BEA WebLogic Integration™

Starting, Stopping, and Customizing BEA WebLogic Integration

Release 7.0
Document Date: June 2002

Copyright

Copyright © 2002 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

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

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

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

Trademarks or Service Marks

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

All other trademarks are the property of their respective companies.

Starting, Stopping, and Customizing BEA WebLogic Integration

Part Number	Date	Software Version
N/A	June 2002	7.0

Contents

About This Document

What You Need to Know viii
e-docs Web Site viii
How to Print the Document viii
Contact Us! ix
Documentation Conventions x

1. Getting Started

WebLogic Server Domains 1-2
WebLogic Integration Samples Domain 1-3
WebLogic Integration Configuration Templates 1-5
Administration and Design Tools 1-7
About the WebLogic Integration Repository 1-9
WebLogic Integration Users and Passwords 1-9
Domain Configuration Requirements 1-11
    Configuring the Database for a Domain 1-13
Configuring and Starting the Samples Domain 1-14
Starting WebLogic Integration 1-17
    Starting WebLogic Integration on Windows 1-17
        Starting WebLogic Integration Through Menus 1-17
        Starting WebLogic Integration from the Command Line 1-18
    Starting WebLogic Integration on UNIX 1-19
    Startup Complete Messages 1-20
Stopping WebLogic Integration 1-20
    Stopping WebLogic Integration on Windows 1-21
    Stopping WebLogic Integration on UNIX 1-22
Next Steps 1-22

2. WebLogic Integration Administration and Design Tools

Web Browser Configuration Requirements 2-2
Accepting Cookies 2-2
Bypassing the Proxy Server 2-3
Bypassing the Proxy Server in Netscape 2-3
Bypassing the Proxy Server in Microsoft Internet Explorer 2-3
Starting the Samples Launcher 2-4
Learning to Use the Samples 2-6
Starting the WebLogic Server Administration Console 2-7
Learning to Use the WebLogic Server Administration Console 2-8
Starting the Application View Console 2-9
Learning to Use Application Views 2-12
Starting the Studio 2-13
Start Procedure for the Studio 2-13
Note About BPM Plug-Ins 2-16
Learning to Use the Studio 2-17
Starting the Worklist 2-18
Learning to Use the Worklist 2-20
Starting the Format Builder 2-21
Learning to Use the Format Builder 2-23
Starting the B2B Console 2-24
Learning to Use the B2B Console 2-26
Starting and Stopping the B2B Engine from the B2B Console 2-26
Stopping the B2B Engine from the B2B Console 2-26
Restarting the B2B Engine from the B2B Console 2-28

3. Customizing WebLogic Integration

WebLogic Integration Commands and Supporting Files 3-2
Specifying a New Database for a Domain 3-2
Initializing the Database for a Domain 3-4
Using the Database Wizard 3-5
Database Connection Information 3-5
Database Wizard Options 3-7
How the Database Wizard Works 3-7
Database Configuration Modes 3-8

Using the Database Wizard in Graphical Mode 3-8
Using the Database Wizard in Console Mode 3-15
Updating the Database Configuration from the WebLogic Server Administration Console 3-17
Updating the JDBC Connection Pool 3-18
Updating the RDBMS Realm Properties 3-19
Database Access Parameters 3-20
Database Access URL Format 3-21
Creating and Customizing a New Domain 3-22
Preparing to Create a Domain 3-22
Procedure for Creating and Customizing a Domain for a Single Server . 3-23
Supporting Directories and Files 3-25
Updating the WebLogic Integration Environment 3-29
Configuring a Custom Java Message Service Queue 3-30
Deploying EJBs and Java Classes for Business Operations 3-34
Adding an EJB to the WebLogic Integration Application Element 3-35
Adding Java Classes to the CLASSPATH 3-38
Adding the JAR File to the setEnv Command File 3-39
Adding the JAR File to the startWebLogic Command File 3-39
Configuring BPM to Support Null Variables 3-40
wli.bpm.server.evaluator.supportsNull=true 3-40
wli.bpm.server.evaluator.supportsNull=false 3-42
Understanding the BPM Security Model 3-43
BPM User Groups and Permissions 3-44
Security Realm Guidelines 3-46
Enabling the RDBMS Security Realm 3-47
Configuring a Custom Security Realm 3-48
Creating the Custom Realm 3-48
Configuring the Caching Realm 3-49
Enabling the WLPI Caching Realm 3-49
Removing the RDBMS Realm 3-50
Updating Passwords 3-50
Updating the system Password 3-51
Updating the BPM Passwords 3-51
Updating the wlisystem Password 3-52

Customizing Mail Session Properties 3-53
Root Directory of a Domain 3-55
Using an Alternate Character Set 3-55
Changing WebLogic Integration Port Numbers 3-56
Step 1. Change the WebLogic Server Listen Port Number 3-56
Changing the Listen Port from the WebLogic Server Administration Console 3-56
Changing the Listen Port by Editing the Configuration File 3-58
Step 2. Changing Other WebLogic Integration Port Numbers 3-58
Changing B2B URI Endpoints 3-59
Changing the Application View Console Port Number 3-60
Changing the B2B Console Port Number 3-61
Changing the WebLogic Integration Shutdown Scripts 3-61
Deploying B2B Without Persistence 3-62

A. WebLogic Integration Sample Configuration Files

config.xml A-2
setEnv A-14
startWeblogic A-15
fileRealm.properties A-17

B. WebLogic Integration Commands

Command List B-2
Environment Variables B-4
Command Descriptions B-11

Index

About This Document

This document provides the basic information you need to get started with WebLogic Integration™. Specifically, it discusses the following topics:

- Chapter 1, "Getting Started," provides basic information about WebLogic Integration including starting and stopping procedures.
- Chapter 2, "WebLogic Integration Administration and Design Tools," describes procedures for starting each design and administration tool and provides references to more detailed information.
- Chapter 3, "Customizing WebLogic Integration," describes how to perform typical customizations.
- Appendix A, "WebLogic Integration Sample Configuration Files," provides developers and system administrators with examples of the files used to configure and deploy the resources and applications that make up WebLogic Integration.
- Appendix B, "WebLogic Integration Commands," provides developers and system administrators with a reference for key WebLogic Integration commands.

What You Need to Know

This document is intended mainly for developers and system administrators responsible for evaluating, deploying, and administering WebLogic Integration solutions.

We assume that you are familiar with WebLogic Server administration and the principles of J2EE application development. Before reading this document, we recommend that you read *Introducing BEA WebLogic Integration*.

e-docs Web Site

BEA product documentation is available on the BEA corporate Web site. From the BEA Home page, click on Product Documentation or go directly to the "e-docs" Product Documentation page at http://e-docs.bea.com.

How to Print the Document

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

A PDF version of this document is available from the BEA WebLogic Integration documentation Home page, which is available on the documentation CD and on the e-docs Web site at http://e-docs.bea.com. You can open the PDF in Adobe Acrobat Reader and print the entire document, or a portion of it, in book format. To access the PDFs, open the BEA WebLogic Integration documentation Home page, click the PDF Files button, and select the document you want to print.

If you do not have the Adobe Acrobat Reader installed, you can download it for free from the Adobe Web site at http://www.adobe.com/.

Contact Us!

Your feedback on the BEA WebLogic Integration documentation is important to us. Send us e-mail at **docsupport@bea.com** if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the WebLogic Integration documentation.

In your e-mail message, please indicate which release of the WebLogic Integration documentation you are using.

If you have any questions about this version of BEA WebLogic Integration, or if you have problems installing and running BEA WebLogic Integration, contact BEA Customer Support through BEA WebSupport at **www.bea.com**. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

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

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

Documentation Conventions

The following documentation conventions are used throughout this document.

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

`monospace text`	Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard.
	Examples:
	`#include <iostream.h> void main ( ) the pointer psz`
	`chmod u+w *`
	`\tux\data\ap`
	`.doc`
	`tux.doc`
	`BITMAP`
	`float`

`monospace boldface text`	Identifies significant words in code. *Example*: `void` **`commit`** `( )`
`monospace italic text`	Identifies variables in code. *Example*: `String` *`expr`*
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. *Example*s: LPT1 SIGNON OR

Convention	Item
{ }	Indicates a set of choices in a syntax line. The braces themselves should never be typed.
[ ]	Indicates optional items in a syntax line. The brackets themselves should never be typed. *Example*: `buildobjclient [-v] [-o name ] [-f file-list]... [-l file-list]...`
|	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.
. . .	Indicates one of the following in a command line: ■ That an argument can be repeated several times in a command line ■ That the statement omits additional optional arguments ■ That you can enter additional parameters, values, or other information The ellipsis itself should never be typed. *Example*: `buildobjclient [-v] [-o name ] [-f file-list]... [-l file-list]...`
. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

CHAPTER

1 Getting Started

BEA WebLogic Integration extends the core infrastructure and features of WebLogic Server to provide a complete solution for developing, deploying, and integrating applications and business processes from within and across the enterprise.

This section provides the information you need to get started with WebLogic Integration. It explains how to configure a WebLogic Server domain to deploy WebLogic Integration resources. It also provides an overview of the components and passwords, and it describes how to start WebLogic Integration in a sample or custom domain.

This section includes the following topics:

- WebLogic Server Domains
- WebLogic Integration Samples Domain
- WebLogic Integration Configuration Templates
- Administration and Design Tools
- About the WebLogic Integration Repository
- WebLogic Integration Users and Passwords
- Domain Configuration Requirements
- Configuring and Starting the Samples Domain
- Starting WebLogic Integration
- Stopping WebLogic Integration
- Next Steps

WebLogic Server Domains

A WebLogic Server domain is a collection of WebLogic Server resources managed as a single unit. A domain includes one or more instances of WebLogic Server and may include WebLogic Server clusters.

WebLogic Integration is a collection of applications and resources—EJBs, Web applications, JDBC connection pools, and so on—that are deployed in a domain to provide a unified platform for developing comprehensive e-business solutions.

Every WebLogic Server domain includes one and only one administration server; any other WebLogic Server instances in the domain are managed servers. Managed servers can be organized into clusters to simplify the task of deploying resources on a set of servers. The configuration for an entire domain is maintained on the administration server. A managed server obtains its configuration at startup from the administration server.

The configuration for a domain is defined in Extensible Markup Language (XML) and stored in a file named `config.xml`. This file is located in the following directory:

`domain``/config.xml`

The name of the directory (that is, the value of *`domain`*) is the name of the domain, which can be located anywhere on your system. For convenience, the scripts required to start and stop an administration server are often located in the same directory as the `config.xml` file.

The administration server is the central point of control for an entire domain. The WebLogic Server Administration Console is a convenient, graphical interface used to carry out administrative tasks. Resource parameters, server instances, and deployments are defined in the `config.xml` file maintained by the administration server, through the WebLogic Server Administration Console.

For general information about the administration of WebLogic Server domains, see "Overview of WebLogic Server Management" in the *BEA WebLogic Server Administration Guide*.

The following sections describe the WebLogic Integration samples domain and the WebLogic Integration configuration templates that can be used with the BEA Configuration Wizard to create custom domains. The samples domain can be used to explore the capabilities and features demonstrated in the sample applications.

As the starting point for developing and testing your own WebLogic Integration solutions, we recommend using a custom domain based on one of the configuration templates provided with WebLogic Integration.

WebLogic Integration Samples Domain

A samples domain is created in which you can run the WebLogic Integration sample applications if you performed one of the following sets of tasks during installation:

- Selected Typical installation
- Selected Custom installation and included WebLogic Integration Samples in your installation

All WebLogic Integration functionality is supported in the samples domain. In addition, Web applications, custom JMS queues, and other resources required to run the samples are configured.

The WebLogic Integration samples domain is configured as follows:

- The domain contains a single WebLogic Server instance. This is the administration server for the domain.

  Note: For information about adding managed servers and deploying WebLogic Integration in a clustered environment, see *Deploying BEA WebLogic Integration Solutions*.

- The default listen port is 7001.
- The PointBase database is installed by default with WebLogic Platform.
- A PointBase database is configured for the sample applications.
- The JDBC connection pool required to connect to the PointBase database is configured.
- A file-based security realm in compatibility mode (FileRealm) is configured to use the PointBase database.

The samples domain directory also contains several files used to control the deployment and configuration of WebLogic Integration resources and applications for the domain:

- `config.xml` file—XML document that describes the configuration of an entire WebLogic Server domain. The resources deployed in the domain are determined by the `config.xml` file.
- `application.xml` file—XML document that determines the deployment order of the J2EE components of WebLogic Integration.
- Start command—Command that starts the server for the domain. The command is platform specific: each domain directory contains either a `startWeblogic.cmd` file (Windows) or a `startWeblogic` file (UNIX).
- Stop command—Command that gracefully shuts down the server for the domain. The command is platform specific: each domain directory contains either a `stopWeblogic.cmd` file (Windows) or a `stopWeblogic` file (UNIX).
- `fileRealm.properties`—File in which the User, Group, and ACL objects that are created when WebLogic Server is started are stored. A file-based security realm in compatibility mode (FileRealm) is the default security realm in a WebLogic Integration environment.

An example of each of these files is provided in Appendix A, "WebLogic Integration Sample Configuration Files."

In addition to the commands described here, a number of others are used to set up the WebLogic Integration environment. These commands are described in Appendix B, "WebLogic Integration Commands."

WebLogic Integration Configuration Templates

The BEA Configuration Wizard, which is described in *Using the Configuration Wizard*, is a tool that can be used to create new WebLogic Server domains. The servers included in a domain created with the Configuration Wizard, and the resources—EJBs, Web applications, JDBC connection pools, and so on—that are deployed on each server, depend on the Configuration Wizard template you select and your responses to the Configuration Wizard prompts.

When you install WebLogic Integration as part of BEA WebLogic Platform, the Configuration Wizard templates described in the following table are available for use with the Configuration Wizard.

Use the template for a . . .	**To configure a domain in which you can develop solutions that use the following functionality provided by WebLogic Integration . . .**
WebLogic Integration (WLI) Domain	Full range: application integration, data integration, business process management (BPM), and B2B integration
Enterprise Application Integration (EAI) Domain	BPM, application integration, and data integration
Business Process Management (BPM) Domain	BPM and data integration

These templates support the creation of the three most commonly used WebLogic Integration development environments. In most situations, you can expect a domain created with one of these templates to meet your evaluation, development, and testing needs.

Detailed information about each template is provided in *Configuration Wizard Template Reference*.

As described in *Using the Configuration Wizard*, when you use the Configuration Wizard to set up the domain configuration stored on the server, you are prompted to assign a domain name (*domain*) and to specify the location in which the *domain* directory will be installed. The files required by the server are installed in the specified location under the *domain* directory.

For example, if you accept the default location, *BEA_HOME*/user_projects, the files required by the administration server are installed in the following directory:

BEA_HOME/user_projects/*domain*

By convention, *DOMAIN_HOME* is used to indicate this complete path to the root of a domain.

Like the WebLogic Integration samples domain, the following files are required by the administration server in a domain created using one of the WebLogic Integration domain templates:

- config.xml file—XML document that describes the configuration of an entire WebLogic Server domain. The configuration of the administration server, managed servers, server clusters, and resources deployed on each server are controlled by the config.xml file stored in the *DOMAIN_HOME* directory on the administration server.
- Start command—Command that starts the administration server for the domain. The command is platform specific: the *DOMAIN_HOME* directory contains either a startWeblogic.cmd file (Windows) or a startWeblogic file (UNIX).
- Stop command—Command that gracefully shuts down the administration server for the domain. The command is platform specific: the *DOMAIN_HOME* directory contains either a stopWeblogic.cmd file (Windows) or a stopWeblogic file (UNIX).
- fileRealm.properties—File in which the User, Group, and ACL objects that are created when WebLogic Server is started are stored. The file realm is the default security realm in a WebLogic Server environment.

An example of each of these files is provided in Appendix A, "WebLogic Integration Sample Configuration Files."

Depending on the template you select to create the domain, additional commands may be included. For a complete description of the files and commands installed by each template, see *Configuration Wizard Template Reference*. For a description of the available WebLogic Integration commands, see Appendix B, "WebLogic Integration Commands."

As you become more familiar with WebLogic Integration, you can customize a domain created with the Configuration Wizard, as described in "Creating and Customizing a New Domain" on page 3-22.

Administration and Design Tools

As described in *Introducing BEA WebLogic Integration*, WebLogic Integration includes the following design and administration tools:

- *WebLogic Server Administration Console*
  The WebLogic Server Administration Console is used to manage the WebLogic Integration resources (applications, EJBs, services, and so forth) deployed on WebLogic Server.
- *Application View Console*
  Application views provide a critical layer of abstraction between the technical implementation of an adapter built with the WebLogic Integration Adapter Development Kit (ADK) and the business data and capabilities of the target application. The Application View Console allows you to access and control adapter resources.
- *WebLogic Integration Studio*
  The Studio is a graphical interface application that facilitates process modeling by making it easy for you to define the actions (tasks, events, and decisions) required to implement the business rules for a process. Actions can invoke applications, interface with human operators, transform data formats, send and receive XML messages, send e-mail, and so on. The Studio allows you to organize actions, using drag-and-drop, into workflows based on a familiar flow-chart paradigm.

- *WebLogic Integration Worklist*
  The Worklist is a graphical application that operates in conjunction with the process engine to notify users of workflow tasks that require action on their part. The Worklist provides workflow developers a convenient way to incorporate user transactions into their workflows.

Note: The Worklist client application is deprecated as of this release of WebLogic Integration. For information about the features that are replacing the Worklist, see the *BEA WebLogic Integration Release Notes.*

- *WebLogic Integration Format Builder*
  Data translation is an integral part of almost any enterprise application integration solution. XML is quickly becoming the standard for exchanging information between applications. The Format Builder application is a tool for creating Message Format Language (MFL) files (or *message formats*) that describe the content and structure of the binary data already used in your organization. In conjunction with the data integration plug-in, these message formats are used by the WebLogic Integration process engine to translate binary data to XML, and vice versa.

- *WebLogic Integration B2B Console*
  The B2B Console is a graphical tool that allows you to define, export, or import the elements required for trading exchange, supply chain management, and collaborative commerce applications. The B2B Console is also used to start and stop the B2B engine and to monitor the state of the B2B engine, trading partner sessions, delivery channels, and other elements involved in an active exchange.

In Chapter 2, "WebLogic Integration Administration and Design Tools," you will find:

- Instructions for starting each tool
- References to the information required to use each tool

About the WebLogic Integration Repository

The WebLogic Integration repository is the database that maintains the information required by WebLogic Integration. The following tables make up the repository.

Table 1-1 WebLogic Integration Repository Tables

This set of tables . . .	Contains . . .
Common tables (also known as the *XML repository*)	XML, DTDs, XLST style sheets, and other entities used by WebLogic Integration
Workflow tables	Static workflows and dynamic run-time information
JMS queue tables	Message state (dynamic)
B2B configuration tables	Static configuration information required to support B2B integration. For more information about the data stored in these tables, see "Working with the Repository" in *Administering B2B Integration*.
Persistence tables	Persistence state (dynamic)

For the information required to initialize the repository for a domain, see "Domain Configuration Requirements" on page 1-11.

WebLogic Integration Users and Passwords

Default users and passwords are configured in two domains:

- The samples domain described in "WebLogic Integration Samples Domain" on page 1-3
- A domain created using one of the templates described in "WebLogic Integration Configuration Templates" on page 1-5

Table 1-2 describes these users and passwords.

Table 1-2 Default User Names and Passwords

User Name	Password	Description	Update Considerations
For the samples domain: `system` You are prompted to specify the server user name when you use the Configuration Wizard to create a domain.	For the samples domain: `security` You are prompted to specify the server password when you use the Configuration Wizard to create a domain.	Login used to: ■ Boot WebLogic Server ■ Access the WebLogic Server Administration Console ■ Access the B2B Console ■ Access the Application View Console	Update this password from the WebLogic Server Administration Console.
`wlisystem`	`wlisystem`	Internal login used by the B2B and process engines	**Warning:** This password *must* be updated using the B2B Console. *Do not* use the WebLogic Server Administration Console.
`admin`	`security`	Default BPM logins used to log on and connect to the process engine and XML repository from the WebLogic Integration Studio, Worklist, and Format Builder clients	Update (add or delete) user names from the Studio. Update an existing user password from the WebLogic Server Administration Console.
`mary`	`password`		
`joe`	`password`		

To maximize system security, you can change the default passwords, as described in "Updating Passwords" on page 3-50.

When you create a domain based on one of the configuration templates, a default FileRealm security realm is configured. The `fileRealm.properties` file reflects the initial configuration of this security realm. In addition to the FileRealm (which always exists) a default RDBMSRealm is also available to support BPM security. For additional information about the BPM security model, and instructions for enabling the RDBMSRealm or setting up an alternate BPM security realm, see "Understanding the BPM Security Model" on page 3-43.

Security issues related to WebLogic Integration B2B functionality are discussed in *Implementing Security with B2B Integration*.

Domain Configuration Requirements

The following database configuration tasks must be performed in any WebLogic Integration domain:

- Define database access information for the domain.

  The `config.xml` file contains domain configuration information which is used to connect to the JDBC connection pools. After a domain is created using the Configuration Wizard, the WebLogic Integration Database Wizard is run, to set the database configuration information. For additional information see the following section, "Configuring the Database for a Domain."

  In the WebLogic Integration samples domain and new domains created with the Configuration Wizard, the JDBC connection pools are configured to use a default PointBase database.

  Warning: Do not update the database configuration information in the `config.xml` file for the domain directly. The `config.xml` file is updated with the database configuration information from the `setDBVars` and `setDBVarsExt` files. These files are located in the *`DOMAIN_HOME`*`\dbinfo\database_type` directory, where *`DOMAIN_HOME`* represents the full pathname of the domain (including the name of the domain) and *`database_type`* specifies the name of the database, for example `oracle`. We recommend that you avoid modifying the `config.xml` file directly because any changes you make may be overwritten if the `switchdb` script or the switch database option of the Database Wizard is run. Update the database configuration information using the Database Wizard. If you must make manual modifications, update the `setDBVars` and `setDBVarsExt` files and then regenerate the `config.xml` file by running the `switchdb` script or the switch database option of the Database Wizard. For more information about the `switchdb` command, see "switchdb" on page B-46.

- Initialize the database repository for the domain.

  Before you can start the server in a WebLogic Integration domain, you must first initialize the database specified in the `config.xml` file. The following table summarizes the initialization requirements.

Table 1-3 Database Initialization Requirements

Task ID	Database Initialization Task	Required for
1	Create the following repository tables: ■ Common tables (also known collectively as the XML repository)—Set of database tables that store XML, DTDs, XLST style sheets, and other entities used by WebLogic Integration ■ Workflow tables—Set of database tables that store static workflows and dynamic run-time information (For a description of the BPM database schema entities see "Database Schema" in *Programming BPM Client Applications*.)	Any domain that supports WebLogic Integration
2	Populate the common tables with the required data (for example XML schemas).	Any domain that supports WebLogic Integration
3	Create the following repository tables: ■ B2B configuration tables—Set of database tables that contain the static information required to configure your environment for B2B integration. The data stored in these tables is discussed in "Working with the Repository" in *Administering B2B Integration*. ■ Persistence tables—Set of database tables in which the persistence state is stored. (dynamic)	Any domain that supports the B2B integration functionality provided by WebLogic Integration
4	Load the system data required to support the WebLogic Integration B2B functionality. Each supported business protocol requires the following: ■ Business protocol definition ■ Set of system logic plug-in definitions	Any domain that supports the B2B integration functionality provided by WebLogic Integration B2B
5	Load the configuration data and workflows required by the samples.	WebLogic Integration samples domain
6	Configure the application view for the End-to-End sample application.	WebLogic Integration samples domain

The following tools are available to assist you in the database initialization process:

- The `RunSamples` command is used to initialize the WebLogic Integration samples database, as described in "Configuring and Starting the Samples Domain" on page 1-14. This command completes all the tasks (1 to 6 in the preceding table) required to initialize the samples database.

- The WebLogic Integration Database Wizard can be used to update the JDBC connection pool or to initialize the database repository. For additional information see the following section, "Configuring the Database for a Domain."

Configuring the Database for a Domain

The WebLogic Integration Database Wizard is a tool that automates database configuration tasks. When you use the Configuration Wizard to create a WebLogic Integration domain, a domain-specific version of the Database Wizard is installed in the new domain. If you have installed the samples domain, a domain-specific version of the Database Wizard is also included in that domain.

The Database Wizard provides the following options:

- *Switch Database*
  This option prompts you to specify a database type and provide other information required to access the database. The wizard updates two items:
  - JDBC connection pools in the `config.xml` file for the domain
  - Domain-specific environment settings that it uses to create the database
- *Create Database*
  This option creates the tables required by the domain and loads data appropriate to the domain.

  For example, if you use the Configuration Wizard to create a domain that supports B2B functionality, the version of the Database Wizard installed in the domain performs Tasks 1, 2, 3, and 4 (described in Table 1-3) when you select this option. If you use the Configuration Wizard to create a domain that supports EAI functionality, the version of the Database Wizard installed in the domain performs Tasks 1 and 2 when you select this option.

  Before creating the required tables, this option also prompts for database access information. The current settings are displayed; you can accept them, or you can specify the information required to connect to and initialize a different database. In other words, this option can be used to both switch and initialize the database for a domain.

  Note: Although a domain-specific version of the Database Wizard is available in the samples domain, the Create Database option does not complete tasks 5 and 6 for that domain. To use the samples domain, you must set up and start the server in that domain by executing the `RunSamples` command. See the

following section, "Configuring and Starting the Samples Domain" on page 1-14.

When you use the Configuration Wizard to create a new domain, you must use the Database Wizard to initialize the repository. For instructions, see "Using the Database Wizard" on page 3-5. After you have initialized the repository, you can start the administration server for the domain, as described in "Starting WebLogic Integration" on page 1-17.

The information you need to set up and start the samples domain is described in the following section.

Configuring and Starting the Samples Domain

When you execute the `RunSamples.cmd` (Windows) or `RunSamples` (UNIX) command for the first time after installing WebLogic Integration, or after you have used the Database Wizard to switch the samples domain to a new database, the command performs the following tasks:

- Creates the required repository tables in the database. See Table 1-3.
- Loads the system data and sample configuration data required for B2B integration
- Starts WebLogic Integration in the samples domain
- Configures the application view for the WebLogic Integration sample application
- Imports the workflow package containing the sample workflows
- Launches your default Web browser and displays the samples launcher page

Note: To successfully display the samples launcher page, your Web browser must meet the requirements described in "Web Browser Configuration Requirements" on page 2-2.

Execute the `RunSamples` command by performing the steps appropriate for your platform:

- On Windows:

  Choose Start→Programs→BEA WebLogic Platform 7.0→WebLogic Integration 7.0→Integration Examples→Start Server and Launch Examples (with dataloader).

  When the `RunSamples` command is executed on a Windows system, a command window is opened.

- On UNIX:

  a. Go to the *SAMPLES_HOME*`/integration/samples/bin` directory, where *SAMPLES_HOME* represents the WebLogic Platform samples directory.

  b. Execute the `RunSamples` command.

  For example, enter the following commands:

  ```
  cd SAMPLES_HOME/integration/samples/bin
  . ./RunSamples
  ```

On both Windows and UNIX platforms, command progress and messages are displayed. Several minutes may be required for all tasks to complete. When the command has completed, the following actions occur:

- The following message is displayed in the command window:

```
Trailer:
REM
REM
=============================================================
REM The program will stop when the server is shut down
REM WARNING: Killing this command will kill the server too!!!
REM
=========================================================+=======
REM

RunSamples execution successful
```

- Your default Web browser is launched, and the samples launcher is displayed.

The `RunSamples` command both configures the samples database as required, and starts WebLogic Integration in the samples domain (the server is booted in the background). After you execute the `RunSamples` command for the first time, and the database is properly configured, you can start WebLogic Integration in the samples domain in either of the following ways:

- Start WebLogic Integration normally, as described in "Starting WebLogic Integration" on page 1-17. After the server starts, you can view the samples launcher page as described in "Starting the Samples Launcher" on page 2-4.
- Execute the `RunSamples` command again.

  When you execute the `RunSamples` command after the samples database is properly configured, you are prompted with the following message:

  ```
  The WebLogic Integration repository has already been created
  and populated, possibly from a previous run of this RunSamples
  script. Do you want to destroy all the current data in the
  repository and create and populate the WebLogic Integration
  repository, again?
  Y for Yes, N for No
  ```

  If you answer `N` to this prompt, the `RunSamples` command skips the steps for creating and populating the repository and runs only the steps for booting WebLogic Server in the samples domain and displaying the samples launcher page in your Web browser.

  If you answer `Y` to this prompt, the `RunSamples` command recreates and repopulates the repository. It then boots WebLogic Server and displays the samples launcher page in your Web browser. When you answer `Y`, the `RunSamples` command destroys all the data currently in the repository and reloads the sample data into the repository. Answer `Y` only when the current sample data has been altered or removed and you want a fresh or unaltered version of the sample data in the repository.

Note: If you quit the browser while the server is still running, you can display the samples launcher page as described in "Starting the Samples Launcher" on page 2-4.

Starting WebLogic Integration

As described in "WebLogic Integration Samples Domain" on page 1-3, when you execute the `startWeblogic.cmd` (Windows) or `startWeblogic` (UNIX) command for a domain, WebLogic Server is started, and the WebLogic Integration applications and resources specified in the `config.xml` file for the domain are deployed on the server.

The following sections provide instructions for executing the start command on a Windows or UNIX system.

Starting WebLogic Integration on Windows

On a Windows system, you can start WebLogic Integration through menus or from the command line.

Note: If you are already running an instance of WebLogic Server that uses the same listen port as the one to be used by the server you are starting, you must stop the first server before executing the start command.

Starting WebLogic Integration Through Menus

To start WebLogic Integration using menus, do one of the following:

- To start the server in the WebLogic Integration samples domain, choose Start→Programs→BEA WebLogic Platform 7.0→WebLogic Integration 7.0→Integration Examples→Start Server (only).

  Note: The first time you start WebLogic Integration in the samples domain, use the `RunSamples` command, as described in "Configuring and Starting the Samples Domain" on page 1-14.

- To start the server in a domain created with the Configuration Wizard, choose Start→Programs→BEA WebLogic Platform 7.0→User Projects→*domain*→Start Server.

A command window is launched, and startup messages are displayed. When the messages described in "Startup Complete Messages" on page 1-20 are displayed, startup has completed successfully.

Starting WebLogic Integration from the Command Line

To start WebLogic Integration from the command line:

1. Choose Start—Run.
2. Enter `cmd` in the Open text box and click OK.
3. Do one of the following:
   - To start the server in a domain created with the Configuration Wizard, go to the *`DOMAIN_HOME`* directory.

     For example, if you created `mydomain` in the `c:\bea\user_projects` directory, enter the following:

     ```
     cd \bea\user_projects\mydomain
     ```

   - To start the server in the WebLogic Integration samples domain, go to the *`SAMPLES_HOME`*`\integration\config\samples` directory, where *`SAMPLES_HOME`* represents the WebLogic Platform samples directory.

     For example, if you installed WebLogic Platform in the `c:\bea\weblogic700` directory, enter the following:

     ```
     cd \bea\weblogic700\samples\integration\config\samples
     ```

4. Start the server by entering:

   ```
   startWeblogic
   ```

When the messages described in "Startup Complete Messages" on page 1-20 are displayed, startup has completed successfully.

Starting WebLogic Integration on UNIX

Note: If you are already running an instance of WebLogic Server that uses the same listen port as the one to be used by the server you are starting, you must stop the first server before executing the start command.

To start WebLogic Integration on a UNIX system:

1. Do one of the following:
   - To start the server in a domain created with the Configuration Wizard, go to the *DOMAIN_HOME* directory.

     For example, if you created `mydomain` in the `/home/bea/user_projects` directory, enter the following:

     ```
     cd /home/bea/user_projects/mydomain
     ```

   - To start the server in the WebLogic Integration samples domain, go to the *SAMPLES_HOME*`/integration/config/samples` directory, where *SAMPLES_HOME* represents the WebLogic Platform samples directory.

     For example, if you installed WebLogic Platform in the `/home/bea/weblogic700` directory, enter the following:

     ```
     cd /home/bea/weblogic700/samples/integration/config/samples
     ```

2. Start the server by entering:

   ```
   . ./startWeblogic
   ```

   Note: If you want to run the server as a background task, use the `nohup` command to ensure that the `startWeblogic` command executes without exiting (or *hanging up*). For example: `nohup startWebLogic &`

When the messages described in the following section are displayed, startup has completed successfully.

Startup Complete Messages

When the following messages are displayed, WebLogic Integration has started successfully:

```
<Month DD, YYYY hh:mm:ss meridianAbbreviation Timezone>  <Notice>
 <BPM> <000000> <Started WebLogic Integration - BPM Server version
7.0>

<Month DD, YYYY hh:mm:ss meridianAbbreviation Timezone>  <Notice>
<Management> <141052> <Application Poller started for development
server.>

<Month DD, YYYY hh:mm:ss meridianAbbreviation Timezone> <Notice>
<WebLogicServer> <000354> <Thread "ListenThread.Default" listening
on port 7001>

<Month DD, YYYY hh:mm:ss meridianAbbreviation Timezone> <Notice>
<WebLogicServer> <000331> <Started WebLogic Admin Server
"myserver" for domain "samples" running in Development Mode>

<Month DD, YYYY hh:mm:ss meridianAbbreviation Timezone> <Notice>
<WebLogicServer> <000365> <Server state changed to RUNNING>

<Month DD, YYYY hh:mm:ss meridianAbbreviation Timezone>  <Notice>
<WebLogicServer> <000360> <Server started in RUNNING mode>
```

Depending on the configuration of the domain, additional messages may be displayed.

