storage directories, and related resources. When installing eaDirect on WebLogic Server, you were asked to specify the owner and group permissions (**userid:groupid**) of the WebSphere application server. If you made a mistake during installation, you must change the owner and group permissions for the directory, including all subfolders, to the application server owner.

This guide uses the example username and password **edxadmin:edxadmin** as the application server owner and group for the eaSuite.

DEFAULT	EXAMPLE	CUSTOM
Specified during installation.	<code>edxadmin:edxadmin</code>	

For details on owner and group permissions for WebSphere, please consult the documentation for your platform.

If your application server administrator uses custom user and group permissions, you can reset these permissions with the `chown` command.

To reset user and group permissions:

1. Switch user to the default owner of your application server home directory.
2. Recursively change the user and group permissions of the application server installation directory and all subdirectories to the application server owner.

```
chown -R edxadmin:edxadmin /opt/WebSphere
```
3. Switch user to your application server owner and configure your application server with your new owner.



Tip

You should also verify the owner information in any profile files used by the database server owner and application server owner. See your server documentation for details.

Starting and Stopping WebSphere

Developers and system administrators will need to be familiar with how to stop and start your WebSphere server and any active web applications for your platform.

For details on starting and stopping your WebSphere server and for verifying startup, please consult the server documentation for your platform.

Starting and Stopping an Active Application Server

Improperly starting or stopping an application server in an active eaDirect production environment can produce unexpected and unintended results. You can create custom startup and shutdown scripts that include all your command parameters, as well as the command used to start or stop the Scheduler.



Caution

WebSphere 4 users must first create an application server instance for eaDirect. See your server documentation for details.



The default command-line startup shell scripts are fine for an inactive production environment where there are no running jobs. However, the startup process will stop immediately if you enter a **Ctrl+C** (often used to force a hard shutdown of the server) in the startup directory, or if you close the terminal session.

UNIX Environment Variables for WebSphere



This section applies to all UNIX platforms.

About UNIX Config Files for Your eaDirect Environment

eaDirect installs several configuration files that you use to define your eaDirect environment.

- **edx_config**: This shell script, `$EDX_HOME/bin/edx_config`, prompts you to specify environment variables for your application server.
- **edx_env**: This configuration file, `$EDX_HOME/config/edx_env`, stores the environment variables you specify in `edx_config`, to pass at application server startup.
- **edx.config**: This script file, `$EDX_HOME/config/edx.config`, passes the environment data in `edx_env` to your application server through your startup script.

Setting UNIX Environment Data with *edx_config*

`edx_config` prompts you to enter values for your Java and database installation, including absolute directory pathnames or user identification information. **You should run this script any time you need to modify your eaDirect environment.**

If you have not already done so, verify that the ownership of the `$EDX_HOME` directory is set to the user and group of the application server owner. If not, change it before running `edx_config`.

Use the [Quick Reference Appendix](#) to enter values for your platform for each of the specified parameters. You may want to print the *Environment Variables* sections for easy reference. You can accept the default values, if appropriate, or enter your own.



Be sure the time zone (TZ) for your server is set to your system time zone. eaDirect jobs can fail if Java system time does not match actual system time.

To set environment data with `edx_config`:

1. Switch user to the application server owner, in this example `edxadmin`.
`su - edxadmin`
2. Navigate to the `bin` directory for eaDirect on your application server, for example
`cd $EDX_HOME/bin`
3. Run the script `edx_config`.
`./edx_config`
4. Enter values as prompted by the script, using the [Quick Reference Appendix](#).



Make sure that the database values you enter in this session are the same values specified during database configuration. Consult your DBA for any custom settings specific to your platform.

Passing UNIX Environment Data to WebSphere



This section applies to all UNIX platforms supported by WebSphere.

To pass your eaDirect environment to your application server, you must edit your application server startup script to:

1. Set values for your eaDirect home, application server home, and Java home directories.
2. Call and process the configuration script `edx.config`. This procedure is called **sourcing** your configuration.



The code examples in the following section are for **WebSphere on Solaris**. Use these procedures with the environment variables for your application server and platform from the [Quick Reference Appendix](#).

To pass your eaDirect environment to WebSphere:

1. Switch user to your application server owner, for example `edxadmin`.
`su - edxadmin`
2. Stop your application server.
3. Change directory to the location of your application server startup script and open the file for editing. For example,

For **WebSphere 4**:

```
cd $WAS_HOME/bin
vi startupServer.sh
```

For WebSphere 5:

```
cd $WAS_HOME/bin
vi startServer.sh
```

4. Declare and initialize the variable `$EDX_HOME` near the beginning of the file with other variable declarations. For example:

```
binDir=`dirname $0`
. $binDir/setupCmdLine.sh

EDX_HOME=/export/home/EDCSbd/
export EDX_HOME
```

For **WebSphere 4**, you must also add the following lines:

```
WAS_HOME=/opt/WebSphere/AppServer/
export WAS_HOME
```

5. In the same file, source `edx.config` just before the command to start the JVM. The dot and space preceding the pathname are a required part of the syntax. For example:


```
. $EDX_HOME/config/edx.config
```
6. Save and close the file.

To edit *launchClient.sh* for eaDirect on WebSphere 4:

You must also define eaDirect to WebSphere 4 by editing *launchClient.sh* in `$WAS_HOME/bin` to add the line `$EDX_OPTS \` to the section following the line `"$JAVA_HOME/bin/java \"`. For example:

Bold indicates text that you should add.

```
$JAVA_HOME/bin/java \
  $WAS_JAVAOPTS \
  $CLIENTSAS \
  $EDX_OPTS \
  -Dserver.root=$WAS_HOME \
  -Dws.ext.dirs=$WAS_EXT_DIRS \
  -Dcom.ibm.CORBA.BootstrapHost=$COMPUTERNAME \
  -Djava.naming.factory.initial=$NAMING_FACTORY \
  -classpath $WAS_CLASSPATH com.ibm.ws.bootstrap.WSLauncher \
  com.ibm.websphere.client.applicationclient.launchClient "$@"
```

Start the Server:

When you have finished the steps to pass your eaDirect environment to WebSphere, restart the server. For example:

```
./startServer.sh server1
```

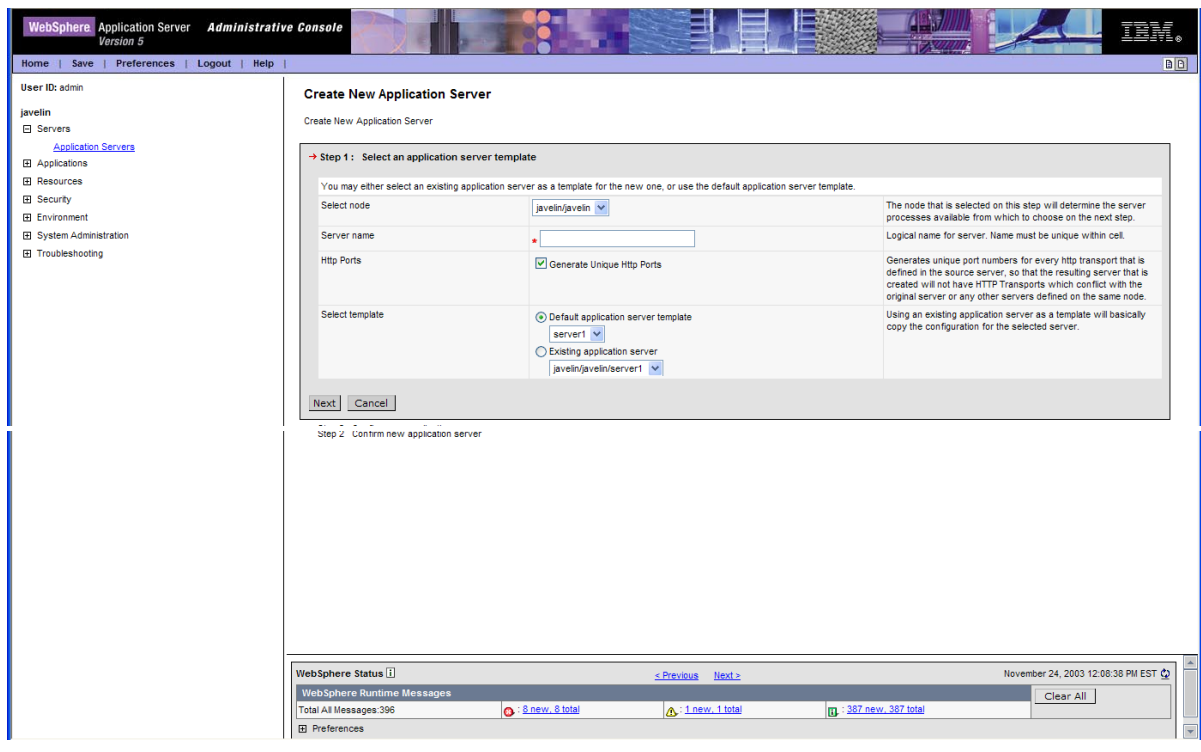
Preparing WebSphere Server for Unix

Creating An Application Server Instance For WebSphere

As a convenience, WebSphere provides a default application server instance that you can use immediately to deploy and install a J2EE application.

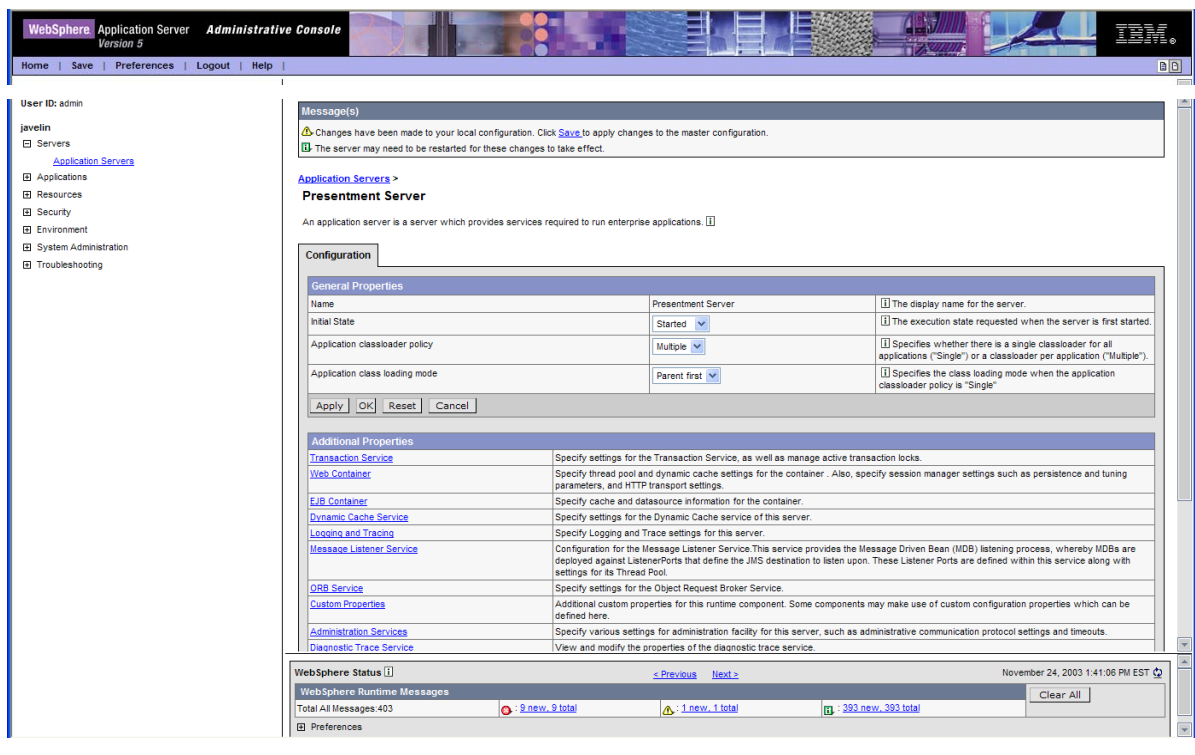
You may choose to create a separate application server instance for a new J2EE application (EAR). For example, you may choose to deploy eaDirect on the default instance, and the eaSample web application on a new instance. See the WebSphere documentation for more information on deployment. To create an application server, follow these steps:

1. Open the WebSphere Administrative Console.
2. Expand the Servers entry by clicking on the + sign, then click on the Application Servers hyperlink.
3. Click on **New**. You will be presented with the following screen:

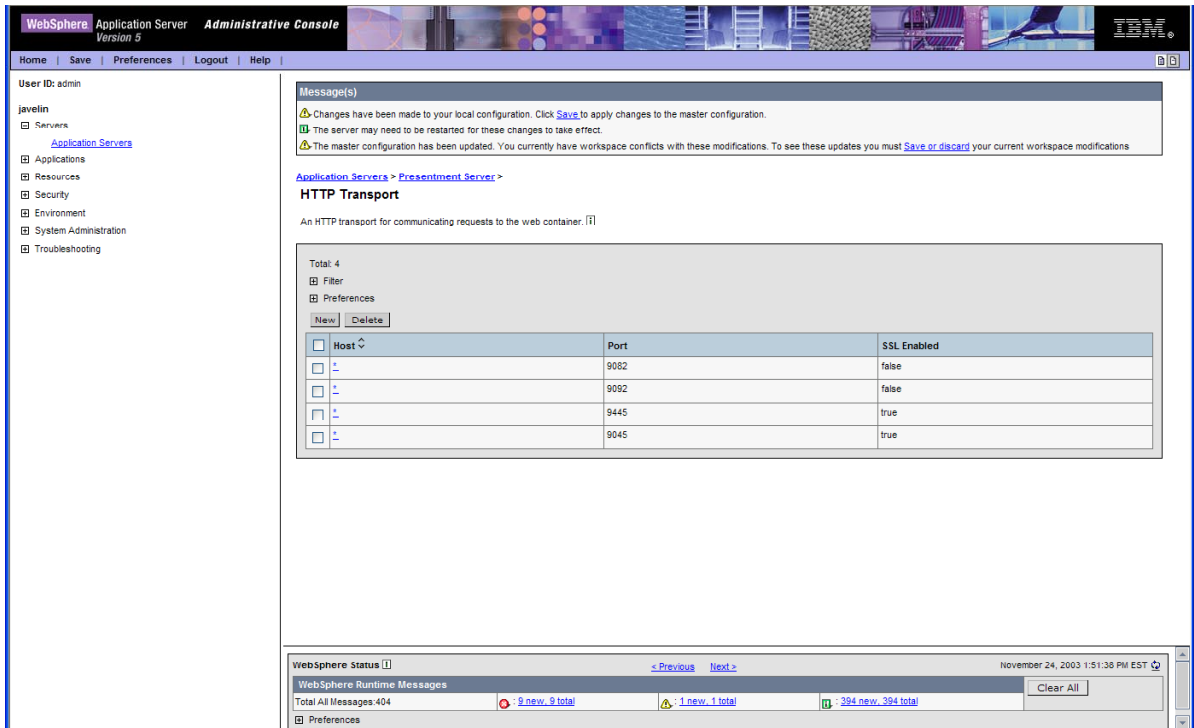


4. Select the Cell / Node combination you wish to create the new server on.
5. Choose a name for the new server.

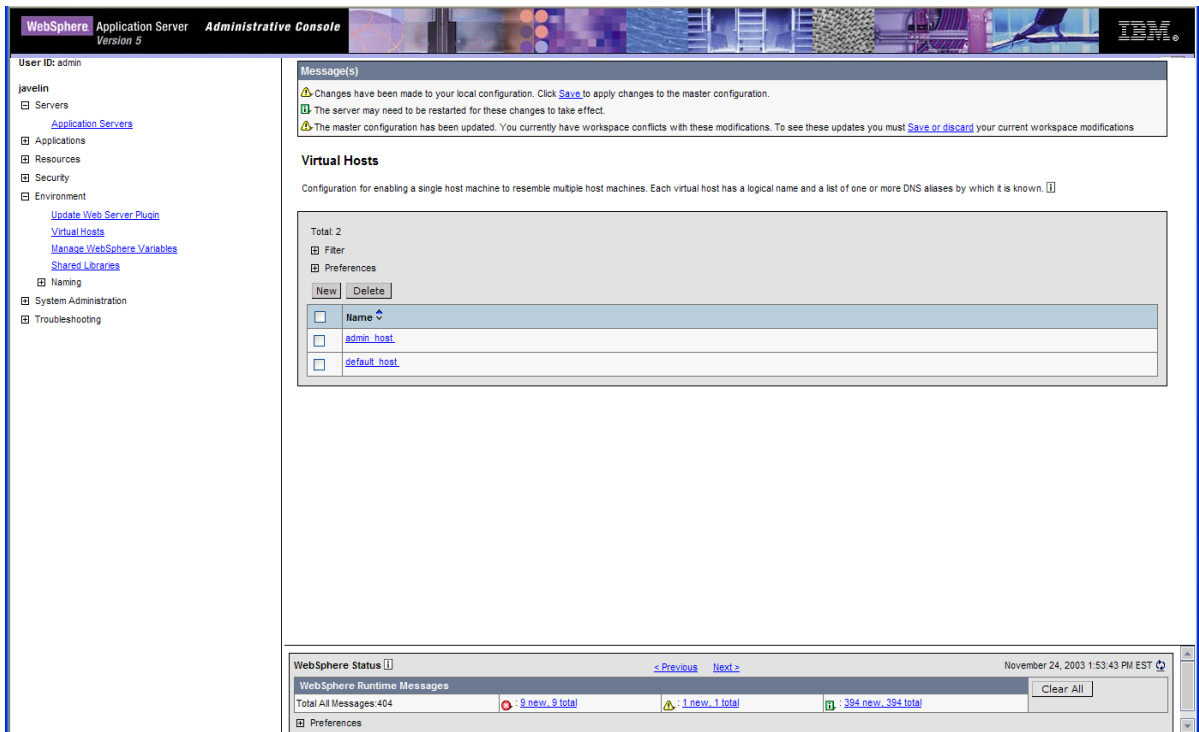
6. Pick a template for the new server, using either server1 as the template or another server you have created. Click the **Next** button. You will be asked to confirm the creation of the new application server. Click **Finish** to confirm the creation of your new server. Save the Master Configuration to preserve the changes you just made.
7. You now need to place your server on a virtual host and open ports so users can access the server. WebSphere recommends you leave the admin_host for server1 and place new servers on another virtual host. The example below will use the default_host.
8. From the Application Servers window, click on the new server you created. You will be presented with the following page:



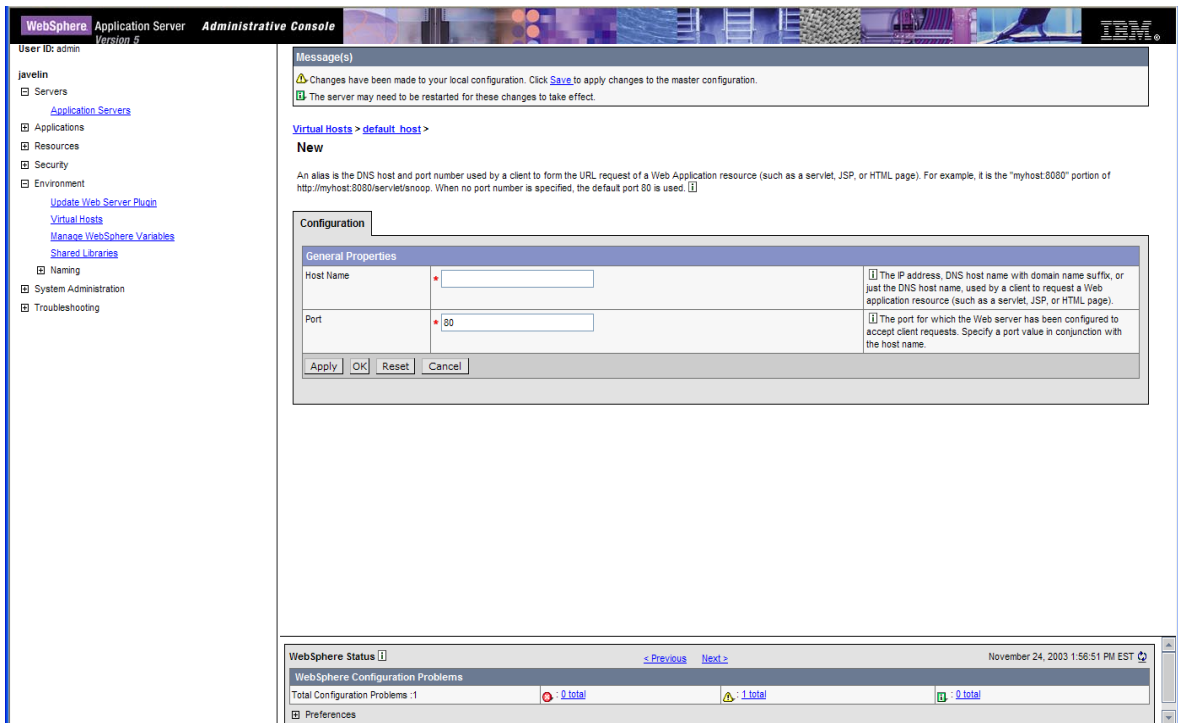
9. You need to find out the ports your new server is running on. Click the Web Container and then the Http Transports links. You will be taken to the following screen:



- The above screenshot shows that ports 9082 and 9092 were opened for http, and ports 9445 and 9045 were opened for https. You now need to open those ports up on the virtual you are placing this server on. To accomplish this, expand Environment in the left pane and choose Virtual Hosts. Your screen will now look like this:



11. Click on default_host to add the new ports. Click on Host Aliases to see what ports are open for the virtual host. On that page, you can click on **New** to add an alias. For example:



12. You may enter an asterisk (*) to allow all traffic on the port you are creating, or you may restrict to an IP address or DNS name. If you are restricting to a particular web server, it is a good idea to explicitly state via IP, DNS, or FQDN; so to accept requests from myserver on port 9082 you could have one of the following 3 entries:

- 192.168.1.100:9081 (for IP)
- myserver:9081 (for DNS)
- myserver.mydomain.com:9081 (for Fully Qualified Domain Name)

13. After you have entered the server and port, click **Apply** then **OK**. After you have finished configuring the virtual host, be sure to click **save** to keep your changes.

Java Virtual Machine (JVM) Settings for WebSphere 5

To configure JVM settings for WebSphere 5:

In the WebSphere Administrative console, click on **Servers**, then **Application Servers**, then the server for eaDirect, then **Process Definition**, then **Java Virtual Machine**.

On the General Properties page:

Property Name	Value
Classpath	Path to <code>EDX_HOME/lib/edx_system.jar</code> . For example: <code>/usr/EDCSbd/lib/edx_system.jar</code>

Under Additional Properties, click on **Custom Properties** and create:

Property Name	Value
<code>edx.home</code>	Value of <code>EDX_HOME</code> , for example: <code>/usr/EDCSbd</code>
<code>java.protocol.handler.pkgs</code>	<code>com.edocs.protocol</code>
<code>com.edocs.tasks.loader.alias</code>	The name of the database. For example: <code>edx0</code>
<code>com.edocs.tasks.loader.password</code>	The password of the database user.
<code>com.edocs.tasks.loader.user</code>	The name of the database user. For example: <code>edx_dba</code>
<code>com.edocs.fs.logging.pool-size</code>	Set this value lower (or equal to) the size of the JMS pool. This limits LogPublisher connections (and eventually the number of JMS connections opened) to improve performance.

After applying the new settings, **make sure you:**

1. Save the master configuration.
2. Restart the server.

Java Virtual Machine (JVM) Settings for WebSphere 4

WebSphere administrators must configure the Java Virtual Machine (JVM) to define your server node, name, and port. Use the shell script `XMLConfig.sh` to configure these settings. This script writes to your configuration file `ws_config.xml` so you need not edit it directly. Your application server **must be running** when you run `XMLConfig.sh`.

SQL Loader system properties are defined in the file `EDX_HOME/config/edx_sqldr.config`. If you customize these system properties, you will have to manually specify the parameters for the JVM of the eaDirect application server.

When you run `edx_config`, the script asks you to enter your database user name, password, and alias. If you are using values other than the defaults, you must also set these values through JVM settings in the System Properties table.

To configure JVM settings for WebSphere 4:

1. Switch user to your application server owner, for example **edxadmin**.
`su - edxadmin`
2. Start your application server.
3. Change directory to the location of your application server startup script, for example:

```
cd $WAS_HOME/bin
```

4. Run the configuration script *XMLConfig.sh*. Use the [Quick Reference Appendix](#) to enter the values for your environment, using the following syntax:

```
XMLConfig.sh -import $EDX_HOME\config\ws_config.xml
-adminNodeName system_name -nameServiceHost host_name
-nameServicePort admin_port -substitute
"NodeName=system_name;ServerName=eaDirect_server;EDX_HOME=$EDX
_HOME;JMS_HOME=mqm_home"
```

For example:

```
./XMLConfig.sh -import /opt/EDCSbd/config/ws_config.xml
-adminNodeName javelin -nameServiceHost javelin
-nameServicePort 900 -substitute
"NodeName=localhost;ServerName=javelin;EDX_HOME=/opt/EDCSbd;JM
S_HOME=/opt/mqm/java"
```

Upon completion, you are returned to the command prompt.



In a multiple server environment, run *XMLConfig.sh* on the master node before cloning to other servers. See the IBM WebSphere documentation for information about cloning servers.

Default Custom Properties

Add the following Custom Properties using the values for your installation:

Name	Default Value
<code>com.edocs.tasks.loader.user</code>	SQL database user. Default: <code>edx_dba</code>
<code>com.edocs.tasks.loader.password</code>	SQL database user password. Default: <code>edx</code>
<code>com.edocs.tasks.loader.alias</code>	SQL server name. Default: <code>doc_frost</code>
<code>edx.home</code>	Path of %EDX_HOME%. Default: <code>/opt/EDCSbd</code>
<code>java.protocol.handler.pkgs</code>	<code>com.edocs.protocol</code>

To configure Custom Properties in WebSphere 4:

1. Select the server which supports the eaDirect application in the WebSphere Administrative Console.
2. Select the JVM Settings tab, and add or update values in the System Properties section.
3. Click **Apply**.

6

Configuring Java Resources for WebSphere for UNIX

Overview

This chapter assumes in-depth understanding of and practical experience with [Application Server Administration](#). It is designed for experienced WebSphere administrators and primarily presents only the steps and settings specific to eaDirect.

See WebSphere Server documentation for detailed step-by-step instructions on Java resource configuration, performance, and tuning. You must also consult your application server administrator for settings that may be specific to your configuration.

You must start WebSphere Server and bring up the Administrative Console (WebSphere 5) or Administrative Client (WebSphere 4) before you begin this chapter.



If you cannot bring up the WebSphere Console/Client, you will be unable to proceed with configuring your application server for eaDirect.

Configuring Java Database Connectivity (JDBC) for eaDirect

After you have successfully configured the eaDirect database, you must configure Java Database Connectivity (JDBC) resources on the eaDirect application server. JDBC Connections on the application server support data retrieval from relational databases and other data sources.

About JDBC Connections for eaDirect

JDBC connection pools contain named groups of JDBC Connections that are created when the connection pool is registered, usually when starting up your application server. Your application server opens JDBC Connections to the database during startup and adds these connections to the pool. A J2EE web application borrows a connection from the pool, uses it, and then returns it to the pool by closing it.

JDBC transaction data sources enable JDBC clients to obtain a connection to a Database Management System (DBMS). Each data source points to the value specified for the Name attribute when a JDBC connection pool was configured.

eaDirect requires three sets of **JDBC Connection Pools** and related **JDBC Transaction (Tx) Data Sources**:

- **edxAdmin** supports the Command Center through the eaDirect web application
- **edxLogger** supports eaDirect logging through the eaDirect web application
- **edxUser** supports user data retrieval through custom web applications

Use the [Quick Reference Appendix JDBC Connections](#) settings for your application server to configure each connection pool.

For details on configuring JDBC Connections, please see the JDBC documentation for your application and database servers.



edxAdmin connection pools support concurrency for scheduling multiple jobs. Tuning **edxAdmin** connection capacity and threads can improve eaDirect email performance.

For details on how to create and configure JDBC Connections, please see your application server documentation.

You will create three sets of JDBC connection pools and three sets of transaction datasources. Their names are specific to edocs across all platforms, but **JDBC properties vary by database server**. Use the [Quick Reference Appendix](#) to select the application server properties for your combination of application server and database server properties (Oracle, DB2 or SQL Server).



Make sure you are using the correct properties for your application server and database combination.

Configuring the JDBC Driver for WebSphere 4

Edit the following files to configure the JDBC driver for WebSphere version 4. Editing these files is not required in WebSphere 5.

