BEA WebLogic Integration™

Starting, Stopping, and Customizing BEA WebLogic Integration

Release 2.1
Document Date: October 2001
Revised: November 30, 2001

Copyright

Restricted Rights Legend

Trademarks or Service Marks

Starting, Stopping, and Customizing BEA WebLogic Integration

Part Number	Date	Software Version
880-001004-001	Release: October 2001 Revised: November 30, 2001	2.1

Contents

1. Getting Started

2. WebLogic Integration Administration and Design Tools

3. Customizing WebLogic Integration

A. WebLogic Integration Sample Configuration Files

B. WebLogic Integration Commands

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 that you are using the documentation for the BEA WebLogic Integration 2.1 release.

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 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. *Examples*: 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 is a single, unified platform that extends the core infrastructure and features of WebLogic Server to provide a comprehensive integration platform.

This section provides the information you need to get started with WebLogic Integration. It describes the WebLogic Server domains that are preconfigured to deploy WebLogic Integration resources, provides an overview of the components and passwords, and describes how to start WebLogic Integration in a preconfigured domain.

This section includes the following topics:

- WebLogic Server Domains
- WebLogic Integration Preconfigured Domains
- 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 structure:

`config/`*`domain_name`*`/config.xml`

The *`config`* directory can be located under the WebLogic Server installation directory or elsewhere on the system. *`domain_name`* is the name of the domain. 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 section describes the preconfigured domains that are installed with WebLogic Integration. These preconfigured domains can serve as your starting point for working with the WebLogic Integration samples provided, or for developing and testing your own WebLogic Integration solutions.

WebLogic Integration Preconfigured Domains

When you install WebLogic Integration, the following preconfigured domains are created and configured:

- *WebLogic Integration domain*
  This domain is configured to support the development of solutions that employ the full range of WebLogic Integration functionality. WebLogic Integration functionality associated with application integration, data integration, business process management (BPM), and B2B integration are supported in this domain.
- *Enterprise application integration domain*
  This domain is configured to support the development of enterprise application integration solutions. The WebLogic Integration functionality associated with BPM, application integration, and data integration are supported in this domain.
- *BPM domain*
  This domain is configured to support the development of business process management solutions. The WebLogic Integration functionality associated with BPM and data integration are supported in this domain.

These preconfigured domains support three of the most common WebLogic Integration development environments. Typically, one of these preconfigured domains will meet your evaluation, development, and testing needs.

If you selected the WebLogic Integration Full Installation with Samples option during installation, a samples domain is also created to run the WebLogic Integration samples. All WebLogic Integration functionality is supported in this domain. In addition, Web applications, custom JMS queues, and other resources required to run the samples are configured.

Each of the preconfigured and sample domains share the following characteristics:

- By default, each 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 JDBC connection pool required to connect to your database is configured based on the database information you provided during installation.
- A default RDBMS security realm is configured to use the same database.

As you become more familiar with WebLogic Integration, you can use a preconfigured domain as the starting point for development and testing, or you can create your own domain by modeling it after one of the preconfigured domains.

Each of the domains described in this section is located under the WebLogic Integration installation directory. Table 1-1 shows the location of each domain when the product is installed in the default location.

Table 1-1 WebLogic Integration Preconfigured Domains

Domain	Location
WebLogic Integration	*BEA_HOME*/wlintegration2.1/config/wlidomain
Enterprise Application Integration	*BEA_HOME*/wlintegration2.1/config/eaidomain
Business Process Management	*BEA_HOME*/wlintegration2.1/config/bpmdomain
Samples	*BEA_HOME*/wlintegration2.1/config/samples This domain is installed only if you select the WebLogic Integration Full Installation with Samples option during installation.

A copy of each of the following files is provided in each preconfigured domain directory. These files control the initial deployment and configuration of the 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 controlled by the `config.xml` file.
- Start command—Command that starts the administration 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 administration 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. The file realm is the default security realm in a WebLogic Server environment.
- `wlai.properties`—File that sets properties required to support application integration functionality. This file is located in the `wlai` subdirectory of any domain that supports application integration. For example, in the WebLogic Integration domain the file is `config/wlidomain/wlai/wlai.properties`.

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."

When you start WebLogic Server by executing the `startWebLogic` command for one of the preconfigured domains (as described in "Starting WebLogic Integration" on page 1-17), the WebLogic Integration resources configured for the domain are deployed.

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 ADK (Adapter Development Kit) 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.

- *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 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 1-2 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 1-10.

WebLogic Integration Users and Passwords

The following users and passwords are configured when you install the product.

Table 1-3 Default User Names and Passwords

User Name	Default Password	Description	Update Considerations
system	security	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.
wlcsystem	wlcsystem	Internal login used by the B2B Engine.	Update this password only from the B2B Console. **Warning:** Do not update this password using the WebLogic Server Administration Console.
wlpisystem	wlpisystem	Internal login used by the process engine.	**Warning:** Do not update this password.
admin	security	Default BPM logins are used to log on and connect to the process engine or 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 ensure system security, you can change the default passwords, as described in "Updating Passwords" on page 3-40.

When you install WebLogic Integration, 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 3-33.