To shut down WebLogic Integration gracefully, run the `stopWebLogic` command, as described in "Stopping WebLogic Integration on Windows" on page 1-21. Do not stop the server by closing the command window or by pressing Ctrl+c.

Stopping WebLogic Integration

You can stop your entire WebLogic Integration system—WebLogic Server and all the WebLogic Integration applications and resources deployed in a preconfigured domain—by executing the `stopWeblogic.cmd` (Windows) or `stopWeblogic` (UNIX) command located in the domain directory. The following sections provide instructions for executing the stop command on a Windows or UNIX system.

Stopping WebLogic Integration on Windows

To stop WebLogic Integration, complete the following procedure:

1. Choose Start—Run.
2. Enter `cmd` in the Open text box, and then click OK.
3. Do one of the following:
   - To stop the server in a domain created with the Configuration Wizard, go to the *`DOMAIN_HOME`* directory.

     For example, if you created `mydomain` in the `c:\bea\user_projects` directory, enter the following:

     ```
     cd \bea\user_projects\mydomain
     ```

   - To stop the server in the WebLogic Integration samples domain, go to the *`SAMPLES_HOME`*`\integration\config\samples` directory, where *`SAMPLES_HOME`* represents the WebLogic Platform samples directory.

     For example, if you installed WebLogic Platform in the `c:\bea\weblogic700` directory, enter the following:

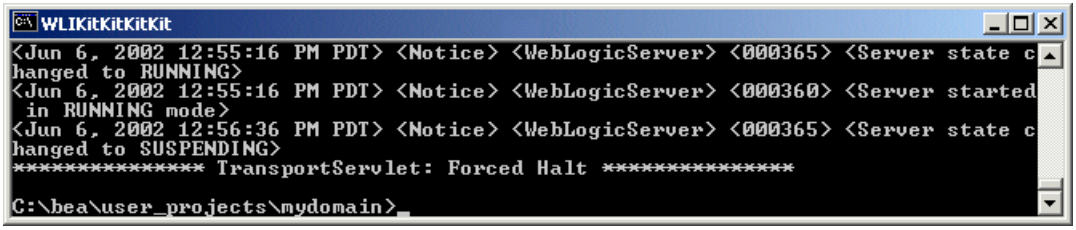
     ```
     cd \bea\weblogic700\samples\integration\config\samples
     ```

4. Stop WebLogic Integration by entering:

   ```
   stopWeblogic
   ```

   Shutdown messages are displayed, as shown in Figure 1-1.

Figure 1-1 Shutdown Sequence Messages



Stopping WebLogic Integration on UNIX

To stop WebLogic Integration on a UNIX system:

1. Do one of the following:
   - To stop the server in a domain created with the Configuration Wizard, go to the *DOMAIN_HOME* directory.

     For example, if you created `mydomain` in the `/home/bea/user_projects` directory, enter the following:

     ```
     cd /home/bea/user_projects/mydomain
     ```

   - To stop the server in the WebLogic Integration samples domain, go to the *SAMPLES_HOME*`/integration/config/samples` directory.

     For example, if you installed WebLogic Platform in the `/home/bea/weblogic700` directory, enter the following:

     ```
     cd /home/bea/weblogic700/samples/integration/config/samples
     ```

2. Stop WebLogic Integration by entering:

   ```
   . ./stopWeblogic
   ```

   Shutdown messages are displayed, as shown in Figure 1-1.

Next Steps

To learn how to start each of the WebLogic Integration administration and design tools, see Chapter 2, "WebLogic Integration Administration and Design Tools."

We highly recommend that you explore the sample applications. This is one of the best ways to learn how you can use WebLogic Integration to connect Web applications, databases, enterprise information systems, processes, and business partners.

If you have not already done so, we recommend that you:

- Configure the samples domain, as described in "Configuring and Starting the Samples Domain."

- Run the WebLogic Integration sample application.

The WebLogic Integration sample (listed as "WLI Sample" on the samples launcher page), showcases how you can use many of the features of WebLogic Integration to integrate existing systems, streamline complex business processes, and connect business partners. *Learning to Use BEA WebLogic Integration* guides you through this example.

CHAPTER

2 WebLogic Integration Administration and Design Tools

This section provides the information you need to get started with the WebLogic Integration design and administration tools. It includes the following topics:

- Web Browser Configuration Requirements
- Starting the Samples Launcher
- Starting the WebLogic Server Administration Console
- Starting the Application View Console
- Starting the Studio
- Starting the Worklist
- Starting the Format Builder
- Starting the B2B Console
- Starting and Stopping the B2B Engine from the B2B Console

Web Browser Configuration Requirements

To use WebLogic Integration you must also use Netscape Navigator 4.7 (or above) or Microsoft Internet Explorer 5.x (or above). In addition you must configure your browser to accept cookies and to bypass the proxy server.

Note: On UNIX systems, you must include the browser executable `netscape` in your `PATH` variable.

Accepting Cookies

Configure your browser to accept cookies by completing the steps for the appropriate browser:

- To configure Netscape Navigator to accept cookies:
  a. Choose Edit—Preferences.

     The Preferences dialog box is displayed.

  b. Select Advanced from the Category list to display the Advanced options.

  c. Select the Accept all cookies option.

- To configure Microsoft Internet Explorer to accept cookies:
  a. Choose Tools—Internet Options.

     The Internet Options dialog box is displayed.

  b. Enable cookies by customizing the security level settings for the applicable Web content zone.

Bypassing the Proxy Server

If your Web browser is configured to use a proxy server to access the Internet, you must bypass the proxy when you access any of the following:

- WebLogic Integration samples launcher page
- WebLogic Server Administration Console
- WebLogic Integration Application View Console
- WebLogic Integration B2B Console

Bypassing the Proxy Server in Netscape

To bypass the proxy server in Netscape:

1. Choose Edit→Preferences to display the Preferences dialog box.
2. In the Category list, click the plus sign to expand the Advanced category.
3. Select Proxies to display the Proxies options.
4. Click View to display the Manual Proxy Configuration options.
5. In the field displayed after Do not use proxy servers for domains beginning with, enter `localhost`, or the IP address of the system running WebLogic Integration.

Bypassing the Proxy Server in Microsoft Internet Explorer

To bypass the proxy server in Microsoft Internet Explorer:

1. Choose Tools→Internet Options to display the Internet Options dialog box.
2. Select the Connections tab.
3. Click the LAN Settings button (at the bottom of the tab) to display the Local Area Network (LAN) dialog box.
4. Check the Bypass proxy server for local addresses option in the Proxy server section.
5. Click OK to save the settings and dismiss the LAN dialog box.
6. Click OK to dismiss the Internet Options dialog box.

Starting the Samples Launcher

The samples launcher page is your gateway to an exploration of the sample applications provided with WebLogic Integration. As described in "Configuring and Starting the Samples Domain" on page 1-14, to start WebLogic Integration in the samples domain for the first time, you must execute the `RunSamples` command. This command configures the samples domain, starts the server, and starts the samples launcher.

Once you have configured your samples domain, you can access the samples launcher at any time. If the server is running, you can open the samples page in your browser or use the shortcuts described in the following procedure.

To start the samples launcher:

1. If it is not already running, start WebLogic Integration in the samples domain, as described in "Starting WebLogic Integration" on page 1-17.
2. Do one of the following:
   - On a Windows or UNIX system, open the following URL in your Web browser:

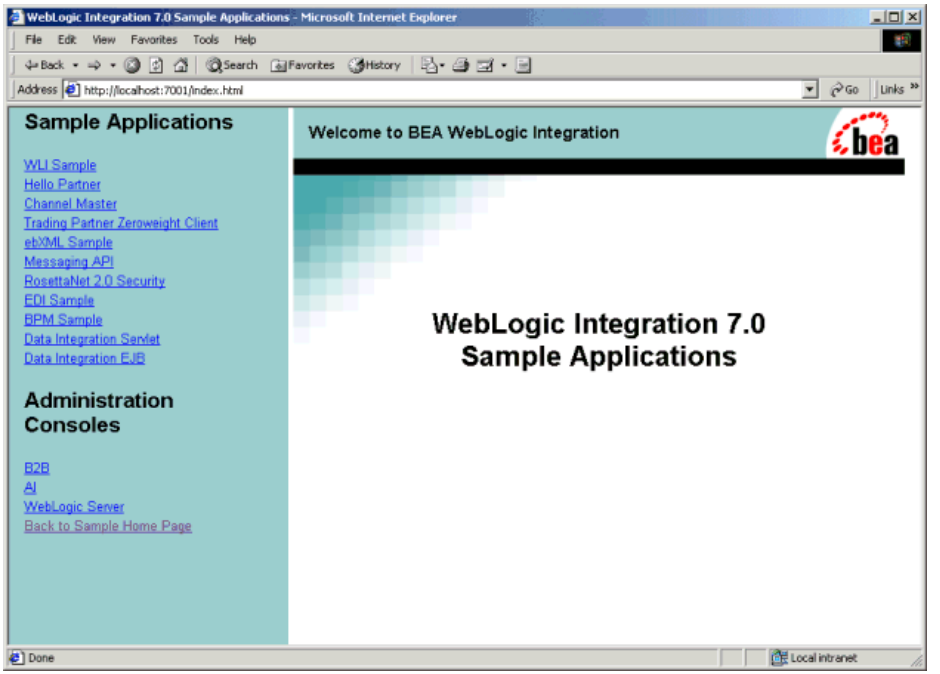
     `http://`*`host`*`:7001/index.html`

     In this URL, *`host`* is the computer name or IP address of the system that is running the server for the samples domain, and `7001` is the listen port configured for the server. The server may be running on any system. If it is running on the local computer, specify `localhost` or `127.0.0.1`.
   - On a Windows system, if the server is running on the local computer, choose Start→Programs→BEA WebLogic Platform 7.0→WebLogic Integration 7.0→Integration Examples→Launcher Examples (start server first).

     This shortcut starts the samples launcher for the server running on the local system.

   The samples launcher page is displayed in your Web browser, as shown in the following figure.

Figure 2-1 WebLogic Integration Samples Launcher



Learning to Use the Samples

The following table lists documents in which you can find information about the sample applications.

For information about this sample . . .	See . . .
WebLogic Integration (WLI)	*Learning to Use BEA WebLogic Integration* **Note:** This sample is based on the XOCP protocol, which is deprecated. See the *BEA WebLogic Integration Release Notes* for details.
Business process management (BPM)	*Learning to Use BPM with WebLogic Integration* (Contains a tutorial for working with business process management (BPM) in WebLogic Integration)
B2B samples: ■ Hello Partner ■ Channel Master ■ Trading Partner Zeroweight Client ■ Messaging API ■ RosettaNet 2.0 Security	*Running the B2B Integration Samples* **Note:** The Hello Partner, Channel Master, and Messaging API samples are based on the XOC protocol, which is deprecated. The Trading Partner Zeroweight Client sample is based on the JSP tag library for the Messaging API, the B2B browser client, and the B2B file-sharing client, all of which are deprecated. See the *BEA WebLogic Integration Release Notes* for details.
Electronic Data Interchange (EDI)	"EDI Sample" in *Using EDI with WebLogic Integration*
Data integration samples: ■ Servlet ■ EJB	"Running the WebLogic Integration Sample Applications" in *Using the Data Integration Plug-In*

Starting the WebLogic Server Administration Console

The WebLogic Server Administration Console allows you to view or modify the configuration of the WebLogic Integration resources deployed on WebLogic Server.

To start the WebLogic Server Administration Console:

1. Start WebLogic Integration, as described in "Starting WebLogic Integration" on page 1-17.
2. Launch the WebLogic Server Administration Console by performing the steps appropriate for your platform:
   - On a Windows or UNIX system, open the following URL in your Web browser:

     `http://host:port/console`

     In this URL, *host* is the computer name or IP address of the system that is running the administration server for the domain, and *port* is the listen port configured for the server. The server may be running on any system. If it is running on the local computer, specify `localhost` or `127.0.0.1`.

     For example, if the administration server is running on the local computer, and it is configured to listen on 7001, enter the following:

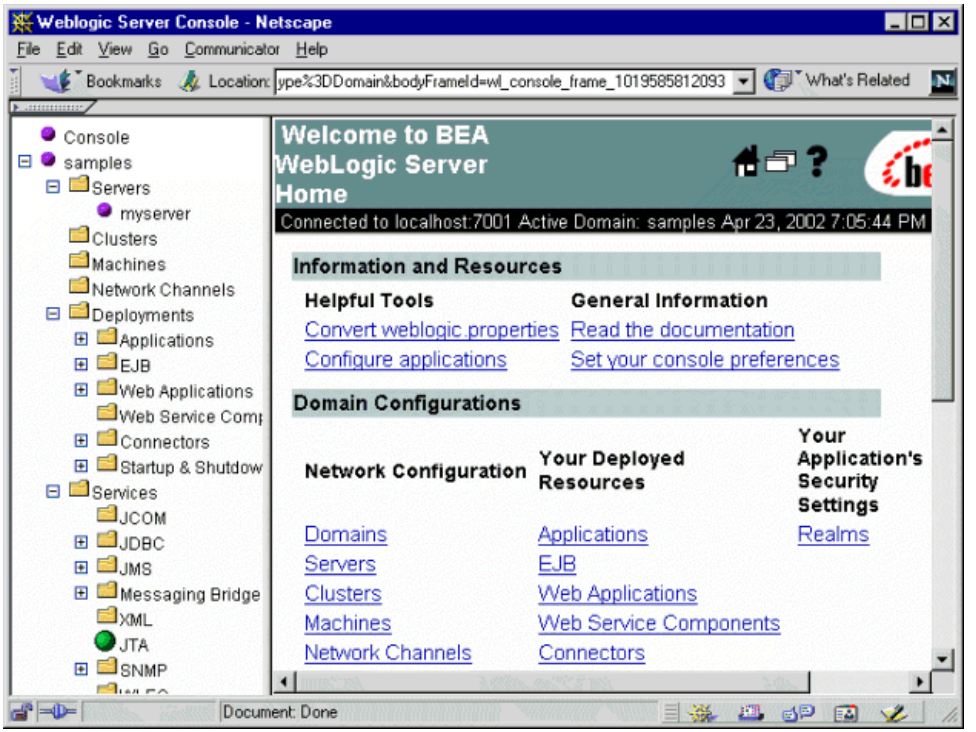
     `http://localhost:7001/console`
   - On a Windows system, if the administration server is running on the local computer, choose Start—Programs—BEA WebLogic Platform 7.0—WebLogic Server 7.0—Server Tour and Examples—Start Admin Console.

     This shortcut starts the console for the administration server running on the local system.
3. When prompted, enter the user name and password for the server.

   Note: See "WebLogic Integration Users and Passwords" on page 1-9 for the default user name and password.

   The WebLogic Server Administration Console is displayed in your Web browser, as shown in the following figure.

Figure 2-2 WebLogic Server Administration Console



Note: Bookmark the WebLogic Server Administration Console for quick access.

Learning to Use the WebLogic Server Administration Console

For detailed information about WebLogic Server administration, see the WebLogic Server 7.0 documentation at the following URL:

`http://e-docs.bea.com/wls/docs70/index.html`

Information about using the WebLogic Server Administration Console is also provided through online help. For help at any time, click the question mark in the upper right-hand corner of the page.

Starting the Application View Console

The Application View Console allows you to define application views for adapters you develop using the WebLogic Integration Adapter Development Kit (ADK). You can define application views only for adapters that are deployed in the active domain. When you define an application view, you are creating an XML-based interface to an enterprise application. In conjunction with the application integration plug-in, this view can then be used to include operations that use the application in the workflows you model in the Studio.

To start the Application View Console:

1. Start WebLogic Integration, as described in "Starting WebLogic Integration" on page 1-17. The domain must be configured to deploy the adapters for which you plan to define views. For information about the deployment requirements for each type of adapter, see *Developing Adapters*.

   Note: The Application View Console is available only if the domain is configured to support application integration functionality.

2. Launch the console by performing the steps appropriate for your platform:

   - On a Windows or UNIX system, open the following URL in your Web browser:

     `http://host:port/wlai`

     In this URL, *host* is the computer name or IP address of the system that is running WebLogic Integration, and *port* is the listen port configured for the server. WebLogic Integration may be running on any machine. If it is running on the local computer, specify `localhost` or `127.0.0.1`.

     For example, if the administration server is running on the local computer, and it is configured to listen on 7001, enter the following:

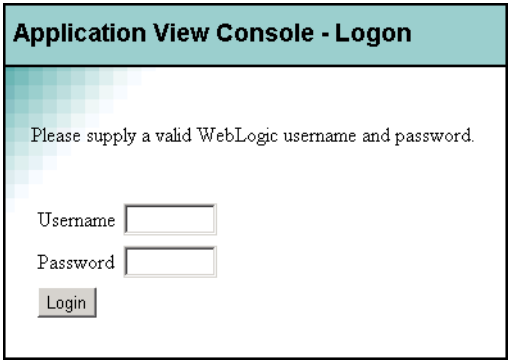
     `http://localhost:7001/wlai`

   - On a Windows system, if WebLogic Integration is running on the local computer, choose Start—Programs—BEA WebLogic Platform 7.0—WebLogic Integration 7.0—Application View Console.

     This shortcut starts the console for the server running on the local system.

3. Enter the server user name and password. Click Login.

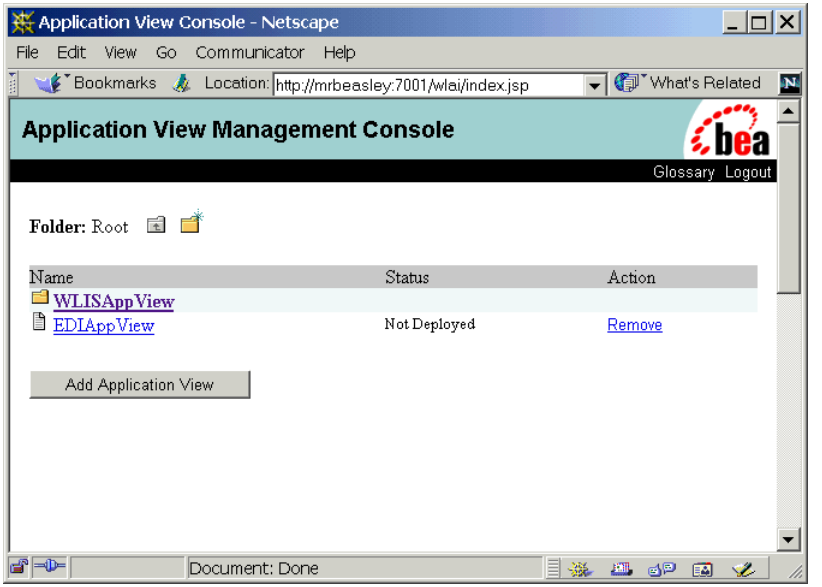
Figure 2-3 Application View Logon



Note: See "WebLogic Integration Users and Passwords" on page 1-9.

The Application View Console home page is displayed. For example, the following figure shows the initial Application View Console page for the samples domain.

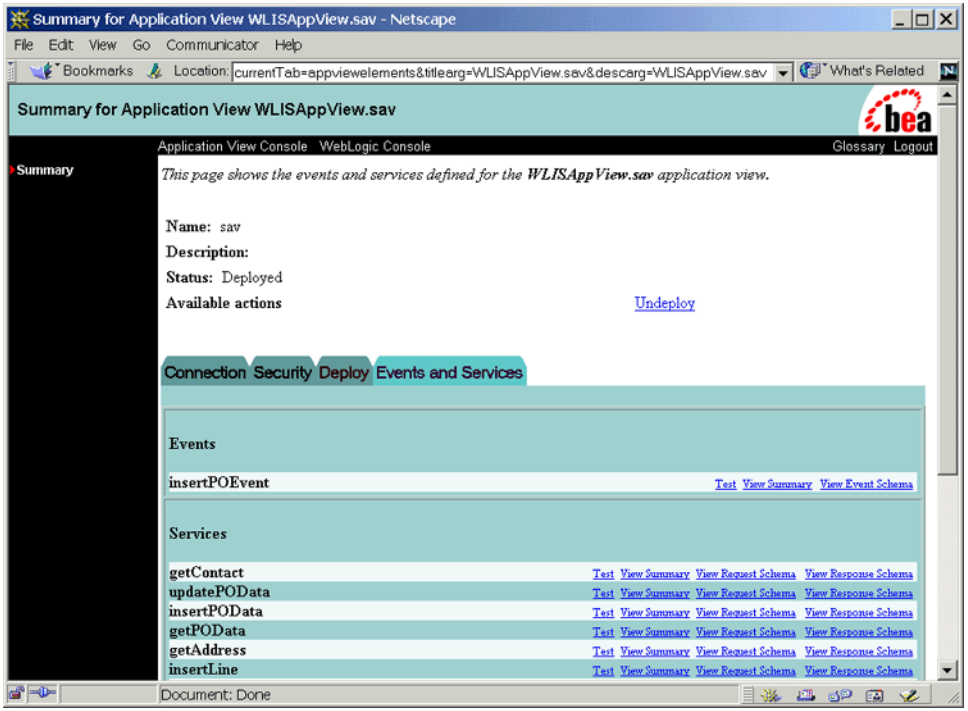
Figure 2-4 Application View Console Page for the Samples Domain



From this page, you can navigate to an existing view or define a new view for a deployed adapter.

Select the name of an Application View to access a page on which a summary of the specified view is displayed. For example, the summary page for the `WLISAppView` application view, which is used in the WebLogic Integration sample, is shown in the following figure.

Figure 2-5 Summary Page for the WLISAppView Application View



Learning to Use Application Views

For information about using adapters built with the BEA WebLogic Integration ADK, see *Using Application Integration*. This document explains how to define and use application view services and events.

Starting the Studio

The WebLogic Integration Studio is the process-modeling client application that allows you to create and monitor workflows. When you execute the `studio.cmd` (Windows) or `studio` (UNIX) command, you are prompted to log in to the process engine running on a local or remote instance of WebLogic Integration.

The following sections provide the information you need to start the Studio:

- Start Procedure for the Studio
- Note About BPM Plug-Ins
- Learning to Use the Studio

Start Procedure for the Studio

To start the Studio:

1. Start WebLogic Integration, as described in "Starting WebLogic Integration" on page 1-17.
2. Launch the Studio by performing the steps appropriate for your platform:
   - Windows:

     Choose Start—Programs—BEA WebLogic Platform 7.0—WebLogic Integration 7.0—Studio.

     A command window is displayed briefly as the `studio.cmd` command is executed. Within a few moments, the Studio window is displayed.
   - UNIX:

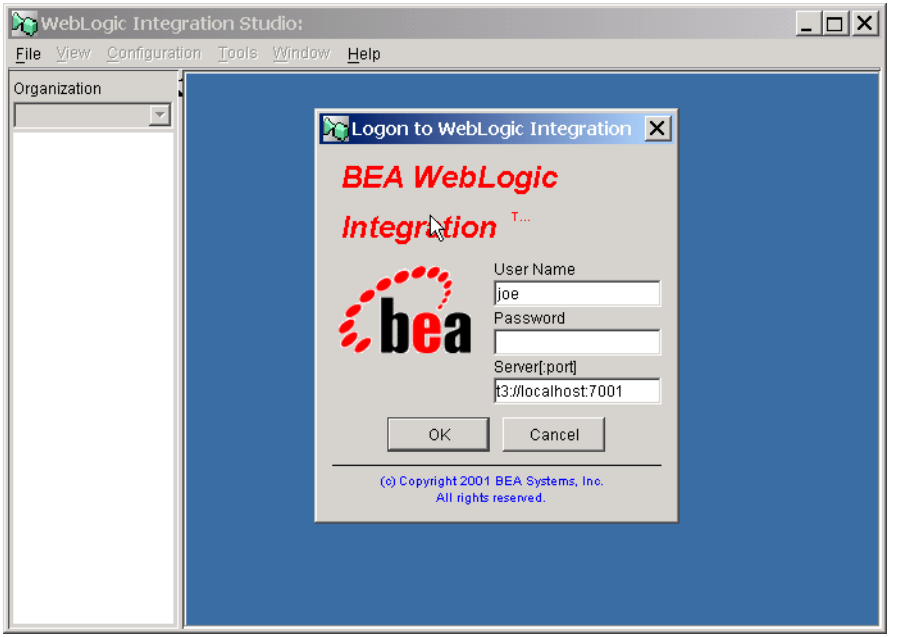
     Go to the *`WLI_HOME`*`/bin` directory and run the `studio` command.

     For example, if WebLogic Integration is installed in the `/home/bea/weblogic700/integration` directory, enter the following:

     ```
     cd /home/bea/weblogic700/integration/bin
     . ./studio
     ```

The logon dialog box is displayed, as shown in the following figure.

Figure 2-6 Studio Logon Dialog Box



3. Enter your user name and password. If you have not been assigned a user name and password for the WebLogic Integration process engine, enter a default user name and password. See "WebLogic Integration Users and Passwords" on page 1-9.

4. In the `Server [:port]` field, specify the system that is running WebLogic Integration, as follows:

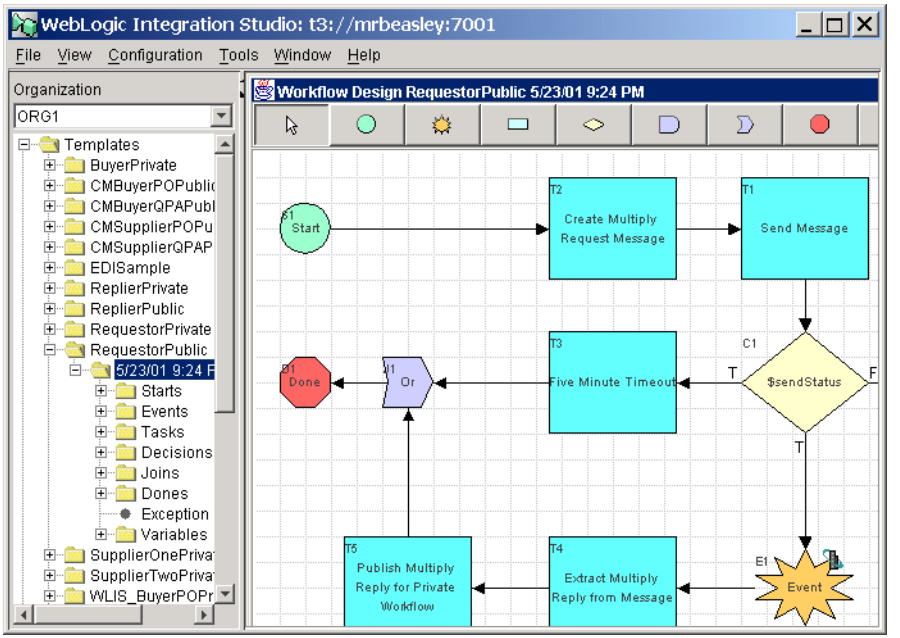
   `t3://`*`host`*`:`*`port`*

   Here, *`host`* is the computer name or IP address of the system, and *`port`* is the listen port configured for the server. If WebLogic Integration is running on the local computer, specify `localhost` or `127.0.0.1`.

5. Click OK.

Once you are connected, the Studio application window is displayed. The following figure shows the Studio application window as it appears when connected to the process engine in the samples domain. In this case, the requestor partner public workflow for the Hello Partner sample is selected.

Figure 2-7 Sample Workflow Displayed in the Studio



Note About BPM Plug-Ins

The functionality of the WebLogic Integration process engine and the Studio client application can be extended by built-in or customer-developed BPM plug-ins. Which plug-ins are deployed depends on the configuration of the active domain. The following table summarizes the deployment of plug-ins in a WebLogic Integration domain.

Table 2-1 Domain-Specific Deployment of Plug-Ins

The plug-in for . . .	Extends Studio functionality so you can . . .	It is deployed in . . .
Data integration	Perform data translation using message formats developed in the Format Builder.	Any WebLogic Integration domain
Application integration	Use application view services and events in a workflow.	Any WebLogic Integration domain that supports application integration functionality
B2B integration	Create workflows for use in trading partner exchanges.	Any WebLogic Integration domain that supports B2B functionality

To view a list of the plug-ins currently deployed, choose Configuration—Plugins from the Studio menu.

For information about developing custom plug-ins, see *Programming BPM Plug-Ins for WebLogic Integration*.

Learning to Use the Studio

Information about using the Studio is provided in the following documents.

To learn . . .	See . . .
The fundamentals	*Using the WebLogic Integration Studio*
	Learning to Use BPM with WebLogic Integration (a step-by-step tutorial)
How to create workflows for use in B2B applications	*Creating Workflows for B2B Integration*
How to use application view services and events in workflows	"Using Application Views in the Studio" in *Using Application Integration*
How to employ data translation in workflow actions	*Using the Data Integration Plug-In*
How to extend functionality by creating your own plug-ins	*Programming BPM Plug-Ins for WebLogic Integration*

Information about using the Studio is also provided in the online help.

For help with . . .	Choose . . .
General information about using the Studio	Help→Help Topics
The application integration plug-in	Help→Plug-In Help→AI Help
The data integration plug-in	Help→Plug-In Help→DI Help

Note: Online help is not available for the B2B integration plug-in. For information on using the B2B integration plug-in, see *Creating Workflows for B2B Integration*.

Starting the Worklist

The Worklist can be used to manage notifications from the system and responses from the user regarding workflow tasks that require human intervention. The following procedure describes how to start the Worklist.

Note: The Worklist client application is deprecated as of this release of WebLogic Integration. For information about the features that are replacing the Worklist, see the *BEA WebLogic Integration Release Notes.*

To start the Worklist:

1. Start WebLogic Integration, as described in "Starting WebLogic Integration" on page 1-17.
2. Launch the Studio by performing the steps appropriate for your platform:
   - Windows:

     Choose Start→Programs→BEA WebLogic Platform 7.0→WebLogic Integration 7.0→Worklist.

     The command window is displayed briefly as the `worklist.cmd` command is executed. Within a few moments, the Worklist application window is displayed.
   - UNIX:

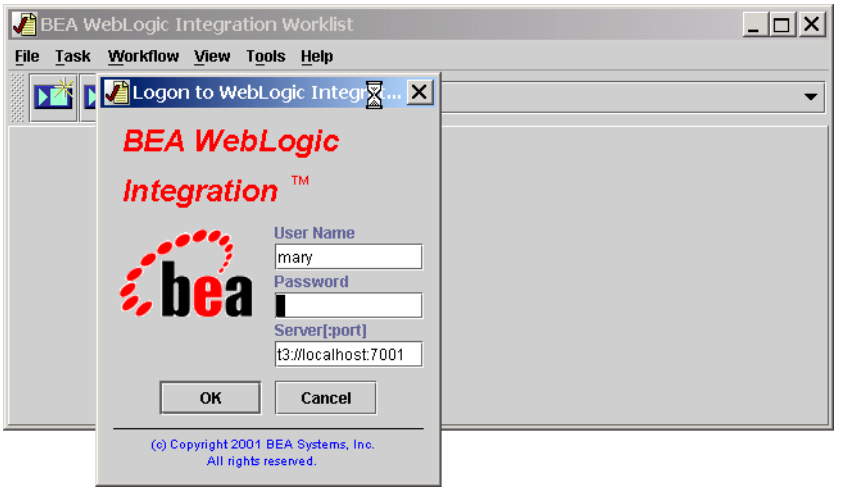
     Go to the *`WLI_HOME`*`/bin` directory and run the `worklist` command.

     For example, if WebLogic Integration is installed in the `/home/bea/weblogic700/integration` directory, enter the following:

     ```
     cd /home/bea/weblogic700/integration/bin
     . ./worklist
     ```

The logon dialog box is displayed, as shown in the following figure.

Figure 2-8 Worklist Logon Dialog Box



3. Enter your user name and password. If you have not yet been assigned a user name and password for the WebLogic Integration process engine, enter a default user name and password. See "WebLogic Integration Users and Passwords" on page 1-9.

4. In the `Server [:port]` field, specify the system that is running WebLogic Integration, as follows:

   `t3://`*`host`*`:`*`port`*

   Here *`host`* is the computer name or IP address of the system, and *`port`* is the listen port configured. If WebLogic Integration is running on the local computer, specify `localhost` or `127.0.0.1`.

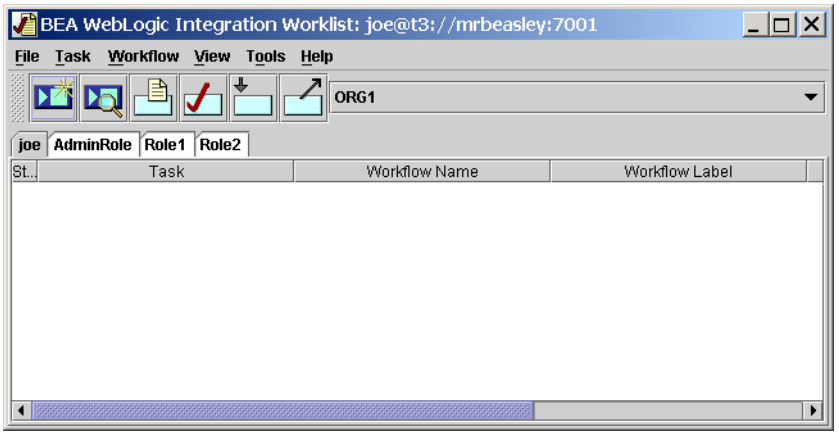
   For example, if the server is running on the local computer, and it is configured to listen on 7001, enter the following:

   ```
   t3://localhost:7001
   ```

5. Click OK.

Once you are connected, the Worklist application window is displayed, as shown in the following figure.