To edit admin.config:

Edit `$WAS_HOME/bin/admin.config`, and add the following entries:

```
com.ibm.ejs.sm.adminServer.dberrorMap=EC2627=com.ibm.ejs.cm.po
rtability.DuplicateKeyException

com.ibm.ejs.sm.adminServer.dbdataSourceClassName=com.ibm.websp
here.jdbcx.sqlserver.SQLServerDataSource

com.ibm.ejs.sm.adminServer.dbportNumber=1433

com.ibm.ejs.sm.adminServer.dbselectMethod=cursor
```

To edit setupCmdLine:

Edit `$WAS_HOME/bin/setupCmdLine.sh`, and add the following entry:

```
SET
DBDRIVER_JARS=$WAS_HOME\lib\ext\base.jar;$WAS_HOME\lib\ext\spy
.jar;$WAS_HOME\lib\ext\sqlserver.jar;$WAS_HOMED\lib\ext\util.j
ar
```

Configuring JDBC Connections for WebSphere 4 and 5

JDBC properties vary by both application and database server. Use the [Quick Reference Appendix](#) to select the properties for your combination:

- WebSphere 5 for Oracle, see page 116.
- WebSphere 4 for Oracle, see page 118.
- WebSphere 5 for DB2, see page 119.
- WebSphere 4 for DB2, see page 121.
- WebSphere 4 for MSSQL, see page 123.

To configure JDBC Connections for WebSphere:

1. Create a JDBC Connection Pool each for `edxAdmin`, `edxLogger`, and `edxUser`. Use the appropriate JDBC values for your database server.
2. Create a JDBC Transaction (Tx) Data Source each for `edxAdmin`, `edxLogger`, and `edxUser`. Use the appropriate JDBC values for your database server.
3. Review your connections. Each data source should target the connection pool of the same name (`Admin`, `User`, or `Logger`).
4. When you are finished, proceed to [Configuring Java Messaging Services \(JMS\) for eaDirect](#).

Configuring Java Messaging Services (JMS) for eaDirect

After you have successfully configured JDBC Connections, you will now configure Java Messaging Services (JMS) on the application server for eaDirect. eaDirect requires three sets of JMS resources:

- `edxAnnotation` supports Line Item Dispute and Annotation features
- `edxDispute` supports Line Item Dispute and Annotation features
- `edxLogger` supports eaDirect logging through the eaDirect web application
edxLogger requires **five** JMS consumers and session pools.

**Tip**

If your web application does not implement Line Item Dispute and Annotation, you need only configure JMS resources for **edxLogger**.

Use the [Quick Reference Appendix JMS Resources](#) settings for your application server to configure each set of JMS resources.

About JMS Resources for eaDirect

JMS enable web application components to asynchronously send and receive messages.

- **JMS Connection Factories** are data objects that enable Java Messaging Service (JMS) clients to create JMS connections. You define and configure one or more connection factories to create connections with predefined attributes. Your application server adds the connection factories to the JNDI space during startup, and each J2EE web application retrieves a connection factory using the JNDI on the application server.
- **JMS Stores** store persistent messages in a database accessed through a designated JDBC connection pool. The JMS database can be any database that is accessible through a WebSphere-supported JDBC driver. When creating a JMS Store, you must define the name of the **backing store**, and the **JDBC connection pool** and **database table name prefix** for use with multiple instances
- **JMS Servers** manage connections and message requests on behalf of clients.
- **JMS Topics** can be one of two destinations that you can configure for a JMS server. The other destination is a JMS queue.
- **JMS Session Pools** allow a JMS listener to have multiple threads that will improve performance under heavy load.

Configuring JMS for WebSphere 5 for UNIX

WebSphere 5 does not require MQ series. If you have installed MQ Series the system where you plan to install WebSphere 5, **uninstall** MQ series first. For detailed information about:

Oracle on Solaris, see *JMS Resources (WebSphere 5, Oracle, Unix)* on page 117.

DB2 on AIX, see *JMS Resources (WebSphere 5, DB2, Unix)* on page 120.

After configuring JMS for WebSphere 5, be sure to **save to the master configuration**.

The next step is to deploy the application, which is described on page **Error! Bookmark not defined.**

Configuring JMS for WebSphere 4 for UNIX

Configuring WebSphere 4 for UNIX

WebSphere 4 uses the third-party tools **JMSAdmin** and **MQSeries** to configure and administer JMS. edocs JMS settings for **edxAdmin**, **edxLogger**, and **edxUser** are configured through the shell script **edx_mqm_config** installed with eaDirect.

Caution

JMS for WebSphere requires several components of third-party software. Check your eaDirect System Requirements in the release notes to ensure you have the correct versions, patches, and JDK for MQSeries on your platform.

JMSAdmin is the WebSphere Java Message Service Administration Tool to administer JMS objects such as connection factories, queues, and topics, binding them to a JNDI name space. **JMSAdmin.exe** is the command-line executable file.

MQSeries is a third-party tool to configure JMS. **JMSAdmin.config** is the default configuration file for the MQSeries Classes for JMSAdmin.

- **MQSeries 5.2 with CSD06.** For information about installing MQSeries 5.2, see [book/pdf/en_US/amqdac03](#) on the MQSeries 5.2 installation CD-ROM.
- **MA0C** (MQSeries Publish and Subscribe package version 1.0.6). For more information, see <http://www-3.ibm.com/software/ts/mqseries/txppacs/ma0c.html>.
- **MQSeries Classes for Java 5.2** (MA88 1.1.4). eaDirect does not use two phase commit, so you do not need to install that feature.

To configure JMS for WebSphere 4:

1. Configure the **JMSAdmin** executable for your eaDirect environment.
2. Configure the **JMSAdmin.config** settings for your eaDirect environment.
3. Create an MQSeries queue manager for eaDirect.
4. Verify that the MQSeries Publish/Subscribe broker is installed and running properly.
5. Configure the JMS connection factory and destinations with **edx_mqm_config**.

The following topics describe each step in detail.

Caution

These examples are for UNIX. Windows users should use the appropriate pathnames for your installation.

To configure JMSAdmin for UNIX for WebSphere 4:

1. Switch user to the MQSeries owner, in this example **edxadmin**.

```
su - edxadmin
```

2. Make the owner of your application server and eaDirect home directories a member of the `mqm` group.

3. Change directory to the Java home directory of your MQM Series installation, for example:

```
cd /opt/mqm/java/bin
```

4. Open the executable `JMSAdmin` (not `JMSAdmin.config`) in a text editor. For example:

```
vi JMSAdmin
```

5. Add the following code to the end of the file. All code following the `PATH` statement must be **on a single line**. Use the [Quick Reference Appendix](#) to edit the application server environment variables `WAS_HOME` and `MQ_JAVA` for your platform.

```
WAS_HOME=/usr/WebSphere/AppServer
MQ_JAVA_INSTALL_PATH=/usr/mqm/java
PATH=$PATH:/usr/WebSphere/AppServer/java/jre/lib/ext
export MQ_JAVA_INSTALL_PATH PATH WAS_HOME
java -DMQJMS_LOG_DIR=$MQ_JAVA_INSTALL_PATH/log
-DMQJMS_TRACE_DIR=$MQ_JAVA_INSTALL_PATH/trace
-DMQJMS_INSTALL_PATH=$MQ_JAVA_INSTALL_PATH
-Djava.ext.dirs=$MQ_JAVA_INSTALL_PATH/lib:$WAS_HOME/java/jre/l
ib/ext:$WAS_HOME/lib com.ibm.mq.jms.admin.JMSAdmin $*
```

6. Save and close the file.

To configure `JMSAdmin.config` for UNIX for WebSphere 4:

1. Switch user to the MQSeries owner, in this example `edxadmin`.

```
su - edxadmin
```

2. Change directory to the Java home directory of your MQM Series installation, for example:

```
cd /opt/mqm/java/bin
```

3. Open `JMSAdmin.config` (not the executable `JMSAdmin`) in a text editor. For example:

```
vi JMSAdmin.config
```

4. From the list of JNDI service providers, **uncomment** (remove `#`) the setting:

```
INITIAL_CONTEXT_FACTORY=com.ibm.websphere.naming.WsnInitialContextFactory
```

5. From the list of JNDI service providers, **uncomment** (remove `#`) the setting:

```
INITIAL_CONTEXT_FACTORY=com.ibm.ejs.ns.jndi.CNInitialContextFactory
```

6. From the list of URLs for JNDI service providers, **uncomment** (remove `#`) the following setting and edit it for your application server and port:

```
PROVIDER_URL=iiop://localhost:900
```

Tip

Port 900 is the default admin port for the administration server. You can specify another port number if necessary, but it must match the port for the admin server.

7. Make sure that no other settings are uncommented for a JNDI service provider and URL.
8. Save and close the file.
9. Make sure that no other settings are uncommented for a JNDI service provider and URL.
10. Save and close the file.

To create, start, and stop an MQSeries Queue Manager for UNIX:

1. Switch user to the MQSeries owner, in this example **edxadmin**.

```
su - edxadmin
```
2. Change directory to the Java home directory of your MQM Series installation, for example:

```
cd /opt/mqm/java/bin
```
3. Create a queue manager called **edxQueueManager** using the **crtmqm** command:

```
crtmqm -u SYSTEM.DEAD.LETTER.QUEUE edxQueueManager
```
4. Start the queue manager using the **strmqm** command:

```
strmqm edxQueueManager
```
5. Verify that **edxQueueManager** is running using the **dspmqm** command:

```
dspmqm -m edxQueueManager
```
6. If **edxQueueManager** is listed with a status of ***Ended***, start it using the **strmqm** command.
7. You can stop the **edxQueueManager** using the **endmqm** command.

```
endmqm edxQueueManager
```

To verify the MQSeries Publish/Subscribe broker for UNIX:

1. Switch user to the MQSeries owner, in this example **edxadmin**.

```
su - edxadmin
```
2. Change directory to the Java home directory of your MQM Series installation, for example:

```
cd /opt/mqm/java/bin
```
3. Make sure that you have the pathname **\$JMS_HOME/bin** in your **PATH**.

4. Verify that `edxQueueManager` is running using the `dspmqm` command:

```
dspmqrk -m edxQueueManager
```
5. If you receive an error message that the operating system cannot run the `dspmqrk` command, confirm that the MQSeries Publish/Subscribe broker is installed properly, and that `$JMS_HOME/bin` is included in the system PATH setting.
6. If the operating system reports that the broker is unavailable (or inactive), start it using the following command:

```
strmqbrk -m edxQueueManager
```
7. Run the following command to verify that the broker has been installed and is running:

```
dspmqrk -m edxQueueManager
```
8. If the broker is running, you should see a message similar to the following:

```
MQSeries message broker for queue manager edxQueueManager  
running
```
9. Change directory to `$JMS_HOME/java/bin`, and create the MQ JMS System queues by running the following command:

```
runmqsc edxQueueManager < MQJMS_PSQ.mqsc
```

To configure JMS topics and destinations for UNIX:

1. Start your application server if it is not already running.
2. Change directory to `$JMS_HOME/java/bin`.
3. Define and export the appropriate value for `EDX_HOME`, for example:

```
EDX_HOME=/opt/EDCSbd; export EDX_HOME
```
4. Then run the following command to configure the JMS topics:

```
$EDX_HOME/config/edx_mqm_config
```



Tip

If your MQSeries server is on a remote machine, see the IBM MQSeries documentation for the appropriate configuration.

Restarting the Application Server

When restarting the application server you should follow this order:

1. Stop the scheduler and logger.
2. Stop the MQ queues using `endmqbrk` and `endmqm`.
3. Stop the eaDirect application server.

When you are ready to start the eaDirect application server again, reverse the steps: start the application server, then start the MQ queues, then start the logger and scheduler.

Where to Go From Here

After configuring JMS for WebSphere4 , the next step is to deploy the application.

Deploying the eaDirect Application

Once your application server is configured successfully, you may proceed to deploying the eaDirect J2EE applications that power the eaSuite. This requires customizing each web application's deployment code for your environment and platform. For details, see your application server documentation and *Deploying and Customizing J2EE Applications*.



Tip

for each web application to be deployed, WebSphere users will need to enter the system classpath for edocs JAR files `edx_common.jar;edx_client.jar;edx_system.jar`. For details, see *Deploying and Customizing J2EE Applications*.

After you have deployed the eaDirect application, continue to the next section in this document to start the scheduler and logger.

Deployment Using WebSphere 5

This section describes how to deploy eaDirect using the WebSphere application server. eaDirect 4.1 supplies a pre-assembled EAR for deployment.



Caution

The specific steps for your application server may differ from the ones described below. You should consult your System Administrator and application server documentation for complete details of how to deploy a J2EE application based on your system's configuration.

To deploy a web application with WebSphere 5:

1. From the WebSphere Administrative Console, expand the Applications branch and click on **Install New Application**.
2. Select "Server Path" and then type in the absolute path to where your application is installed, for example,


```
/opt/EDCSbd/J2EEApps/websphere5/Deployed_ear-eadirect.ear
```
3. Keep clicking **Next** until you reach the last page, and then click **Finish**. Use the default settings for each step.
4. Save to the Master Configuration.

See the Deployment Guide for information about deploying additional edocs applications.

Deployment Using WebSphere 4

This section describes how to deploy eaDirect using the WebSphere 4 application server.

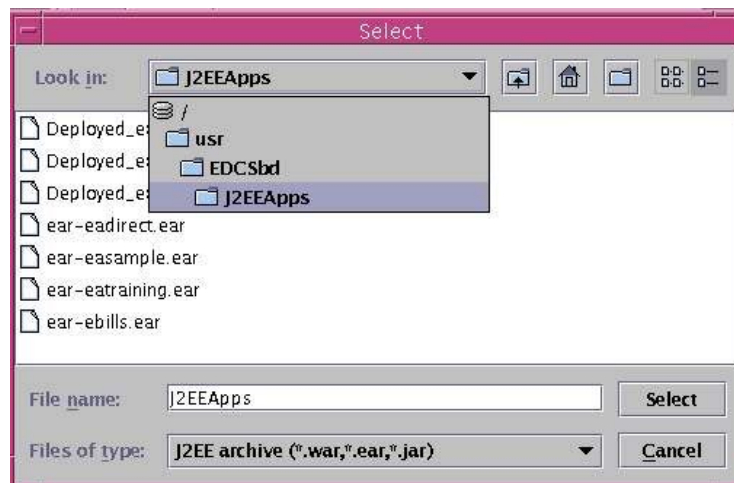


The specific steps for your application server may differ from the ones described below. You should consult your System Administrator and application server documentation for complete details of how to deploy a J2EE application based on your system's configuration.

The first step in setting up the application server is to generate the deployment code for the eaDirect application that you will deploy on WebSphere. This is done through the Application Assembly Tool.

To generate deployment code with WebSphere 4:

1. Start the Application Assembly Tool and click the **Existing** tab.
2. Click **Browse** at the bottom of the dialog, and navigate to the directory that contains the eaDirect J2EE applications. For example:



3. Select **ear-eadirect.ear** from the list of J2EE applications and click **Select**.
4. Click **OK**. The eaDirect application opens in the Application Assembly Tool.
5. In the left pane, right click on **eadirect**, and select **Generate code for deployment** from the menu.