Figure 2-9 Worklist Application Window



Learning to Use the Worklist

To learn how to use the Worklist, see *Using the WebLogic Integration Worklist*.

Starting the Format Builder

The Format Builder allows you to create Message Format Language (MFL) files—also known as message formats—that describe the content and structure of the binary data used by other systems in your organization. In conjunction with the data integration plug-in, these message formats are used by the process engine to translate binary data to XML, and vice versa. This translation capability simplifies the task of incorporating, into your workflows, actions that send or receive data from these systems.

To start the Format Builder:

1. Start WebLogic Integration, as described in "Starting WebLogic Integration" on page 1-17.
2. Launch the Format Builder by performing the steps appropriate for your platform:
   - Windows:

     Choose Start→Programs→BEA WebLogic Platform 7.0→WebLogic Integration 7.0→Format Builder.

     The command window is displayed briefly as the `fb.cmd` command is executed. Within a few moments, the Format Builder application window is displayed.
   - UNIX:

     Go to the *`WLI_HOME`*`/bin` directory and run the `fb` command.

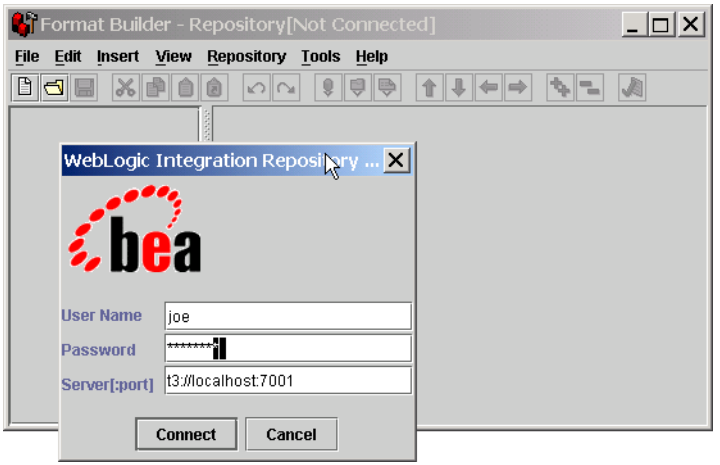
     For example, if WebLogic Integration is installed in the `/home/bea/weblogic700/integration` directory, enter the following:

     ```
     cd /home/bea/weblogic700/integration/bin
     . ./fb
     ```

   The Format Builder application window is displayed.
3. Choose Repository→Logon.

   The logon dialog box is displayed, as shown in the following figure.

Figure 2-10 Format Builder Logon Dialog Box



4. Enter your user name and password. If you have not yet been assigned a user name and password for the XML repository, enter a default user name and password. See "WebLogic Integration Users and Passwords" on page 1-9.

5. In the `Server [:port]` field, specify the system that is running WebLogic Integration as follows:

   `t3://`*`host`*`:`*`port`*

   Here *`host`* is the computer name or IP address of the system and *`port`* is the listen port configured for the server. If WebLogic Integration is running on the local computer, specify `localhost` or `127.0.0.1`.

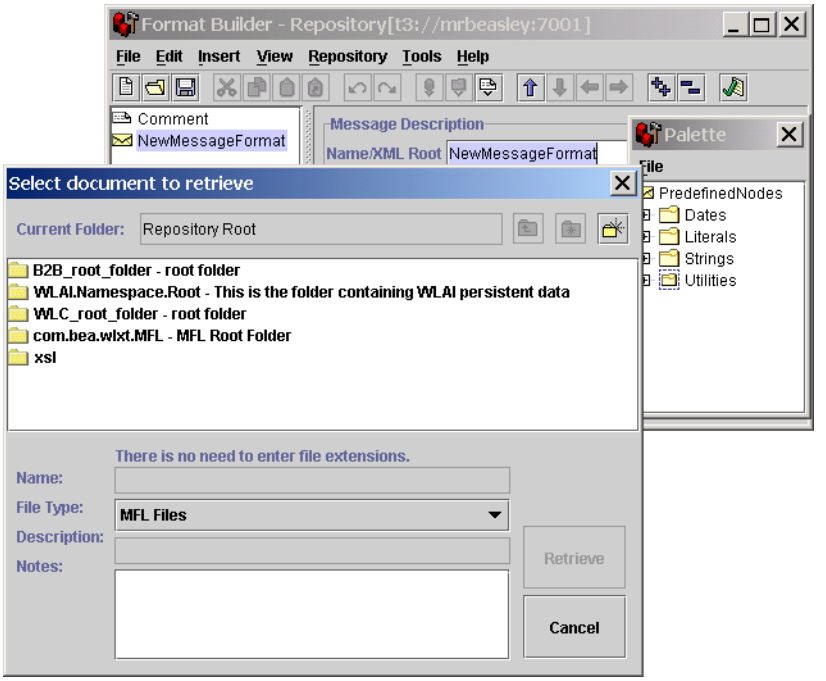
   For example, if the server is running on the local computer, and it is configured to listen on 7001, enter the following:

   ```
   t3://localhost:7001
   ```

6. Click Connect.

Once you are connected, the application window is displayed. You can create new message formats, or retrieve and modify existing formats. For example, to retrieve items from the XML repository, Choose Repository—Retrieve. The retrieve dialog box is displayed, as shown in the following figure.

Figure 2-11 Retrieve Dialog Box in the Format Builder



Learning to Use the Format Builder

To learn how to use the Format Builder, see *Translating Data with WebLogic Integration*.

Information about using the Format Builder is also provided in the online help provided with the Format Builder. To view the help, choose Help→Help Topics.

Starting the B2B Console

The WebLogic Integration B2B Console allows you to:

- Configure B2B engine preferences, trading partners, conversation definitions, collaboration agreements, business protocol definitions, and logic plug-ins
- Export and import configured elements
- Monitor the B2B engine, trading partner sessions, conversations, and collaboration agreements
- Start and stop the B2B engine

To start the WebLogic Integration B2B Console:

1. Start WebLogic Integration, as described in "Starting WebLogic Integration" on page 1-17.
2. Do one of the following:
   - On a Windows or UNIX system, open the following URL in your Web browser:

     `http://host:port/b2bconsole`

     In this URL, *host* is the computer name or IP address of the system that is running WebLogic Integration and *port* is the listen port configured for the server. WebLogic Integration may be running on any system. If it is running on the local computer, specify `localhost` or `127.0.0.1`.

     For example, if the administration server is running on the local computer, and it is configured to listen on 7001, enter the following:

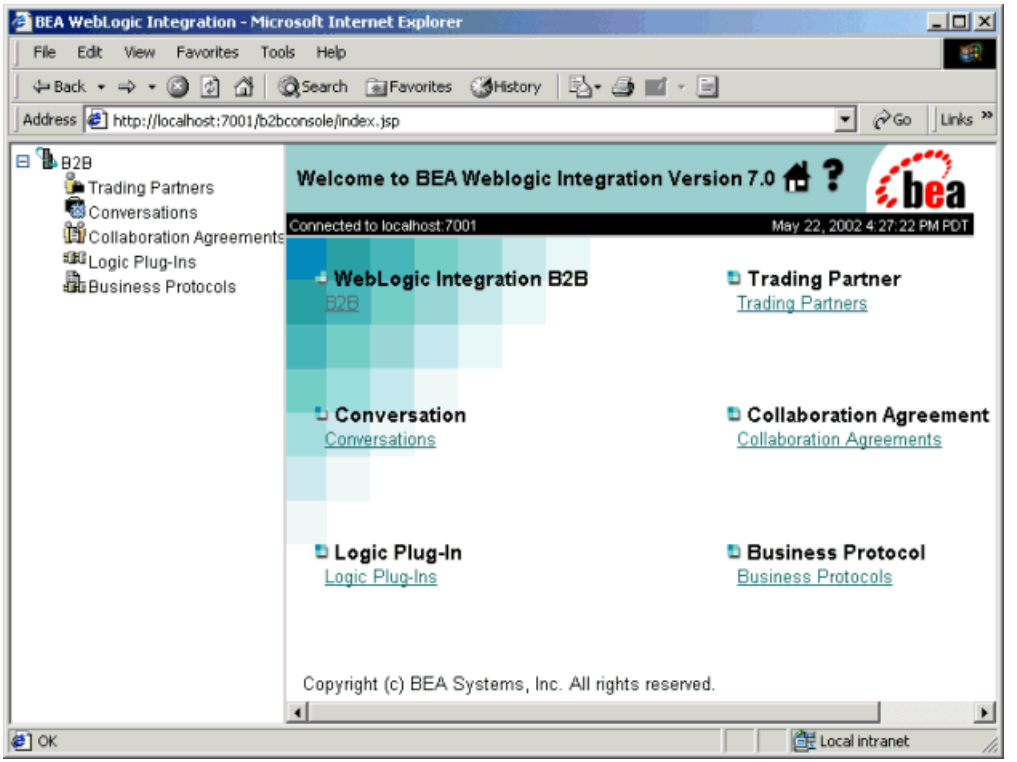
     `http://localhost:7001/b2bconsole`
   - On a Windows system, if WebLogic Integration is running on the local computer, choose Start—Programs—BEA WebLogic Platform 7.0—WebLogic Integration 7.0—B2B Console.

     This shortcut starts the console for the server running on the local system.
3. When prompted, enter the user name and password for the server.

Note: See "WebLogic Integration Users and Passwords" on page 1-9 for the default user name and password.

The B2B Console is displayed in your Web browser, as shown in the following figure.

Figure 2-12 Initial View of the B2B Console



Notes: Bookmark the B2B Console for quick access.

Which items are displayed in the navigation tree (in the left pane) is determined by the B2B preference settings. For example, if the Display advanced configuration controls option is disabled, the Logic Plug-Ins are not displayed. For information about the B2B preferences, see "Setting Preferences" in "Configuring B2B Integration" in *Online Help for the WebLogic Integration B2B Console*.

Learning to Use the B2B Console

For an overview of the configuration requirements for the B2B integration functionality provided by WebLogic Integration, a tour of the B2B console, and information about the B2B repository, see *Administering B2B Integration*.

Information about using the B2B Console, including descriptions of the fields and options on each page, is provided in the online help. For help at any time, click the question mark in the upper right-hand corner of any page displayed by the B2B Console.

Note: The information in the online help is also available as a document entitled *Online Help for the WebLogic Integration B2B Console*.

Starting and Stopping the B2B Engine from the B2B Console

This section provides instructions for shutting down and restarting the B2B engine from the B2B Console.

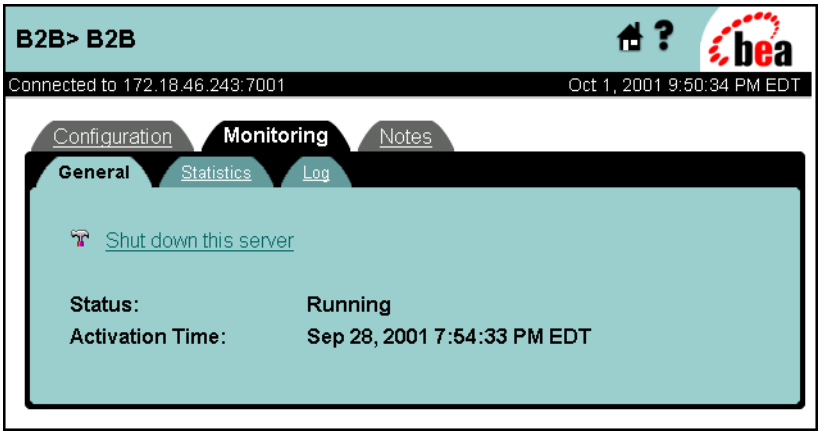
When you use the B2B Console to shut down or restart the B2B engine, only the B2B engine is affected; all other WebLogic Integration applications, including the web server that contains the B2B Console web application itself, continue to run.

Stopping the B2B Engine from the B2B Console

To shut down the B2B engine, while keeping other WebLogic Integration applications and services running:

1. Select B2B from the navigation tree to display the B2B page.
2. First select the high-level Monitoring tab, then the nested General tab, as shown in the following figure.

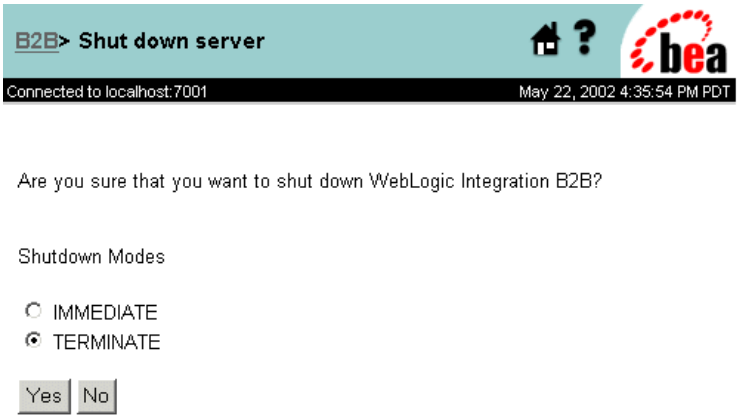
Figure 2-13 General Tab for B2B Monitoring



3. Select Shut down this server.

   The shutdown confirmation page is displayed.

Figure 2-14 Shutdown Confirmation Page



4. Select an appropriate shutdown mode, based on the information in Table 2-2, and click Yes to confirm that you want to shut down the B2B engine.

Table 2-2 Shutdown Mode

Selection	Description
Immediate	Stops all activity immediately.
Terminate	Shuts down the B2B engine, triggers a shutdown of the trading partner sessions, causes all conversations to terminate, and removes all message queues. **Note:** Select Terminate as the shutdown method if you plan on importing repository data using the BulkLoader utility or the B2B Console after the B2B engine has been shut down.

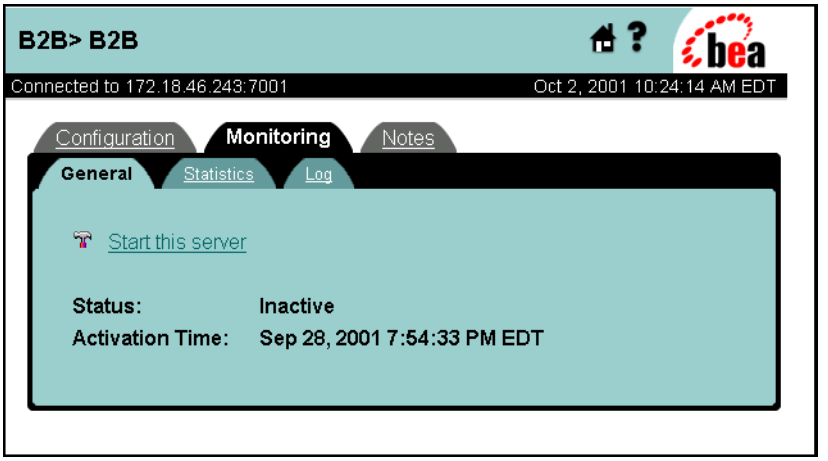
5. The B2B engine is shut down. WebLogic Server and all other WebLogic Integration applications and resources remain active. When you have shut down the B2B engine from the B2B Console, you can restart it, as described in the following procedure.

Restarting the B2B Engine from the B2B Console

To restart the B2B engine after shutting it down from the B2B Console:

1. First select the high-level Monitoring tab, then the nested General tab, as shown in the following figure.

Figure 2-15 General Tab for B2B Monitoring



2. Select Start this server.

   The startup confirmation page is displayed.

Figure 2-16 Startup Confirmation Page

3. Click Yes.

The restart process may take a couple of minutes. You are returned to the Monitoring/General tab when the process is complete.

CHAPTER

3 Customizing WebLogic Integration

The following sections describe how to modify the default installation of WebLogic Integration:

- WebLogic Integration Commands and Supporting Files
- Specifying a New Database for a Domain
- Initializing the Database for a Domain
- Using the Database Wizard
- Updating the Database Configuration from the WebLogic Server Administration Console
- Creating and Customizing a New Domain
- Updating the WebLogic Integration Environment
- Configuring a Custom Java Message Service Queue
- Deploying EJBs and Java Classes for Business Operations
- Configuring BPM to Support Null Variables
- Understanding the BPM Security Model
- Updating Passwords
- Customizing Mail Session Properties
- Root Directory of a Domain
- Using an Alternate Character Set

- Changing WebLogic Integration Port Numbers
- Deploying B2B Without Persistence

For information about more advanced configuration options, such as clustering, see *Deploying BEA WebLogic Integration Solutions*.

WebLogic Integration Commands and Supporting Files

A number of commands and supporting files are provided with WebLogic Integration. Many of the commands are used by the Database Wizard and the `RunSamples` script described in "Domain Configuration Requirements" on page 1-11. Although such commands are not normally invoked in isolation, in some situations you may need to modify a command or troubleshoot the execution of a command. To help you familiarize yourself with the use of these commands, a command reference is provided in Appendix B, "WebLogic Integration Commands."

In addition, examples of the files that control the configuration and startup of a typical WebLogic Integration domain are provided in Appendix A, "WebLogic Integration Sample Configuration Files."

Specifying a New Database for a Domain

Which method you use to specify a new database for a domain depends on:

- Whether the database you are replacing is used for the samples domain or for a custom domain you created using the Configuration Wizard

  Note: For information about creating and customizing a WebLogic Integration domain, see "Creating and Customizing a New Domain" on page 3-22.

- Whether or not you need to preserve the WebLogic Integration repository data stored in the current database for the domain

The following table summarizes the methods used in each situation.

Table 3-1 Database Update Methods

If you are updating the database for . . .	And you . . .	Then . . .
A custom domain	Do not want to preserve repository data in the existing database	1. Start the Database Wizard for the domain. 2. Select the Create Database option. This option prompts for the new database access information, and initializes the database.
	Want to preserve repository data in the existing database	1. Export the information required. 2. Start the Database Wizard for the domain. 3. Select the Create Database option. This option prompts for the new database access information, and initializes the database. 4. Import the information required. See **Note**
The samples domain	Do not want to preserve repository data in the existing database	1. Start the Database Wizard for the domain. 2. Select the Switch Database option to update the configuration. 3. Execute the `RunSamples` command to initialize the new samples database.
	Want to preserve repository data in the existing database	1. Export the information required. 2. Start the Database Wizard for the domain. 3. Select the Switch Database option to update the configuration. 4. Execute the `RunSamples` command to initialize the new samples database. 5. Import the information required.

Note
You also have the option of preserving the existing domain. To do so, simply configure a new domain to use the new database, as described in "Creating and Customizing a New Domain" on page 3-22. You can then export workflow packages or B2B configuration elements, as required, for import to the new domain. When it is no longer needed, you can delete the obsolete domain.

The following table provides the location of the information required to perform the preceding tasks.

Table 3-2 Finding Instructions for Updating a Database

To learn how to perform this task . . .	Refer to . . .
Start the Database Wizard	"Using the Database Wizard" on page 3-5
Execute the `RunSamples` command	"Configuring and Starting the Samples Domain" on page 1-14
Export and import B2B configuration elements	"Importing and Exporting B2B Integration Components" in *Administering B2B Integration*
Export and import workflow packages	"Importing and Exporting Workflow Packages" in *Using the WebLogic Integration Studio*

Initializing the Database for a Domain

Which method you use to initialize the database for a domain depends on whether the database will be used for the samples domain or for a custom domain created with the Configuration Wizard:

- If you are initializing a database for the samples domain, use the `RunSamples` command to initialize the database, as described in "Configuring and Starting the Samples Domain" on page 1-14.
- If you are initializing a database for a domain created with the Configuration Wizard, use the Database Wizard. Select the Create Database option to create and initialize the database and then load the system data into the database. See the following section, "Using the Database Wizard."

  Note: For information about creating and customizing a domain, see "Creating and Customizing a New Domain" on page 3-22.

For additional information about the tasks performed by the `RunSamples` command and the Database Wizard, see "Domain Configuration Requirements" on page 1-11.

Using the Database Wizard

As described in "Configuring the Database for a Domain" on page 1-13, the WebLogic Integration Database Wizard is provided to automate database configuration tasks. A domain-specific version of the command is installed in each WebLogic Integration domain.

The following sections provide the information you need to use the Database Wizard to initialize the database for a domain, specify a new database for a domain, or migrate a WebLogic Integration 2.1 database to WebLogic Integration 7.0:

- Database Connection Information
- Database Wizard Options
- How the Database Wizard Works
- Database Configuration Modes
- Using the Database Wizard in Graphical Mode
- Using the Database Wizard in Console Mode

Database Connection Information

The following table summarizes the information required to connect to each supported database type. The Database Wizard prompts you to provide the values required to connect to the database you are configuring.

Table 3-3 Database Access Information

Database Type	Connection Parameters	Description
PointBase	None	A PointBase database is created for the domain in the *DOMAIN_HOME*\dbInfo\pointbase\db directory. Default access parameters are used.

Table 3-3 Database Access Information (Continued)

Database Type	Connection Parameters	Description
Oracle	Server Hostname	Name of the system that hosts the Oracle Server
	Server Port Number	Oracle port number (the default is 1521)
	User	User ID (schema)
	Password	User password
	Oracle SID	Oracle system identifier
	Net Service Name	Name of the database as it appears in `tnsnames.ora` or in the Names server
Microsoft SQL Server	Hostname	Name of the system that hosts the Microsoft SQL Server
	Port	Microsoft port number (the default is 1433)
	User	Account login name
	Password	Account password
	Database Name	Name of the database defined on the Microsoft SQL Server
	Server Name	The server alias as it appears on the General tab in your Client Network Utility. If no alias is configured, the hostname is used.
Sybase	Hostname	Name of the system that hosts the Sybase server
	Port	Sybase port number (the default is 5000)
	User	Account login name
	Password	Account password
	Database Name	Name of the database defined on the Sybase server
	Server Name	The server name assigned in the Sybase client configuration. This name is set using the Directory Services Editor (DSEDIT1).

Database Wizard Options

The Database Wizard provides the following options:

- *Switch Database*
  Select this option to switch the database for the domain. The environment variables used by commands that are invoked to initialize the database (commands such as `CreateDB` and `RunSamples`) are updated, and the `config.xml` file is modified to reflect the new settings. This option does not initialize the database; it merely configures the environment in preparation for database initialization.
- *Create Database*
  Select this option to initialize the database currently specified, or to switch to a new database and initialize it.

  Note: Although this option is available for the samples domain, it does not complete all the tasks required to initialize the samples database. To initialize the database for the samples domain, you must execute the `RunSamples` command, as described in "Configuring and Starting the Samples Domain" on page 1-14.

How the Database Wizard Works

The WebLogic Integration Database Wizard updates the environment variables set by the `setDBVars` and `setDBVarsExt` commands for the domain, updates the `config.xml` file to reflect the database access information provided, and invokes the commands required to perform the selected tasks.

For detailed information about the environment variables and command files used, see the `wliconfig` command in Appendix B, "WebLogic Integration Commands."

Database Configuration Modes

Like the WebLogic Integration installer, the WebLogic Integration Database Wizard supports the following modes:

- *Graphical mode*
  Both the Windows and UNIX versions of the wizard provide an interactive graphical mode. This mode requires a graphics (windowing) terminal or workstation.
- *Console mode*
  The UNIX version of the wizard provides an interactive, text-only method that can be used to configure the database for a domain on a system that does not include a graphics terminal or workstation.

Follow the procedure appropriate for your system, as described in the following sections:

- Using the Database Wizard in Graphical Mode
- Using the Database Wizard in Console Mode

Using the Database Wizard in Graphical Mode

The following procedure describes how to update your database configuration using the WebLogic Integration Database Wizard in graphical mode.

To configure the database for a domain:

1. If you are using a UNIX platform, go to step 2. If you are using a Windows platform, complete the appropriate step from the following table.

To configure the database for . . .	Choose . . .
The WebLogic Integration samples domain	Start→Programs→BEA WebLogic Platform 7.0→WebLogic Integration 7.0→Integration Examples→Integration Database Wizard
A domain created with the Configuration Wizard	Start→Programs→BEA WebLogic Platform 7.0→User Projects→*domain*→Integration Database Wizard

The Choose Configuration Option dialog box is displayed. Go to step 3.

2. On a UNIX platform, do the following:

   a. Go to the *DOMAIN_HOME* directory.

      For example, if you created `mydomain` in the `/home/bea/user_projects` directory, enter the following:

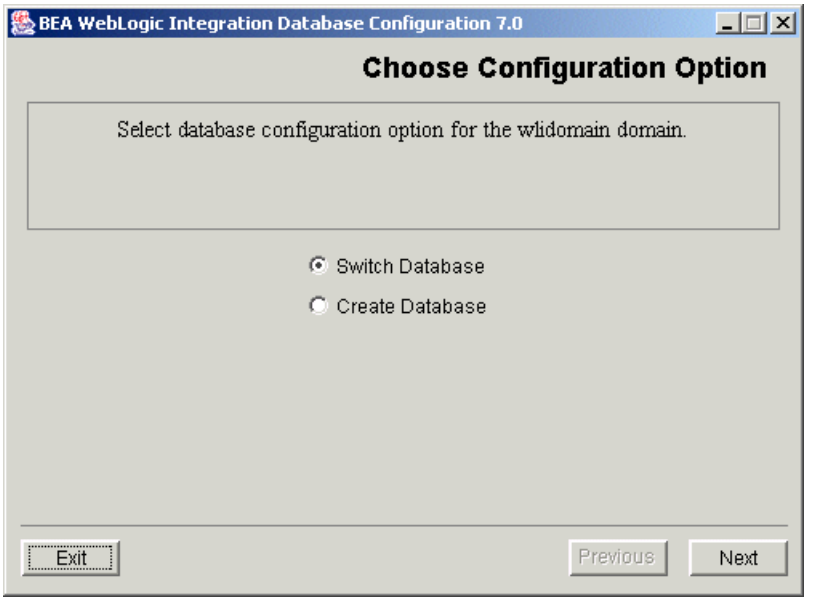
      ```
      cd /home/bea/user_projects/mydomain
      ```

   b. Execute the following command:

      ```
      wliconfig
      ```

   The Choose Configuration Option dialog box is displayed, as shown in the following figure.

Figure 3-1 Choose Configuration Option Dialog Box

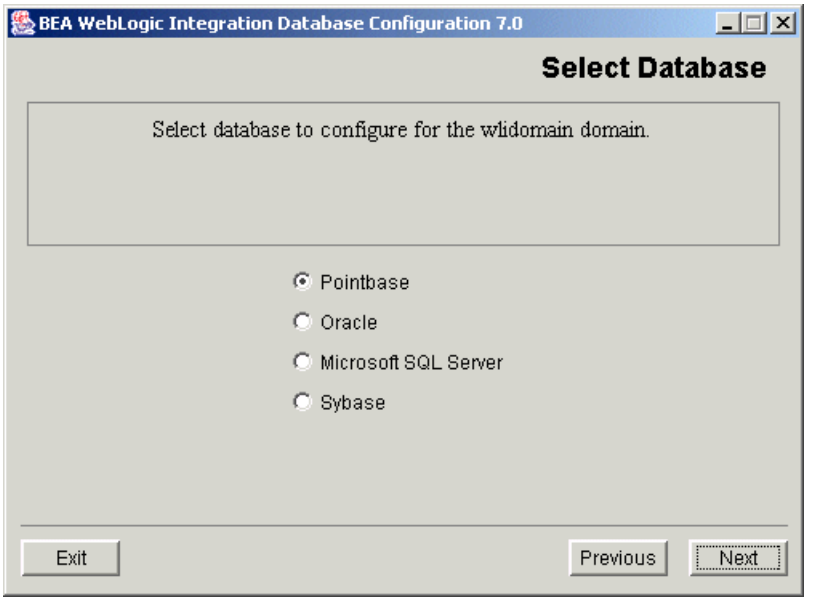


Note: If you are on a UNIX system, and are connecting to a Microft SQL Server database, you will be unable use the wizard to initialize the database because there is no Microsoft SQL client for UNIX. Instead, you will need to initialize the database from a Windows system.

3. Select the appropriate option. See "Database Wizard Options" on page 3-7.
4. Click Next.

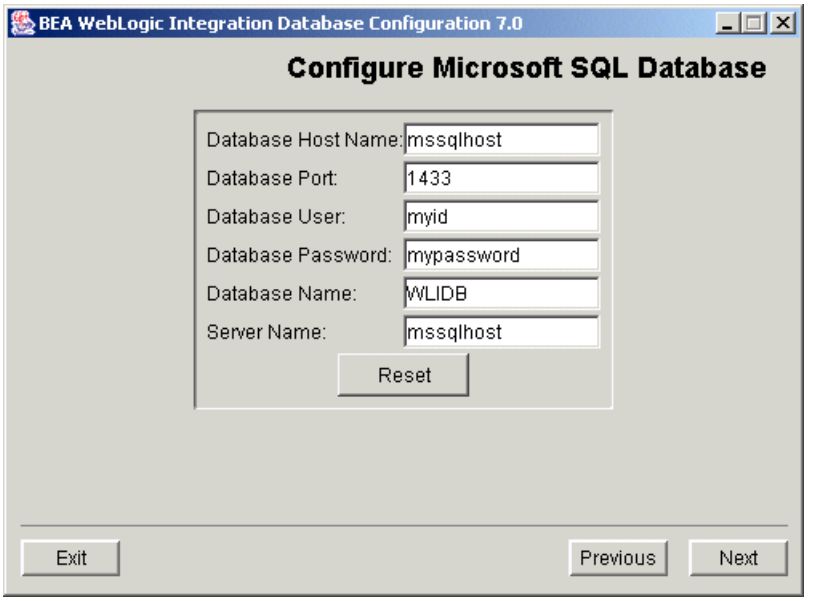
   The Database Selection dialog box is displayed. This dialog box reflects the database that is currently configured for the domain. For example, if you specified the connection parameters for a Pointbase database when you created the domain, Pointbase is selected, as shown in the following figure.

Figure 3-2 Database Selection Dialog Box



5. Do one of the following:
   - To accept the current setting, click Next.
   - To change to a new database, select the desired database, and then click Next.
   - The Database Configuration dialog box is displayed, as shown in the following figure. This dialog box reflects information that was provided when you created the domain, or when you last used the wizard to specify a new database.

Figure 3-3 Database Configuration Dialog Box



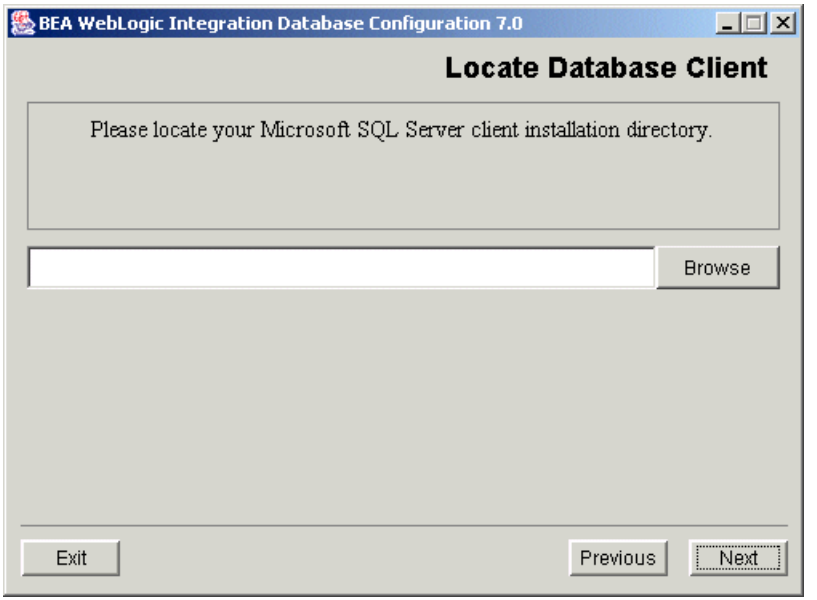
6. Do one of the following:

   - To accept the current values, click Next.
   - To change to a new database, provide the information required to connect to the database, and then click Next. For a summary of the information required for each database type, see "Database Connection Information" on page 3-5.

   The results are as follows

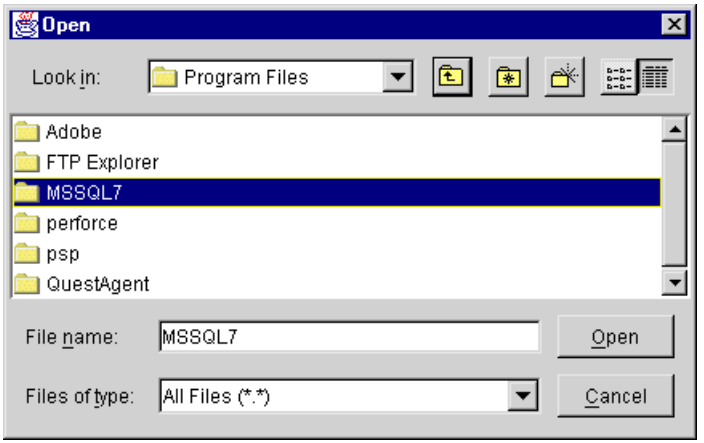
   - If you selected Switch Database, the selected action proceeds. When the changes are complete, the Changes Successful dialog box is displayed, as shown in Figure 3-7. Go to step 12.
   - If you selected Create Database, you are prompted to locate the client installation directory for the database, as shown in the following figure.

Figure 3-4 Locate Database Client Dialog Box



7. Click Browse to display the Open dialog box.
8. Navigate to your database client installation directory. For example, if you are using a Microsoft SQL Server 7.0 client, navigate to the MSQL7 directory, as shown in the following figure. (In this example, the database client installation directory is `c:\Program Files\MSSQL7`.)

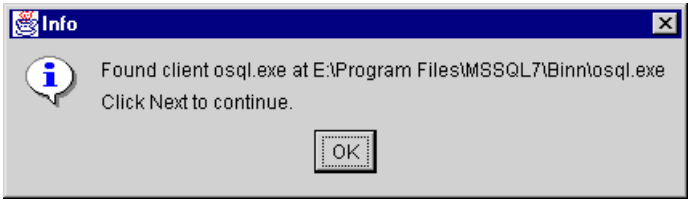
Figure 3-5 Open Dialog Box



9. Click Open.

   A message confirming that the appropriate client application was found is displayed, as shown in the following figure.

Figure 3-6 Confirmation Message



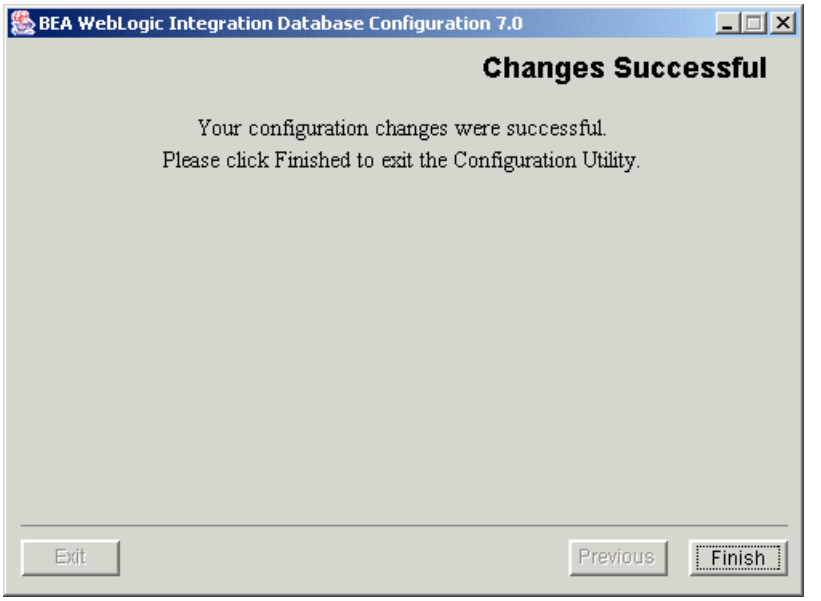
Note: If an error message is displayed indicating that the client application was not found, click OK to dismiss the message, and then Click Browse to locate the correct directory.

10. Click OK to dismiss the message and return to the Locate Database Client dialog box.

11. Click Next to continue.

    When the changes are complete, the Changes Successful dialog box is displayed, as shown in the following figure.

Figure 3-7 Changes Successful Dialog Box



Note: If there is an error in the execution of the selected action, an Error dialog box is displayed. This dialog box indicates the location of a log file. View the log file to determine how to address the problem.

12. Click Finish to exit the installer.

Using the Database Wizard in Console Mode

This section summarizes the console-mode database configuration procedure, a set of steps that emulates the graphical installation procedure.

To start the Database Wizard in console mode, enter the following commands at the prompt:

```
cd DOMAIN_HOME
wliconfig console
```

The following listing shows the console-mode prompts and responses for the Create Database option described in "Database Wizard Options" on page 3-7. (Responses are indicated in bold.)

The prompts that are presented will vary from those shown, depending on your responses to the `Please Select Configuration Option:` and `Please specify the database type to configure:` prompts.

Listing 3-1 Database Wizard Procedure in Console Mode

```
================================================================================
BEA WebLogic Integration Database Wizard 7.0
--------------------------------------------------------------------------------

================================================================================
Configuration Selection
-----------------------------------------------

          1- Switch Database
          2- Create Database

Please Select Configuration Option: 2


================================================================================
Database Selection
-----------------------------------------------


Please specify the database type to configure.

          1- Oracle
         ->2- Microsoft SQL Server
          3- Sybase

Select a number or <Enter> for default(2): 1
Oracle Server Hostname: oraclehost
Oracle Server Port Number: 1521
Oracle User: myuser
Oracle Password: mypassword
Oracle SID: nj908
Oracle Net Service Name: oraclehost.world

Please enter the location of your Oracle client installation: c:\oracle
Searching for Oracle client(Sqlplus.exe)...
```

```
Found client Sqlplus.exe at c:\oracle\sqlplus.exe

        ->1- Use Located Client
          2- Choose new Location

Please choose a number(1): 1

================================================================================
Switching Database please wait...
------------------------------------------------


================================================================================
Creating Database please wait...
------------------------------------------------


================================================================================
Finalizing changes, please wait...
------------------------------------------------


Your configuration changes were successful. Press Enter to exit the Configuration
Utility.
```

Updating the Database Configuration from the WebLogic Server Administration Console

Although the Database Wizard should be used to update the database configuration for any WebLogic Integration domain, there may be circumstances under which it is necessary to update the configuration from the WebLogic Server Administration Console. This section describes the procedure for doing so.

You should be aware that if you use this method to change to a new database, you will be unable to use the wizard, the `createDB` command, or the `RunSamples` command to create the tables in, or to populate, the database. These commands rely on settings for certain environment variables and the database access information stored in the *`DOMAIN_HOME`*`/dbInfo` directory. Unless you update this information manually, the wizard, the `createDB` command, and the `RunSamples` command will not work on the new database.

If you use the WebLogic Server Administration Console to update the database configuration, you can create the tables by executing the database-specific SQL scripts that are located in the *`WLI_HOME`*`/dbscripts` directory. Once you have created the tables, you can use the Bulk Loader or the B2B Console to import system data from the *`WLI_HOME`*`/dbscripts/SystemRepData.xml` file.

Updating the JDBC Connection Pool

To update the WebLogic Integration JDBC connection pool:

1. Start the WebLogic Server Administration Console, as described in "Starting the WebLogic Server Administration Console" on page 2-7.
2. In the navigation tree, choose Services→JDBC→Connection Pools→wlipool.
3. Select the high-level Configuration tab.
4. If it is not already displayed, select the nested General tab.
5. Edit the URL, Driver Classname, and Properties, as required, to customize the settings for your JDBC connection pool. For additional information, see "Database Access Parameters" on page 3-20.
6. Click Apply to save your changes.
7. Close the WebLogic Server Administration Console.
8. Shut down and restart WebLogic Integration to initiate the new settings.

Updating the RDBMS Realm Properties

By default, the RDBMS realm is not configured for the samples domain or a custom domain created with the Configuration Wizard. To configure an RDBMS realm for a domain, you must manually add elements to the `config.xml` file. This procedure is described in "Migrating from the RDBMS Realm" in "Migrating WebLogic Integration 2.1 to WebLogic Integration 7.0" in *BEA WebLogic Integration Migration Guide*. An RDBMS realm must be configured before the RDBMS realm properties can be updated.

To update the RDBMS realm properties:

1. Start the WebLogic Server Administration Console, as described in "Starting the WebLogic Server Administration Console" on page 2-7.
2. In the navigation tree, choose Security→Realms→wlpiRDBMSRealm.
3. Select the high-level Configuration tab.
4. Select the nested Database tab.
5. Edit the Driver, URL, Username, and Password, as required, to customize the settings for your database. For additional information, see "Database Access Parameters" in the following section.
6. Click Apply to save your changes.
7. Close the WebLogic Server Administration Console.
8. Shut down and restart WebLogic Integration to initiate the new settings.

Database Access Parameters

The following table summarizes the database access parameter information required to configure the JDBC connection pool from the WebLogic Server Administration Console.

Table 3-4 JDBC Connection Pool Parameters

Parameter	Description
JDBC Driver	JDBC driver to be used to connect to the database: ■ If you are using the Oracle thin driver, enter: `oracle.jdbc.driver.OracleDriver` ■ If you are using the Sybase jConnect driver, enter: `com.sybase.jdbc.SybDriver` ■ If you are using the PointBase driver, enter: `com.pointbase.jdbc.jdbcUniversalDriver` ■ If you are using the WebLogic jDriver for SQL Server, enter: `weblogic.jdbc.mssqlserver4.Driver`
Database User	Account login name required for connecting to the database server.
Database Password	Password required for connecting to the database server.
URL	URL for the database, as specified in the JDBC driver documentation. The format for the JDBC connection pool URL is discussed in the following section.

Database Access URL Format

Each JDBC connection pool URL includes the following:

- *host* specifies the name of the database server host.
- *port* specifies the port to be used to connect to the database server.
- *database* specifies the database containing the repository tables on the database server.

The following table provides a sample URL for each supported database.

For the following driver . . .	The URL is . . .
Oracle thin driver	`jdbc:oracle:thin:@`*host*`:`*port*`:`*database* For example: `jdbc:oracle:thin:@rdbmshost:1521:wlidb`

WebLogic jDriver for Microft SQL Server	`jdbc:weblogic:mssqlserver4:`*database*`@`*host*`:`*port*`?sql7=true`
	For example:
	`jdbc:weblogic:mssqlserver4:wlidb@rdbmshost:1433?sql7=true`

Sybase jConnect driver	`jdbc:sybase:Tds:`*host*`:5000/`*database* For example: `jdbc:sybase:Tds:rdbmshost:5000/wlidb`
PointBase driver	`jdbc:pointbase://`*host*`:`*port*`/`*database* For example: `jdbc:pointbase://localhost:9092/WLIDB`

Creating and Customizing a New Domain

This section outlines the basic procedure for creating and customizing a WebLogic Integration domain. Detailed information is found in the following documents:

- *Deploying BEA WebLogic Integration Solutions*
- *Using the Configuration Wizard*
- *Configuration Wizard Template Reference*

Preparing to Create a Domain

Before you create a new domain, you should, at a minimum, complete the following tasks:

- Identify the configuration template that most accurately meets your application requirements.

  A set of domain configuration templates is included in your WebLogic Platform installation. These templates are used by the Configuration Wizard to create new WebLogic Server domains. To help you determine which configuration template most accurately meets your application requirements, review "WebLogic Integration Configuration Templates" on page 1-5 and *Configuration Wizard Template Reference*.

  Note: Make sure you select a template that supports WebLogic Integration functionality. By using this type of template, you can ensure that the domain created is based on the WebLogic Server 6.x file-based realm in compatibility mode (FileRealm). The new WebLogic Server 7.0 LDAP-based realm is not supported with WebLogic Integration. If you create a new domain by selecting a WebLogic Server or WebLogic Portal template, the WebLogic Server 7.0 LDAP-based realm will be used in the new domain. For directions on configuring an RDBMS realm for WebLogic Integration, see "Migrating from the RDBMS Realm" in "Migrating WebLogic Integration 2.1 to WebLogic Integration 7.0" in *BEA WebLogic Integration Migration Guide*.

- Review *Using the Configuration Wizard* to find out what kind of information you will need to provide.
- Determine whether this domain is intended for a single-server (standalone) or a multiple-server configuration. The following procedure allows you to create a new domain for a single-server (standalone) configuration. For directions on creating a multiple-server configuration, see "Step 2. Create a WebLogic Integration Domain" in "Configuring a Clustered Deployment" in *Deploying BEA WebLogic Integration Solutions*.

Procedure for Creating and Customizing a Domain for a Single Server

To create and customize a WebLogic Integration domain for a single server:

1. Run the Configuration Wizard by completing the procedure appropriate for your platform:

- On a Windows system you have a choice of three methods:
  - To start the Configuration Wizard from a menu:

    Choose Start→Programs→BEA WebLogic Platform 7.0→Domain Configuration Wizard.
  - To start the Configuration Wizard in graphical mode from the command line:

```
cd c:\bea\weblogic700\integration
setenv.cmd
cd c:\bea\weblogic700\common\bin
dmwiz
```

  - To start the Configuration Wizard in console or text based mode:

```
cd c:\bea\weblogic700\integration
setenv.cmd
cd c:\bea\weblogic700\common\bin
dmwiz console
```

- On a UNIX system you have a choice of two methods:
  - To start the Configuration Wizard in graphical mode:

```
cd /home/joe/bea/weblogic700/integration
. setenv.sh
```

```
cd /home/joe/bea/weblogic700/common/bin
dmwiz.sh
```

- To start the Configuration Wizard in console or text based mode:

```
cd /home/joe/bea/weblogic700/integration
. setenv.sh
cd /home/joe/bea/weblogic700/common/bin
dmwiz.sh console
```

The Choose Domain Type and Name window is displayed.

For more information, see *Using the Configuration Wizard* in the BEA WebLogic Platform document set, available at the following URL:

`http://edocs.bea.com/platform/docs70/confgwiz/index.html`

2. In the Choose Domain Type and Name dialog box, select a domain template that contains the components of WebLogic Integration required for your application.
3. In the Choose Server Type dialog box, select Single Server (Standalone Server).
4. When prompted, continue to provide the information required to create the new domain. When the process is complete, select End Configuration Wizard in the Configuration Wizard Complete dialog box, and then click the Done button to dismiss the Configuration Wizard.
5. (Optional) Save a copy of the generated `config.xml` file. The generated `config.xml` file contains configuration information in the comments. This information is deleted when the instance of WebLogic Server is started. To make a copy of the `config.xml` file, enter the command appropriate for your operating system:

   - On a Windows system:

     ```
     copy config.xml config.xml.backup
     ```

   - On a UNIX system:

     ```
     cp config.xml config.xml.backup
     ```

6. Configure the database connection information for the new domain. Start the WebLogic Integration Database Wizard (`wliconfig`) in the new domain created in steps 1 to 4.

   For example, if you created a domain named `mydomain` in the default location, enter the commands appropriate for your operating system:

   - On a Windows system:

```
cd %BEA_HOME%\user_projects\mydomain
wliconfig
```

- On a UNIX system:

```
cd $BEA_HOME/user_projects/mydomain
wliconfig
```

The Choose Configuration Option window is displayed.

7. Continue to provide the information required to configure the database. For detailed directions, see "Using the Database Wizard in Graphical Mode" on page 3-8.

The following section provides additional information about the directories and files installed when you create a WebLogic Integration domain.

Supporting Directories and Files

The following table lists the directories and files that are included in a domain that supports WebLogic Integration. All the scripts and directories listed in Table 3-5 are located in the domain directory. For example, if you created domain using the Configuration Wizard in the `c:\bea\user_projects\mydomain` directory, the applications directory path would be `c:\bea\user_projects\mydomain\applications`.

Note: Each item is included in *all* domains that support WebLogic Integration, unless otherwise specified.

Table 3-5 Contents of Domain Directory

Directory/File	Content Description
`applications` directory	Application-specific information: the content of this directory depends on your application requirements. For more information about setting up the applications directory, see "Directory Structure" in "Web Applications Basics" in *Assembling and Configuring Web Applications* in the BEA WebLogic Server documentation at the following URL: `http://e-docs.bea.com/wls/docs70/webapp/basics.html`

Table 3-5 Contents of Domain Directory (Continued)

Directory/File	Content Description
`dbInfo` directory	Database-specific information used when creating and initializing the database. **Note:** If you need to update the configuration information in the database variable files in each subdirectory, use the Database Wizard to do so. For more information, see "Using the Database Wizard" on page 3-5. This directory contains the following subdirectories (corresponding to any databases that are supported): ■ `oracle`: Oracle database ■ `pointbase`: PointBase database ■ `mssql`: Microsoft SQL database ■ `sybase`: Sybase database Within each database directory, the following files are defined: ■ `setDBVars` command: basic database variables for the domain ■ `setDBVarsExt` command: extended database variables for the domain For more information about the `setDBVars` and `setDBVarsExt` commands, see "setDBVars setDBVarsExt" on page B-33.
`scripts` directory	Script used to change the current database setting and to process domain-specific files. It contains a single ant call that references the top-level script, *`WLI_HOME`*`/lib/scripts/SwitchDB.xml`. This script is called when the Database Wizard executes the `switchdb` command to update the database configured for the domain. For more information about the `switchdb` command, see "switchdb" on page B-46.
`config.xml` file	Configuration information that describes the WebLogic Server domain and controls the deployed resources. For more information about the contents of the `config.xml` file, see "config.xml" on page A-2.
`fileRealm.properties` file	Default security realm in which the User, Group, and ACL objects that are created when WebLogic Server is started are stored. For more information about the contents of this file, see "fileRealm.properties" on page A-17. For more information about the BPM security model, see "Understanding the BPM Security Model" on page 3-43.

Table 3-5 Contents of Domain Directory (Continued)

Directory/File	Content Description
`SerializedSystemIni.dat` file	Data file used to hash system passwords. This file is associated with the corresponding `fileRealm.properties` file. For more information about this file, see "Protecting User Accounts" in "Configuring WebLogic Security" which, in turn, is available in the *Managing WebLogic Security*. The guide is available from the BEA WebLogic Server documentation set at the following URL: `http://e-docs.bea.com/wls/docs70/secmanage/security7.html`
Start script For example: `startWeblogic.cmd` (Windows) or `startWeblogic.exe` (UNIX)	Script used to start the instance of WebLogic Server in the current domain. Although you can modify the functionality as desired, this script typically performs the following tasks: ■ Sets up the environment by calling the `setenv` command ■ Sets the current domain by calling the appropriate command ■ Loads the database-specific variables into the current process using the `setdbdata` command ■ Removes old log files ■ Executes the Java command that starts WebLogic Server in the domain and deploys WebLogic Integration applications and resources as specified in the `config.xml` file (described previously in this table) **Note:** For more information about the start script and the commands that might be defined in the script, see "startWeblogic" on page B-42. **Note:** You may want to update or add other command-line options, as required by your application. For more information about WebLogic Server command-line options, see "Starting the WebLogic Administration Server from the Command Line" in "Starting the WebLogic Administration Server" which, in turn, is available in "Starting and Stopping WebLogic Servers" in the *WebLogic Server Administration Guide*. The guide is provided, with the BEA WebLogic Server documentation set, at the following URL: `http://e-docs.bea.com/wls/docs70/adminguide/startstop.html`

Table 3-5 Contents of Domain Directory (Continued)

Directory/File	Content Description
Stop script For example: `stopWeblogic.cmd` (Windows) or `stopWeblogic.exe` (UNIX)	Script used for graceful shutdown of the instance of WebLogic Server for the current domain. Although you can modify the functionality as desired, this script typically performs the following tasks: ■ Sets up the environment by calling the `setenv` command ■ Issues a shutdown request to the instance of WebLogic Server running in the domain For more information about the stop script and commands that might be defined in it, see "stopWeblogic" on page B-44. You may want to or add or update command-line options, as required by your application. For more information about WebLogic Server command-line options, see "Starting the WebLogic Administration Server from the Command Line" in "Starting the WebLogic Administration Server," which, in turn, is available in the "Starting and Stopping WebLogic Servers" section of the *WebLogic Server Administration Guide*. The guide is provided in the BEA WebLogic Server documentation set at the following URL: http://e-docs.bea.com/wls/docs70/adminguide/startstop.html
Customized versions of other WebLogic Integration commands	WebLogic Integration commands that may be customized within your application environment. For example, depending on the template used to create the domain, customized versions of the following WebLogic Integration commands may be included in the domain: ■ `checkdomain`: Customized version reflects the current domain in the call to `setdomain` (within the `echo` line). This command is called by the `setdbdata` command. ■ `setdbdata`: Customized version calls the customized version of the `checkdomain` command. This command is called by the `set`*`DomainType`*`Data` command. ■ `set`*`DomainType`*`Data`: Customized version sets environment variables specific to the domain, and calls the customized version of the `setdbdata` command. Use of this file is recommended as a way of setting application integration-specific variables. This command is called by the corresponding start script, described previously in this table. For more information, see "WebLogic Integration Commands" on page B-1.

Updating the WebLogic Integration Environment

The environment variables used by WebLogic Integration are set by the `setEnv.cmd` (Windows) or `setenv.sh` (UNIX) file. This file is located in the WebLogic Integration installation directory (*`WLI_HOME`*). An example of this file is provided in Appendix A, "WebLogic Integration Sample Configuration Files."

The variables in the `setEnv` file are set when you install WebLogic Integration and normally do not need to be updated. If you must update the environment, however, you can do so by completing the following procedure.

To update the WebLogic Integration environment:

1. Do one of the following:
   - On Windows:

     Navigate to the WebLogic Integration installation directory, right-click `setEnv.cmd`, and then select Edit from the shortcut menu to open the file in Notepad.
   - On UNIX:

     Go to the appropriate domain directory and open `setenv.sh` in your preferred text editor.
2. Set the following variables to values appropriate for your environment:
   - `JAVA_HOME`
     Location of your JDK 1.3.1 installation. For example: `C:\bea\jdk131_03`.
   - `BEA_HOME`
     Location of your BEA Home directory. For example: `C:\bea`.
   - `WL_HOME`
     Location of your WebLogic Platform installation. For example: `C:\bea\weblogic700\server`.
   - `WLI_HOME`
     Location of your WebLogic Integration installation. For example: `C:\bea\weblogic700\integration`.

- `SAMPLES_HOME`
  Location of your WebLogic Integration samples. For example: `C:\bea\weblogic700\samples`. For additional information, see Appendix A, "WebLogic Integration Sample Configuration Files."

3. Save your changes and close the file.

When you execute the `startWeblogic` command (as described in "Starting WebLogic Integration" on page 1-17) the `setEnv` command is invoked and the environment variables become effective.

Configuring a Custom Java Message Service Queue

You can create custom Java Message Service (JMS) queues and run the message-driven bean generator utility to generate a deployable Java Archive (JAR) file that listens on the custom queue. Configuring a custom queue involves the following steps:

- Using the WebLogic Server Administration Console to create the custom JMS queue for the selected domain
- Running the `mdbgenerator` utility to generate a message-driven bean (MDB) to listen on the queue
- Updating the WebLogic Integration configuration to recognize the new MDB

For more information about creating JMS queues, see your WebLogic Server 7.0 documentation.

To create the custom JMS queue:

1. Start WebLogic Integration, as described in "Starting WebLogic Integration" on page 1-17.
2. Start the WebLogic Server Administration Console, as described in "Starting the WebLogic Server Administration Console" on page 2-7.

3. In the navigation tree, choose JMS→Servers→WLIJMSServer→Destinations. To create a new JMSQueue, right-click Destinations and select Configure a new JMSQueue. Specify a name and JNDI name for the queue. Accept the default settings for the other fields, or see your WebLogic Server 7.0 documentation for other options.

   Note: Do not use priority settings for ordered queues.

4. Shut down WebLogic Integration, as described in "Stopping WebLogic Integration" on page 1-20.

5. Restart WebLogic Integration, as described in "Starting WebLogic Integration" on page 1-17.

6. When the server is running again, verify that the queue has been created by restarting the WebLogic Server Administration Console and choosing JMS→Servers→WLIJMSServer→Destinations.

To run the `mdbgenerator` utility:

1. Open a command window, and go to the `bin` subdirectory of the WebLogic Integration installation directory.

2. At the command prompt, enter the following:

   `mdbgenerator -queue` *`queue_name`* [`-min` *`number`*] [`-max` *`number`*] [`-order` *`number`*] [`-transact`] [`-validate`] [`-timeout` *`seconds`*] [`-help`]

   The options for the `mdbgenerator` command are listed in the following table.

Table 3-6 mdbgenerator Command-Line Options

Option	Description
`-queue` *`queue_name`*	Name of the custom queue for which you want to generate the MDB. This option is required.
`-min` *`number`*	Minimum number of unordered listeners.
`-max` *`number`*	Maximum number of unordered listeners.
`-order` *`number`*	Number of ordered listeners. The number must be prime and less than or equal to 31.
`-transact`	Sets the transaction to required.

Table 3-6 mdbgenerator Command-Line Options

Option	Description
`-validate`	Turns on XML validation.
`-timeout` *seconds*	Sets the transaction timeout in seconds. This option defaults to 30 seconds. This value is used only if the `transact` flag is not set. If the `transact` flag is set, the utility uses the WebLogic Server transaction timeout value, which can be set using the Administration Console and defaults to 30.
`-help`	Displays command usage syntax.

The *queue_name*`-mdb-generator.jar` file is created in the `bin` directory.

3. Move the generated file to the `lib` subdirectory of your WebLogic Integration installation directory.
4. If it is running in the domain, shut down WebLogic Integration, as described in "Stopping WebLogic Integration" on page 1-20.
5. Open the `config.xml` file for the domain.
6. Locate the following XML Tag:

   ```
   <Application Name="WebLogic Integration"
   Path="c:/bea/weblogic700/integration/lib>
   ```

   After all other `<EJB Component>` elements, add the following line:

   ```
   <EJBComponent Name="MyComponent" Targets="myserver"
   URI="queue_name-mdb-generator.jar"/>
   ```

   Note: The `config.xml` file is case-sensitive. Be sure to enter text using the proper case.

7. Update the `application.xml` file, located in the *path*`/META-INF` directory, where *path* represents the location specified by the path attribute of the application element as shown (highlighted in bold) in step 6. For example, in step 6 the `application.xml` file is located in the following directory:

   ```
   c:\bea\weblogic700\integration\lib\META-INF
   ```

   The order in which the J2EE components are deployed is determined by the order of the components in the `application.xml` file.

Warning: Do not remove or change the order of any WebLogic Integration components listed in the `application.xml` file.

Add `<module>` and `<ejb>` elements for the JAR file, right before the `bpm-init-ejb.jar` module element near the end of the `application` element in the `application.xml` file. For an example, see the code lines highlighted in bold in Listing 3-4.

Listing 3-2 Sample Domain application.xml File

```
<!DOCTYPE application PUBLIC '-//Sun Microsystems, Inc.//DTD J2EE Application
1.2//EN' 'http://java.sun.com/j2ee/dtds/application_1_2.dtd'>
<application>
    <display-name>WebLogic Integration</display-name>
    .
    .
    .
    <module>
       <ejb>queue_name-mdb-generator.jar</ejb>
    </module>

    <!--BPM Initialization Bean must be deployed after BPM plug-ins-->
    <module>
       <ejb>bpm-init-ejb.jar</ejb>
    </module>
</application>
```

8. Save the file.
9. Restart WebLogic Integration, as described in "Starting WebLogic Integration" on page 1-17.

Deploying EJBs and Java Classes for Business Operations

To deploy EJBs and make the associated business operations available for use in WebLogic Integration, you must perform the following steps:

1. Copy or move the EJB JAR file to the *WLI_HOME*/lib directory.
2. Add a deployment descriptor for the EJB to the WebLogic Integration application element.

   For more information about the EJB deployment descriptor files, see "Deploying EJBs to WebLogic Server" in *Programming WebLogic Enterprise Java Beans* in the BEA WebLogic Server documentation set at the following URL:

   http://e-docs.bea.com/wls/docs70/ejb/deploy.html
3. Add the EJB to the WebLogic Integration application element by manually updating the configuration file, config.xml. This file is located in the directory indicated by *DOMAIN_HOME*, where *DOMAIN_HOME* is the full pathname for the root of the domain (for example, c:\bea\user_projects*domain*).

   This step is described in "Adding an EJB to the WebLogic Integration Application Element" on page 3-35. The initial configuration of the WebLogic Integration application element is shown in Figure A-3. (See Appendix A, "WebLogic Integration Sample Configuration Files.")

To deploy custom Java classes and make them available in your WebLogic Integration applications, you must add the Java Archive (JAR) file containing the custom Java classes to your CLASSPATH. This procedure is described in "Adding Java Classes to the CLASSPATH" on page 3-38.

Note: The term *WebLogic Integration application* refers to an application that is developed by customers using WebLogic Integration. Do not confuse this term with the WebLogic Integration or WLI application elements in the config.xml file. The WebLogic Integration application element contains the collection of J2EE components that make up WebLogic Integration. In the WebLogic Integration sample config.xml file, WLI is the name of the application element that contains the collection of J2EE components that make up WebLogic Integration. In config.xml files generated by the

Configuration Wizard, `WebLogic Integration` is the name of the application element that contains the collection of J2EE components that make up WebLogic Integration.

Adding an EJB to the WebLogic Integration Application Element

To add an EJB to the `WebLogic Integration` application element, complete the following steps:

1. Update `config.xml` (the configuration file, located in the *`DOMAIN_HOME`* directory) to specify the deployment descriptor files associated with the `WebLogic Integration` application element.

   To specify the EJB descriptor files, use the `<EJBComponent>` element.

   The following code listing is excerpted from the `config.xml` file for the samples domain. It shows the information required to deploy an EJB called `MyEJB.jar`. This file is located in the *`SAMPLES_HOME`*`/integration/config/samples` directory, where *`SAMPLES_HOME`* represents the WebLogic Platform samples directory. Notable lines of code are shown in **bold**.

Listing 3-3 Adding an EJB to the WebLogic Integration Application Element

```
<Application Name="WebLogic Integration"
  Path="c:/bea/weblogic700/integration/lib" TwoPhase="true">

    <!--Repository-->
    <EJBComponent Name="repository-ejb.jar" Targets="myserver"
       URI="repository-ejb.jar" />

    <!--B2B-->
    <EJBComponent Name="WLI-B2B Startup" Targets="myserver"
       URI="b2b-startup.jar" />
    <WebAppComponent Name="TransportServlet" Targets="myserver"
       URI="b2b.war" />
    <EJBComponent Name="WLI-B2B RN MDB" Targets="myserver"
       URI="b2b-rosettanet.jar" />
    <WebAppComponent Name="b2bconsole" ServletReloadCheckSecs="1"
       Targets="myserver" URI="b2bconsole.war"/>
```

```
    <!--AI-->
    <EJBComponent Name="WLI-AI Server" Targets="myserver"
       URI="wlai-server-ejb.jar"/>
    <WebAppComponent Name="wlai" URI="wlai.war" Targets="myserver"/>
    <EJBComponent Name="WLI-AI Async Processor" Targets="myserver"
       URI="wlai-asyncprocessor-ejb.jar"/>
    <EJBComponent Name="WLI-AI Event Processor" Targets="myserver"
       URI="wlai-eventprocessor-ejb.jar"/>

    <!--BPM-->
    <EJBComponent Name="wlpi-ejb.jar" Targets="myserver" URI="wlpi-ejb.jar" />
    <EJBComponent Name="wlpi-master-ejb.jar" Targets="myserver"
       URI="wlpi-master-ejb.jar" />
    <EJBComponent Name="wlpi-mdb-ejb.jar" Targets="myserver"
       URI="wlpi-mdb-ejb.jar" />

    <!--XT-->
    <EJBComponent Name="wlxtpi.jar" Targets="myserver" URI="wlxtpi.jar"/>

    <!--Plugins-->
    <EJBComponent Name="WLI-B2B RN Plugin for BPM" Targets="myserver"
       URI="wlc-wlpi-plugin.jar" />
    <EJBComponent Name="WLI-B2B EBXML Plugin for BPM" Targets="myserver"
       URI="ebxml-bpm-plugin.jar"/>
    <EJBComponent Name="WLI-AI Plug-In for BPM" Targets="myserver"
       URI="wlai-plugin-ejb.jar"/>
    <WebAppComponent Name="WLAIPlugin" URI="wlai-plugin.war" Targets="myserver"/>
    <WebAppComponent Name="XTPlugin" Targets="myserver" URI="wlxtpi.war"/>

    <!--Samples-->
    <WebAppComponent Name="com.bea.wlpi.SamplePlugin" Targets=""
       URI="sampleplugin.war"/>
    <EJBComponent Name="sampleplugin-ejb.jar" Targets=""
       URI="sampleplugin-ejb.jar"/>
    <EJBComponent Name="wlxtejb.jar" Targets="" URI="WLXTEJB.jar"/>
    <EJBComponent Name="pobean.jar" Targets="myserver" URI="pobean.jar"/>

    <!--BPM Initialization-->
    <EJBComponent Name="WLI-BPM Initialization" Targets="myserver"
       URI="bpm-init-ejb.jar"/>
    <EJBComponent Name="WLI Error Listener" Targets="myserver"
       URI="wli-errorlistener-mdb.jar"/>
</Application>
```

Note: Add your EJBs to the end of the `WebLogic Integration` or `WLI` application element list (before the `</Application>` tag).

Warning: In the sample `config.xml` file, `WLI` is the name of the application element that contains the collection of J2EE components that make up WebLogic Integration. In `config.xml` files generated by the Configuration Wizard, `WebLogic Integration` is the name of the application element that contains the collection of J2EE components that make up WebLogic Integration.

For more information about updating the `config.xml` file, see *BEA WebLogic Server Configuration Reference* in the BEA WebLogic Server documentation set at the following URL:

`http://e-docs.bea.com/wls/docs70/config_xml/index.html`

2. Update the `application.xml` file. This file is located in the *`path`*`/META-INF` directory, where *`path`* represents the location specified by the path attribute of the application element, as shown (highlighted in bold) in Listing 3-3. In the example shown in Listing 3-3, the `application.xml` file is located in the following directory:

   ```
   c:\bea\weblogic700\integration\lib\META-INF
   ```

   The following code listing is excerpted from the `application.xml` file for the samples domain. It shows the information required to deploy an EJB called `MyEJB.jar`.

   The order in which the EJB JAR files are deployed is determined by the order in which the EJB JAR files are listed in the `application.xml` file. If you are adding more than one EJB JAR file, keep in mind the following rule of thumb: in general, if EJB A is dependent on EJB B, then EJB B must be deployed first.