The **Generate Code for deployment** dialog appears.

6. In the **Dependent classpath** field, enter the java classpath for the edocs system, client, and common JAR files separated by colons, for example:

```
/opt/EDCSbd/lib/edx_system.jar:/opt/EDCSbd/lib/edx_client.jar:/opt/EDCSbd/lib/edx_common.jar
```

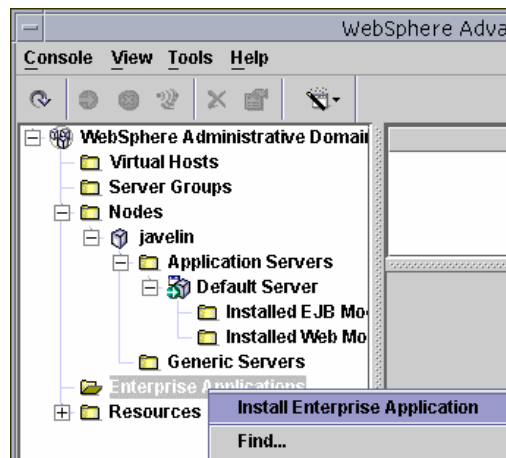
Tip

Be sure to include the leading slashes so that these classpaths are taken from the root.

7. From the Database type drop-down menu, select the database you are using, such as Oracle or DB2.
8. Click **Generate Now**. During generation of the deployment code, status information is shown in the window at the bottom of the dialog.
9. Wait for the progress bar at the bottom of the Application Assembly Tool to complete. Some applications might take several minutes to deploy, depending on the speed of your machine.
10. Click **Close** to exit from the Generate code for deployment dialog.
11. Click **Exit** in the File menu to close the Application Assembly Tool.

To deploy a web application with WebSphere 4:

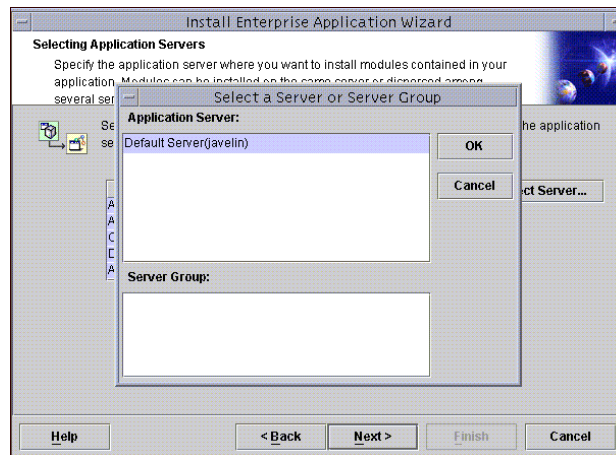
1. From the WebSphere Administrative Console, expand the WebSphere Administrative Domain view.
2. Right-click the **Enterprise Applications** folder, and then select **Install Enterprise Applications** on the menu.



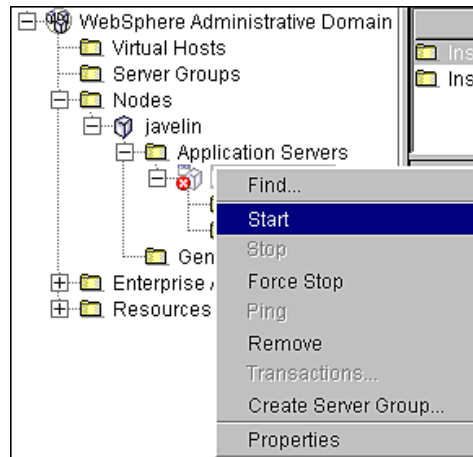
The Install Enterprise Application Wizard dialog appears.

3. Click the **Install Application** radio button, and confirm that the correct node has been chosen in the Browse for file on node field.
4. Click **Browse**. The Open dialog appears.
5. Navigate to `$EDX_HOME/J2EEApps/websphere`, and select `Deployed_ear-eadirect.ear`. The name of the file appears in the File name field.

6. Click **Open**. The Install Enterprise Application Wizard dialog appears with the name of the deployed EAR file in the Install Application (*.ear) Path field.
7. Click **Next** about nine times until you come to the Selecting Application Servers screen. Highlight all the modules for selection by clicking the first and last module in the list, while holding down the **Shift** key.
8. Click **Select Server**. The Select a Server or Server Group dialog appears.
9. Select the Default Server. In the sample screen below, the default server is named javelin.



10. Click **OK** to close the Select a Server of Server Group dialog. The Install Enterprise Application Wizard dialog lists the modules and the server on which they will be installed.
11. Click **Next** and then **Finish**.
12. When asked if you want to regenerate code, click **No**. The installation of eaDirect takes place on the application server.
13. Click **OK** to close the Information dialog reporting that the installation was successful.
14. Stop the server on which you installed the eaDirect J2EE application. Select (expand) **Nodes** and **<your_node>** and **Application Servers**, and then right-click the server name in the left pane. Select **Stop** on the menu.
15. An Information dialog notifies you that the application server has stopped successfully.
16. Click **OK** to close the Information dialog.
17. Restart the application server by right-clicking on the its name, and selecting **Start** on the menu.



18. An Information dialog notifies you that the application server has started successfully.
19. Click **OK** to close the Information dialog.
20. Start the eaDirect J2EE application by expanding the Enterprise Applications folder, right clicking on the application name and then selecting **Start** from the menu.
21. An Information dialog notifies you that the application has started successfully. Click **OK** to close the Information dialog.

Starting the Scheduler and Logger

Starting the eaDirect Logger

The eaDirect Logger generates log files for jobs being run in the Command Center. eaDirect copies all log files to **logs** directory under **EDX_HOME**. The log files are presented in text format.

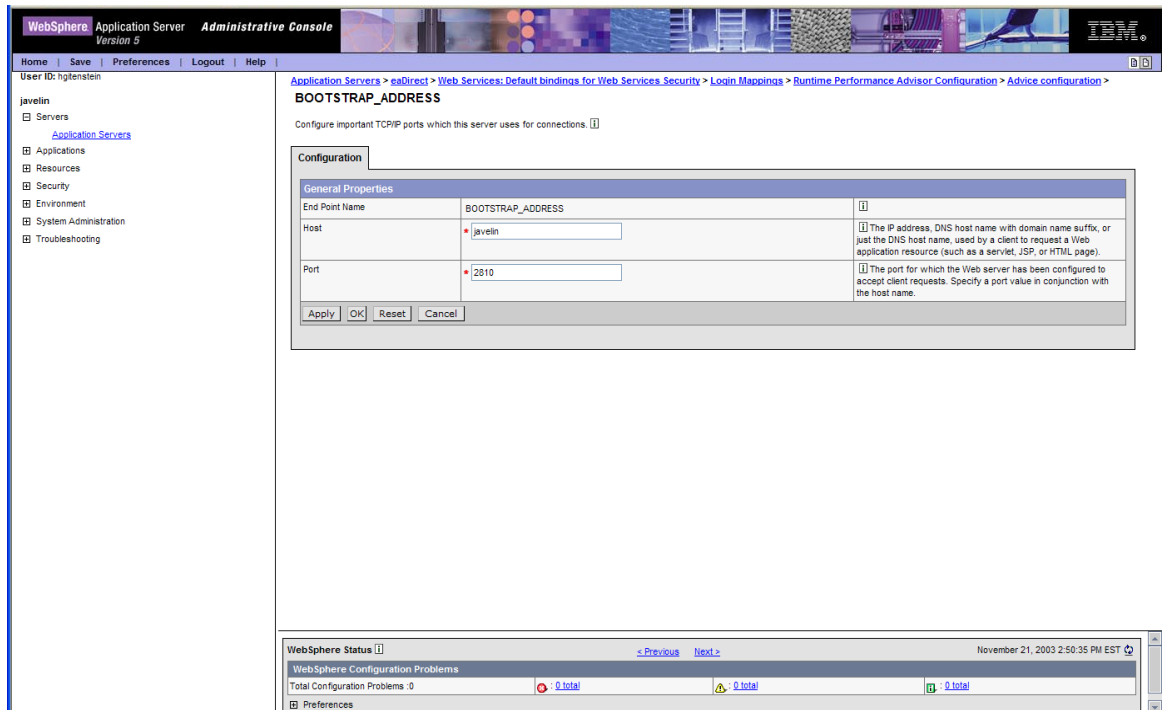
You may start the Logger from a command line, or as a Windows service.

To start the eaDirect Logger from the UNIX command line:

1. Switch user to the application server owner.
2. Change directory to the **bin** directory of your eaDirect installation, **\$EDX_HOME/bin**.
3. Run the Logger command for your application server (**ws_logger**), host, and port. This example shows the Logger command for **WebSphere 5**:

```
./ws_logger -start -url iiop://localhost:2809
```

Note that the JNDI port for your server may be found by opening the WebSphere 5 Administrative Console and navigating through the pages Application Servers > eaDirect > Web Services: Default bindings for Web Services Security > Login Mappings > Runtime Performance Advisor Configuration > Advice configuration > BOOTSTRAP_ADDRESS:



Here is an example for **WebSphere 4**:

```
./ws_logger -start -url iiop://localhost:900
```



Tip

This example uses the default port for the administration server. You can specify another port number if necessary, but it must match the port for the admin server.

4. You can stop the Logger by replacing the **-start** parameter with the **-stop** parameter.

Starting the eaDirect Scheduler

You must start the eaDirect Scheduler in order to schedule and run jobs in the eaDirect Command Center. If you attempt to run a new job with the Scheduler not running, the job will not run and you will see 'Not yet started' as its status.

The eaDirect scheduler does not need to be started until after you have deployed the EAR file.

To start the eaDirect Scheduler from the UNIX command line:

1. Switch user to the application server owner.

```
su - edxadmin
```

2. Change directory to the `bin` directory of your eaDirect installation, `$EDX_HOME/bin`.

3. Run the Scheduler command for your application server, host, and port. This example shows the Scheduler command for **WebSphere 5**:

```
./ws_scheduler -start -url iiop://localhost:2809
```

Note that the JNDI port for your server may be found by opening the WebSphere 5 Administrative Console and navigating through the pages Application Servers > eaDirect > Web Services: Default bindings for Web Services Security > Login Mappings > Runtime Performance Advisor Configuration > Advice configuration > BOOTSTRAP_ADDRESS (see the previous section for an example).

Here is an example for **WebSphere 4**:

```
./ws_scheduler -start -url iiop://localhost:900
```

Tip

These examples use the default port for the administration server. You can specify another port number if necessary, but it must match the port for the admin server.

4. You can stop the Scheduler by replacing the `-start` parameter with the `-stop` parameter.



Configuring WebSphere Server for Windows

Windows Environment Variables for WebSphere

About eaDirect Config Files for Your Windows Environment

eaDirect installs several configuration files that you use to define your eaDirect environment.

- **edx_env.bat**: This configuration file, `%EDX_HOME%\config\edx_env.bat`, stores environment variables to pass at application server startup.
- **edx_load.config.bat**: This configuration file, `%EDX_HOME%\config\edx_load.config.bat`, stores database Java options to pass at application server startup.
- **edx.config.bat**: This shell script, `%EDX_HOME%\bin\edx.config.bat`, passes the environment data in `edx_env.bat` and `edx_load.config.bat` to your application server through your startup script.

Setting Windows Environment Data

You must edit the configuration file `edx_env.bat` to set values for your eaDirect home, application server home, and Java home directories.

You must also edit the configuration file `edx_load.config.bat` to set values for your database user, password, and server name. This file is called in turn by `edx_env.bat`.

Use the [Quick Reference Appendix](#) to enter application server environment variables for your platform for each of the specified parameters. You may want to print the *Environment Variables* sections for easy reference. You can accept the default values, if appropriate, or enter your own.



The following procedure shows example variables for **WebSphere** on **Windows**. Application server administrators on other platforms should consult the [Quick Reference Appendix](#) for appropriate variables. For example, WebSphere requires `%WAS_HOME%`, and requires `%JMS_HOME%` for MQSeries.

To edit Windows environment data with `edx_env.bat`:

1. Navigate to `%EDX_HOME%\config` and open `edx_env.bat`.
2. Modify the default settings to reflect your eaDirect environment. For example:

```
@rem define APP_SERVER
@set APP_SERVER=WS
@rem define EDX_HOME
...
@rem define EDX_HOME
@set EDX_HOME=C:\EDCSbd
@rem define EDX_HOME
...
@rem define JAVA_HOME
@set JAVA_HOME=C:\WebSphere\AppServer\java
@rem define JAVA_HOME
...
@rem define JMS_HOME
@set JMS_HOME=C:\Program Files\ibm\MQSeries\Java
@rem define JMS_HOME
```



Make sure you set all paths to the appropriate point releases/patches for your application server and JDK, if necessary. Check the Release Notes and your system documentation for updated requirements.

3. Save and close the file.



The following procedure shows example variables for **WebSphere** on **Windows** Application server administrators on other platforms should consult the [Quick Reference Appendix](#) for appropriate variables.

To edit Windows Java Options with `edx_load.config.bat`:

1. Navigate to `%EDX_HOME%\config` and open `edx_load.config.bat`.
2. Modify the default settings for `com.edocs.tasks.loader` to reflect your database user, password, and server name. Use the settings for your database from the [Quick Reference Appendix](#). For example:

```
@set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.edocs.tasks.loader.user=edx_dba
@set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.edocs.tasks.loader.password=edx
@set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.edocs.tasks.loader.alias=doc_frost
```

3. Save and close the file.



Make sure you set all paths to the appropriate point releases/patches for your application server and JDK, if necessary. Check the Release Notes and your system documentation for updated requirements.

To edit Windows Java Options with `edx.config.bat`:

1. Navigate to `%EDX_HOME%\config` and open `edx.config.bat`.
2. Modify the default settings for the `_AppServer` to WebSphere For example:

```
@set _AppServer=ws
```

3. Save and close the file.

Passing Windows Environment Data to WebSphere

To pass your eaDirect environment to your application server, you must edit your application server startup scripts to set variables for eaDirect. edocs recommends starting WebSphere as a Windows Service, and modifying the script that calls that service.

To edit `adminserver.bat` for eaDirect:

Edit `adminserver.bat` (which is located in the `%WAS_HOME%` subdirectory of your application server home directory). Before editing `adminserver.bat`, be sure to save a backup copy **in a different directory**.

1. Set an environment variable for your eaDirect home. and
2. Call and process the configuration scripts `edx.config.bat`. This procedure is called **sourcing** your configuration. For example:

Bold indicates text that you should add.

```
...
call "%~dp0setupCmdLine.bat"
...
@rem edocs additions
@set EDX_HOME=C:\EDCSbd
@call %EDX_HOME%\config\edx.config.bat
...
```

Preparing WebSphere Server for Windows

Creating An Application Server Instance For WebSphere

As a convenience, WebSphere provides a default application server instance that you can use immediately to deploy and install a J2EE application.