   Warning: Do not remove or change the order of any of the WebLogic Integration EJB JAR or WAR files listed in the `application.xml` file.

   To specify EJB JAR files, add a `<module>` element and an `<ejb>` element for every EJB JAR file, right before the`bpm-init-ejb.jar` module element near the end of the `application` element in the `application.xml` file. For an example, see the code lines highlighted in bold in Listing 3-4.

Listing 3-4 Sample Domain application.xml File

```
<!DOCTYPE application PUBLIC '-//Sun Microsystems, Inc.//DTD J2EE Application
  1.2//EN' 'http://java.sun.com/j2ee/dtds/application_1_2.dtd'>
<application>
    <display-name>WebLogic Integration</display-name>
    .
    .
    .
    <module>
       <ejb>MyEJB.jar</ejb>
    </module>

    <!--BPM Initialization Bean must be deployed after BPM plug-ins-->
    <module>
       <ejb>bpm-init-ejb.jar</ejb>
    </module>
</application>
```

3. Restart the server, as described in "Getting Started" on page 1-1.

Adding Java Classes to the CLASSPATH

You can add the JAR file for your custom Java classes to the server `CLASSPATH` in one of the following ways:

- *Add the JAR file to the* `setenv` *command file.*
  If you add the JAR file to the `WLISERVERCP` environment variable in the `setEnv` command file, it will be available to any WebLogic Integration domain installed on the machine; each time `startWeblogic` is executed, it invokes the `setEnv` command before issuing the Java command that starts the server.
- *Add the JAR file to the* `startWeblogic` *command file.*
  If you add the JAR file to the `CLASSPATH` specified in the `startWeblogic` command file, the Java classes will be available only on the server started by that command file.

Adding the JAR File to the setEnv Command File

To add the JAR file to the `setEnv` command file:

1. Open *`WLI_HOME`*`/setEnv.cmd` (Windows) or *`WLI_HOME`*`/setenv.sh` (UNIX) in your preferred text editor.
2. Locate the following line:

   ```
   set WLISERVERCP=%WLISERVERCP%;%JAVA_HOME%\lib\tools.jar
   ```

3. Add the complete pathname to the end of this statement, using a semi-colon to separate the new entry from the existing entries.
4. Save and close the file.

To make a Java class available to client applications, you can also add a Java class to the `WLICP` variable that is used to set the `CLASSPATH` for WebLogic Integration clients.

Note: You may notice that the `WLISERVERCP` environment variable does not appear in the `startWeblogic` command file. For each domain, a `set`*`DomainType`*`Data` file is invoked after the `setEnv` command file. The value of the `WLISERVERCP` environment variable is transferred to the `SVRCP` environment variable in this command file.

Adding the JAR File to the startWebLogic Command File

To add the JAR file to the `startWeblogic` command file:

1. Open the `startWeblogic.cmd` (Windows) or `startWeblogic` (UNIX) command file in your preferred text editor.
2. Locate the Java command that starts the server:

   ```
   REM Start weblogic
   %JAVA_HOME%\bin\java %DB_JVMARGS% -Xmx256m -classpath %SVRCP%...
   ```

3. Add the following command just above the start server command:

   ```
   set SVRCP=%SVRCP%;MyJAR
   ```

   Here, *`MyJAR`* is the complete pathname for the JAR file that contains the Java classes.
4. Save and close the file.

Configuring BPM to Support Null Variables

Support for null variables can be enabled at server startup by modifying the Java command in the `startWeblogic` command to include the following:

```
-Dwli.bpm.server.evaluator.supportsNull=true
```

When added to the Java command that starts the server, this option turns on null variable support for BPM. If this option is set to `false`, or if it is omitted from the command, then support for null variables is disabled.

After the server has started in a domain, the first time an expression is evaluated, a message indicating the status of null variable support is displayed:

- If null support is enabled, the message `WebLogic Integration BPM server supports NULL` is displayed.
- If null support is disabled, the message `WebLogic Integration BPM server does not support NULL` is displayed.

The server displays this message only once, the first time an expression is evaluated.

The following sections summarize how variable initial values and expression evaluation work when null support is enabled, and when it is disabled.

wli.bpm.server.evaluator.supportsNull=true

When `wli.bpm.server.evaluator.supportsNull=true`, the following rules apply:

- The value `NULL` (case-insensitive) is a valid constant in an expression that yields `null`. For example, set workflow variable `$a` expression `NULL`.

- The operators = and <> can be used to compare null values. All other comparison operations (>=, < ...) on any null value yield false.

  For example, if the workflow variables `$a` and `$c` have a null value, and `$b` has a non-null value, then:

  ```
  $a=NULL       true
  $a=$c         true
  $a<>NULL      false
  $b<>NULL      true
  $b>$a         false
  ```

- Null values cannot be used in arithmetic expressions. For example, `$a * NULL` causes an exception.

- Null values cannot be used in logical expressions. For example, `$a XOR NULL` causes an exception.

- Null values can be used in string concatenation (`StrCat`). For example: `"wli "+ NULL` yields `"wli null"`.

- With regard to variable initial values: *NO variable* is assigned an initial value; all variables have null as the initial value. This differs from the functionality that applies when null support is disabled.

- Any type of variable can be assigned a `NULL` value.

- Null variable values can be monitored and updated in the Studio. When you are monitoring a workflow instance, a null variable value is displayed as `null`. You can set a variable value to `null` in the Update Variable dialog box.

The following section describes the usage rules that apply when `wli.bpm.server.evaluator.supportsNull=false`. It is important that you understand the differences before you set this option to `true`.

If you have workflows that were defined in a previous version, or with null support disabled, these workflows will not work as they did earlier if they rely, in any way, on functionality that is affected. You must reexamine any legacy workflows and initialize appropriate workflow variable values. If you do not do so, the workflow may not run when you enable null variable support.

wli.bpm.server.evaluator.supportsNull=false

When `wli.bpm.server.evaluator.supportsNull=false`, the WebLogic Integration process engine operates exactly as it has operated in previous releases. The following rules apply:

- The value `NULL` (case-insensitive) is *not* valid in any expression. For example, `$a * NULL` is invalid.
- All comparison operations (`=`, `<>`, `>=`, `<` ...) on any null value yield false.

  For example, if the workflow variables `$a` and `$c` have a null value, and `$b` has a non-null value, then:

  ```
  $a=$c       false
  $a<>$c      false
  $b<>$c      false
  $b>$a       false
  ```

- Null values cannot be used in arithmetic expressions. For example, `$b * NULL` causes an exception.
- Null values cannot be used in logical expressions. For example, `$a XOR NULL` causes an exception.
- Null values can be used in string concatenation (`StrCat`). For example: `"wli "+ EventAttribute("not-existing-attribute")` yields `"wli null"`.
- Variables are set to the following initial values:

  ```
  String       ""
  Boolean      false
  Integer      0
  Double       0.0
  Date         Current date
  Entity       null
  Session      null
  Object       null
  XML          null
  ```

- Only string variables can be assigned a null value. Assigning a null value to any other variable type causes an exception.
- You cannot monitor or update null variable values.

Understanding the BPM Security Model

The security model provided by WebLogic Integration for business process management (BPM) functions is distinguished by the following characteristics:

- Users and groups are maintained in a BEA WebLogic Server security realm.
- Organizations are BPM-specific entities that exist outside the security realm.
- Roles map to WebLogic Server groups.

Note: For important background information about the WebLogic Server Security Service, see the following documents:

- *Introducing WebLogic Platform 7.0 Security*—This document provides important information if you are planning to use BPM with other BEA WebLogic Platform components.
- "Using WebLogic Integration Security" (in *Deploying WebLogic Integration Solutions*)—This topic provides an overview of WebLogic Integration security and gives practical advice about configuring the security for your WebLogic Integration domain.

When you use the Configuration Wizard to create a domain that supports BPM functionality, the domain is configured to use a FileRealm security realm. In this type of realm, default users, groups, and access control lists (ACLs) are maintained in the `fileRealm.properties` file, which is located in the domain directory.

A default RDBMSRealm can also be configured. To configure an RDBMS realm for a domain, you need to do the following:

- Manually add elements to the `config.xml` file. This procedure is described in "Migrating from the RDBMS Realm" in "Migrating WebLogic Integration 2.1 to WebLogic Integration 7.0" in *BEA WebLogic Integration Migration Guide*.
- Add the path for the `rdbmsrealm.jar` file to your classpath. By default, this JAR file is located in the *`WLI_HOME`*`\lib` directory.

If you configure this realm, then users and groups are maintained in the database configured for the WebLogic Integration repository.

The BPM-defined ACL objects are managed differently from users and groups. Even if you enable the RDBMSRealm, ACL objects are stored in the FileRealm (the `fileRealm.properties` file) and must be managed through the WebLogic Server Administration Console.

You can use the existing FileRealm, enable the RDBMSRealm, or configure an alternate security realm. For an overview of the supported WebLogic Server security realm types, see "Security Fundamentals" at the following URL:

`http://e-docs.bea.com/wls/docs70/security/concepts.html`

No matter what type of security realm you configure, you can continue to create users, organizations, and roles through the WebLogic Integration Studio, as long as the security realm you configure conforms with the guidelines described in "Security Realm Guidelines" on page 3-46.

Note: WebLogic Integration and WebLogic Server user names and passwords (users created using the WebLogic Integration Studio) can contain any character from the JDK-supported character sets, including international characters.

The procedure for enabling the RDBMSRealm is provided in "Enabling the RDBMS Security Realm" on page 3-47.

The procedure for creating a custom security realm is provided in "Configuring a Custom Security Realm" on page 3-48. If you are creating another type of security realm (for example, NT or UNIX), see the WebLogic Server documentation for instructions.

BPM User Groups and Permissions

To support BPM functionality, default users (described in "WebLogic Integration Users and Passwords" on page 1-9) are organized into three types of user groups:

- Permission groups—Members are defined by the types of operations they can perform.
- System groups—These are required by the process engine.
- Role groups—Members are defined by their BPM-defined roles. When you create a role in the Studio, a corresponding role-specific group is created in the security realm.

The following tables list the default groups and the members assigned to them.

Table 3-7 Permission Groups and Members

Permission Group	Members
AdministerUser	admin, joe, mary, guest, wlisystem
ConfigureComponents	admin, joe, mary, guest, wlisystem
ConfigureSystem	admin, joe, mary, guest, wlisystem
CreateTemplate	admin, joe, mary, guest, wlisystem
DeleteTemplate	admin, joe, mary, guest, wlisystem
ExecuteTemplate	admin, joe, mary, guest, wlisystem
MonitorInstance	admin, joe, mary, guest, wlisystem
UpdateTemplate	admin, joe, mary, guest, wlisystem

Table 3-8 System Groups and Members

System Group	Members
everyone	admin, joe, system, mary, guest, wlisystem
wlpiAdministrators	admin, joe, system, mary, guest, wlisystem
wlpiUsers	admin, joe, system, mary, guest, wlisystem

Table 3-9 Role Groups and Members

Role Group	Members
AccountingCDE	admin, joe
CustomerServiceCDE	admin
Role1Org1	admin, joe, mary
Role2Org1	admin, joe, mary

Table 3-9 Role Groups and Members (Continued)

Role Group	Members
Role1Org2	admin, joe, mary
Role2Org2	admin, joe, mary
ShippingCDE	admin, mary

Security Realm Guidelines

You can create an alternate security realm as long as it conforms to the following guidelines:

- BPM users must be WebLogic Server users.
- You must create the following users: `system`, `admin`, and `wlisystem`.
- You must create the following WebLogic Server groups:
  - All permission groups, as listed in Table 3-7
  - `wlpiUsers` group
- The `wlisystem` user must be a member of all the required groups.
- Any user who requires access to the WebLogic Integration process engine or the XML repository must be a member of the `wlpiUsers` group.

For procedures for creating an alternate security realm, see your WebLogic Server 7.0 documentation.

Once a manageable security realm is populated in conformance with the preceding guidelines, and configured as described in the following section, you can create additional users, roles, and organizations, and assign permissions to users and roles through the WebLogic Integration Studio. For information and procedures, see "Administering Data" in *Using the WebLogic Integration Studio*.

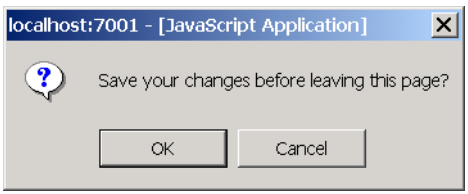
Enabling the RDBMS Security Realm

To change the security realm from the default FileRealm to the RDBMSRealm:

1. Make sure that the database used for the RDBMS realm is also populated with the WebLogic Integration repository. (You can populate the database with the repository by using the Integration Database Wizard, which you can start by choosing Start→Programs→BEA WebLogic Platform 7.0→WebLogic Integration 7.0→Integration Examples→Integration Database Wizard.)
2. Configure the RDBMSRealm by manually adding elements to the `config.xml` file. This procedure is described in "Migrating from the RDBMS Realm" in "Migrating WebLogic Integration 2.1 to WebLogic Integration 7.0" in *BEA WebLogic Integration Migration Guide*.
3. Start WebLogic Integration, as described in "Starting WebLogic Integration" on page 1-17.
4. Start the WebLogic Server Administration Console, as described in "Starting the WebLogic Server Administration Console" on page 2-7.
5. In the navigation tree, select Compatibility Security.
6. Select the Filerealm tab.

   Note: If the following message is displayed, click Cancel to dismiss.

   Figure 3-8 Leaving the Page? Confirmation

7. From the Caching Realm drop-down list, select `wlpiCachingRealm`.
8. Click Apply.
9. Shut down WebLogic Integration, as described in "Stopping WebLogic Integration" on page 1-20.
10. Go to the *`DOMAIN_HOME`* directory for the domain you are updating.

11. Back up the file `fileRealm.properties` by copying and renaming it.
12. Open the original `fileRealm.properties` file in a text editor.
13. Delete all entries beginning with `user` and `group`, and save the file.

    Note: Do not delete any entries that begin with `acl`. ACL objects are always maintained in the `fileRealm.properties` file, whether or not the RDBMSRealm is enabled.

14. Restart WebLogic Integration, as described in "Starting WebLogic Integration" on page 1-17.

Configuring a Custom Security Realm

Configuring a custom security realm involves the following tasks:

1. Creating the custom realm.
2. Configuring the caching realm (`wlpiCachingRealm`).
3. Enabling the caching realm.
4. Removing the RDBMSRealm (optional).

The following sections provide a procedure for each of these tasks.

Creating the Custom Realm

To create a custom security realm:

1. Start WebLogic Integration, as described in "Starting WebLogic Integration" on page 1-17.
2. Start the WebLogic Server Administration Console, as described in "Starting the WebLogic Server Administration Console" on page 2-7.
3. In the navigation tree, choose Compatibility Security—Realms and click Realms.
4. In the right pane, select Configure a new Custom Realm link.

   The Configuration tab for the new realm is displayed.

5. Enter a name for the custom realm. For example, `CUSTOMRealmForNetscapeDirSvr`.

   Note: You can ignore the other tabs and attributes; the settings are not used.

6. Create the custom realm, as described in "Installing a Custom Security Realm" in "Compatibility Security" in the WebLogic Server *Administration Console Online Help*.

7. Click Create.

Configuring the Caching Realm

To configure the caching realm:

1. In the navigation tree, choose Compatibility Security→Caching Realms→wlpiCachingRealm.
2. Select the realm you created from the Basic Realm drop-down list.
3. In the right pane, select Configure a New Caching Realm.
4. Deselect the Case Sensitive Cache option.
5. Click Apply.

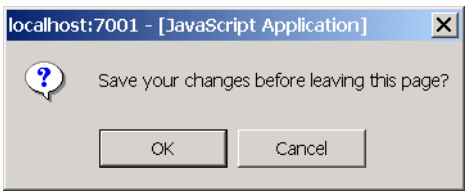
Enabling the WLPI Caching Realm

To enable the WLPI caching realm:

1. In the navigation tree, select Security.
2. Select the Filerealm tab.

   Note: If the following message is displayed, click Cancel to dismiss.

   Figure 3-9 Leaving the Page? Confirmation



3. Select `wlpiCachingRealm` from the Caching Realm drop-down list.
4. Click Apply.

Removing the RDBMS Realm

Note: This step is optional.

To remove the RDMBS realm:

1. In the navigation tree, choose Security—Realms.
2. Click the garbage can icon to the right of `wlpiRDBMSRealm`.

Note: If you remove a RDBMS realm that is referenced by a Caching Realm, you might need to edit the `config.xml` file and modify the value of the CachingRealm's `BasicRealm` attribute to point to your new custom realm.

Updating Passwords

To ensure system security, update the default passwords listed in "WebLogic Integration Users and Passwords" on page 1-9, as described in the following procedures.

Warning: The `wlisystem` user name and password are stored in the WebLogic Integration repository and used in the WebLogic Integration run-time environment. This type of password *must* be updated using the B2B Console; d*o not* use the WebLogic Server Administration Console for this purpose. If you modify the `wlisystem` password through the B2B Console, the password is updated in both the WebLogic Integration repository and the security realm.

Note: WebLogic Integration and WebLogic Server user names and passwords (users created using the WebLogic Integration Studio) can contain any character from the JDK-supported character sets, including international characters.

Updating the system Password

The password for the `system` login for the active domain can be changed through the WebLogic Server Administration Console, as described in the following procedure.

To change the password, complete the following procedure:

1. Start the WebLogic Server Administration Console, as described in "Starting the WebLogic Server Administration Console" on page 2-7.
2. Select Users from the navigation tree to open the Users page.
3. In the Change a User's Password section, enter, in the name field, the name of the user who owns the password you want to change.

   For example, to change the system password, enter `system`.
4. Enter the existing password in the Old Password field.
5. Enter the new password in the New Password field.
6. Retype the new password in the Confirm Password field.
7. Click Change to update the password.
8. Modify the password specified in the `startWebLogic.cmd` (Windows) or `startWebLogic` (UNIX) for the domain.

Warning: The `startWeblogic.cmd` (Windows) or `startWeblogic` (UNIX) command is configured for automatic login. If you change the password for `system`, you must modify the password specified in this file. See the sample `startWebLogic` file in Appendix A, "WebLogic Integration Sample Configuration Files."

Updating the BPM Passwords

You can update the passwords for the BPM users `admin`, `joe`, `mary`, and any new users you create in the Studio application, through the WebLogic Server Administration Console, as described in the previous section.

Updating the wlisystem Password

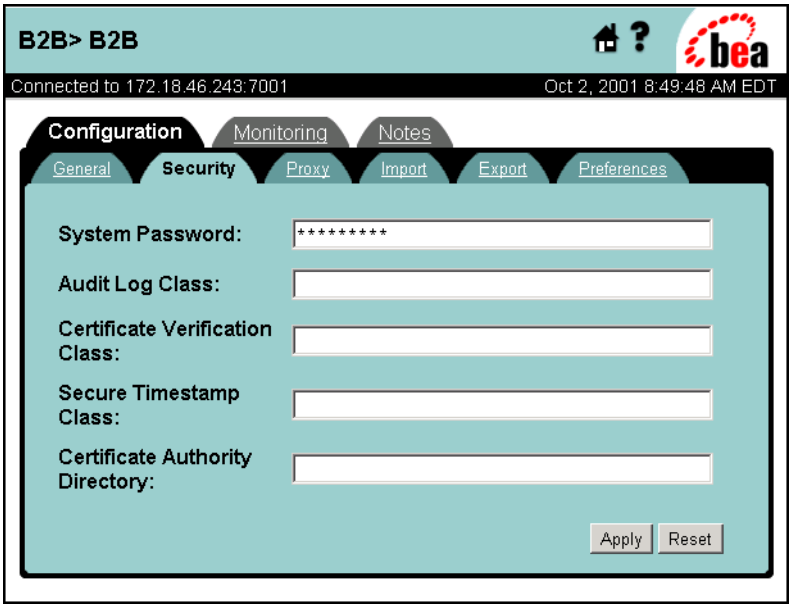
The password for the `wlisystem` user can be changed only through the B2B Console, as described in the following procedure. *Do not* update this password from the WebLogic Server Administration Console.

To update the password for the `wlisystem` user:

1. Start the B2B Console, as described in "Starting the B2B Console" on page 2-24.
2. Select B2B from the navigation tree.
3. If it is not already selected, select the high-level Configuration tab, and then select the nested Security tab.

   The Security tab is displayed, as shown in the following figure. The content of the System Password field is the password for the `wlisystem` user.

Figure 3-10 B2B Configuration Security Tab



4. To update the `wlisystem` password, highlight the current content of the System Password field, and then carefully type the new password over it.

5. Click Apply to update the `wlisystem` password.

   This procedure updates the `wlisystem` password in both the WebLogic Integration repository and the security realm.

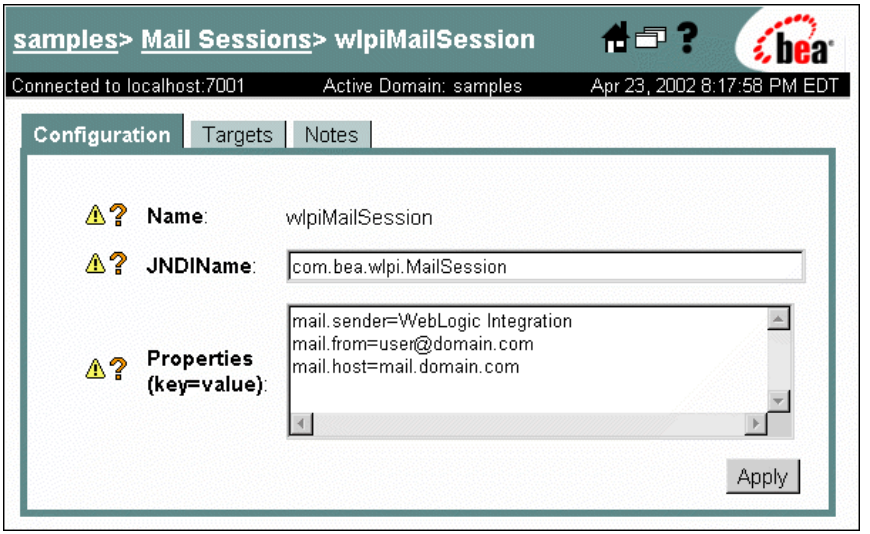
Customizing Mail Session Properties

The mail session properties you specify when you use the Configuration Wizard to create a domain are used to set up a basic mail session configuration for the domain. You can modify the mail session configuration by updating existing settings, or by adding properties. The following procedure describes how to update the mail session properties for any domain.

To update the mail session properties for any domain, complete the following procedure:

1. Start the WebLogic Server Administration Console for the domain, as described in "Starting WebLogic Integration" on page 1-17.
2. Choose Services→Mail→wlpiMailSession. The wlpiMailSession Configuration tab is displayed, as shown in the following figure.

Figure 3-11 wlpiMailSession Configuration Tab



3. Edit the contents of the Properties field as required.

   The following table provides descriptions of the properties that can be set.

Table 3-10 Mail Session Properties

Property	Description
`mail.from`	E-mail address from which e-mail is sent by a workflow.
`mail.host`	Domain name of the mail or SMTP server.
`mail.sender`	Name of the sender that appears in an e-mail when it is received. The default is WebLogic Integration.
`mail.user`	If required by the mail server, the user name used for logging on.
`mail.password`	If required by the mail server, the password used for logon authentication.

4. Click Apply when your changes are complete.

Root Directory of a Domain

Some components of WebLogic Integration need access to particular files, such as the XML DTD, in the root directory. By default, the `startWeblogic` script for WebLogic Integration uses the home directory of the domain (*`DOMAIN_HOME`*) as the domain's root directory.

For example, suppose that, using the Configuration Wizard, you create a domain in `c:/bea/user_projects/mydomain`. That directory is now known as *`DOMAIN_HOME`*. The `startWeblogic.cmd` script, located in the *`DOMAIN_HOME`* directory, sets the root directory (also known as the run-time directory) to the same directory (`c:/bea/user_projects/mydomain`) using the `weblogic.RootDirectory` property.

Using an Alternate Character Set

If you are running WebLogic Integration in another locale, no special configuration is required, as long as your database, your operating system, WebLogic Integration, and the WebLogic Integration client applications (the Studio, Worklist, and Format Builder), are all running in the same locale.

The only change required after installation is to your B2B preferences. To enable the B2B Console to allow input from an alternate character set, you must verify that language and character set preferences are properly defined. For details about setting the language and the default character set for the language, see "Setting Preferences" in "Configuring B2B Integration" in the *Online Help for the WebLogic Integration B2B Console*.

Changing WebLogic Integration Port Numbers

Two WebLogic Integration instances running on a single machine cannot share the default port number (7001). If you are running two instances on one machine, you must change the port number used by one of them.

Warning: You may need to change parameters other than the WebLogic Server listen port number. For more information, see Table 3-11.

The following steps contain instructions for changing the WebLogic Integration port numbers in your application:

- Step 1. Change the WebLogic Server Listen Port Number
- Step 2. Changing Other WebLogic Integration Port Numbers

Step 1. Change the WebLogic Server Listen Port Number

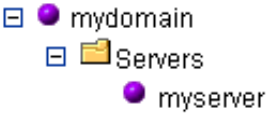
The WebLogic Server listen port number specifies the dedicated TCP/IP port at which the server listens for connections. To change the listen port configured for a server, follow the steps in "Changing the Listen Port from the WebLogic Server Administration Console" on page 3-56 or "Changing the Listen Port by Editing the Configuration File" on page 3-58.

Changing the Listen Port from the WebLogic Server Administration Console

To change the listen port from the WebLogic Server Administration Console:

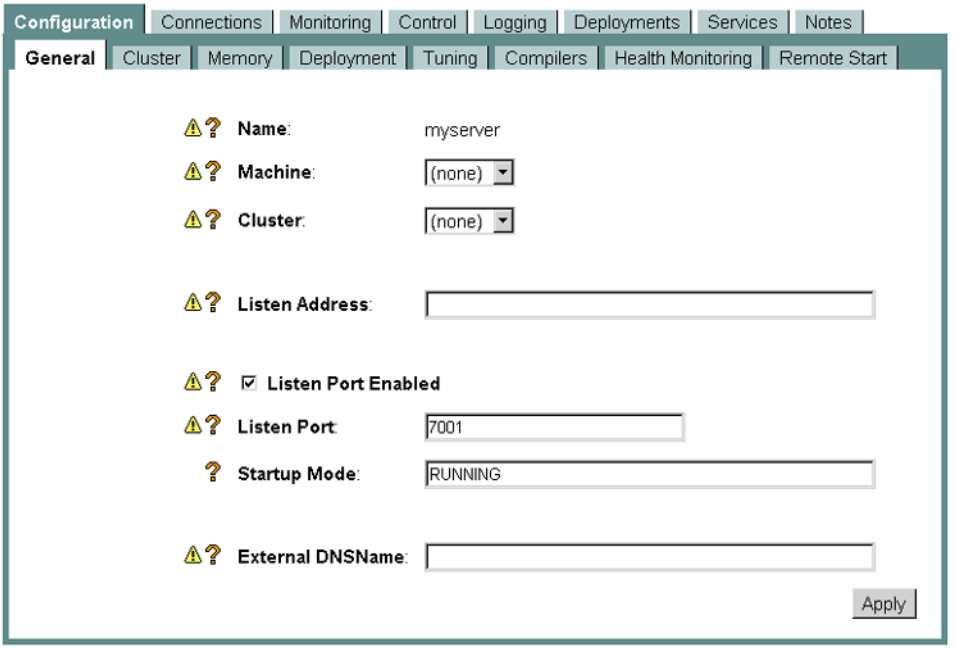
1. Start the WebLogic Server Administration Console as described in "Starting the WebLogic Server Administration Console" on page 2-7.
2. In the navigation tree (in the left pane) of the WebLogic Server Administration Console, choose Servers→*myserver*, where *myserver* is the name of the WebLogic Server with the listen port you want to change, as shown in the following figure.

Figure 3-12 Choosing a Server



The Configuration page for WebLogic Server is displayed, as shown in the following figure.

Figure 3-13 WebLogic Server Administration Console Configuration Page



3. Enter the new port number in the Listen Port field, and then click Apply.
4. Restart the server to activate the new setting.

Warning: You must change the listen port using the WebLogic Server Administration Console. Do not change the port number using the B2B Console.

Changing the Listen Port by Editing the Configuration File

Note: Do not edit the `config.xml` file while the server is running. If the server is running, update the listen port as described in the preceding procedure.

To change the listen port by editing the `config.xml` file:

1. Open the administration server *`DOMAIN_HOME`*`/config.xml` file in your preferred text editor.

   Here *`DOMAIN_HOME`* is the complete path to the root of the domain (for example, the `c:\bea\user_projects\`*`domain`* directory).

2. In the `config.xml` file, locate the `ListenPort` setting for the server targeted for the listen port number change (shown in bold in Listing 3-5).

3. Set the `ListenPort` to a unique number that is not used by other WebLogic Server instances running on the same machine.

Listing 3-5 Changing the ListenPort Element in the config.xml File

```
<Server
 ListenPort="7001"
 Name="myserver"
 TransactionLogFilePrefix="c:\bea\weblogic700\samples/integration/config/sample
s/logs/"
 StdoutDebugEnabled="true"
 StdoutEnabled="true"
 StdoutSeverityLevel="64" NativeIOEnabled="true">
```

Step 2. Changing Other WebLogic Integration Port Numbers

Which steps you must complete to update the remainder of the WebLogic Integration port numbers is determined by which features of WebLogic Integration are used in the application. To find out which steps are required to change the port number for your application, refer to the following table.

Table 3-11 Changing WebLogic Integration Port Numbers

If your application implements this WebLogic Integration functionality . . .	Complete the procedure in the following section . . .
B2B security	"Configuring the SSL Protocol and Mutual Authentication" in "Configuring Security" in *Implementing Security with B2B Integration*
Application integration	"Changing the Application View Console Port Number" on page 3-60
B2B engine	"Changing B2B URI Endpoints" on page 3-59
B2B Administration Console	"Changing the B2B Console Port Number" on page 3-61
Application intergration, B2B, or BPM	"Changing the WebLogic Integration Shutdown Scripts" on page 3-61

Warning: The samples contain many files in which the port number is set. To avoid introducing errors, we recommend that you do not change the default port number in the samples.

Changing B2B URI Endpoints

A WebLogic Integration URI endpoint specifies the location at which a trading partner listens for a B2B message. If you have already changed the listen port for your WebLogic Server, as described in "Step 1. Change the WebLogic Server Listen Port Number" on page 3-56, and you are using WebLogic Integration B2B functionality, you must change the URI Endpoints for your B2B trading partners.

To change the port number in the URI Endpoint:

1. Start the B2B Console, as described in "Starting the B2B Console" on page 2-24.
2. In the navigation tree (in the left pane) of the B2B Console, choose B2B→Trading Partners.

3. From the list shown in the right pane, select a trading partner that is involved in your WebLogic Integration application.
4. In the right pane, click the URI, update the port number, and then click Add/Apply. Repeat this step for all URIs in the chain.
5. Repeat steps 2-4 for all trading partners that are configured.

Warning: URI endpoints are also specified in the Bulk Loader repository data file. If you change the port number in the URI endpoint in the repository and you want your Bulk Loader repository to reflect that change, make sure you export the current repository information using the B2B Console or the command-line Bulk Loader. For more information, see "Importing and Exporting B2B Integration Components" in *Administering B2B Integration*.

Changing the Application View Console Port Number

If you have changed the listen port for your WebLogic Server, as described in "Changing B2B URI Endpoints" on page 3-59, and you are using the WebLogic Integration Application View Console, you must update the port number specified in the `startAIconsole` command.

To update the port number specified in the `startAIconsole` command:

1. In your preferred text editor, open the appropriate command file: *`WLI_HOME`*`/bin/startAiconsole.cmd` (Windows) or *`WLI_HOME`*`/bin/startAIconsole.sh` (UNIX).
2. Set the port number to the value you assigned when updating the listen port for the server.
3. Save the file.

Changing the B2B Console Port Number

If you have changed the listen port for WebLogic Server, as described in "Changing B2B URI Endpoints" on page 3-59, and you are using the WebLogic Integration B2B Console, you must update the port number specified in the `startB2Bconsole` command.

To update the port number specified in the `startB2Bconsole` command:

1. In your preferred text editor, open the appropriate command file: *`WLI_HOME`*`/startB2bconsole.cmd` (Windows) or *`WLI_HOME`*`/startB2bconsole.sh` (UNIX.
2. Set the port number to the value you assigned when updating the listen port for the server.
3. Save the file.

Changing the WebLogic Integration Shutdown Scripts

If you have changed the listen port for WebLogic Server, as described in "Changing B2B URI Endpoints" on page 3-59, you must change the port number in the shutdown script for the server.

To change the port number in a shutdown script:

1. In your preferred text editor, open the file for the appropriate server shutdown command: *`DOMAIN_HOME`*`\stopWeblogic.cmd` (Windows) or *`DOMAIN_HOME`*`/stopWeblogic.cmd` (UNIX).

   For example, if you are defining a server port number for a domain called `mydomain` in the `c:\bea\user_projects` directory, open the `c:\bea\user_projects\mydomain\stopWebLogic.cmd` file.
2. Set the port number to the value you set when updating the listen port for the server.
3. Save the file.

Deploying B2B Without Persistence

By default, WebLogic Integration is deployed in persistence mode, which means that WebLogic Integration can recover in the event of hardware or network failures. For more information about the benefits of deploying WebLogic Integration in persistence mode, see "What to Expect from WebLogic Integration Recovery" in "Understanding WebLogic Integration High Availability" in *Deploying BEA WebLogic Integration Solutions*.