You may choose to create a separate application server instance for a new J2EE application **EAR**. For example, you may choose to deploy eaDirect on the default instance, and the eaSample web application on a new instance. See your application server documentation for more information on deployment.

Java Virtual Machine (JVM) Settings

WebSphere administrators will need to configure the Java Virtual Machine (JVM) to define your server node, name, and port. Use the command file **XMLConfig.bat** to configure these settings. Your application server **must be running** when you run **XMLConfig.bat**.

SQL Loader system properties are defined in the file `%EDX_HOME%\config\edx_sqldr.config`. If you customize these system properties, you will must manually specify the parameters for the JVM of the eaDirect application server.

When you run `edx_config.bat`, the script asks you to enter your database user name, password, and alias. If you are using values other than the defaults, you must also set these values through JVM settings in the System Properties table.

To configure JVM settings for WebSphere 4:

1. Make sure your application server is running.
2. Open a command prompt.
3. Change your working directory to the location of your application server configuration script, `%WAS_HOME%\config`. For example:


```
cd \WebSphere\AppServer\config
```
4. Edit `ws_config.xml` and replace every occurrence of the following variables with the actual values:


```
$NodeName$
$ServerName$
$EDX_HOME$
$JMS_HOME$
DB_ASIA$
DB_USER
DB_PWD
```
5. Change your working directory to the location of your application server startup script, `%WAS_HOME%\bin`. For example:


```
cd WebSphere\AppServer\bin
```
6. Run the configuration script **XMLConfig.bat**. Use the [Quick Reference Appendix](#) to enter the values for your environment, using the syntax shown in the following example:

```
XMLConfig.bat -import %EDX_HOME%\config\ws_config.xml
-adminNodeName system_name -nameServiceHost host_name
-nameServicePort admin_port -substitute
"NodeName=system_name;ServerName=eaDirect_server;EDX_HOME=%EDX_HOME%;JMS_HOME=mqm_home"
```

For example:

```
XMLConfig.bat -import C:\EDCSbd\config\ws_config.xml
-adminNodeName doc_frost -nameServiceHost doc_frost
-nameServicePort 900 -substitute
NodeName=doc_frost;ServerName=doc_frost;EDX_HOME=C:\EDCSbd;JMS
_HOME=C:\Program Files\ibm\MQSeries\Java"
```

Upon completion, you are returned to the command prompt.

Default Custom Properties:

The following values are created by the `XMLConfig.sh` script. If you use a different database user, password, or server, or have a different `EDX_HOME` then the ones shown in this table, then you must add these Custom Properties with your values:

Name	Default Value
<code>com.edocs.tasks.loader.user</code>	SQL database user. Default: <code>edx_dba</code>
<code>com.edocs.tasks.loader.password</code>	SQL database user password. Default: <code>edx</code>
<code>com.edocs.tasks.loader.alias</code>	SQL server database name. Default: <code>edx.db</code>
<code>edx.home</code>	Path of %EDX_HOME%. Default: <code>/opt/EDCSbd</code>
<code>java.protocol.handler.pkgs</code>	<code>com.edocs.protocol</code>

To configure Custom Properties in WebSphere 4:

1. Select the server which supports the eaDirect application in the WebSphere Administrative Console.
2. Select the JVM Settings tab, and add or update values in the System Properties section.
3. Click **Apply**.

8

Configuring Java Resources for WebSphere for Windows

Overview

This chapter assumes in-depth understanding of and practical experience with [Application Server Administration](#). It is designed for experienced WebSphere administrators and primarily presents only the steps and settings specific to eaDirect.

See WebSphere Server documentation for detailed step-by-step instructions on Java resource configuration, performance, and tuning. You must also consult your application server administrator for settings that may be specific to your configuration.

You must start WebSphere Server and bring up the Administrative Console (WebSphere 5) or Administrative Client (WebSphere 4) before you begin this chapter.



If you cannot bring up the WebSphere Console/Client, you will be unable to proceed with configuring your application server for eaDirect.

Configuring Java Database Connectivity (JDBC) for eaDirect

After you have successfully configured the eaDirect database, you must configure Java Database Connectivity (JDBC) resources on the eaDirect application server. JDBC Connections on the application server support data retrieval from relational databases and other data sources.

About JDBC Connections for eaDirect

JDBC connection pools contain named groups of JDBC Connections that are created when the connection pool is registered, usually when starting up your application server. Your application server opens JDBC Connections to the database during startup and adds these connections to the pool. A J2EE web application borrows a connection from the pool, uses it, and then returns it to the pool by closing it.

JDBC transaction data sources enable JDBC clients to obtain a connection to a Database Management System (DBMS). Each data source points to the value specified for the Name attribute when a JDBC connection pool was configured.

eaDirect requires three sets of **JDBC Connection Pools** and related **JDBC Transaction (Tx) Data Sources**:

- **edxAdmin** supports the Command Center through the eaDirect web application
- **edxLogger** supports eaDirect logging through the eaDirect web application
- **edxUser** supports user data retrieval through custom web applications

Use the *Quick Reference Appendix JDBC Connections* settings for your application server to configure each connection pool.

For details on configuring JDBC Connections, please see the JDBC documentation for your application and database servers.



edxAdmin connection pools support concurrency for scheduling multiple jobs. Tuning **edxAdmin** connection capacity and threads can improve eaDirect email performance.

For details on how to create and configure JDBC Connections, please see your application server documentation.

You will create three sets of JDBC connection pools and three sets of transaction datasources. Their names are specific to edocs across all platforms, but **JDBC properties vary by database server**. Use the *Quick Reference Appendix* to select the application server properties for your combination of application server and database server properties (Oracle, DB2 or SQL Server).



Make sure you are using the correct properties for your application server and database combination.

Configuring the JDBC Driver for WebSphere 4

Edit the following files to configure the JDBC driver for WebSphere version 4. Editing these files is not required in WebSphere 5.

To edit admin.config

Edit `%WAS_HOME%\bin\admin.config` for Windows or `$WAS_HOME/bin/admin.config` for Unix, and add the following entries:

```
com.ibm.ejs.sm.adminServer.dberrorMap=EC2627=com.ibm.ejs.cm.po
rtability.DuplicateKeyException

com.ibm.ejs.sm.adminServer.dbdataSourceClassName=com.ibm.websp
here.jdbcx.sqlserver.SQLServerDataSource

com.ibm.ejs.sm.adminServer.dbportNumber=1433
```



```
com.ibm.ejs.sm.adminServer.dbselectMethod=cursor
```

To edit setupCmdLine:

Edit `%WAS_HOME%\bin\setupCmdLine.bat`, and add the following entry:

```
SET
DBDRIVER_JARS=%WAS_HOME%\lib\ext\base.jar;%WAS_HOME%\lib\ext\s
py.jar;%WAS_HOME%\lib\ext\sqlserver.jar;%WAS_HOME%\lib\ext\uti
l.jar
```

Configuring JDBC Connections for WebSphere 4 and 5

JDBC properties vary by both application and database server. Use the [Quick Reference Appendix](#) to select the properties for your combination:

- WebSphere 5 for Oracle, see page 116.
- WebSphere 4 for Oracle, see page 118.
- WebSphere 5 for DB2, see page 119.
- WebSphere 4 for DB2, see page 121.
- WebSphere 4 for MSSQL, see page 123.

To configure JDBC Connections for WebSphere:

5. Create a JDBC Connection Pool each for `edxAdmin`, `edxLogger`, and `edxUser`. Use the appropriate JDBC values for your database server.
6. Create a JDBC Transaction (Tx) Data Source each for `edxAdmin`, `edxLogger`, and `edxUser`. Use the appropriate JDBC values for your database server.
7. Review your connections. Each data source should target the connection pool of the same name (`Admin`, `User`, or `Logger`).
8. When you are finished, proceed to [Configuring Java Messaging Services \(JMS\) for eaDirect](#).

Configuring Java Messaging Services (JMS) for eaDirect

After you have successfully configured JDBC Connections, you will now configure Java Messaging Services (JMS) on the application server for eaDirect. eaDirect requires three sets of JMS resources:

- **edxAnnotation** supports Line Item Dispute and Annotation features
- **edxDispute** supports Line Item Dispute and Annotation features
- **edxLogger** supports eaDirect logging through the eaDirect web application

edxLogger requires **five** JMS consumers and session pools.



Tip

If your web application does not implement Line Item Dispute and Annotation, you need only configure JMS resources for **edxLogger**.

Use the [Quick Reference Appendix JMS Resources](#) settings for your application server to configure each set of JMS resources.

About JMS Resources for eaDirect

JMS enable web application components to asynchronously send and receive messages.

- **JMS Connection Factories** are data objects that enable Java Messaging Service (JMS) clients to create JMS connections. You define and configure one or more connection factories to create connections with predefined attributes. Your application server adds the connection factories to the JNDI space during startup, and each J2EE web application retrieves a connection factory using the JNDI on the application server.
- **JMS Stores** store persistent messages in a database accessed through a designated JDBC connection pool. The JMS database can be any database that is accessible through a WebSphere-supported JDBC driver. When creating a JMS Store, you must define the name of the **backing store**, and the **JDBC connection pool** and **database table name prefix** for use with multiple instances
- **JMS Servers** manage connections and message requests on behalf of clients.
- **JMS Topics** can be one of two destinations that you can configure for a JMS server. The other destination is a JMS queue.
- **JMS Session Pools** allow a JMS listener to have multiple threads that will improve performance under heavy load.

Configuring JMS for WebSphere 4 for Windows

WebSphere 4 uses the third-party tools **JMSAdmin** and **MQSeries** to configure and administer JMS. edocs JMS settings for **adxAdmin**, **edxLogger**, and **edxUser** are configured through the shell script **edx_mqm_config** installed with eaDirect.



Caution

JMS for WebSphere requires several components of third-party software. Check your [eaDirect System Requirements](#) to ensure you have the correct versions, patches, and JDK for MQSeries on your platform.

JMSAdmin is the WebSphere Java Message Service Administration Tool to administer JMS objects such as connection factories, queues, and topics, binding them to a JNDI name space. **JMSAdmin.bat** is the command-line executable file.

MQSeries is a third-party tool to configure JMS. **JMSAdmin.config** is the default configuration file for the MQSeries Classes for JMSAdmin.

- **MQSeries 5.2 with CSD06.** For information about installing MQSeries 5.2, see [book/pdf/en_US/amqdac03](#) on the MQSeries 5.2 installation CD-ROM.

- **MA0C** (MQSeries Publish and Subscribe package version 1.0.6). For more information, see <http://www-3.ibm.com/software/ts/mqseries/txppacs/ma0c.html>.
- **MQSeries Classes for Java 5.2** (MA88 1.1.4). eaDirect does not use two phase commit, so you do not need to install that feature.

To configure JMS for WebSphere 4:

1. Configure the **JMSAdmin** executable for your eaDirect environment.
2. Configure the **JMSAdmin.config** settings for your eaDirect environment.
3. Create an MQSeries queue manager for eaDirect.
4. Verify that the MQSeries Publish/Subscribe broker is installed and running properly.
5. Configure the JMS connection factory and destinations with **edx_mqm_config**.

The following topics describe each step in detail.

To configure JMSAdmin for Windows for WebSphere 4:

1. Change your working directory to the Java home directory of your MQM Series installation, for example:

```
C:\Program Files\ibm\MQSeries\Java\bin
```
2. Open the executable **JMSAdmin.bat** in a text editor.
3. Add the following code to the end of the file. All code following the PATH statement must be **on a single line**. Use the [Quick Reference Appendix](#) to edit the application server environment variables **WAS_HOME** and **MQ_JAVA** for your platform.

```
@set WAS_HOME=C:\WebSphere\AppServer
@set MQ_JAVA_INSTALL_PATH=C:\Program Files\ibm\MQSeries\Java
@set PATH=%PATH%;C:\WebSphere\AppServer\java\jre\lib\ext

java -DMQJMS_LOG_DIR=$MQ_JAVA_INSTALL_PATH\log
-DMQJMS_TRACE_DIR=$MQ_JAVA_INSTALL_PATH\trace
-DMQJMS_INSTALL_PATH=$MQ_JAVA_INSTALL_PATH
-Djava.ext.dirs=$MQ_JAVA_INSTALL_PATH\lib:$WAS_HOME\java\jre\lib\ext:
$WAS_HOME\lib com.ibm.mq.jms.admin.JMSAdmin $*
```

4. Comment out the old java command line (prefix it with **rem**).
5. Save and close the file.

To configure JMSAdmin.config for Windows for WebSphere 4:

1. Change directory to the Java home directory of your MQM Series installation, for example:

```
C:\Program Files\ibm\MQSeries\Java
```

2. Open **JMSAdmin.config** in a text editor.

3. Comment out the line (add a #):

```
INITIAL_CONTEXT_FACTORY=com.sun.jndi.ldap.LdapCtxFactory
```

4. Uncomment the line (remove the #):

```
INITIAL_CONTEXT_FACTORY=com.ibm.ejs.ns.jndi.CNInitialContextFactory
```

5. From the list of URLs for JNDI service providers, **uncomment** (remove #) the following setting and edit it for your application server and port:

```
PROVIDER_URL=iiop://localhost:900
```



Port 900 is the default admin port for the administration server. You can specify another port number if necessary, but it must match the port for the admin server.

6. Make sure that no other settings are uncommented for a JNDI service provider or URL.
7. Save and close the file.

To create, start, and stop an MQSeries Queue Manager for Windows:

1. Open a command prompt window.
2. Change directory to the Java home directory of your MQM Series installation, for example:

```
cd C:\Program Files\ibm\MQSeries\Java\bin
```

3. Create a queue manager called **edxQueueManager** using the **crtmqm** command:

```
crtmqm -u SYSTEM.DEAD.LETTER.QUEUE edxQueueManager
```

4. Start the queue manager using the **strmqm** command:

```
strmqm edxQueueManager
```

5. Verify that **edxQueueManager** is running using the **dspmq** command:

```
dspmq -m edxQueueManager
```

6. If **edxQueueManager** is listed with a status of ***Ended***, start it using the **strmqm** command.

7. You can stop the **edxQueueManager** using the **endmqm** command.

```
endmqm edxQueueManager
```

To verify the MQSeries Publish/Subscribe broker for Windows:

1. Open a command prompt window.
2. Change directory to the Java home directory of your MQM Series installation, for example:

```
cd C:\Program Files\ibm\MQSeries\Java\bin
```

3. Make sure that you have the pathname `%JMS_HOME%\java\bin` in your PATH.
4. Verify that `edxQueueManager` is running using the `dspmqrk` command:


```
dspmqrk -m edxQueueManager
```
5. If you receive an error message that the operating system cannot run the `dspmqrk` command, confirm that the MQSeries Publish/Subscribe broker is installed properly, and that `%JMS_HOME%/bin` is included in the system PATH setting.
6. If the operating system reports that the broker is unavailable (or inactive), start it using the following command:


```
strmqbrk -m edxQueueManager
```
7. Run the following command to verify that the broker has been installed and is running:


```
dspmqrk -m edxQueueManager
```
8. If the broker is running, you should see a message similar to the following:


```
MQSeries message broker for queue manager edxQueueManager
running
```
9. Change directory to `%JMS_HOME%\java\bin`, and create the MQ JMS System queues by running the following command:


```
runmqsc edxQueueManager < MQJMS_PSQ.mqsc
```

To configure JMS topics and destinations for Windows:

1. Start your application server if it is not already running.
2. Change directory to `%JMS_HOME%\java\bin`.
3. Define and export the appropriate value for `EDX_HOME`, for example:


```
set EDX_HOME=C:\EDCSbd
```
4. Then run the following command to configure the JMS topics:


```
%EDX_HOME%\config\edx_mqm_config
```


Tip

If your MQSeries server is on a remote machine, see the IBM MQSeries documentation for the appropriate configuration.

Restarting the Application Server

When restarting the application server you should follow this order:

1. Stop the scheduler and logger.
2. Stop the MQ queues using `endmqbrk` and `endmqm`.

3. Stop the eaDirect application server.

When you are ready to start the eaDirect application server again, reverse the steps: start the application server, then start the MQ queues, then start the logger and scheduler.

Where to Go From Here

After configuring JMS for WebSphere4 , the next step is to deploy the application.

Deploying the eaDirect Application

Once your application server is configured successfully, you may proceed to deploying the eaDirect J2EE applications that power the eaSuite. This requires customizing each web application's deployment code for your environment and platform. For details, see your application server documentation and *Deploying and Customizing J2EE Applications*.



Tip

for each web application to be deployed, WebSphere users will need to enter the system classpath for edocs JAR files `edx_common.jar;edx_client.jar;edx_system.jar`. For details, see *Deploying and Customizing J2EE Applications*.

After you have deployed the eaDirect application, continue to the next section in this document to start the scheduler and logger.

Deployment with WebSphere 5

This section describes how to deploy eaDirect using the WebSphere application server. eaDirect 4.1 supplies a pre-assembled EAR for deployment.



Caution

The specific steps for your application server may differ from the ones described below. You should consult your System Administrator and application server documentation for complete details of how to deploy a J2EE application based on your system's configuration.

To deploy a web application with WebSphere 5:

1. From the WebSphere Administrative Console, expand the Applications branch and click on **Install New Application**.
2. Select "Server Path" and then type in the absolute path to where your application is installed, for example,


```
/opt/EDCSbd/J2EEApps/websphere5/Deployed_ear-eadirect.ear
```
3. Keep clicking **Next** until you reach the last page, and then click **Finish**. Use the default settings for each step.
4. Save to the Master Configuration.

See the Deployment Guide for information about deploying additional edocs applications.

Deployment with WebSphere 4

This section describes how to deploy eaDirect using the WebSphere 4 application server.

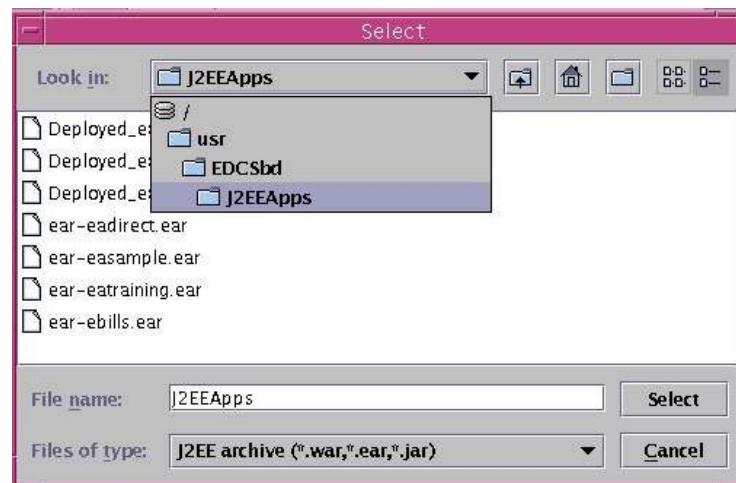


The specific steps for your application server may differ from the ones described below. You should consult your System Administrator and application server documentation for complete details of how to deploy a J2EE application based on your system's configuration.

The first step in setting up the application server is to generate the deployment code for the eaDirect application that you will deploy on WebSphere. This is done through the Application Assembly Tool.

To generate deployment code with WebSphere 4:

1. Start the Application Assembly Tool and click the **Existing** tab.
2. Click **Browse** at the bottom of the dialog, and navigate to the directory that contains the eaDirect J2EE applications. For example:



3. Select **ear-eadirect.ear** from the list of J2EE applications and click **Select**.
4. Click **OK**. The eaDirect application opens in the Application Assembly Tool.
5. In the left pane, right click on **eadirect**, and select **Generate code for deployment** from the menu.

The **Generate Code for deployment** dialog appears.

6. In the **Dependent classpath** field, enter the java classpath for the edocs system, client, and common JAR files separated by colons, for example:

```
/opt/EDCSbd/lib/edx_system.jar:/opt/EDCSbd/lib/edx_client.jar:/opt/EDCSbd/lib/edx_common.jar
```



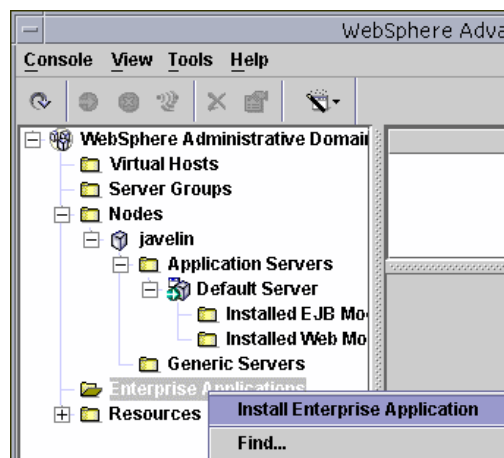
Tip

Be sure to include the leading slashes so that these classpaths are taken from the root.

7. From the Database type drop-down menu, select the database you are using, such as Oracle or DB2.
8. Click **Generate Now**. During generation of the deployment code, status information is shown in the window at the bottom of the dialog.
9. Wait for the progress bar at the bottom of the Application Assembly Tool to complete. Some applications might take several minutes to deploy, depending on the speed of your machine.
10. Click **Close** to exit from the Generate code for deployment dialog.
11. Click **Exit** in the File menu to close the Application Assembly Tool.

To deploy a web application with WebSphere 4:

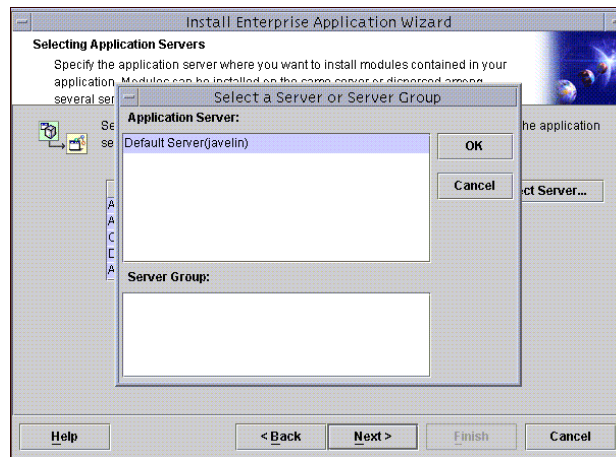
1. From the WebSphere Administrative Console, expand the WebSphere Administrative Domain view.
2. Right-click the **Enterprise Applications** folder, and then select **Install Enterprise Applications** on the menu.



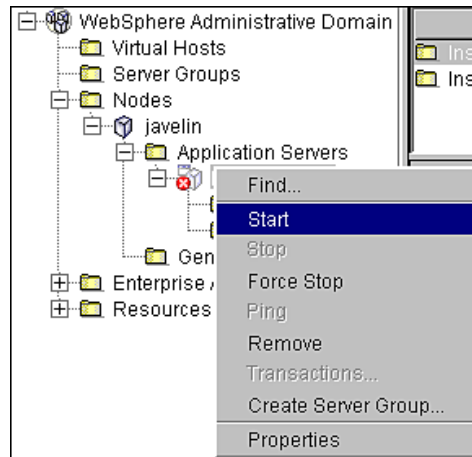
The Install Enterprise Application Wizard dialog appears.

3. Click the **Install Application** radio button, and confirm that the correct node has been chosen in the Browse for file on node field.
4. Click **Browse**. The Open dialog appears.
5. Navigate to `$EDX_HOME/J2EEApps/websphere`, and select `Deployed_ear-eadirect.ear`. The name of the file appears in the File name field.

6. Click **Open**. The Install Enterprise Application Wizard dialog appears with the name of the deployed EAR file in the Install Application (*.ear) Path field.
7. Click **Next** about nine times until you come to the Selecting Application Servers screen. Highlight all the modules for selection by clicking the first and last module in the list, while holding down the **Shift** key.
8. Click **Select Server**. The Select a Server or Server Group dialog appears.
9. Select the Default Server. In the sample screen below, the default server is named javelin.



10. Click **OK** to close the Select a Server of Server Group dialog. The Install Enterprise Application Wizard dialog lists the modules and the server on which they will be installed.
11. Click **Next** and then **Finish**.
12. When asked if you want to regenerate code, click **No**. The installation of eaDirect takes place on the application server.
13. Click **OK** to close the Information dialog reporting that the installation was successful.
14. Stop the server on which you installed the eaDirect J2EE application. Select (expand) **Nodes** and **<your_node>** and **Application Servers**, and then right-click the server name in the left pane. Select **Stop** on the menu.
15. An Information dialog notifies you that the application server has stopped successfully.
16. Click **OK** to close the Information dialog.
17. Restart the application server by right-clicking on the its name, and selecting **Start** on the menu.



18. An Information dialog notifies you that the application server has started successfully.
19. Click **OK** to close the Information dialog.
20. Start the eaDirect J2EE application by expanding the Enterprise Applications folder, right clicking on the application name and then selecting **Start** from the menu.
21. An Information dialog notifies you that the application has started successfully. Click **OK** to close the Information dialog.

Starting the Scheduler and Logger

Starting the eaDirect Logger

The eaDirect Logger generates log files for jobs being run in the Command Center. eaDirect copies all log files to **logs** directory under **EDX_HOME**. The log files are presented in text format.

You may start the Logger from a command line, or as a Windows service.

To start the eaDirect Logger from a Windows command prompt:

1. Open a command prompt window and change directory to the **bin** directory of your eaDirect installation, **%EDX_HOME%\bin**.
2. Run the Logger command for your application server (**ws_logger**), host, and port. This example shows the Logger command for WebSphere:

```
C:\> ws_logger -start -url iiop://localhost:900
```

3. If the Logger starts successfully, a start-up message with the name of the log file appears in the command prompt window. **Do not close this window while eaDirect is running**, as closing it will stop the Logger. A log file is created in `%EDX_HOME%\Logs`.
4. You can stop the Logger by replacing the `-start` parameter with the `-stop` parameter, or simply by closing the command prompt window.

To install the eaDirect Logger as a Windows service:

1. Navigate to the `bin` directory for eaDirect, or `%EDX_HOME%\bin`.
2. Open the Logger script file `LGR.txt` and modify the Java classpath information to reflect the current environment. For example:


```
classpath=c:\jdk1.3.1\lib\tools.jar;c:\bea\wlserver\lib\weblog
ic.jar;c:\edcsbd\lib\edx_client.jar;c:\edcsbd\lib\edx_common.j
ar
```

Caution

Make sure you set all paths to the appropriate point releases/patches for your application server and JDK, if necessary. Check the Release Notes and your system documentation for updated requirements.

3. Confirm that the following line of code is present in the file for your host and port:


```
-Djava.naming.provider.url=t3://localhost:9080
```
4. Confirm that all the directory references in `LGR.txt` are correct.
5. Save and close `LGR.txt`.
6. Open a command window, and then change directory to `%EDX_HOME%\bin`. Use the `install` command to install the Logger as a Windows Service, for example:


```
C:\> Loggersvc -install C:\EDCSbd\bin\LGR.txt
```
7. If the Logger service is installed successfully, a confirmation message appears.

To start the eaDirect Logger as a Windows Service:

1. Install the Logger as a Windows Service if you have not already done so.
2. Start your application server if it is not already running.
3. From the Start menu, select **Settings>Control Panel**.
4. Double-click **Administrative Tools**, then double-click **Services**.
5. Right-click **Logger Service** and select **Start**. You may also click the Start icon.
6. To start the Logger automatically at startup, right-click the service and select **Properties**. From the Startup Type menu, select **Automatic**.

Starting the eaDirect Scheduler

You must start the eaDirect Scheduler in order to schedule and run jobs in the eaDirect Command Center. If you attempt to run a new job with the Scheduler not running, the job will not run and you will see ‘Not yet started’ as its status.

The eaDirect scheduler does not need to be started until after you have deployed the EAR file.

You may start the Scheduler from a command line, or as a Windows service.

To start the Scheduler from a Windows command prompt:

1. Open a command prompt window and change directory to the `bin` directory of your eaDirect installation, `%EDX_HOME%\bin`.
2. Run the Scheduler command for your application server (`ws_scheduler.bat`), with the proper host, and port. This example shows the Scheduler command for WebSphere:

```
C:\> ws_scheduler -start -url iiop://localhost:900
```

3. If the Scheduler starts successfully, a start-up message with the name of the log file appears in the command prompt window. **Do not close this window while eaDirect is running**, as closing it will stop the Scheduler. A log file is created in `%EDX_HOME%\Logs`.
4. You can stop the Scheduler by replacing the `-start` parameter with the `-stop` parameter, or simply by closing the command prompt window.

To install the Scheduler as a Windows Service:

To install the Scheduler as a Windows Service, you must modify the Scheduler template file `SCH.txt`, installed to the `bin` directory for eaDirect.

1. Navigate to the `bin` directory for eaDirect, or `%EDX_HOME%\bin`.
2. Open the Scheduler template file `SCH.txt` and modify the Java classpath to reflect your active Java environment. For example:

```
-classpath=C:\jdk1.3.1_07\lib\tools.jar;  
C:\WebSphere\AppServer\lib\websphere.jar;  
c:\edcsbd\lib\edx_client.jar;  
c:\edcsbd\lib\edx_common.jar;  
C:\EDCSbd\lib\edx_common.jar
```



Make sure you set all paths to the appropriate point releases/patches for your application server and JDK, if necessary. Check the Release Notes and your system documentation for updated requirements.