The ability to turn off persistence is supported only for the B2B functionality provided by WebLogic Integration. Persistence is turned off using the WebLogic Server Administration console. For more information, see "Configuring Persistence and Recovery" in *Administering B2B Integration*.

A WebLogic Integration Sample Configuration Files

This section provides developers and system administrators with sample configuration files for BEA WebLogic Integration. During installation, these files are automatically customized, so they will work on any supported computer system and network. They are provided here, along with descriptions of the information required in each.

This section includes the following topics:

- config.xml
- setEnv
- startWeblogic
- fileRealm.properties

Each file includes sample values for parameters that are updated by the installation program. These sample values are highlighted in **bold**.

Note: Some acronyms used in the `config.xml` and other files are a legacy from a previous release: the acronym WLC corresponds to B2B; WLPI corresponds to business process management (BPM); WLAI corresponds to application integration; and WLXT corresponds to data integration.

config.xml

A samples domain is created in which you can run the WebLogic Integration sample applications if you performed one of the following sets of tasks during installation:

- Selected the Typical installation option
- Selected the Custom installation option and included WebLogic Integration Samples in your installation

The configuration for domains is defined in eXtensible Markup Language (XML). Persistent storage for the samples domain configuration is provided by the `config.xml` file located in *`SAMPLES_HOME`*`/samples/integration/config`, where *`SAMPLES_HOME`* is the WebLogic Platform samples directory. This section describes the contents of the WebLogic Integration samples domain `config.xml` file.

You can view and modify the settings captured in `config.xml` through the WebLogic Server Administration Console. When you start the WebLogic Server Administration Console, as described in "Starting the WebLogic Server Administration Console" on page 2-7, the server home page is displayed. Items listed in the navigation tree, shown in the left pane, correspond to the elements and attributes defined in the `config.xml` file.

The root element of the `config.xml` file is `Domain`. As shown in the following figure, this section of the file deploys the application integration DBMS sample adapter, the BEA WebLogic Adapter for Peregrine Power.Enterprise!, and the default Web Application. (The default Web Application responds to HTTP requests that cannot be resolved by any other deployed Web applications.)

Figure A-1 config.xml File: Part 1

```
<?xml version="1.0" encoding="UTF-8"?>
<!--If your domain is active, please do not edit the config.xml file. Any chang\
es made to that file while the domain is active will not have any effect on the\
 domain's configuration and are likely to be lost. If your domain is inactive, \
you may edit this file with an XML editor. If you do so, please refer to the BE\
A Weblogic Server Configuration Reference  documentation available from http://\
edocs.bea.com/wls/docs70/config_xml.  In general, we recommend that changes to \
your configuration file be made through the Administration Console.-->
<Domain Name="samples">
    <Application Deployed="true" Name="BEA_POWERENTERPRISE_3_0"
        Path="C:\bea\weblogic700\integration\adapters\powerenterprise\lib\BEA_P\
OWERENTERPRISE_3_0_EAR.ear" TwoPhase="true">
        <ConnectorComponent Name="BEA_POWERENTERPRISE_3_0"
            Targets="myserver" URI="BEA_POWERENTERPRISE_3_0.rar"/>
        <WebAppComponent Name="BEA_POWERENTERPRISE_3_0_EventRouter"
            Targets="myserver" URI="BEA_POWERENTERPRISE_3_0_EventRouter.war"/>
        <WebAppComponent Name="BEA_POWERENTERPRISE_3_0_Web"
            Targets="myserver" URI="BEA_POWERENTERPRISE_3_0_Web.war"/>
    </Application>
    <Application Deployed="true" Name="BEA_WLS_DBMS_ADK"
        Path="C:\bea\weblogic700\integration\adapters\dbms\lib\BEA_WLS_DBMS_ADK\
.ear" TwoPhase="true">
        <ConnectorComponent Name="BEA_WLS_DBMS_ADK" Targets="myserver" URI="BEA\
_WLS_DBMS_ADK.rar"/>
        <WebAppComponent Name="BEA_WLS_DBMS_ADK_Web" Targets="myserver" URI="BE\
A_WLS_DBMS_ADK_Web.war"/>
        <WebAppComponent Name="DbmsEventRouter" Targets="myserver" URI="BEA_WLS\
_DBMS_ADK_EventRouter.war"/>
    </Application>
    <Application Deployed="true" Name="DefaultWebApp_myserver"
        Path="C:\bea\weblogic700\samples\integration\config\samples\application\
s" TwoPhase="false">
        <WebAppComponent IndexDirectoryEnabled="true"
            Name="DefaultWebApp_myserver" Targets="myserver" URI="DefaultWebApp\
_myserver"/>
    </Application>
```

As shown in the following two figures, the next section deploys the EJBs and Web applications required for the WebLogic Integration application.

The `path` attribute is used by WebLogic Server to locate the WebLogic Integration exploded EAR and JAR files and to locate the WLI `application.xml` and `weblogic-application.xml` files. (The `application.xml` and `weblogic-application.xml` files are located in the `path\META-INF` directory.)

WebLogic Integration J2EE components are now deployed using the exploded form. For more information about deploying J2EE components using the exploded form, see *Assembling and Configuring Web Applications* in the BEA WebLogic Server documentation at the following URL:

http://e-docs.bea.com/wls/docs70/webapp/basics.html

When you components are deployed in the exploded form, the `DeploymentOrder` attribute is not specified. Instead, the deployment order is determined by the order of the J2EE components in the `application.xml` file. If you do not want to deploy a component of the `WLI` application, do not remove the element for the component in the `application.xml` file, instead remove the server name from the Targets attribute (`Targets=""`).

Figure A-2 config.xml File: Part 2

The path attribute of the Application element may be different for your installation.

WLI Application element - definitions for all the jar and war files that make up the WLI application.

```
    <Application Deployed="true" Name="WLI"
        Path="C:\bea\weblogic700\integration\lib" TwoPhase="true">
        <EJBComponent Name="Sample BPM Plug-in" Targets="myserver" URI="samplep\
lugin-ejb.jar"/>
        <EJBComponent Name="Sample EJB" Targets="myserver" URI="pobean.jar"/>
        <EJBComponent Name="WLI Repository" Targets="myserver" URI="repository-\
ejb.jar"/>
        <EJBComponent Name="WLI-AI Async Processor" Targets="myserver" URI="wla\
i-asyncprocessor-ejb.jar"/>
        <EJBComponent Name="WLI-AI BPM Plug-in" Targets="myserver" URI="wlai-pl\
ugin-ejb.jar"/>
        <EJBComponent Name="WLI-AI Event Processor" Targets="myserver" URI="wla\
i-eventprocessor-ejb.jar"/>
        <EJBComponent Name="WLI-AI Server" Targets="myserver" URI="wlai-server-\
ejb.jar"/>
        <EJBComponent Name="WLI-B2B RN BPM Plug-in" Targets="myserver" URI="wlc\
-wlpi-plugin.jar"/>
        <EJBComponent Name="WLI-B2B RN MDB" Targets="myserver" URI="b2b-rosetta\
net.jar"/>
        <EJBComponent Name="WLI-D2D Startup" Targets="myserver" URI="b2b-startu\
p.jar"/>
        <EJBComponent Name="WLI-B2B ebXML BPM Plug-in"
            Targets="myserver" URI="ebxml-bpm-plugin.jar"/>
        <EJBComponent Name="WLI-BPM Event Processor" Targets="myserver" URI="wl\
pi-mdb-ejb.jar"/>
        <EJBComponent Name="WLI-BPM Initialization" Targets="myserver" URI="bpm\
-init-ejb.jar"/>
        <EJBComponent Name="WLI-BPM Plugin Manager" Targets="myserver" URI="wlp\
i-master-ejb.jar"/>
        <EJBComponent Name="WLI-BPM Server" Targets="myserver" URI="wlpi-ejb.ja\
r"/>
        <EJBComponent Name="WLI-DI BPM Plug-in" Targets="myserver" URI="wlxtpi.\
jar"/>
        <EJBComponent Name="WLI-DI Server" Targets="myserver" URI="WLXTEJB.jar"\
/>
        <WebAppComponent Name="Sample BPM Plug-in Help"
            Targets="myserver" URI="sampleplugin.war"/>
        <WebAppComponent Name="TransportServlet" Targets="myserver" URI="b2b.wa\
r"/>
```

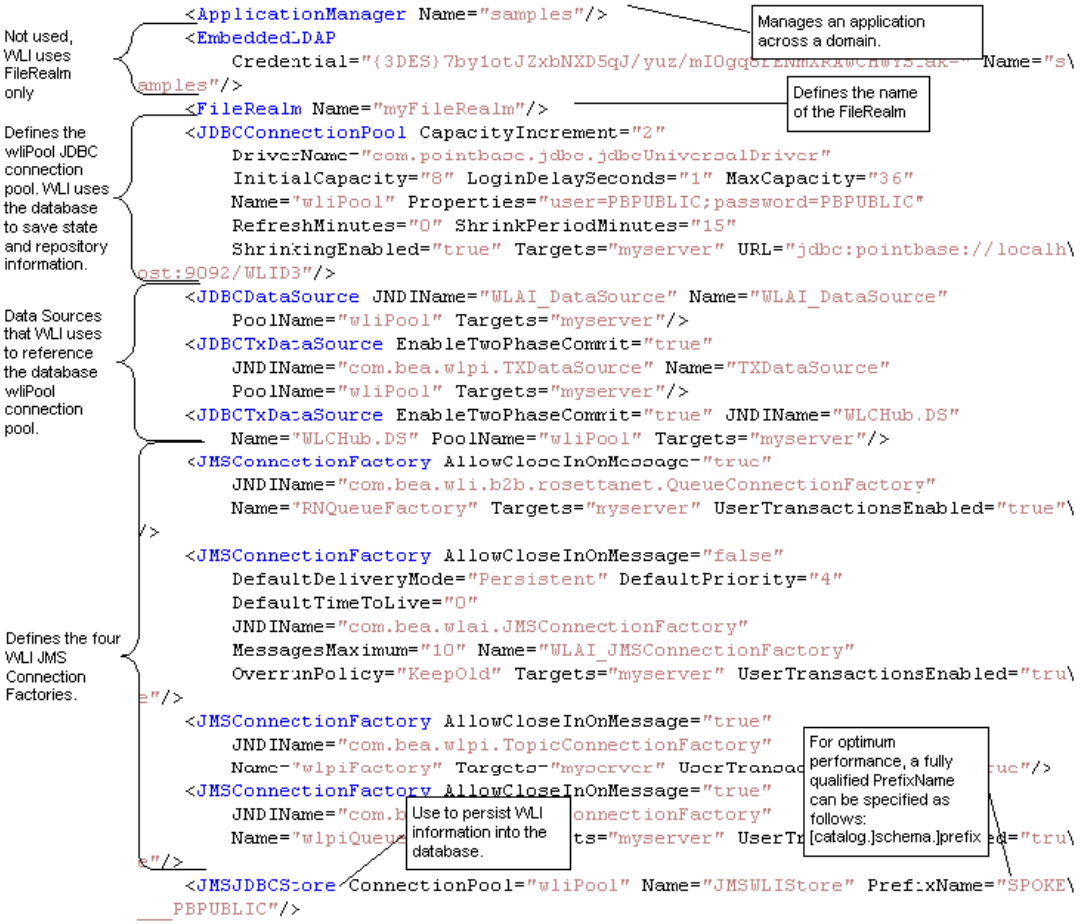
As shown in the following figure, the next section of the `config.xml` file contains the elements related to application integration functionality.

Figure A-3 config.xml File: Part 3

```
        <WebAppComponent Name="TransportServlet" Targets="myserver" URI="b2b.wa\
r"/>
        <WebAppComponent Name="WLI-AI BPM Plug-in Help"
            Targets="myserver" URI="wlai-plugin.war"/>
        <WebAppComponent Name="WLI-DI BPM Plug-in Help"
            Targets="myserver" URI="wlxtpi.war"/>
        <WebAppComponent Name="b2bconsole" ServletReloadCheckSecs="1"
            Targets="myserver" URI="b2bconsole.war"/>
        <WebAppComponent Name="wlai" Targets="myserver" URI="wlai.war"/>
    </Application>
    <Application Deployed="true"
        Name="WLISAppView.sav_connectionFactory"
        Path="C:\bea\weblogic700\samples\integration\config\samples\wlai\deploy\
"
        StagingMode="nostage" TwoPhase="true">
        <ConnectorComponent Name="WLISAppView.sav_connectionFactory"
            Targets="myserver" URI="WLISAppView.sav_connectionFactory.rar"/>
    </Application>
    <Application Deployed="true" Name=" appsdir WLPI sample war"
        Path="C:\bea\weblogic700\samples\integration\config\samples\application\
s"
        StagedTargets="myserver" StagingMode="stage" TwoPhase="true">
        <WebAppComponent Name="WLPI_sample" Targets="myserver" URI="WLPI_sample\
.war"/>
    </Application>
    <Application Deployed="true" Name="_appsdir_lwcWebApp_war"
        Path="C:\bea\weblogic700\samples\integration\config\samples\application\
s"
        StagedTargets="myserver" StagingMode="stage" TwoPhase="true">
        <WebAppComponent Name="lwcWebApp" Targets="myserver" URI="lwcWebApp.war\
"/>
    </Application>
    <Application Deployed="true" Name="wlisWebApp"
        Path="C:\bea\weblogic700\samples\integration\config\samples\application\
s" TwoPhase="false">
        <WebAppComponent Name="wlisWebApp" Targets="myserver" URI="wlisWebApp.w\
ar"/>
    </Application>
```

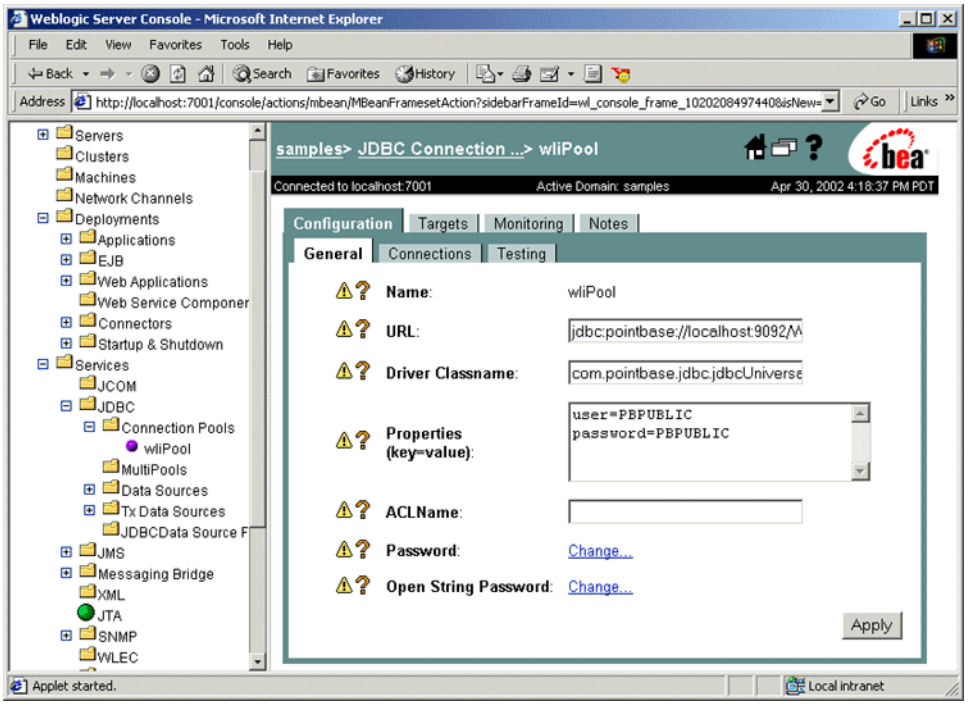
As shown in the following figure, the next section of the `config.xml` file defines the JDBC connection pool (called `wliPool`), the JDBC data sources, and the JMS connection factories for WebLogic Integration.

Figure A-4 config.xml File: Part 4



You can also view and update attributes for the WebLogic Integration JDBC connection pool (called `wliPool`) by choosing Services→JDBC→Connection Pools→wliPool from the WebLogic Server Administration Console navigation tree, as shown in the following figure.

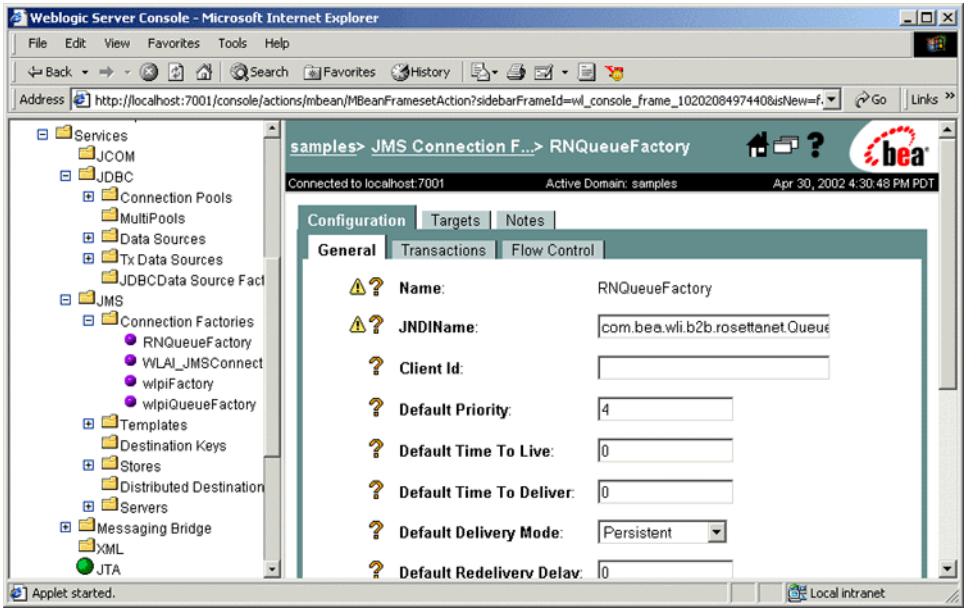
Figure A-5 WebLogic Integration JDBC Connection Pool



For information about updating the JDBC configuration, see "Specifying a New Database for a Domain" on page 3-2.

You can also view the four WebLogic Integration connection factories by choosing Services—JMS—Connection Factories from the WebLogic Server Administration Console navigation tree, as shown in the following figure.

Figure A-6 WebLogic Integration Connection Factories



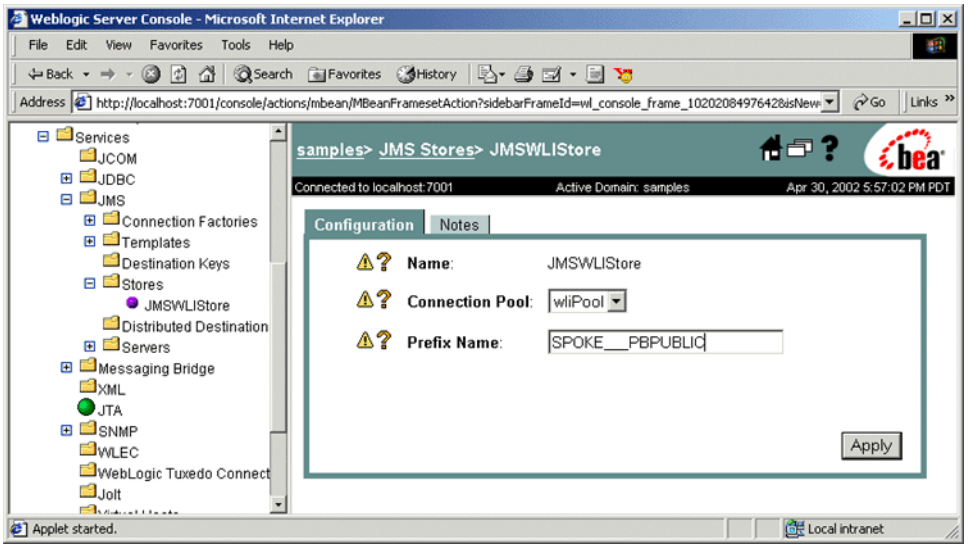
Warning: To avoid unexpected consequences, do not change the attributes for the WebLogic Integration connection factories unless instructed to do so by the WebLogic Integration documentation. For example, if you change the Default Delivery Mode for the connection factories, your ability to recover WebLogic Integration in the case of a system or network failure may be compromised.

To optimize performance, customize the prefix name for the JMS JDBC store to suit your database. Use the following syntax for a prefix name:

`[[`*`catalog`*`.]`*`schema`*`.]`*`prefix`*`]`

To update the JMS store prefix name, choose Services→JMS→Stores→JMSWLIStore. The JMSWLIStore page is displayed, as shown in the following figure.

Figure A-7 JMS JDBC Store Page



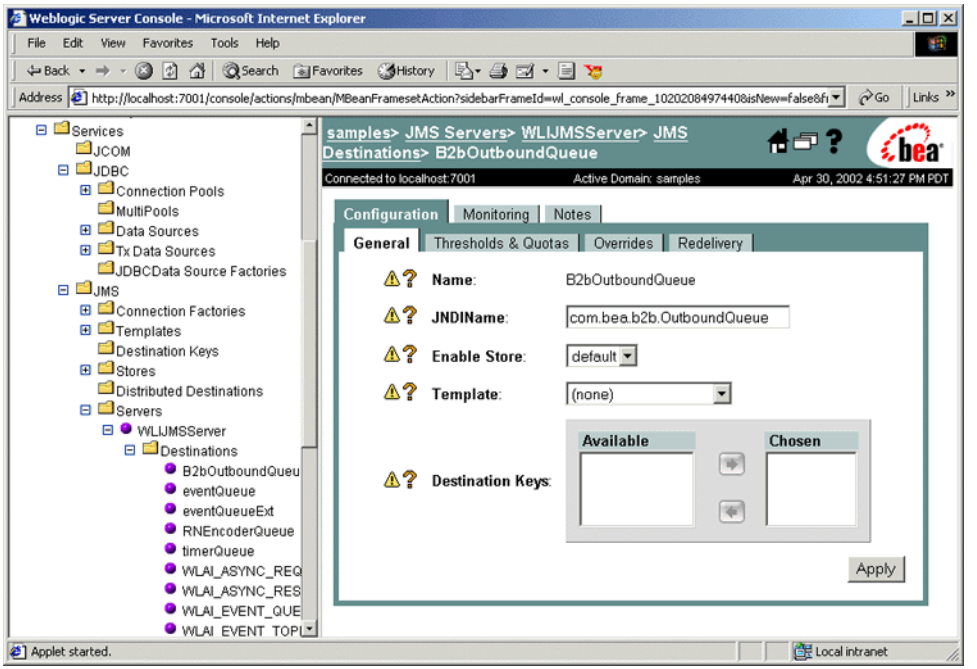
Enter the qualified prefix name (with the appropriate schema and catalog prefixes for your database configuration) in the Prefix Name field. See the *DOMAIN_HOME*\dbinfo*database_type*\setDBVarsExt for the appropriate schema and catalog prefixes for your database configuration. *DOMAIN_HOME* represents the complete path to the root of a domain and *database_type* represents the your database: oracle, sybase, db2, mssql, or pointbase.

As shown in the following figure, the next section of the config.xml file defines the JMS Server for WebLogic Integration.

Figure A-8 config.xml File: Part 5

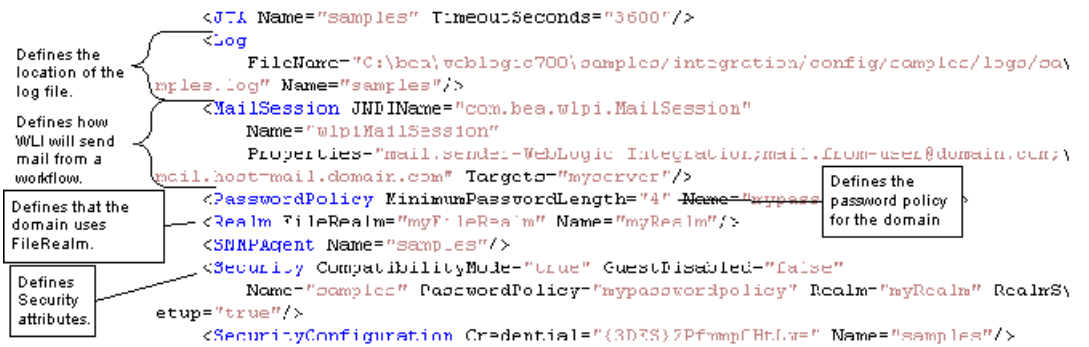
You can also view the WebLogic Integration JMS Server (called `WLIJMSServer`) by choosing Services→JMS→Connection Factories from the WebLogic Server Administration Console navigation tree, as shown in the following figure.

Figure A-9 WebLogic Integration JMS Server



As shown in the following figure, the next section of the `config.xml` file includes the Realm, Security, and other elements.

Figure A-10 config.xml File: Part 6



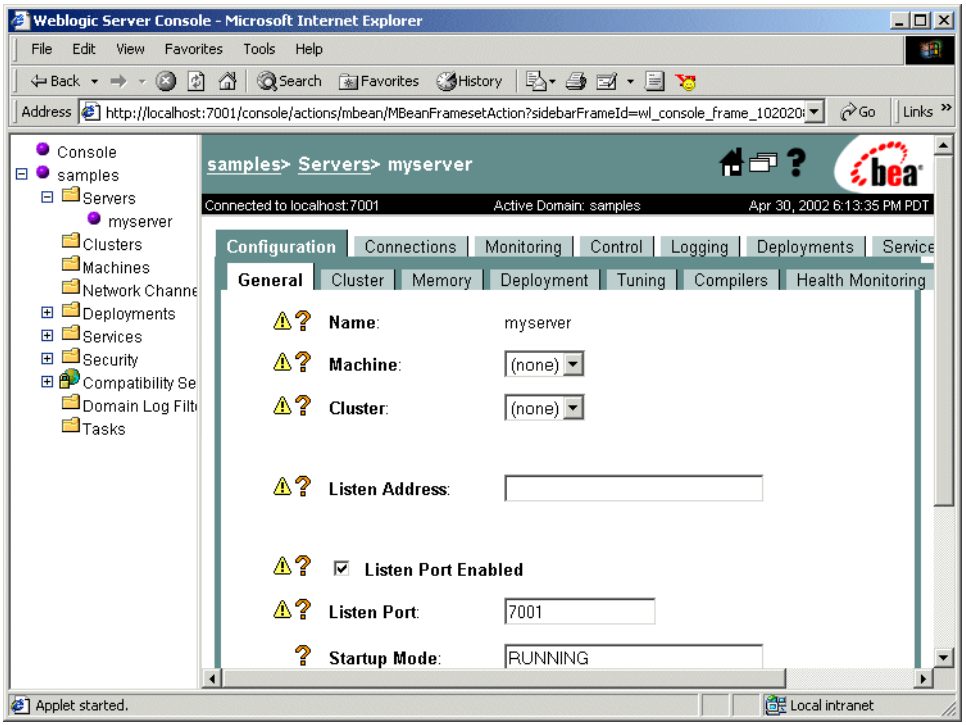
For information about security realms, see “Understanding the BPM Security Model” on page 3-43.

As shown in the following figure, the next section of the `config.xml` file includes the Server elements and attributes and the WebLogic Integration B2B shutdown class.

Figure A-11 config.xml File: Part 7

The domain name specified by the domain element corresponds to the domain name specified in the WebLogic Server Administration Console. For example, if you select myserver from the navigation tree in the console, the server page is displayed, as shown in the following figure.

Figure A-12 WebLogic Server Administration Console Server Page



By selecting the appropriate tab on this page, you can update the listen port, log file name, log level and output, and other server configuration parameters. For help with any settings, click the question mark in the upper right corner to view the online help, or see the WebLogic Server documentation set at the following URL:

`http://e-docs.bea.com/wls/docs70/index.html`

setEnv

This executable file is used to set the environment variables for WebLogic Integration. An instance of this file resides in the WebLogic Integration installation directory, and in the `bin` directory under the WebLogic Integration installation directory. This file is called by the `startWeblogic` command file and other command files provided with WebLogic Integration.

The first section of the setenv script for the samples domain (either `setEnv.cmd` if you are running on Windows or `setenv.sh` if you are running on UNIX) is shown in the following figure. (The script is located in the *`WLI_HOME`* directory, where *`WLI_HOME`* represents the directory in which you installed WebLogic Integration.) The values in bold are set during installation, based on the location of your installation.

For additional information about this command and the environment variables set by it, see Appendix B, "WebLogic Integration Commands."

Listing A-1 setEnv Command

```
@echo off
.
.
.
set JAVA_HOME=C:\bea\jdk131_03
set PATH=%PATH%;%JAVA_HOME%\bin
set BEA_HOME=C:\bea
set WL_HOME=C:\bea\weblogic700\server
set WLI_HOME=C:\bea\weblogic700\integration
set SAMPLES_HOME=C:\bea\weblogic700\samples
set JAVA_OPTIONS="-XX:MaxPermSize=131072k"
REM set DB2_HOME=

:checkJDK
if exist %JAVA_HOME%\lib\nul goto setCP
echo.
echo The JDK wasn't found in directory %JAVA_HOME%.
echo Please edit the setEnv.cmd script so that the JAVA_HOME
echo variable points to the location of your JDK.
goto error

:setCP
```

```
REM Set the WLI common Classpath
set WLICOMMONCP=

for %%a in (
      wlicommon.jar,
) do call :wlicmnlib %%a

.
.
.
```

startWeblogic

This executable command file starts WebLogic Server. Using the content of the `config.xml` file, it deploys the WebLogic Integration applications and resources specified in the `config.xml`file for the domain. It is installed in each domain directory. For example, if, on a Windows system, a new domain (called `mydomain`) is created in the default location, the directory location for the WebLogic Integration `startWeblogic.cmd` for that domain is:

BEA_HOME`/user_projects/mydomain`

In this pathname *BEA_HOME* represents the location of your BEA Home directory, for example: `C:\bea`.

The `startWeblogic.cmd` (Windows) for the samples domain is shown in the following listing. The `startWebLogic.cmd` command for the preconfigured samples domain is located in the *SAMPLES_HOME*`/integration/config/samples`, where *SAMPLES_HOME* represents the WebLogic Platform samples directory.

Listing A-2 startWebLogic.cmd Command

```
@echo off
.
.
.
if /I "%WLI_HOME%" == "" call c:\bea\weblogic700\integration\setEnv.cmd

setlocal
```

```
call %SAMPLES_HOME%\integration\samples\bin\SetSampleData || goto :EOF
if "%DB_TYPE%" == "pointbase" goto pointbase
goto run

:pointbase
REM Invoke ant script to finish up work
set SCRIPT=%WLI_HOME%\lib\scripts\PointbaseChecker.xml

if exist %SCRIPT% goto ant
echo WARNING: No ant script to check for pointbase (%SCRIPT%)
goto finish

:ant
%JAVA_HOME%\bin\java -classpath %WLICP% -Xmx256m org.apache.tools.ant.Main \
-buildfile %SCRIPT% || goto error

:run
REM Remove old logs.
del /f /q %WLI_SAMPLES_HOME%\*.log* > nul 2>&1
del /f /q %WLI_SAMPLES_HOME%\logs\* > nul 2>&1

REM Change directory to where DTD files are located
cd /d %WLI_SAMPLES_HOME%

REM WLIS data directory
if not exist %WLI_SAMPLES_HOME%\data mkdir %WLI_SAMPLES_HOME%\data || goto finish

REM Start weblogic
%JAVA_HOME%\bin\java %JAVA_OPTIONS% %DB_JVMARGS% -Xmx256m -classpath %SVRCP% \
 -Dbea.home=%BEA_HOME% -Dweblogic.home=%WL_HOME% \
 -Dweblogic.system.home=%SAMPLES_HOME%\integration -Dweblogic.Domain=samples \
 -Dweblogic.management.username=system -Dweblogic.management.password=security \
 -Dweblogic.Name=myserver -Dweblogic.RootDirectory=%SAMPLES_HOME%\integration \
 -Djava.security.policy=%WL_HOME%\lib\weblogic.policy \
 -Dweblogic.management.discover=false \
 -Dweblogic.SystemDataStoreConfigDirectory=%WLI_HOME%\lib \
 -Dweblogic.servlet.ClasspathServlet.disableStrictCheck=true weblogic.Server

:finish
endlocal
```

The `-Dweblogic.management.password=security` option in the `java` command that starts WebLogic Server provides the password. If this option is removed, you are prompted for a password. If you change the password, you must remove or update this option.

fileRealm.properties

This properties file controls the User, Group, and ACL objects that are created when WebLogic Server is started.

The `fileRealm.properties` file for the samples domain is shown in the following listing.

Listing A-3 fileRealm.properties File

```
#Fri Apr 12 22:21:45 BST 2002
acl.access.weblogic.admin.mbean.MBeanHome=wlcSamplesUser,everyone,guest
acl.boot.weblogic.server=system,everyone
acl.create.weblogic.jms.ServerSessionPool=everyone
acl.enablermonitor.WLCAdmin=admin
acl.execute.weblogic.servlet.AdminClients=system
acl.execute.weblogic.servlet.AdminConnections=system
acl.execute.weblogic.servlet.AdminEvents=system
acl.execute.weblogic.servlet.AdminJDBC=system
acl.execute.weblogic.servlet.AdminLicense=system
acl.execute.weblogic.servlet.AdminMain=system
acl.execute.weblogic.servlet.AdminProps=system
acl.execute.weblogic.servlet.AdminRealm=system
acl.execute.weblogic.servlet.AdminThreads=system
acl.execute.weblogic.servlet.AdminVersion=system
acl.execute.weblogic.servlet.Certificate=system
acl.execute.weblogic.servlet.ConsoleHelp=everyone
acl.execute.weblogic.servlet.T3AdminMain=system
acl.execute.weblogic.servlet.classes=everyone
acl.execute.weblogic.servlet.containerManaged=everyone
acl.execute.weblogic.servlet=system,everyone
acl.hubconfig.WLCAdmin=admin
acl.list.weblogic.jndi.weblogic.ejb=system,everyone
acl.list.weblogic.jndi.weblogic.fileSystem=everyone
acl.list.weblogic.jndi.weblogic.rmi=everyone
acl.list.weblogic.jndi.weblogic=system,everyone
acl.list.weblogic.jndi=everyone
acl.lockServer.weblogic.admin=system
acl.lookup.weblogic.admin.mbean.MBeanHome=wlcSamplesUser,everyone,guest
acl.lookup.weblogic.jndi.weblogic.ejb=system,everyone,guest
acl.lookup.weblogic.jndi.weblogic.fileSystem=everyone
acl.lookup.weblogic.jndi.weblogic.rmi=everyone
acl.lookup.weblogic.jndi.weblogic=everyone
```

```
acl.lookup.weblogic.jndi=everyone
acl.modify.weblogic.admin.acl=system
acl.modify.weblogic.jndi.weblogic.ejb=system,everyone,guest
acl.modify.weblogic.jndi.weblogic.fileSystem=everyone
acl.modify.weblogic.jndi.weblogic.rmi=everyone
acl.modify.weblogic.jndi.weblogic=system,everyone,guest
acl.modify.weblogic.jndi=everyone
acl.read.managedObject=system
acl.read.weblogic.workspace=system,everyone
acl.receive.weblogic.jms=everyone
acl.reserve.weblogic.jdbc.connectionPool.wliPool=wlisystem,everyone
acl.reserve.weblogic.jdbc.connectionPool=system
acl.reset.weblogic.jdbc.connectionPool.examplesPool=wlcSamplesUser,admin,guest
acl.reset.weblogic.jdbc.connectionPool.wliPool=wlcSamplesUser,wlisystem,admin,\
guest
acl.reset.weblogic.jdbc.connectionPool.wlpiPool=wlisystem,admin,guest
acl.reset.weblogic.jdbc.connectionPool=system
acl.send.weblogic.jms=everyone
acl.shrink.weblogic.jdbc.connectionPool.examplesPool=everyone
acl.shrink.weblogic.jdbc.connectionPool.wliPool=wlcSamplesUser,wlisystem,\
admin.guest
acl.shrink.weblogic.jdbc.connectionPool.wlpiPool=wlisystem,admin,guest
acl.shutdown.weblogic.admin=system
acl.unlockServer.weblogic.admin=system
acl.write.managedObject=system
acl.write.weblogic.workspace=system,everyone

group.AccountingCDE=admin,joe
group.AdministerUser=admin,joe,mary,guest,wlisystem
group.Administrators=ConfigureComponents,system
group.ConfigureComponents=admin,joe,wlisystem,mary,guest
group.ConfigureSystem=admin,joe,mary,guest,wlisystem
group.CreateTemplate=admin,joe,wlisystem,mary,guest
group.CustomerServiceCDE=admin
group.DeleteTemplate=admin,joe,wlisystem,mary,guest
group.Deployers=Administrators
group.ExecuteTemplate=admin,joe,wlisystem,mary,guest
group.MonitorInstance=admin,joe,wlisystem,mary,guest
group.Monitors=Administrators
group.Operators=Administrators
group.Role1Org1=admin,joe,mary
group.Role1Org2=admin,joe,mary
group.Role2Org1=admin,joe,mary
group.Role2Org2=admin,joe,mary
group.ShippingCDE=admin,mary
group.UpdateTemplate=admin,joe,mary,guest,wlisystem
group.adapter=admin,wlcSamplesUser,joe,hub,system,mary,guest,wlisystem
group.adminGroup=admin,joe,mary
group.wlcSamplesGroup=wlcSamplesUser,guest,wlisystem
```

```
group.wlpiAdministrators=admin,wlcSamplesUser,joe,wlisystem,system,mary,guest
group.wlpiUsers=admin,joe,wlisystem,system,mary,guest

user.admin=0xab3a488db0652704287970cdf97854812feea77b
user.hub=0x6e4b0d35e609e96e749330f5086bb21f08a013d5
user.joe=0xa078cb45e6f6c4eefdd1f14495ff739b5536904c
user.mary=0xa078cb45e6f6c4eefdd1f14495ff739b5536904c
user.system=0xab3a488db0652704287970cdf97854812feea77b
user.wlcSamplesUser=0x010e807f5525398753170a0fd0cf8694241286bc
user.wlisystem=0x8510e22f04b4a7ebde6837902aa590a31e234be2
```

B WebLogic Integration Commands

This section provides system administrators and developers with information about key WebLogic Integration commands. It is organized into the following topics:

- Command List—an alphabetical list of the commands with hypertext links for quick access to detailed descriptions.
- Environment Variables—an alphabetical list, with brief descriptions, of the environment variables set or referenced by the commands.
- Command Descriptions—reference pages, containing detailed descriptions of the commands.

Command List

The following table provides the name and a brief description of each command documented in this section.

Table B-1 WebLogic Integration Commands

This command . . .	Allows you to . . .
bulkimporter	Import workflow objects from Java Archive (JAR) files
bulkloader	Import, export, or delete repository data
bulkmigrator	Migrate a repository data file
checkdomain	Validate a domain and set WLI_DOMAIN for the calling process
createdb	Create repository tables and load system data
createmssql createsybase	Create a Microsoft SQL or Sybase database, and execute a script in that database
domainname	Set or verify the current domain set by the setdomain command
fb	Start Format Builder
launchSamplesPage	Open the samples launcher page in the default Web browser
mdbgenerator	Generate a deployable JAR file to listen on a custom queue
migratedb	Migrate a database from WebLogic Integration 2.1 to 7.0
PIMigrator	Migrate a workflow template
RunSamples	Configure the samples domain and start WebLogic Server in the samples domain
setdbdata	Load the database-specific variables into the current process
setdbtype	Set the database type for the domain
setDBVars setDBVarsExt	Set the basic and extended database variables for the domain

Table B-1 WebLogic Integration Commands (Continued)

This command . . .	Allows you to . . .
setdomain	Set the current domain
setenv	Set common WebLogic Integration environment variables
SetDomainTypeData SetSampleData	Set domain-specific environment variables
setupAppView setupEDIAppView	Set up the sample adapter or EDI adapter application view
showdomain	Validate and display the current domain
startAiconsole startB2bconsole	Start the Application View Console or B2B Console
startpointbasecommander	Start the PointBase administrative tool
startWeblogic	Start WebLogic Server and deploy WebLogic Integration resources
stopWeblogic	Shut down WebLogic Server
studio	Start the WebLogic Integration Studio
switchdb	Update the database configured for the current domain
wliconfig	Launch the WebLogic Integration Database Wizard
worklist	Start the WebLogic Integration Worklist

Environment Variables

The following table provides the name and a brief description of each environment variable referenced by a WebLogic Integration command.

Note: The italicized string *DOMAIN_NAME* is used as a placeholder for a pathname in this document. Specifically, it represents the full pathname for the root of this domain. Unlike other terms shown in italic monospace font (such as WLI_HOME), *DOMAIN_HOME* is not an environment variable.

Table B-2 WebLogic Integration Environment Variables

Variable	Description
CP	The classpath set by the Set*DomainType*Data command and referenced by the setupAppView and setupEDIAppView commands. This variable is reset, but not exported, by the fb, worklist, and studio commands.
CMNCP	The common classpath set by the Set*DomainType*Data command. For example, in the SetSampleData command, CMNCP is set as follows: CMNCP=$WLI_SAMPLES_HOME/lib/samples.jar;$WLI_HOME/lib/wli-aux.jar
DB_CATALOG	Name of the database catalog. See the description of the DB_PINGTABLE environment variable (in this table) and the setDBVarsExt and setupAppView commands.
DB_CODESET	The type of encoding used. This environment variable is not set if the encoding is US-ASCII or UTF-8. See the setDBVars commands.
DB_COMMAND	The command line required to invoke the client for the database. For example, if the current domain is configured to use an Oracle database, DB_COMMAND is set as follows: DB_COMMAND="sqlplus *username/password@oracle_sid* @" See the createdb, setdbdata, and setDBVars commands.
DB_CREATE	Invokes a database-specific command script. If the specified database does not exist, this command creates the database. See the createdb, setdbdata, and setDBVars commands.

Table B-2 WebLogic Integration Environment Variables (Continued)

Variable	Description
DB_DIR	The location of the setDBVars and setDBVarsExt commands for the current domain. For example, if the current domain is configured to use an Oracle database, this variable is set as follows: DB_DIR=$DBINFO/oracle See the description of the DBINFO environment variable (in this table) and the setdbdata command.
DB_DRIVER	The JDBC driver required to connect to the configured database. For example, if the current domain is configured to use an Oracle database, then DB_DRIVER is set as follows: DB_DRIVER="oracle.jdbc.driver.OracleDriver" See the bulkloader, migratedb, and setDBVars commands.
DB_JVMARGS	When the domain is configured to use a PointBase database, this variable is set as follows: DB_JVMARGS="-Dtransaction.isolationlevel=TRANSACTION_READ_COMMITTED -Ddatabase.pagesize=4096 -Dcache.size=2063 -Dsort.size=1024" If a Microsoft SQL, Sybase, or Oracle database is configured, DB_JVMARGS is not set. See the bulkloader, createdb, migratedb, startWeblogic, stopWeblogic, setDBVars, startpointbasecommander, setupAppView, and setupEDIAppView commands.
DB_PASSWORD	The password required to connect to the database configured for the current domain. See the bulkloader, migratedb, and setDBVars commands.
DB_PINGTABLE	Specifies a table in the database. For example, if a Microsoft SQL database is configured for the samples domain on a Windows system, DB_PINGTABLE is set by the setDBVarsExt command as follows: DB_CATALOG=WLIDB DB_SCHEMA=dbo DB_PINGTABLE=%DB_CATALOG%.%DB_SCHEMA%.EVENT The DB_PINGTABLE variable is then invoked by the setupAppView command used to set up the sample application view as follows: %JAVA_HOME%\bin\java . . . -cp %CP% examples.wlis.wlai.WLISAppViewDeployer . . . -pingTable %DB_PINGTABLE% . . . (Only selected parts of the actual java command are shown.)

Table B-2 WebLogic Integration Environment Variables (Continued)

Variable	Description
DB_PROPERTIES	Used with the switchdb command to update the JDBC connection pool properties for the current domain. For example, if the current domain is configured to use an Oracle database, then DB_PROPERTIES is set as follows: On UNIX: DB_PROPERTIES="user=$DB_USER;password=$DB_PASSWORD" On Windows: DB_PROPERTIES="user=%DB_USER%;password=%DB_PASSWORD%" See the setdbdata command and the switchDB.xml file.
DB_SCHEMA	The database schema name. For Microsoft SQL and Sybase this is the database name; for Oracle it is the user name. For the default PointBase database the following setting is used: DB_SCHEMA=PBPUBLIC See the description of the DB_PINGTABLE environment variable (in this table) and the setDBVarsExt and setupAppView commands.
DB_SCRIPT_DIR	The location of the scripts for database configured for the current domain. For example, if the database for the current domain is an Oracle database, then DB_SCRIPT_DIR is set as follows: On UNIX: DB_SCRIPT_DIR="$WLI_HOME/dbscripts/oracle" On Windows: DB_SCRIPT_DIR="%WLI_HOME%/dbscripts/oracle" See the createdb and setDBVars commands.
DB_SERVERNAME	The Microsoft SQL server name. See the migratedb and setDBVars commands.
DB_TYPE	The type of database for the current domain. Valid values are: oracle, mssql, sybase, and pointbase. See the createdb, migratedb, startpointbasecommander, startWeblogic, switchdb, setdbdata, setdbtype, and RunSamples commands.

Table B-2 WebLogic Integration Environment Variables (Continued)

Variable	Description
DB_URL	The URL required by the database driver. The URL is constructed based on information provided during installation or during database configuration via the Database Wizard. For example, if DB_DRIVER="oracle.jdbc.driver.OracleDriver" then DB_URL="jdbc:oracle:thin:@(description=(address=(host=*host*)(protocol=tcp)(port=1521))(connect_data=(sid=*oraclesid*)))" See the bulkloader, migratedb, and setDBVars commands.
DB_USER	The user name required to connect to the database configured for the current domain. See the bulkloader, migratedb, and setDBVars commands.
DBINFO	The location of the dbInfo directory for the current domain. This variable is set as follows: On UNIX: DBINFO=$WLI_DOMAIN/dbInfo On Windows DBINFO=%WLI_DOMAIN%\dbInfo See the setdbdata and switchdb commands.
EVENTROUTER_URL	The URL for the event router for the domain. For example: EVENTROUTER_URL="http://$WLIS_HOST_AND_PORT/DbmsEventRouter/EventRouter" See the Set*DomainType*Data and setupAppView commands.
JAVA	The location of the java command. For example, if you install WebLogic Server in the default location on a Solaris system, JAVA is set as follows: JAVA=$JAVA_BIN/java See the PIMigrator command.
JAVA_HOME	The root directory of the Java Development Kit (JDK) installation. For example, if you install WebLogic Server in the default location on a Windows system, JAVA_HOME is set as follows: JAVA_HOME=C:\bea\jdk131 See the bulkimporter, bulkloader, bulkmigrator, createdb, fb, mdbgenerator, migratedb, setenv, startpointbasecommander, studio, switchdb, wliconfig, worklist, startWeblogic, stopWeblogic, setupAppView, setupEDIAppView, and RunSamples commands.

Table B-2 WebLogic Integration Environment Variables (Continued)

Variable	Description
JAVA_OPTIONS	The run-time options for the `java` command.
JAVACLASSPATH	The classpath variable invoked by the `bulkmigrator` command.
JDBC_DATASOURCE_JNDI_NAME	The JDBC data source JNDI name for the domain. For example: `JDBC_DATASOURCE_JNDI_NAME=WLAI_DataSource` See the `Set`*`DomainType`*`Data` and `setupAppView` commands.
POINTBASE_HOME	For a domain configured to use a PointBase database, this variable sets the location of the PointBase database as follows: `$WLI_DOMAIN/dbInfo/pointbase` See the `setDBVars`, `startpointbasecommander`, and `creatdb` commands.
SAMPLES_HOME	The WebLogic Platform samples directory. For example, if you install WebLogic Platform in the default location on a Windows system, `SAMPLES_HOME` is set as follows: `SAMPLES_HOME=C:\bea\weblogic700\samples` See the `createdb`, `setenv`, `startWeblogic`, `stopWeblogic`, `Set`*`DomainType`*`Data`, `setDBVars`, and `RunSamples` commands.
SQLARGS	The set of options required by a Microsoft SQL or Sybase database. For example, if a Sybase database has been configured for the current domain, `SQLARGS` is set as follows: `SQLARGS="-b -e -n -S` *`servername`* `-U` *`username`* `-P \"`*`password`*`\""` See the `setDBVars`, `createmssql`, and `createsybase` commands.
SVRCP	Server classpath set by the `Set`*`DomainType`*`Data` command. For example, when the `SetSampleData` command is run, `SVRCP` is set as follows: `SVRCP=$WLISERVERCP:$CMNCP`
WL_HOME	The WebLogic Server installation directory. For example, if you install WebLogic Platform in the default location on a Windows system, `WL_HOME` is set as follows: `WL_HOME=C:\bea\weblogic700\server` See the `bulkimporter`, `mdbgenerator`, `setenv`, `startWeblogic`, `stopWeblogic`, and `RunSamples` commands.

Table B-2 WebLogic Integration Environment Variables (Continued)

Variable	Description
WLI_DOMAIN	The currently set domain. See the checkdomain, createdb, migratedb, setdbdata, setdomain, domainname, showdomain, switchdb, Set*DomainType*Data, and startWeblogic commands.
WLI_HOME	The WebLogic Integration installation directory. For example, if you install WebLogic Platform in the default location on a Windows system, WLI_HOME is set as follows: WLI_HOME=C:\bea\weblogic700\integration. See the bulkimporter, bulkloader, checkdomain, createdb, fb, mdbgenerator, migratedb, RunSamples, setdomain, setenv, Set*DomainType*Data, setupAppView, setupEDIAppView, showdomain, startpointbasecommander, startWeblogic, stopWeblogic, studio, switchdb, and worklist commands.
WLI_SAMPLES_HOME	The root of the domain. For example: WLI_SAMPLES_HOME=$SAMPLES_HOME/integration/config/samples See the Set*DomainType*Data command.
WLICOMMONCP	WebLogic Integration common classpath (for both client and server). See the setenv command.
WLICP	The WebLogic Integration client classpath. See the bulkloader, fb, migratedb, setenv, PIMigrator, studio, switchdb, worklist, Set*DomainType*Data, setDBVars, and RunSamples commands.
WLIS_HOST_AND_PORT	The host and listen port for the domain administration server. For example: WLIS_HOST_AND_PORT=localhost:7001 See the Set*DomainType*Data, setupAppView, and setupEDIAppView commands.
WLIS_PASSWORD	The WebLogic Server password for the domain. For example: WLIS_USER_NAME=system See the Set*DomainType*Data, setupAppView, and setupEDIAppView commands.
WLIS_USER_NAME	The WebLogic Server username for the domain. For example: WLIS_PASSWORD=security See the Set*DomainType*Data, setupAppView, and setupEDIAppView commands.

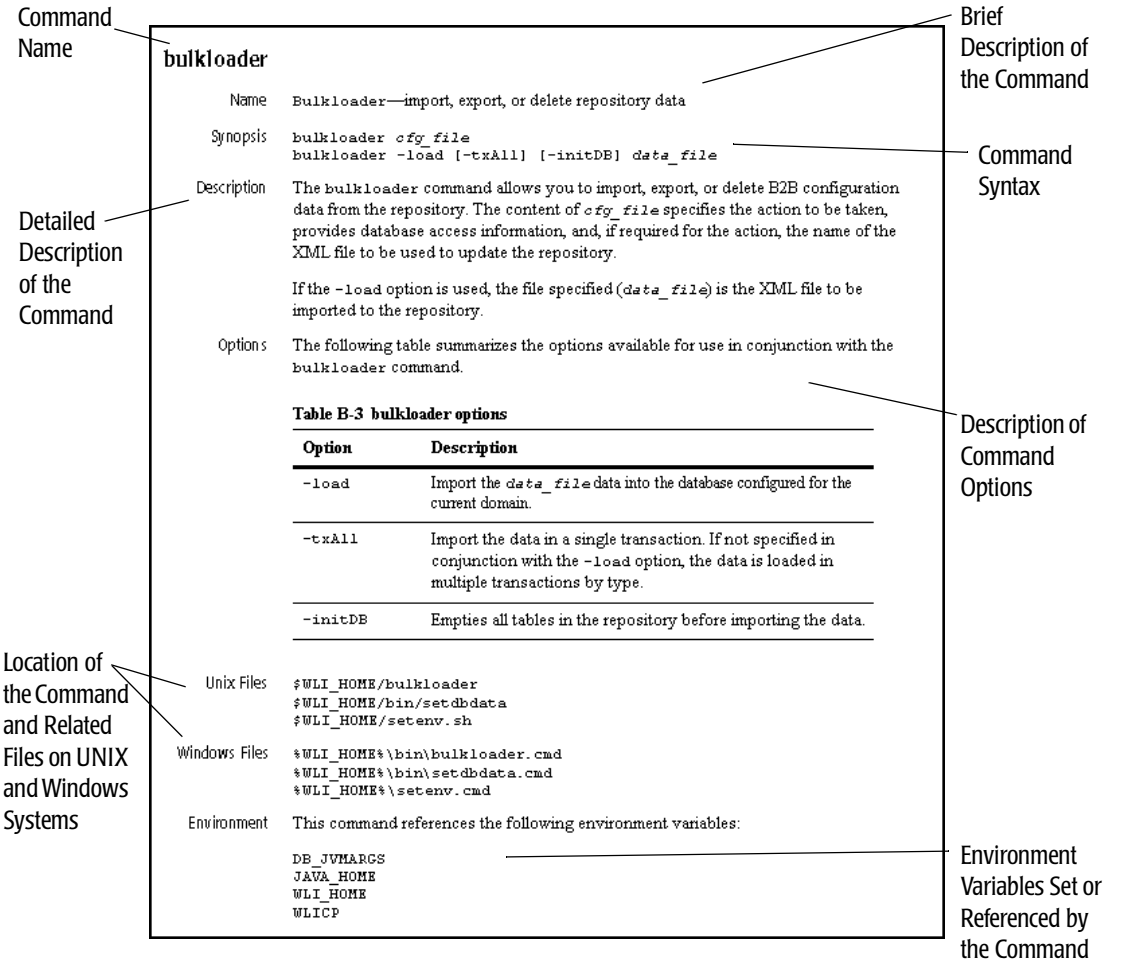
Table B-2 WebLogic Integration Environment Variables (Continued)

Variable	Description
WLISERVERCP	WebLogic Integration server classpath. See the `setenv`, `Set`*`DomainType`*`Data`, and `stopWeblogic` commands.

Command Descriptions

This section presents descriptions of the WebLogic Integration commands, arranged in alphabetical order. The following figure summarizes the format of each command description.

Figure B-1 Command Description Format



bulkimporter

Name `bulkimporter`—import workflow objects from Java Archive (JAR) files

Synopsis

```
bulkimporter -user bpm_user -password bpm_password -url
 wls_server_url -orgid org_name {-file file_name |-dir
 full_dir_path | -dir full_dir_path -file file_name}
[-activate] [-packagepwd password] [-help] [-quiet]
```

Description The `bulkimporter` command allows you to import workflow packages from the command line without using the WebLogic Integration Studio. A workflow package contains the following workflow objects:

- template definitions
- business operations
- business calendars
- event key tables
- XML repository items

Organizations, users, and roles cannot be imported.

At least one jar file must be specified on the command line but many jar files can be imported in one command as shown in the following supported combinations:

- One jar file using the `-file file_name` option.
- A set of jar files in a directory using the `-dir full_dir_path` option.
- One jar file using the `-file file_name` option and a set of jar files in a directory using the `-dir full_dir_path` option on a single command line.

If the `-quiet` option is not specified, then a listing of all the objects imported into WebLogic Integration are displayed.

You can also import workflow objects using the WebLogic Integration Studio. For more information, see "Importing and Exporting Workflow Packages" in *Using the WebLogic Integration Studio*.

Options The following table summarizes the options for the `bulkimporter` command.

Table B-3 bulkimporter Options

Option	Description
`-user` *`bpm_user`*	Specify the WebLogic Integration BPM user name. This is the same password that you use to login to the Studio. For a list of default BPM users, see "BPM User Groups and Permissions" on page 3-44. This option is required.
`-password` *`bpm_password`*	Specify the password for the BPM username specified by the `-user` option. This option is required.
`-url` *`wls_server_url`*	Specify the WebLogic Server URL. To connect to an instance of the WebLogic Server on a local machine with the default port number of 7001, set `wls_server_url` to `t3//localhost:7001`. This option is required.
`-orgid` *`org_name`*	Specify the BPM organization name. For a list of default BPM organizations, see option. This option is required.
`-file` *`file_name`*	Jar file which contains the workflow objects to import. File must end in the `.jar` extension. Either this option, the `-dir` *`full_dir_path`* option, or both the `-file` *`file_name`* and `-dir` *`full_dir_path`* options together must be specified.
`-dir` *`full_dir_path`*	The full directory path of the directory which contains the jar files to import. All the files that end in the `.jar` extension in this directory are imported. Either this option, the `-file` *`full_dir_path`* option, or both the `-file` *`file_name`* and `-dir` *`full_dir_path`* options together must be specified.
`-activate`	Activate the workflows after importing.
`-packagepwd` *`password`*	Specify the workflow password if the workflow package has been password protected.
`-help`	Display all the supported options.
`-quiet`	Run the command quietly, do display any messages while running the command.

UNIX Files `$WLI_HOME/bin/bulkimporter`

Windows Files `%WLI_HOME%\bin\bulkimporter.cmd`

Environment

This command references the following environment variables:

```
WL_HOME
WLI_HOME
JAVA_HOME
```

If `WLI_HOME` is not set, this command executes the `setenv` command.

bulkloader

Name `bulkloader`—import, export, or delete repository data

Synopsis
```
bulkloader [-wlibc] cfg_file
bulkloader -load [-txAll] [-initDB][-wlibc] data_file
```

Description The `bulkloader` command allows you to import, export, or delete B2B configuration data from the repository. The content of *`cfg_file`* specifies the action to be taken, provides database access information, and, if required for the action, supplies the name of the XML file to be used to update the repository.

If the `-load` option is used, the file specified (*`data_file`*) is the XML file to be imported to the repository. The database information is obtained from settings in the `setDBVars` command for the current domain. The current domain is set by the `setdomain` command. You can verify the current domain with the `checkdomain` command.

For detailed information about the Bulk Loader, and the contents of the *`cfg_file`* and *`data_file`* files, see "Working with the Bulk Loader" in *Administering B2B Integration*.

Note: By default, the Bulk Loader does not import, export, or delete data while the B2B engine is running. If, for some reason, the B2B engine does not shut down normally, the Bulk Loader fails with the following error:

```
ERROR: WLC is still running or was shut down abnormally.
```

To override this error and force the command to execute by modifying the Bulk Loader command file, see "Forcing the Bulkloader" in "Working with the Bulk Loader" in *Administering B2B Integration*.

Options The following table summarizes the options for the `bulkloader` command.

Table B-4 bulkloader Options

Option	Description
`-wlibc`	Use the `-wlibc` flag when you import an XML file created by a trading partner using the lightweight client, WebLogic Integration – Business Connect
`-load`	Import the *`data_file`* data into the database configured for the current domain.

Table B-4 bulkloader Options (Continued)

Option	Description
`-txAll`	Import the data in a single transaction. When not specified, a transaction is initiated upon import or delete of each of the following entities: trading partners, conversation definitions, collaboration agreements, business protocol definitions, and logic plug-ins. If invalid data is detected during a transaction for any entity, the import is rolled back for the current transaction only; importing continues for the next transaction.
`-initDB`	Empties all tables in the repository before importing the data. **Warning:** When you specify the `-initDB` option the existing data in the database is destroyed. If the data you are loading does not include the required system data, this option renders the system unusable. Because data created by a WebLogic Integration – Business Connect trading partner does not include system data, never use this option in conjunction with the `-wlibc` option. Do not specify `-initDB` unless you intend to reinitialize the entire database.

UNIX Files

```
$WLI_HOME/bulkloader
$WLI_HOME/bin/setdbdata
$WLI_HOME/setenv.sh
```

Windows Files

```
%WLI_HOME%\bin\bulkloader.cmd
%WLI_HOME%\bin\setdbdata.cmd
%WLI_HOME%\setEnv.cmd
```

Environment

This command references the following environment variables:

```
DB_JVMARGS
JAVA_HOME
WLI_HOME
WLICP
```

When executed with the `-load` option, this command references:

```
DB_DRIVER
DB_URL
DB_USER
DB_PASSWORD
```

If `WLI_HOME` is not set, this command executes the `setenv` command.

bulkmigrator

Name `bulkmigrator`—migrate a repository data file

Synopsis `bulkmigrator` *`collaborate1.0_file wlintegration2.0_file`*

Description The `bulkmigrator` command converts an XML data file exported from WebLogic Collaborate 1.0.x (*`collaborate1.0_file`*) to the format required for WebLogic Integration 2.0 (*`wlintegration2.0_file`*). Some additional steps must be taken to make the file compatible for import to WebLogic Integration 7.0.

For detailed instructions, see "Migrating from Releases Prior to WebLogic Integration 2.1" in "Introduction to Migration" in *BEA WebLogic Integration Migration Guide*.

Options None.

UNIX Files `$WLI_HOME/bin/bulkmigrator`

Windows Files `%WLI_HOME%\bin\bulkmigrator.cmd`

Environment This command references the following environment variables:

```
JAVA_HOME
JAVACLASSPATH
```

checkdomain

Name `checkdomain`—validate a domain and set `WLI_DOMAIN` for the calling process

Synopsis

```
checkdomain
```

Description The `checkdomain` command is used in conjunction with `setdomain` and other command scripts. The `checkdomain` command validates that a domain has the directories and files required for other commands to run, and, if the domain is valid, it executes the `domainname` command to set `WLI_DOMAIN` for the calling process.

Note: The `domainname` command is created when the `setdomain` command is executed. This command stores the current `WLI_DOMAIN` setting.

Options None.

UNIX Files

```
$WLI_HOME/bin/checkdomain
DOMAIN_HOME/checkdomain
$WLI_HOME/bin/domainname
$WLI_HOME/setenv.sh
```

Windows Files

```
%WLI_HOME%\bin\checkdomain.cmd
DOMAIN_HOME\checkdomain.cmd
%WLI_HOME%\bin\domainname.cmd
%WLI_HOME%\setEnv.cmd
```

Environment This command sets `WLI_DOMAIN` and references `WLI_HOME`.

If `WLI_HOME` is not set, this command executes the `setenv` command.

createdb

Name `createdb`—create repository tables and load system data

Synopsis
```
createdb [-f] [-noinit]
```

Description The `createdb` command creates the WebLogic Integration repository tables in the database configured for the current domain, and then, if required, loads the system data into that database. The command calls the `setdbdata` command to load the current database access information and the `bulkloader` command to load the data.

The tables created are dependent on the requirements of the current domain. (For a description of the required tables, see Table 1-3 in "Domain Configuration Requirements" on page 1-11.) For example, if the current domain does not support B2B functionality, then the B2B tables are not created, and the B2B system data is not loaded. If the current domain is the WebLogic Integration samples domain, additional, sample-specific tables are created.

Note: Because the `osql` client used to create tables in a Microsoft SQL database is not available on UNIX, you cannot use the `createdb` command to create tables in a Microsoft SQL database from a UNIX system.

Options The following table summarizes the options for the `createdb` command.

Table B-5 createdb Options

Option	Description
`-f`	Do not prompt for confirmation. By default, the command prompts for confirmation before proceeding.
`-noInit`	Create tables only. Do not populate the repository with system data. By default, `createdb` loads system data into the repository.

UNIX Files
```
$WLI_HOME/bin/createdb
$WLI_HOME/bin/setdbdata
$WLI_HOME/bin/bulkloader
$WLI_HOME/dbscripts/$DB_TYPE/REPOSITORY_SCHEMA_BASE.sql
$WLI_HOME/dbscripts/$DB_TYPE/REPOSITORY_SCHEMA.sql
$WLI_HOME/dbscripts/$DB_TYPE/PERSISTENCE_SCHEMA.sql
$WLI_HOME/dbscripts/$DB_TYPE/BPM_SCHEMA.sql
$SAMPLES_HOME/integration/samples/wlis/dbscripts/$DB_TYPE/WLIS_SCHEM
A.sql
$SAMPLES_HOME/integration/samples/e2e/dbscripts/$DB_TYPE/E2E_SCHEMA.
sql
$WLI_HOME/adapters/dbms/src/sql/msql/cust.sql
```

```
$WLI_HOME/adapters/dbms/src/sql/msql/evnt.sql
$WLI_HOME/adapters/dbms/src/sql/oracle/cust.sql
$WLI_HOME/adapters/dbms/src/sql/oracle/evnt.sql
$WLI_HOME/adapters/dbms/src/sql/pointbase/cust.sql
$WLI_HOME/adapters/dbms/src/sql/pointbase/evnt.sql
$WLI_HOME/adapters/dbms/src/sql/sybase/cust.sql
$WLI_HOME/adapters/dbms/src/sql/sybase/evnt.sql
$WLI_HOME/dbscripts/SystemRepData.xml
$WLI_HOME/setenv.sh
```

Windows Files

```
%WLI_HOME%\bin\createDB.cmd
%WLI_HOME%\bin\setdbdata.cmd
%WLI_HOME%\bin\bulkloader.cmd
%WLI_HOME%\dbscripts\%DB_TYPE%\REPOSITORY_SCHEMA_BASE.sql
%WLI_HOME%\dbscripts\%DB_TYPE%\REPOSITORY_SCHEMA.sql
%WLI_HOME%\dbscripts\%DB_TYPE%\PERSISTENCE_SCHEMA.sql
%WLI_HOME%\dbscripts\%DB_TYPE%\BPM_SCHEMA.sql
%SAMPLES_HOME%\integration\samples\wlis\dbscripts\%DB_TYPE%\WLIS_SCH
EMA.sql
%SAMPLES_HOME%\integration\samples\e2e\dbscripts\%DB_TYPE%\E2E_SCHEM
A.sql
%WLI_HOME%\adapters\dbms\src\sql\msql\cust.sql
%WLI_HOME%\adapters\dbms\src\sql\msql\evnt.sql
%WLI_HOME%\adapters\dbms\src\sql\oracle\cust.sql
%WLI_HOME%\adapters\dbms\src\sql\oracle\evnt.sql
%WLI_HOME%\adapters\dbms\src\sql\pointbase\cust.sql
%WLI_HOME%\adapters\dbms\src\sql\pointbase\evnt.sql
%WLI_HOME%\adapters\dbms\src\sql\sybase\cust.sql
%WLI_HOME%\adapters\dbms\src\sql\sybase\evnt.sql
%WLI_HOME%\dbscripts\SystemRepData.xml\
%WLI_HOME%\setEnv.cmd
```

Environment

The `createdb` command references the following environment variables:

```
CLCP
DB_COMMAND
DB_CREATE
DB_JVMARGS
DB_PORT
DB_SCRIPT_DIR
DB_TYPE
JAVA_HOME
POINTBASE_HOME
SAMPLES_HOME
WLI_DOMAIN
WLI_HOME
```

If `WLI_HOME` is not set, this command executes the `setenv` command.

createmssql
createsybase

Name createmssql—create Microsoft SQL database and execute a script in that database
createsybase—create Sybase database and execute a script in that database

Synopsis
```
createmssql database_name sql_script
createsybase database_name sql_script
```

Description The createmssql and createsybase commands check for the existence of the specified database. If the database exists, the command then runs the specified SQL script. (This action is equivalent to invoking the DB_COMMAND environment variable, which also executes the specified *sql_script*.) If the database does not exist, it is created, and the specified script is executed in the newly created database.