3. Confirm that the following line of code is present in the file for your host and port:

```
-Djava.naming.provider.url=t3://localhost:9082
```

4. If you want the Scheduler to log information to a file rather than to the console, add the following value in **SCH.txt**:

```
-Dcom.edocs.pwc.debug=true <scheduler_logfile>
```
5. Confirm that all the directory references in **SCH.txt** are correct.
6. Save and close **SCH.txt**.

**Tip**




If you have a 2.x or earlier version of the Scheduler installed as a Windows service, you must remove it before installing the new version of the Scheduler.

7. Open a command window, and then change directory to **%EDX_HOME%\bin**. Use the **install** command to install the Scheduler as a Windows Service, for example:

```
C:\> Schedulersvc -install C:\EDCSbd\bin\SCH.txt
```
8. If the Scheduler service is installed successfully, a confirmation message appears.

To start the eaDirect Scheduler as a Windows Service:

1. Install the Scheduler as a Windows Service if you have not already done so.
2. Start your application server if it is not already running.
3. From the Start menu, select **Settings>Control Panel**.
4. Double-click **Administrative Tools**, then double-click **Services**.
5. Right-click **Scheduler Service** and select **Start**. You may also click the Start icon.

 Routing and Remot...	Offers routing services to businesses in local area an...	Disabled	LocalSystem
 RunAs Service	Enables starting processes under alternate credentials	Started	Automatic
 Scheduler Service	eDocs Scheduler service	Started	Manual

6. To start the Scheduler automatically at startup, right-click the service and select **Properties**. From the Startup Type menu, select **Automatic**.

Appendix A: WebLogic Reference

Java DataBase Connectivity (JDBC)

**Tip**

Connection pool size increased in 3.4 and later versions to support concurrency for multiple jobs.

You will be entering the same information six times: one connection pool and one Tx data source each for **Admin**, **User**, and **Logger**. Make sure you have chosen the correct properties for your application server and database server, and that each datasource and its properties maps to the connection pool of the same name.

For details of how to configure JDBC connections, see your application server documentation. For the procedure to create connections for eaDirect, see the topic for your application server:

- [Configuring JDBC Connections for WebSphere](#)

**Caution**

Make sure you are using the correct properties for your application server, database, and JDBC resource.

WebLogic Environment Variables

WebLogic 6.1 and 7.0

**Caution**

Make sure you set all paths to the appropriate point releases/patches for WebLogic Server and JDK, if necessary. Check the Release Notes and your system documentation for updated requirements to these environment variables.

VARIABLE	DESCRIPTION	WINDOWS	SOLARIS	HP-UX	CUSTOM
WL_HOME	app server home	C:\bea\wlserver	/export/home/ /bea/wlserver	/export/home/ /bea/wlserver	
APP_OWNER	app server owner		edxadmin	edxadmin	
APP_GROUP	app server group		edxadmin	edxadmin	
APP_PORT	app server port	7001	7001	7001	
ADMIN_PORT	app server admin port	7002	7002	7002	
JAVA_HOME	Java home directory	%WL_HOME%\jdk131	\$WL_HOME/jdk131	\$WL_HOME/jdk131	

Java DataBase Connectivity (JDBC) for WebLogic

WebLogic 6.1 and 7.0 for SQL Server (Windows)



Tip

WebLogic 7 users may **clone** additional Java resources. Right-click a resource and select **Clone <name>**, then change the resource name and properties as required.

JDBC Connection Pools

Create three JDBC Connection Pools, using WebLogic Server documentation at <http://bea.com>. Use the same **Properties** for all three connection pools. Remember to move the Targets-Server from **Available** to **Chosen** for each data source as you configure it.

Pool 1: Admin	Pool 2: User	Pool 3: Logger
edxAdminConnectionPool	edxUserConnectionPool	edxLoggerConnectionPool

General Tab	
URL	jdbc:inetpool:inetdae7://localhost:1433
Driver Classname	com.inet.pool.PoolDriver
Properties	poolurl=jdbc:inetpool:inetdae7://localhost:1433 user=edx_dba password=edx

Connections Tab	
Initial Capacity	1
Maximum Capacity	20
Capacity Increment	5
Login Delay Seconds	1
Refresh Period	1
Supports Local Transaction	False (box unchecked)
Allow Shrinking	True (box checked)
Shrink Period	15
Prepared Statement Cache Size	300
(WL 7 only)	
XAPrepared Statement Cache Size	300

JDBC Connection Pool Tab Settings, continued:

Testing Tab	
Test Table Name	job
Test Reserved Connections	TRUE (checked)
Test Released Connections	FALSE (unchecked)

Targets Tab	
Targets-Server	[move myserver from Available to Chosen]

JDBC Tx Data Sources

Create a transaction data source each as shown for **edxAdmin**, **edxLogger**, and **edxUser**, using WebLogic Server documentation at <http://bea.com>. Remember to move the Targets-Server from **Available** to **Chosen** for each data source as you configure it.

	Datasource 1: Admin	Datasource 2: User	Datasource 3: Logger
Name	edxAdminDataSource	edxUserDataSource	edxLoggerDataSource
JNDI Name	edx.databasePool	edx.user.databasePool	edx.logger.databasePool
Pool Name	edxAdminConnectionPool	edxUserConnectionPool	edxLoggerConnectionPool

Configuration Tab (use defaults)	
Emulate Two-Phase Commit for non-XA Driver	FALSE (unchecked)
Row Prefetch Enabled	FALSE (unchecked)
Row Prefetch Size:	48
Stream Chunk Size: bytes	256

Targets Tab	
Targets-Server	[move myserver from Available to Chosen]

WebLogic 6.1 and 7.0 for Oracle (Solaris and HP-UX)

JDBC Connection Pools

Create three JDBC Connection Pools, using WebLogic Server documentation at <http://bea.com>. Use the same **Properties** for all three connection pools. Remember to move the Targets-Server from **Available** to **Chosen** for each data source as you configure it.

Pool 1: Admin	Pool 2: User	Pool 3: Logger
edxAdminConnectionPool	edxUserConnectionPool	edxLoggerConnectionPool

General Tab	
URL	jdbc:oracle:thin:@DB_host:DB_port:eaDirect_DB:SID. For example, jdbc:oracle:thin:@localhost:1521:edx0
Driver Classname	oracle.jdbc.driver.OracleDriver
Properties	Substitute the localhost:port:SID, user and password as required: poolurl=jdbc:oracle:thin:@localhost:1521:edx0 user=edx_dba password=edx

Connections Tab	
Initial Capacity	1
Maximum Capacity	20
Capacity Increment	5
Login Delay Seconds	1
Refresh Period	1
Supports Local Transaction	False (box unchecked)
Allow Shrinking	True (box checked)
Shrink Period	15
Prepared Statement Cache Size	300
(WL 7 only)	
XAPrepared Statement Cache Size	300

JDBC Connection Pool Tabs for WebLogic 6 continued:

Testing Tab	
Test Table Name	dual
Test Reserved Connections	TRUE (checked)
Test Released Connections	FALSE (unchecked)

Targets Tab	
Targets-Server	[move myserver from Available to Chosen]

JDBC Tx Data Sources

Create three transaction data sources, using WebLogic Server documentation at <http://bea.com>. Remember to move the Targets-Server from **Available** to **Chosen** for each data source as you configure it.

	Datasource 1: Admin	Datasource 2: User	Datasource 3: Logger
Name	edxAdminDataSource	edxUserDataSource	edxLoggerDataSource
JNDI Name	edx.databasePool	edx.user.databasePool	edx.logger.databasePool
Pool Name	edxAdminConnectionPool	edxUserConnectionPool	edxLoggerConnectionPool

Configuration Tab (use defaults)	
Emulate Two-Phase Commit for non-XA Driver	FALSE (unchecked)
Row Prefetch Enabled	FALSE (unchecked)
Row Prefetch Size:	48
Stream Chunk Size: bytes	256

Targets Tab	
Targets-Server	[move myserver from Available to Chosen]

Java Messaging Services for WebLogic 6.1 and 7.0

WebLogic 7 users may **clone** additional Java resources. Right-click a resource and select **Clone <name>**, then change the resource name and properties as required.

You will be entering very similar information many times: one set of JMS resources each for annotation and dispute, and FIVE sets for logging. Make sure you have chosen the correct properties for the resource you are creating, and that each resource maps to others of the **same name**.

JMS Connection Factories

Create three JMS connection factories, using WebLogic Server documentation at <http://bea.com>. You may accept the default **Properties** for all three connection factories, or consult your application server administrator to tune these values.

	1: Annotation	2: Dispute	3: Logger
Name	<code>edxAnnotationTCF</code>	<code>edxDisputeTCF</code>	<code>edxLoggerTCF</code>
JNDI Name	<code>edx/tcf/annotate</code>	<code>edx/tcf/dispute</code>	<code>edx/tcf/log</code>

Targets Tab	
Targets-Server	[move <code>myserver</code> from Available to Chosen]

JMS (JDBC) Stores

Create three JMS JDBC Stores, using WebLogic Server documentation at <http://bea.com>. You may accept the default **Prefix Name=<NULL>** for all three stores, or consult your application server administrator to tune these values.

Name (of JMS Store)	Connection Pool
<code>edxAnnotationStore</code>	<code>edxUserConnectionPool</code>
<code>edxDisputeStore</code>	<code>edxUserConnectionPool</code>
<code>edxLoggerStore</code>	<code>edxLoggerConnectionPool</code>

JMS Servers

Create three JMS Servers, using WebLogic Server documentation at <http://bea.com>. You may accept the default **Properties** for all three servers, or consult your application server administrator to tune these values.

	1: Annotation	2: Dispute	3: Logger
Name	edxAnnotationServer	edxDisputeServer	edxLoggerServer
Store	edxAnnotationStore	edxDisputeStore	edxLoggerStore

Targets Tab	
Targets-Server	[select myserver from drop-down menu]

JMS Topics

Create three JMS Topics, using WebLogic Server documentation at <http://bea.com>. For WebLogic 7 select **Destinations** under each defined Server, then click on **Configure a new JMSTopic**. Make sure to create the matching topic for each server. Accept the default **Properties** for all three topics, or consult your application server administrator to tune these values.

	1: Annotation	2: Dispute	C3: Logger
Name	edxAnnotationTopic	edxDisputeTopic	edxLoggerTopic
JNDI Name	edx/jms/annotate	edx/jms/dispute	edx/jms/log

JMS Session Pools and Consumers for Annotation and Dispute


Tip

If your deployment does not use annotation and dispute, you can skip to configuring session pools and consumers for Logger.

Create one pair of JMS Session Pools and Consumers each for Annotation and Dispute, using WebLogic Server documentation at <http://bea.com>. Set **Acknowledge Mode** to **auto** and **Sessions Maximum** to **-1** for all three Session Pools.

Session Pool	1: Annotation	2: Dispute
Name	edxAnnotationPool	edxDisputePool
Connection Factory	edx/tcf/annotate	edx/tcf/dispute
Listener Class	com.edocs.services.annotation.Listener	com.edocs.services.dispute.Listener



For each session pool, -1 specifies no session maximum. Tune each Session Maximum to the maximum number of threads for each pool.

Consumer	1: Annotation	2: Dispute
Name	<code>edxAnnotationConsumer</code>	<code>edxDisputeConsumer</code>
Messages Maximum	10	10
Selector	<code>JMSType='USER'</code>	<code>JMSType='USER'</code>
Destination	<code>edx/jms/annotate</code>	<code>edx/jms/dispute</code>

JMS Session Pools and Consumers for Logging

Create FIVE pairs of **JMS Session Pools and Consumers** for **Logger**, using WebLogic Server documentation at <http://bea.com>. Set **Acknowledge Mode** to **auto** and **Sessions Maximum** to **-1** for all five Session Pools.



For each session pool, -1 specifies no session maximum. Tune each Session Maximum to the maximum number of threads for each pool.

1) Admin Activity

JMS session pool - Configuration Tab

Property	Value
Name	<code>edxLoggerAdminActivityPool</code>
Connection Factory	<code>edx/tcf/log</code>
Listener Class	<code>com.edocs.fs.logging.sub.AdminActivityListener</code>
Acknowledge Mode	<code>auto</code>
Sessions Maximum	<code>-1</code>

JMS Consumer- Configuration Tab

Property	Value
Name	<code>edxLoggerAdminActivityConsumer</code>
Messages Maximum	10
Selector	<code>JMSType='ADM'</code>
Destination	<code>edx/jms/log</code>

2) CSR Activity

JMS session pool- Configuration Tab

Property	Value
Name	<code>edxLoggerCSRActivityPool</code>
Connection Factory	<code>edx/tcf/log</code>
Listener Class	<code>com.edocs.fs.logging.sub.CSRActivityListener</code>
Acknowledge Mode	<code>auto</code>
Sessions Maximum	<code>-1</code>

JMS consumer- Configuration Tab

Property	Value
Name	<code>edxLoggerCSRActivityConsumer</code>
Messages Maximum	<code>10</code>
Selector	<code>JMSType= 'CSR'</code>
Destination	<code>edx/jms/log</code>

3) Message Log

JMS session pool- Configuration Tab

Property	Value
Name	<code>edxLoggerMessageLogPool</code>
Connection Factory	<code>edx/tcf/log</code>
Listener Class	<code>com.edocs.fs.logging.sub.MessageLogListener</code>
Acknowledge Mode	<code>auto</code>
Sessions Maximum	<code>-1</code>

JMS consumer- Configuration Tab

Property	Value
Name	<code>edxLoggerMessageLogConsumer</code>
Messages Maximum	<code>10</code>
Selector	<code>JMSType= 'MSG'</code>
Destination	<code>edx/jms/log</code>

4) System Activity

JMS session pool- Configuration Tab

Property	Value
Name	<code>edxLoggerSystemActivityPool</code>
Connection Factory	<code>edx/tcf/log</code>
Listener Class	<code>com.edocs.fs.logging.sub.SystemActivityListener</code>
Acknowledge Mode	<code>auto</code>
Sessions Maximum	<code>-1</code>

JMS consumer

Property	Value
Name	<code>edxLoggerSystemActivityConsumer</code>
Messages Maximum	<code>10</code>
Selector	<code>JMSType= 'SYS'</code>
Destination	<code>edx/jms/log</code>

5) User Activity

JMS session pool

Property	Value
Name	<code>edxLoggerUserActivityPool</code>
Connection Factory	<code>edx/tcf/log</code>
Listener Class	<code>com.edocs.fs.logging.sub.UserActivityListener</code>
Acknowledge Mode	<code>auto</code>
Sessions Maximum	<code>-1</code>

JMS consumer

Property	Value
Name	<code>edxLoggerUserActivityConsumer</code>
Messages Maximum	<code>10</code>
Selector	<code>JMSType= 'USER'</code>
Destination	<code>edx/jms/log</code>

Appendix B: WebSphere Reference

WebSphere Environment Variables

- **WebSphere 4 for Windows on MSSQL** requires a WebSphere branded version of a JDBC driver, **Connect 3.1**. This version supports WebSphere 4.0.5, 4.0.6, and 5.0, 5.0.1 and 5.0.2. Version 2.2 does not support WAS 5.0 or higher, so edocs recommends version 3.1.
- **WebSphere 4 for Windows on MSSQL** also requires the stored procedure patch for JTA, downloadable from IBM WebSphere.
- WebSphere 4.06 FixPak 4 and FixPak 5 use the same version JDK. IBM does not support changing the JDK from the version installed with each fix pack.