For example, if a Microsoft SQL database named WLIDB is configured for the current domain, the DB_CREATE environment variable in the %WLI_DOMAIN%\dbInfo\mssql\setDBVars command file is set as follows:

```
DB_CREATE=call %WLI_HOME%\dbscripts\mssql\createmssql WLIDB
```

The DB_CREATE environment variable is invoked by the createdb command as follows:

```
%DB_CREATE%REPOSITORY_SCHEMA_BASE.sql
```

If the WLIDB database already exists, invoking the DB_CREATE environment variable is equivalent to invoking the DB_COMMAND environment variable. If the WLIDB database does not exist, it is created, and the specified script is executed in that database.

Note: Because the osql client is not available on UNIX, the createmssql command is not available on UNIX. You cannot use the createdb command on a UNIX system to create tables in a Microsoft SQL database.

Options None.

UNIX Files
```
$WLI_HOME/dbscripts/sybase/createsybase
```

Windows Files
```
%WLI_HOME%\dbscripts\mssql\createmssql.cmd
%WLI_HOME%\dbscripts\sybase\createsybase.cmd
```

Environment These commands reference the following environment variables:

```
SQLARGS
DB_COMMAND
```

domainname

Name `domainname`—set or verify the current domain set by the `setdomain` command

Synopsis `domainname`

Description This machine-generated command stores the current domain setting. This file is generated when the `setdomain` command is executed. The `checkdomain` command uses the setting in this file to verify the domain and set `WLI_DOMAIN` for the calling process.

Options None.

UNIX Files `$WLI_HOME/bin/domainname`

Windows Files `%WLI_HOME%\bin\domainname.cmd`

Environment This command sets the `WLI_DOMAIN` environment variable.

fb

Name `fb`—start Format Builder

Synopsis

```
fb
```

Description This command starts the Format Builder client. A shortcut to the `fb.cmd` command is provided on the Windows Start menu.

Options None.

UNIX Files

```
$WLI_HOME/bin/fb
$WLI_HOME/setenv.sh
```

Windows Files

```
%WLI_HOME%\bin\fb.cmd
%WLI_HOME%\setEnv.cmd
```

Environment The `fb` command sets but does not export the `CP` environment variable:

The `fb` command references the following environment variables:

```
WLICP
WLI_HOME
JAVA_HOME
```

If `WLI_HOME` is not set, this command executes the `setenv` command.

launchSamplesPage

Name `launchSamplesPage`—open the samples launcher page in the default Web browser

Synopsis `launchSamplesPage`

Description This command opens `http://localhost:7001/index.html`. If the administration server for the samples domain is running on the local machine, this command opens a page from which the samples can be run.

Note: On UNIX systems, the browser executable `netscape` must be included in the system path variable.

Options None.

UNIX Files `$SAMPLES_HOME/integration/samples/bin/launchSamplesPage`

Windows Files `%SAMPLES_HOME%\integration\samples\bin\launchSamplesPage.cmd`

Environment No environment variables are referenced or set.

mdbgenerator

Name `mdbgenerator`—generate a deployable JAR file to listen on a custom queue

Synopsis `mdbgenerator -queue` *`queue_name`* [`-min` *`number`*] [`-max` *`number`*] [`-order` *`number`*] [`-transact`] [`-validate`] [`-timeout` *`seconds`*] [`-help`]

Description The message driven bean (MDB) generator utility (the `mdbgenerator` command) generates a deployable Java Archive (JAR) file that can be used to listen on a custom Java Message Service (JMS) queue.

Options The following table summarizes the options for the `mdbgenerator` command.

Table B-6 mdbgenerator Options

Option	Description
`-queue` *`queue_name`*	Name of the custom queue for which you want to generate the MDB. This option is required.
`-min` *`number`*	Minimum number of unordered listeners.
`-max` *`number`*	Maximum number of unordered listeners.
`-order` *`number`*	Number of ordered listeners. The number must be prime and less than or equal to 31.
`-transact`	Sets the transaction to required.
`-validate`	Turns on XML validation.
`-timeout` *`seconds`*	Sets the transaction timeout in seconds. This option defaults to 30 seconds. This value is only used if the `transact` flag is not set. If the `transact` flag is set, the utility uses the WebLogic Server transaction timeout value, which can be set using the Administration Console and defaults to 30.
`-help`	Displays command usage syntax.

UNIX Files
```
$WLI_HOME/bin/mdbgenerator
$WLI_HOME/setenv.sh
```

Windows Files
```
%WLI_HOME%\bin\mdbgenerator
%WLI_HOME%\setEnv.cmd
```

Environment

This command references the following environment variables:

```
WLI_HOME
WL_HOME
JAVA_HOME
```

If `WLI_HOME` is not set, this command executes the `setenv` command.

migratedb

Name `migratedb`—migrate a database from WebLogic Integration 2.1 or WebLogic Integration 2.1 Service Pack 1 (SP1) to WebLogic Integration 7.0

Synopsis

```
migratedb
```

Description The `migratedb` command creates the scripts required to migrate the database configured for the current domain, and then executes those scripts against the database. The command updates WebLogic Integration 2.1 or WebLogic Integration 2.1 SP1 database tables and schema as required for compatibility with WebLogic Integration 7.0.

For additional information see *BEA WebLogic Integration Migration Guide*.

Options None.

UNIX Files

```
$WLI_HOME/bin/migratedb
$WLI_HOME/bin/checkdomain
$WLI_HOME/bin/setdbdata
$WLI_HOME/bin/bulkloader
$WLI_HOME/setenv.sh
$WLI_HOME%/dbscripts/migrate/SystemRepData.xml
```

Windows Files

```
%WLI_HOME%\bin\migratedb.cmd
%WLI_HOME%\bin\checkdomain.cmd
%WLI_HOME%\bin\setdbdata.cmd
%WLI_HOME%\bin\bulkloader.cmd
%WLI_HOME%\setEnv.cmd
%WLI_HOME%\dbscripts\migrate\SystemRepData.xml
```

Environment This command references the following environment variables:

```
DB_DRIVER
DB_NAME
DB_PASSWORD
DB_SERVERNAME
DB_TYPE
DB_URL
DB_USER
JAVA_HOME
WLI_DOMAIN
WLI_HOME
WLICP
```

If `WLI_HOME` is not set, this command executes the `setenv` command.

PIMigrator

Name `PIMigrator`—migrate a workflow template

Synopsis `PIMigrator` *`InputTemplate.xml OutputTemplate.xml`*

Description The `PIMigrator` command updates a template created for use with Collaborate 1.0 to the format required by WebLogic Integration 2.0/2.1.

Options None.

UNIX Files

```
$WLI_HOME/bin/PIMigrator
$WLI_HOME/lib/dtd/PITemplateMigration.xsl
```

Windows Files

```
%WLI_HOME%\bin\PIMigrator.cmd
%WLI_HOME%\lib\dtd\PITemplateMigration.xsl
```

Environment This command references the following environment variables:

```
JAVA
WLC_HOME
WLICP
```

RunSamples

Name `RunSamples`—configure the samples database and start WebLogic Server in the samples domain

Synopsis `RunSamples [option . . . ]`

Description The `RunSamples` command performs the following tasks for the samples domain:

- Creates the required repository tables in the samples database repository. See Table 1-3 in "Domain Configuration Requirements" on page 1-11.
- Loads the B2B integration system data and sample configuration data
- Starts WebLogic Integration in the samples domain
- Configures the application view for the End-to-End sample application
- Imports the workflow package containing the sample workflows
- Launches your default Web browser and displays the Samples Launcher page

When the `RunSamples` command is executed with the default option, `all`, or with no options, its behavior depends on whether it is being run for the first time against the database configured for the samples domain. For additional information see "Configuring and Starting the Samples Domain" on page 1-14.

Options The following table summarizes the options for the `RunSamples` command.

Table B-7 RunSamples Options

Option	Description
`all`	Perform all steps necessary to run the samples. This option is invoked by default.
`BulkLoader`	Run the Bulk Loader for the samples
`CreateDB`	Create the database for the samples
`ImportPackages`	Import BPM packages into the server
`SampleDB`	Create sample specific database tables
`SetupAppView`	Set up the application view for the sample adapter

Table B-7 RunSamples Options (Continued)

Option	Description
`StartServer`	Start the sample server

UNIX Files

```
$SAMPLES_HOME/integration/samples/bin/RunSamples
$WLI_HOME/bin/setdomain
$SAMPLES_HOME/integration/samples/bin/SetSampleData
```

Windows Files

```
%SAMPLES_HOME%\integration\samples\bin\RunSamples.cmd
%WLI_HOME%\bin\setdomain.cmd
%SAMPLES_HOME%\integration\samples\bin\SetSampleData.cmd
```

Environment

This command references the following environment variables:

```
WLI_HOME
WLICP
WL_HOME
JAVA_HOME
DB_TYPE
```

If `WLI_HOME` is not set, this command executes the `setenv` command.

setdbdata

Name `setdbdata`—load the database-specific variables into the current process

Synopsis `setdbdata`

Description The `setdbdata` command loads the database-specific variables into the current process by:

- Calling the `checkdomain` command to load the current domain information.
- Calling the `setdbtype` command to set the database type. The `setdbtype` command is generated based on information provided when you use the Configuration Wizard to create a domain, execute the `switchdb` command, or use the WebLogic Integration Database Wizard (the `wliconfig` command) to update the database for a domain.
- Executing the `setDBVars` command.
- Executing the `setDBVarsExt` command, if it exists. (The `setDBVarsExt` command is optional. No error is generated if it does not exist.)

Options None.

UNIX Files

```
$WLI_HOME/bin/setdbdata
$WLI_HOME/bin/checkdomain
DOMAIN_HOME/setdbdata
DOMAIN_HOME/checkdomain
DOMAIN_HOME/dbInfo/$DB_TYPE/setDBVars
DOMAIN_HOME/dbInfo/$DB_TYPE/setDBVarsExt
DOMAIN_HOME/dbInfo/setdbtype
```

Windows Files

```
%WLI_HOME%\bin\setdbdata.cmd
%WLI_HOME%\bin\checkdomain.cmd
DOMAIN_HOME\setdbdata.cmd
DOMAIN_HOMEn\checkdomain.cmd
DOMAIN_HOME\dbInfo\%DB_TYPE%\setDBVars
DOMAIN_HOME\dbInfo\%DB_TYPE%\setDBVarsExt
DOMAIN_HOME\dbInfo\setdbtype
```

Environment In addition to the variables set by `setdbtype` and `setDBVars`, the `setdbdata` command sets the following:

```
DBINFO
DB_CREATE
DB_DIR
DB_PROPERTIES
```

Note: `DB_CREATE` is set to `DB_COMMAND` if it is not defined by `setDBVars`.

setdbtype

Name `setdbtype`—set the database type for the domain

Synopsis `setdbtype`

Description This command is generated based on information provided when you use the Configuration Wizard to create a domain, execute the `switchdb` command, or use the WebLogic Integration Database Wizard (the `wliconfig` command) to update the database for a domain.

Options None.

UNIX Files *DOMAIN_HOME*`/dbInfo/setdbtype`

Windows Files *DOMAIN_HOME*`\dbInfo\setdbtype.cmd`

Environment This command sets the `DB_TYPE` environment variable.

setDBVars
setDBVarsExt

Name `setDBVars`—set the basic database variables for the domain
`setDBVarsExt`—set the extended database variables for the domain

Synopsis
```
setDBVars
setDBVarsExt
```

Description The `setDBVars` and `setDBVarsExt` commands for a domain are updated based on the information provided when you use the Configuration Wizard to create a domain, execute the `switchdb` command, or use the WebLogic Integration Database Wizard (the `wliconfig` command) to update the database for a domain.

Options None.

UNIX Files
```
$WLI_DOMAIN/dbInfo/$DB_TYPE/setDBVars
$WLI_DOMAIN/dbInfo/$DB_TYPE/setDBVarsExt
```

Windows Files
```
%WLI_DOMAIN%\dbInfo\%DB_TYPE%\setDBVars
%WLI_DOMAIN%\dbInfo\%DB_TYPE%\setDBVarsExt
```

Environment The `setDBVars` command sets the following environment variables:

```
DB_JVMARGS
DB_USER
DB_PASSWORD
DB_CODESET
DB_SERVERNAME
DB_NAME
DB_URL
DB_DRIVER
DB_SCRIPT_DIR
SQLARGS
DB_COMMAND
DB_CREATE
CLCP
```

The `setDBVarsExt` command sets the following environment variables:

```
DB_CATALOG
DB_SCHEMA
DB_PINGTABLE
```

Note: Some variables are specific to database type. For example, `CLCP` is set for PointBase only.

setdomain

Name `setdomain`—set the current domain

Synopsis `setdomain` *`domain_path`*

Here, *`domain_path`* is the path to the domain relative to the current directory. For example, if are on a UNIX system, and execute the command from the `$BEA_HOME` directory to set the current domain to `$BEA_HOME/user_projects/mydomain`, enter the following:

```
setdomain user_projects/mydomain
```

If you execute the command from the domain directory (for example, from the `$BEA_HOME/user_projects/mydomain` directory), you need not specify *`domain_path`*.

Note: You can execute the `setdomain` command from any directory if you first execute the `setenv` command.

Description The `setdomain` command verifies that the specified directory is a valid domain. If the domain is valid, the command sets the `WLI_DOMAIN` environment variable to the specified domain and creates the `domainname` command to store the setting. The current domain must be set for the `checkdomain`, `setdbdata`, `RunSamples`, and `switchdb` commands.

Options None.

UNIX Files

```
$WLI_HOME/bin/setdomain
$WLI_HOME/setenv.sh
$WLI_HOME/bin/domainname
```

Windows Files

```
%WLI_HOME%\bin\setdomain.cmd
%WLI_HOME%\setEnv.cmd
%WLI_HOME%\bin\domainname.cmd
```

Environment The `setdomain` command sets the `WLI_DOMAIN` environment variable and references the `WLI_HOME` environment variable.

If `WLI_HOME` is not set, this command executes the `setenv` command.

setenv

Name `setenv`—set common WebLogic Integration environment variables

Synopsis

```
setenv
```

Description The `setenv` command sets the basic WebLogic Integration environment settings.

Options None.

UNIX Files

```
$WLI_HOME/setenv.sh
$WLI_HOME/bin/setenv
```

Windows Files

```
%WLI_HOME%\setEnv.cmd
%WLI_HOME%\bin\setenv.cmd
%WLI_HOME%\samples\bpm_api\commandline\setEnv.cmd
%WLI_HOME%\samples\bpm_api\plugin\setEnv.cmd
```

Environment The `setenv` command sets the following environment variables:

```
BEA_HOME
JAVA_HOME
SAMPLES_HOME
WL_HOME
WLI_HOME
WLICOMMONCP
WLICP
WLISERVERCP
```

The command also adds `%JAVA_HOME%\bin` to the existing `PATH` environment variable.

SetDomainTypeData
SetSampleData

Name

Set*DomainType*Data—set environment variables for the *DomainType* domain
SetSampleData—set environment variables for the samples domain

Synopsis

```
Set DomainType Data
SetSampleData
```

Description

The Set*DomainType*Data command sets environment variables for a custom domain created with the Configuration Wizard.

Here the value of *DomainType* is determined by which Configuration Wizard template is used to create the custom domain. For example, if you used the WebLogic Integration template, the command is SetWliDomainData.

The SetSampleData command sets environment variables for the WebLogic Integration samples domain.

Options

None.

UNIX Files

```
DOMAIN_HOME/SetDomainTypeData
$SAMPLES_HOME/integration/samples/bin/SetSampleData
$WLI_HOME/setenv.sh
$WLI_HOME/bin/setdbdata
```

Windows Files

```
DOMAIN_HOME\SetDomainTypeData
%SAMPLES_HOME%\integration\samples\bin\SetSampleData.cmd
%WLI_HOME%\setEnv.cmd
%WLI_HOME%\setdbdata.cmd
```

Environment

Two commands, Set*DomainType*Data and SetSampleData set the following environment variables for the custom domain and the WebLogic Integration sample domain, respectively.

```
WLI_SAMPLES_HOME (SetSampleData)
WLI_DOMAIN_HOME (SetDomainTypeData)
WLIS_HOST_AND_PORT
WLIS_USER_NAME
WLIS_PASSWORD
JDBC_DATASOURCE_JNDI_NAME
EVENTROUTER_URL
CMNCP
CP
SVRCP
```

The `Set`*`DomainType`*`Data` command references the following environment variables:

```
WLICP
WLISERVERCP
WLI_HOME
CMNCP
SVRCP
WLI_DOMAIN_HOME
WLIS_HOST_AND_PORT
```

setupAppView
setupEDIAppView

Name

`setupAppView`—set up the application view for the sample adapter
`setupEDIAppView`—set up the application view for the EDI adapter

Synopsis

```
setupAppView
setupEDIAppView
```

Description

The `setupAppView` command sets up the application view for sample adapter (`WLISAppView`). The `setupEDIAppView` command sets up the application view for the EDI adapter (`EDIAppView`).

Both commands execute the `SetSampleData` command before issuing the `java` command that sets up the application view.

Options

None.

UNIX Files

```
$SAMPLES_HOME/integration/samples/bin/setupAppView
$SAMPLES_HOME/integration/samples/bin/setupEDIAppView
$SAMPLES_HOME/integration/samples/bin/SetSampleData
$WLI_HOME/setenv.sh
```

Windows Files

```
%SAMPLES_HOME%\integration\samples\bin\setupAppView.cmd
%SAMPLES_HOME%\integration\samples\bin\setupEDIAppView.cmd
%SAMPLES_HOME%\integration\samples\bin\SetSampleData.cmd
%WLI_HOME%\setEnv.cmd
```

Environment

The `setupAppView` and the `setupEDIAppView` commands reference the following environment variables:

```
CP
DB_JVMARGS
JAVA_HOME
WLI_HOME
WLIS_HOST_AND_PORT
WLIS_PASSWORD
WLIS_USER_NAME
```

In addition, the `setupAppView` command references the following environment variables:

```
DB_PINGTABLE
DB_CATALOG
DB_SCHEMA
EVENTROUTER_URL
JDBC_DATASOURCE_JNDI_NAME
```

showdomain

Name `showdomain`—display the current domain

Synopsis

```
showdomain
```

Description This command verifies that the current domain is valid, and it displays the current domain setting.

Options None.

UNIX Files

```
$WLI_HOME/bin/showdomain
$WLI_HOME/bin/checkdomain
$WLI_HOME/setenv.sh
```

Windows Files

```
%WLI_HOME%\bin\showdomain.cmd
%WLI_HOME%\bin\checkdomain.cmd
%WLI_HOME%\setEnv.cmd
```

Environment The `showdomain` command references the `WLI_HOME` environment variable.

If `WLI_HOME` is not set, this command executes the `setenv` command.

startAiconsole
startB2bconsole

Name `startAiconsole`—start the Application View Console
`startB2bconsole`—start the B2B Console

Synopsis
```
startAiconsole
startB2bconsole
```

Description The `startAiconsole` command launches your default browser and opens the URL for the Application View Console (`http://localhost:7001/wlai`). This command can only be used to access the Application View Console if the administration server for the domain is running on the local host.

The `startB2bconsole` command launches your default browser and opens the URL for the B2B Console (`http://localhost:7001/b2bconsole`). This command can only be used to access the B2B Console if the administration server for the domain is running on the local host.

Shortcuts to these commands are provided on the Windows Start menu.

Note: On UNIX systems, the browser executable `netscape` must be included in the system path variable.

Options None.

UNIX Files These command are not available on UNIX systems.

Windows Files
```
%WLI_HOME%\bin\startAiconsole.cmd
%WLI_HOME%\bin\startB2bconsole.cmd
```

Environment This command references the `$WLI_HOME` environment variable.

startpointbasecommander

Name `startpointbasecommander`—start the PointBase administrative tool

Synopsis

```
startpointbasecommander
```

Description The `startpointbasecommander` command verifies the current database setting, and if the setting is PointBase, it launches PointBase commander, the PointBase administrative tool.

Options None.

UNIX Files

```
$WLI_HOME/bin/startpointbasecommander
$WLI_HOME/bin/setdbdata
```

Windows Files

```
%WLI_HOME%\bin\startpointbasecommander.cmd
%WLI_HOME%\bin\setdbdata.cmd
```

Environment This command references the following environment variables:

```
CLCP
DB_JVMARGS
DB_PORT
DB_TYPE
JAVA_HOME
POINTBASE_HOME
```

If `WLI_HOME` is not set, this command executes the `setenv` command.

startWeblogic

Name `startWeblogic`—start WebLogic Server and deploy WebLogic Integration resources

Synopsis `startWeblogic`

Description The `startWeblogic` command:

- Executes the `setenv`, `setdomain`, and `Set`*`DomainType`*`Data` commands to set the domain environment.
- Removes old log files.
- Verifies whether or not the database type has been set for the domain. If the domain is configured to use a default PointBase database, launches PointBase commander, the PointBase administrative tool.
- Completes server specific tasks. For example, if the server supports application integration functionality, this command creates the application integration deployment directory (*`DOMAIN_HOME`*`/wlai`) if it does not exist. If the `startWebLogic` command for the WebLogic Integration samples server is executed, the WebLogic Integration sample data directory (`$SAMPLES_HOME/integration/config/samples/data`) is created if it does not exist.
- Executes the `java` command that starts WebLogic Server in the domain and deploys WebLogic Integration applications and resources as specified in the `config.xml` file.

Options None.

UNIX Files
```
DOMAIN_HOME/startWeblogic
DOMAIN_HOME/SetDomainTypeData
DOMAIN_HOME/dbinfo/setdbdata
$SAMPLES_HOME/integration/samples/bin/SetSampleData
$WLI_HOME/setenv.sh
```

Windows Files
```
DOMAIN_HOME\startWeblogic.cmd
DOMAIN_HOME\SetDomainTypeData.cmd
DOMAIN_HOME\dbInfo\setdbdata.cmd
%SAMPLES_HOME%\integration\samples\bin\SetSampleData.cmd
%WLI_HOME%\setEnv.cmd
```

Environment The `startWeblogic` command references the following environment variables:

```
CLCP
DB_JVMARGS
DB_PORT
DB_TYPE
JAVA_HOME
SVRCP
WL_HOME
WLI_DOMAIN_HOME
WLI_HOME
```

stopWeblogic

Name `stopWeblogic`—shut down WebLogic Server

Synopsis

```
stopWeblogic
```

Description The `stopWeblogic` command executes the `setenv` command, and then issues a shutdown request to the WebLogic Server running in the domain.

Options None.

UNIX Files

```
DOMAIN_HOME/stopWeblogic
$WLI_HOME/setenv.sh
```

Windows Files

```
DOMAIN_HOME\stopWeblogic.cmd
%WLI_HOME%\setEnv.cmd
```

Environment The `stopWeblogic` command references the following environment variables:

```
BEA_HOME
DB_JVMARGS
JAVA_HOME
WL_HOME
WLI_HOME
WLISERVERCP
```

studio

Name `studio`—start the WebLogic Integration Studio

Synopsis

```
studio
```

Description The `studio` command starts the WebLogic Integration Studio client.

Options None.

UNIX Files

```
$WLI_HOME/bin/studio
$WLI_HOME/setenv.sh
```

Windows Files

```
%WLI_HOME%\bin\studio.cmd
%WLI_HOME%\setEnv.cmd
```

Environment The `studio` command sets `CP=$WLI_HOME/lib/wlpi-studio.jar:$WLICP` (UNIX) or `CP=%WLI_HOME%\lib\wlpi-studio.jar:%WLICP%` (Windows), and references the following environment variables:

```
CP
JAVA_HOME
WLI_HOME
WLICP
```

If `WLI_HOME` is not set, this command executes the `setenv` command.

switchdb

Name `switchdb`—update the database configured for the current domain

Synopsis

```
switchdb mssql|oracle|pointbase|sybase
```

Description The `switchdb` command updates the database configuration for the current domain by:

- Executing `checkdomain` to verify the current domain.
- Creating or recreating the `dbInfo/setdbtype` command for the domain.
- Updating the JDBC settings in the `config.xml` file based on the current settings in the `setDBVars` command for the specified database type.

The `switchdb` command does not update the `setDBVars` file. If the database configuration reflected in the `setDBVars` file requires modification, use the WebLogic Integration Database Wizard (the `wliconfig` command) to update the configuration.

Options None.

UNIX Files

```
$WLI_HOME/bin/switchdb
$WLI_HOME/bin/checkdomain
$WLI_HOME/setenv.sh
$WLI_DOMAIN/dbInfo/setdbtype
$WLI_DOMAIN/dbInfo/$DB_TYPE/setDBVars
$WLI_DOMAIN/scripts/SwitchDB.xml
$WLI_HOME/lib/scripts/SwitchDB.xml
```

Windows Files

```
%WLI_HOME%\bin\switchdb.cmd
%WLI_HOME%\bin\checkdomain.cmd
%WLI_HOME%\setEnv.cmd
%WLI_DOMAIN%\dbInfo\setdbtype.cmd
%WLI_DOMAIN%\dbInfo\%DB_TYPE%\setDBVars.cmd
%WLI_DOMAIN%\scripts\SwitchDB.xml
%WLI_HOME%\lib\scripts\SwitchDB.xml
```

Environment The switchdb command sets the `DBINFO` and `DB_TYPE` environment variables, and references the `WLI_DOMAIN` and `JAVA_HOME` environment variables.

If `WLI_HOME` is not set, this command executes the `setenv` command.

worklist

Name `worklist`—start the WebLogic Integration Worklist

Synopsis

```
worklist
```

Description The `worklist` command starts the WebLogic Integration Worklist client.

Options None.

UNIX Files

```
$WLI_HOME/bin/worklist
$WLI_HOME/setenv.sh
```

Windows Files

```
%WLI_HOME%\bin\worklist.cmd
%WLI_HOME%\setEnv.cmd
```

Environment The `worklist` command sets `CP=$WLI_HOME/lib/wlpi-worklist.jar:$WLICP` (UNIX) or `CP=%WLI_HOME%\lib\wlpi-worklist.jar:%WLICP%` (Windows), and references the following environment variables:

```
CP
JAVA_HOME
WLI_HOME
WLICP
```

If `WLI_HOME` is not set, this command executes the `setenv` command.

wliconfig

Name `wliconfig`—launch the WebLogic Integration Database Wizard

Synopsis

```
wliconfig
```

Description This command launches the WebLogic Integration Database Wizard. For additional information, see "Using the Database Wizard" on page 3-5.

Options No options are currently supported. The wizard prompts for the required information.

UNIX Files

```
DOMAIN_HOME/wliconfig
$WLI_HOME/setenv.sh
```

Windows Files

```
DOMAIN_HOME\wliconfig.cmd
%WLI_HOME%\setEnv.cmd
```

Environment The `wliconfig` command references the following environment variables:

```
$JAVA_HOME
$WLI_HOME
```

If `WLI_HOME` is not set, this command executes the `setenv` command.

Index

A

adapters, developing 2-12
admin login 1-9
administration server 1-2
alternate character sets 3-55
alternate security realm, configuring 3-46
application integration plug-in 2-16
Application View Console
    changing port number 3-60
    overview 1-7
    starting 2-9
application views 2-12

B

B2B Console
    changing the port number 3-61
    overview 1-7
    starting 2-24
B2B engine
    starting 2-26
    stopping 2-26
B2B integration plug-in 2-16
B2B sample applications 2-6
B2B URI endpoints, changing 3-59
BPM null variables 3-40
BPM password, changing 3-51
BPM plug-ins 2-16
BPM sample 2-6
browser, configuration requirements 2-2
business process management, security 3-43

C

channel master sample 2-6
classes,startup and shutdown A-11
clustering 1-2
commands
    RunSamples 1-14
    setDBVars 3-7
    setDBVarsExt 3-7
    setEnv 3-29
    setEnv sample A-14
    startWeblogic 1-17
    startWeblogic sample A-15
config.xml file A-2
configuring
    alternate security realm 3-46
    custom security realm 3-48
    database 3-5
    JMS queues 3-30
    RDBMS realm 3-47
    requirements 1-11
    samples domain 1-14
    security realm 3-43
    the repository 1-13
connection parameters 3-5
create database 3-5
custom security realm, configuring 3-48
customer support contact information ix
customizing
    JMS queues 3-30
    mail session settings 3-53

D

data integration
- plug-in 2-16
- samples 2-6

database
- access parameters 3-20
- changing 3-2
- connection parameters 3-5
- create 3-5
- initializing 1-13, 3-4
- migrate 3-5
- populating 1-13
- switch 3-5

database configuration wizard
- console mode 3-15
- graphical mode 3-8
- modes 3-8

Database Wizard 3-5

deploying
- B2B without persistence 3-62
- EJBs 3-34
- Java classes 3-34

documentation, where to find it viii

domains
- creating new 3-22
- preconfigured 1-3
- root directory 3-55
- WebLogic Server 1-2

E

EDI sample 2-6

EJBs
- deploying 3-34
- deployments, WLI application A-5

elements
- in the repository 1-9
- relationships 1-9

e-mail, customizing settings 3-53

environment, setEnv command 3-29

F

fileRealm.properties file
- description 3-43
- sample A-17

Format Builder
- overview 1-7
- starting 2-21

G

groups
- permission 3-45
- role 3-45
- system 3-45

H

hello partner sample 2-6

I

I18N support 3-55

initializing database repository 3-4

internationalization 3-55

J

Java classes, deploying 3-34

JDBC connection parameters 3-5

JDBC connection pool A-12

JMS JDBC store page A-9

JMS JDBC store prefix name A-8

JMS queue, customizing 3-30

L

loading system data 3-4

logins, default 1-9

M

mail session

customizing settings 3-53
properties 3-54
managed servers 1-2
MDBGenerator utility 3-30
message formats 2-21
messages, startup complete 1-20
messaging API sample 2-6
Microsoft SQL Server, required connection parameters 3-5
migrate database 3-5

N

null variable support 3-40

O

Oracle, required connection parameters 3-5

P

passwords
changing 3-50
changing BPM passwords 3-51
changing the system password 3-51
changing the wlcsystem password 3-52
default 1-9
permission groups 3-45
plug-ins
application integration plug-in 2-16
B2B integration plug-in 2-16
data integration plug-in 2-16
port numbers, changing
Application View Console 3-60
B2B console 3-61
B2B URI endpoints 3-59
WebLogic Integration 3-56
WebLogic Server listen 3-56
preconfigured domains 1-3
printing product documentation viii
proxy, bypassing 2-2

Q

queue, customizing 3-30

R

RDBMS realm, configuring 3-47
repository
about the 1-9
initializing 1-13
loading system data 3-4
specifying new 3-2
role groups 3-45
root directory, domain 3-55
RosettaNet security sample 2-6
RunSamples command 1-14

S

sample applications
running 2-4
summary 2-6
samples domain 1-3, 1-14
samples launcher page 2-4
samples server 1-14
security realm
alternate 3-46
configuring 3-43
guidelines 3-46
setDBVars command 3-7
setDBVarsExt command 3-7
setEnv sample A-14
shut down
B2B engine 2-26
changing scripts 3-61
WebLogic Integration 1-20
Shutdown classes A-11
starting
Application View Console 2-9
B2B Console 2-24
B2B engine 2-26
Format Builder 2-21

Studio 2-13
WebLogic Server Administration Console 2-7
Worklist 2-18
Startup classes A-11
startup complete messages 1-20
startWeblogic command 1-17
startWeblogic sample A-15
stopping
B2B engine 2-26
WebLogic Integration 1-20
stopWeblogic command 1-20
Studio
overview 1-7
plug-ins 2-16
starting 2-13
support, technical ix
switch database 3-5
Sybase, required connection parameters 3-5
system groups 3-45
system login 1-9
system password, changing 3-51

T

tables, WebLogic Integration repository 1-9

U

updating passwords 3-50
users 3-50

V

variables, support for null 3-40

W

Web browser, configuration requirements 2-2
WebLogic Integration
changing port numbers 3-56
changing shutdown scripts 3-61
Database Wizard 3-5
repository 1-11
sample application 1-22
starting 1-17
stopping 1-20
WebLogic Server
administration server 1-2
changing listen port number 3-56
domains 1-2
managed server 1-2
WebLogic Server Administration Console
starting 2-7
updating JDBC configuration from 3-17
updating RDBMS realm from 3-19
wlcsystem password, changing 3-52
WLI application, EJB deployments A-5
wli.bpm.server.evaluator.supportsNull option 3-40
wlpiCachingRealm, enabling 3-47
Worklist
overview 1-7
starting 2-18

X

XML repository 1-9

Z

zeroweight client sample 2-6