VARIABLE	DESCRIPTION	WINDOWS	SOLARIS	AIX	CUSTOM
WAS_HOME	Application Server home	C:\WebSphere\AppServer	/opt/WebSphere/AppServer	/usr/WebSphere/AppServer	
APP_OWNER	app server owner		edxadmin	edxadmin	
APP_GROUP	app server group		edxadmin	edxadmin	
APP_PORT	app server port	9080	9080	9080	
ADMIN_PORT	app server admin port	900	900	900	
JAVA_HOME	Java home directory	C:\WebSphere\AppServer\java	\$WAS_HOME/java	\$WAS_HOME/java	
JMS_HOME	MQSeries java client directory	C:\Program Files\IBM\MQSeries	/opt/mqm/java	/usr/mqm/java	



Make sure you set all paths to the appropriate point releases/patches for your application server and JDK, if necessary. Check the Release Notes and your system documentation for updated requirements.

WebSphere 5 for Oracle (Unix)

JDBC Connection Pools

Configure a new **JDBC Provider**, choose **Oracle JDBC Driver** from the JDBC Providers drop-down list.

Also configure the following properties:

Name	Value
Classpath	\$ORACLE_HOME/jdbc/lib/classes12.zip Enter the explicit path; do not use the variable.

Data Sources

Create the following **Data Sources (Version 4)** for the new JDBC Provider:

Name	edxAdminDataSource	edxUserDataSource	edxLoggerDataSource
JNDI Name	edx.databasePool	edx.user.databasePool	edx.logger.databasePool

For each **Datasource**, enter the following fields:

Database Name	The name of the eaDirect database. For example, edx0
Default User ID	The eaDirect database user. For example, edx_dba
Default Password	The password for the eaDirect database user. For example, edx.

For **Connect Pool**:

Maximum Pool Size	20
-------------------	----

Custom Properties:

URL	jdbc:oracle:thin:@DB_host:DB_port:eaDirect_DB. For example, jdbc:oracle:thin:@localhost:1521:edx0
-----	--

JMS Resources (WebSphere 5, Oracle, Unix)

Click on **Resources**, then **WebSphere JMS Provider**, and create three new **WebSphere Topic Connection Factories**:

Name	edxLoggerTCF
JNDI name	edx.tcf.log
Port	Queued

Name	edxAnnotationTCF
JNDI name	edx.tcf.annotate
Port	Queued

Name	edxDisputeTCF
JNDI name	edx.tcf.dispute
Port	Queued

WebSphere Topic Destinations (WebSphere 5, Oracle, Unix)

Click on **Resources**, then **WebSphere JMS Provider**, and create three new **WebSphere Topic Destinations**:

Name	edxLoggerTopic
JNDI Name	edx.jms.log
Topic	edxLoggerTopic
Persistence	Application Defined
Priority	Application Defined

Name	edxAnnotationTopic
JNDI Name	edx.jms.annotate
Topic	edxAnnotationTopic

Name	<code>edxAnnotationTopic</code>
Persistence	Application Defined
Priority	Application Defined

Name	<code>edxDisputeTopic</code>
JNDI Name	<code>edx.jms.dispute</code>
Topic	<code>edxDisputeTopic</code>
Persistence	Application Defined
Priority	Application Defined

After configuring JMS for WebSphere 5, be sure to **save to the master configuration**.

WebSphere 4 for Oracle (Unix)

JDBC Connection Pools (WebSphere 4, Oracle, Unix)

Create three connection pools, using your application server documentation. Use the same **Properties** for all three connection pools.

- `edxAdminConnectionPool`
- `edxUserConnectionPool`
- `edxLoggerConnectionPool`

Properties	
Implementation Class	<code>oracle.jdbc.pool.OracleConnectionPoolDataSource</code>
Node	<code>localhost</code>
Specify Driver/Add Driver	<code>\$ORACLE_HOME/jdbc/lib/classes12.zip</code> Enter the explicit path for \$ORACLE_HOME; do not use the variable.

JDBC Tx Data Sources (WebSphere 4, Oracle, Unix)

Create three transaction data sources, using your application server documentation. Use the same **Custom Properties** for all three datasources.

edxAdminDataSource	edxUserDataSource	edxLoggerDataSource
edx.databasePool	edx.user.databasePool	edx.logger.databasePool
edxAdminConnectionPool	edxUserConnectionPool	edxLoggerConnectionPool

Custom Properties	
URL	jdbc:oracle:thin:@DB_host:DB_port:eaDirect_DB. For example, jdbc:oracle:thin:@localhost:1521:edx0
user	The name of the eaDirect database. For example, edx_dba
password	The eaDirect database user. For example, edx
databaseName	The password for the eaDirect database user. For example, edx0
Connection Pooling Maximum Pool Size	50
Disable AutoConnection	FALSE (unchecked)

WebSphere 5 for DB2 (Unix)

JDBC Connection Pools

Configure a new **JDBC Provider**; expand Resources, JDBC Providers and select **New**. Then choose **DB2 JDBC provider** from the JDBC Providers drop-down list.

Property	Value
Classpath	\$DB_HOME/db2inst1/sqllib/java12/db2java.zip Enter the explicit path for \$DB_HOME; do not use the variable.
Implementation Classname	COM.ibm.db2.jdbc.DB2ConnectionPoolDataSource

Expand Resources, then JDBC Profilers, then DB2 JDBC Provider, then Data Sources (Version 4). Create the following **Data Sources (Version 4)** for the new JDBC Provider:

Name	edxAdminDataSource	edxUserDataSource	edxLoggerDataSource
JNDI Name	edx.databasePool	edx.user.databasePool	edx.logger.databasePool

For each **Datasource**, enter the following fields:

Database Name	The name of the eaDirect database. For example, edx0
Default User ID	The eaDirect database user. For example, edx_dba
Default Password	The password for the eaDirect database user. For example, edx .

Expand Resources, then JDBC Profilers, then DB2 JDBC Provider, then Data Sources (Version 4), then Data Source Name, then Connection Pool. For **Connect Pool** set:

Maximum Pool Size	20
-------------------	----

Expand Resources, then JDBC Profilers, then DB2 JDBC Provider, then Data Sources (Version 4), then Data Source Name, then Custom Properties. For **Custom Properties** set:

connectionAttribute	cursorhold=1
---------------------	--------------



Remember to **save** to the Master Configuration.

JMS Resources (WebSphere 5, DB2, Unix)

Click on **Resources**, then **WebSphere JMS Provider**, and create new **WebSphere Topic Connection Factories**:

Name	edxLoggerTCF
JNDI name	edx.tcf.log
Port	Queued

Name	edxAnnotationTCF
JNDI name	edx.tcf.annotate
Port	Queued

Name	edxDisputeTCF
JNDI name	edx.tcf.dispute
Port	Queued

Topic Destinations (WebSphere 5, DB2, Unix)

Click on **Resources**, then **WebSphere JMS Provider**, and create new **WebSphere Topic Destinations**:

Name	<code>edxLoggerTopic</code>
JNDI Name	<code>edx.jms.log</code>
Topic	<code>edxLoggerTopic</code>
Persistence	<code>Application Defined</code>
Priority	<code>Application Defined</code>

Name	<code>edxAnnotationTopic</code>
JNDI Name	<code>edx.jms.annotate</code>
Topic	<code>edxAnnotationTopic</code>
Persistence	<code>Application Defined</code>
Priority	<code>Application Defined</code>

Name	<code>edxDisputeTopic</code>
JNDI Name	<code>edx.jms.dispute</code>
Topic	<code>edxDisputeTopic</code>
Persistence	<code>Application Defined</code>
Priority	<code>Application Defined</code>

WebSphere 4 for DB2 (Unix)

JDBC Connection Pools

Create three connection pools, using your application server documentation.

- `edxAdminConnectionPool`
- `edxUserConnectionPool`
- `edxLoggerConnectionPool`

Use the same **Properties** for all three connection pools.

Implementation Class	<code>COM.ibm.db2.jdbc.DB2ConnectionPoolDataSource</code>
Node	<code>localhost</code>

Specify Driver/Add Driver	\$DB2_HOME/ sql1lib/java12/db2java.zip Enter the explicit path for \$DB2_HOME; do not use the variable.
---------------------------	---

JDBC Tx Data Sources (WebSphere 4, DB2, Unix)

Create three transaction data sources, using your application server documentation.

Name	edxAdminDataSource	edxUserDataSource	edxLoggerDataSource
JNDI Name	edx.databasePool	edx.user.databasePool	edx.logger.databasePool
Pool Name	edxAdminConnectionPool	edxUserConnectionPool	edxLoggerConnectionPool

Use the same **Custom Properties** for all three datasources.

user	edx_dba
password	edx
database Name	edx0
Connection Pooling Maximum Pool Size	50
Disable AutoConnection	FALSE (unchecked)

WebSphere 4 for SQL Server (Windows)

JDBC Connection Pools

Create three connection pools, using your application server documentation. Use the same **Properties** for all three connection pools.

- `edxAdminConnectionPool`
- `edxUserConnectionPool`
- `edxLoggerConnectionPool`

Use the same **Custom Properties** for all three datasources.

Implementation Class	<code>com.ibm.websphere.jdbcx.sqlserver.SQLServerDataSource</code>
Node	<code>localhost</code>
Specify Driver/Add Driver	<code>%WAS_HOME%\lib\sljc.jar</code> and <code>%WAS_HOME%\lib\sljcx.jar</code>

JDBC Tx Data Source (WebSphere 4, MSSQL, Windows)

Create three transaction data sources, using your application server documentation.

Name	<code>edxAdminDataSource</code>	<code>edxUserDataSource</code>	<code>edxLoggerDataSource</code>
JNDI Name	<code>edx.databasePool</code>	<code>edx.user.databasePool</code>	<code>edx.logger.databasePool</code>
Pool Name	<code>edxAdminConnectionPool</code>	<code>edxUserConnectionPool</code>	<code>edxLoggerConnectionPool</code>

Use the same **Custom Properties** for all three datasources, substituting your database server, port, username and password if different.

Server Name	<code>localhost</code>
port number	<code>1433</code>
user	<code>edx_dba</code>
password	<code>edx</code>
database Name	<code>edx0</code>
selectMethod	<code>cursor</code>
disable3phase (add this as a new custom property)	<code>true</code>

Index

A

About this Guide, 9
 administration, 48
 AIX, 115
 Annotation, 45, 46, 67, 68, 89, 90
 Application, 17, 27, 43, 65, 87
 Application Server, 17, 27, 28, 43, 65, 87, 103
 creating, 56, 83
 attributes, 46, 68, 90

C

Capturing Information about Your eaDirect Environment Using `edx_config`, 21, 23, 32, 35, 53, 54, 61
 Checking `.log` files for errors, 48, 99
 Command Center, 19, 40, 44, 66, 88
 Configuring, 17, 27, 43, 44, 45, 46, 67, 89
 Customer Self-Service, 7

D

Database Connectivity, 43
 Dispute, 45, 46, 67, 68, 89, 90
 documentation, 9, 17, 18, 19, 20, 27, 28, 43, 46, 49, 72, 93

E

eaDirect, 16, 17, 18, 19, 20, 21, 27, 28, 29, 30, 31, 32, 33, 34, 35, 37, 40, 43, 44, 45, 48, 49, 51, 53, 54, 67, 69, 89, 90, 91
 eaSample, 56, 83
 eaSuite, 7, 16, 18, 40, 49, 51
`edx_client.jar`, 40, 73, 94
`edx_common.jar`, 39, 73, 94
`edx_config`
 matching JVM values, 84
`edxAdmin`, 44, 45, 66, 67, 88, 89, 106
`edxAnnotation`, 45, 46, 47, 67, 89
`edxDispute`, 45, 46, 47, 67, 89
`edxLogger`, 44, 45, 46, 47, 66, 67, 68, 69, 88, 89, 90, 106, 113
`edxUser`, 44, 45, 66, 67, 69, 88, 89, 90, 105, 106, 107
 Environment Data, 21, 31, 53, 81
 Application Server, 53, 103, 115
 Application Server, Windows, 30, 81
 Passing UNIX to Your Application Server, 22, 54
 UNIX with `edx_config`, 20, 53
 Windows with `edx_env.bat` and `edx_bcp.config.bat`, 30

- Windows with `edx_env.bat`
and `edx_load.config.bat`,
81
- G**
- Groups, 104
- H**
- Help
 - documentation, 10
 - technical support, 11
- HP/UX, 104
- J**
- JDBC, 43
- JMS, 45, 67, 89
 - WebLogic, 46, 109
 - WebSphere with MQSeries,
69, 90
- JVM
 - matching `edx_config`, 84
 - WebSphere
- windows, 84
 - WebSphere, 61
- L**
- Logging into the Command
Center, 40
- logs, 77, 98
- M**
- Message, 112
- Migrating
 - eaDirect Applications, 16
- P**
- Passing Windows
Environment Data to Your
Application Server, 32, 35,
83
- Permissions
 - Application Server, 18
- S**
- Scheduling jobs, 19, 28, 40,
47, 48, 49, 52, 101
- Session, 46, 47, 111
- Solaris, 19, 28, 31
- Starting and Stopping
 - Active Application Server, 28,
52
 - Application Server, 19, 28,
52
 - Logger, 77, 98
 - Scheduler, 78, 100
 - WebLogic Server, 19, 29
- T**
- Troubleshooting
 - WebLogic on Windows, 41
- Type, 49
- U**
- UNIX Config Files, 21
- User, 113
- V**
- Verify, 69, 71, 72, 91, 92, 93
- W**
- WebLogic, 15, 19, 20, 24, 28,
29, 30, 32, 36, 39, 41, 45,
46, 48, 103, 109
- WebSphere, 70
- Windows, 19, 28, 29, 30, 31,
32, 35, 40, 47, 48, 99, 101