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

Domain Configuration Requirements

Before you can start WebLogic Integration in a preconfigured domain, you must configure the database repository for the domain appropriately. When you install WebLogic Integration, you must specify the access information required to connect to the following two databases:

- *Samples database*
  The access information provided during installation is used to configure the JDBC connection pool and the RDBMSRealm security realm for the samples domain. The JDBC and RDBMS configuration for the specified database is defined in the `config.xml` file for the domain.
- *Production database*
  The access information provided during installation is used to configure the JDBC connection pool and the RDBMSRealm for the WebLogic Integration domain (`config\wlidomain`), enterprise application integration domain (`config\eaidomain`), and BPM domain (`config\bpmdomain`). The JDBC and RDBMS configuration for the specified database is defined in the `config.xml` file for each of these domains.

Typically, you will be using only two of the domains described here: the samples domain and the preconfigured production domain that most closely meets your development and testing requirements. In such cases you need to configure only two domains.

If you plan to use more than one of the preconfigured production domains, you may want to specify a different database for the additional production domain you will use. Although sharing a database may not be problematic, we recommend that you keep the data for different domains segregated in discrete databases.

For example, if you plan to use the WebLogic Integration domain as your primary development and testing domain, and the BPM domain as your secondary development and testing domain, you may want the BPM domain to use a database other than the production database specified during installation.

The WebLogic Integration database configuration wizard automates the process of updating the configuration for a domain. The procedure for updating to a new database is described in "Specifying a New Database for a Domain" on page 3-2.

Before you can start WebLogic Integration in a given domain, you must initialize the database for the domain. Some database configuration tasks are required by all domains; others are domain-specific. The following table summarizes the requirements for all domains.

Table 1-4 Database Configuration Requirements

Task ID	Database Configuration 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
	■ JMS Queue tables—Set of database tables that store the message state (dynamic)
		All preconfigured domains:
		`config\samples`
		`config\wlidomain`
		`config\eaidomain`
		`config\bpmdomain`

2	Populate the common tables with the required data (for example XML schemas).	All preconfigured domains:
		`config\samples`
		`config\wlidomain`
		`config\eaidomain`
		`config\bpmdomain`

Table 1-4 Database Configuration Requirements

Task ID	Database Configuration Task	Required for
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)	`config\samples` `config\wlidomain`
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	`config\samples` `config\wlidomain`
5	Load the configuration data and workflows required by the samples.	`config\samples`
6	Configure the application view for the End-to-End sample application.	`config\samples`

Configuring the Database for a Domain

The WebLogic Integration database configuration wizard is a tool that enables you to automate database configuration tasks. The 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 the JDBC connection pool and RDBMSRealm configurations in the `config.xml` file for the domain. It also updates the domain-specific environment settings used by the wizard to create the database.
- *Create Database*
  This option creates the tables required by the domain and loads data appropriate to the domain. For example, for the WebLogic Integration domain, the command performs Tasks 1, 2, 3, and 4 (described in Table 1-4). For the BPM domain, this option performs Tasks 1 and 2. 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 command can be used to both switch and initialize the database for a domain.

  Note: Although the WebLogic Integration database configuration wizard is available for the samples domain, the Create Database option does not complete tasks 5 and 6 for the samples domain. In order to use the samples domain, you must set up and start the server in the samples domain by executing the `RunSamples` command. See the following section, "Configuring and Starting the Samples Domain" on page 1-14.
- *Migrate Database*
  This option updates a selected WebLogic Integration 2.0 database, as required, to work with WebLogic Integration 2.1. For information about migrating from WebLogic Integration 2.0 or a previous release, see *Migrating to BEA WebLogic Integration Release 2.1*.

Before starting WebLogic Integration in the WebLogic Integration domain, enterprise application integration domain, or BPM domain, as described in "Starting WebLogic Integration" on page 1-17, you must use the WebLogic Integration database configuration wizard to create and initialize the required database. For instructions, see "Using the Database Configuration Wizard" on page 3-6.

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 switched the samples domain to a new database, the command performs the following tasks:

- Creates the required repository tables in the database. See Table 1-4.
- 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 E-Business Platform→WebLogic Integration 2.1→Samples→Run Samples.

- On UNIX:

  a. Go to the `samples/bin` directory (under the WebLogic Integration installation directory).

  b. Execute the `RunSamples` command.

     Example: If WebLogic Integration is installed in the default location, enter the following commands:

     ```
     cd BEA_HOME/wlintegration2.1/samples/bin
     . ./RunSamples
     ```

When the `RunSamples` command is executed on a Windows system, a command window is opened. On both Windows and UNIX platforms, command progress and messages are displayed in a command window. 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 the 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 Preconfigured Domains" 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 domain (`config\wlidomain`), choose Start→Programs→BEA WebLogic E-Business Platform→WebLogic Integration 2.1→Start Server.
- To start the server in the enterprise application integration domain (`config\eaidomain`), choose Start→Programs→BEA WebLogic E-Business Platform→WebLogic Integration 2.1→Additional Preconfigured Domain→EAI Domain→Start Server.
- To start the server in the BPM domain (`config\bpmdomain`), choose Start→Programs→BEA WebLogic E-Business Platform→WebLogic Integration 2.1→Additional Preconfigured Domain→BPM Domain→Start Server.

- To start the server in the samples domain (`config\samples`), choose Start→Programs→BEA WebLogic E-Business Platform→WebLogic Integration 2.1→Samples→Start Server.

  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.

A command window is launched, and startup messages are displayed. When the messages described in "Startup Complete Messages" on page 1-19 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. In the command window, go to the directory for the appropriate domain.

   For example, if WebLogic Integration is installed in the default location, and you want to start the WebLogic Integration domain, enter the following:

   ```
   cd \bea\wlintegration2.1\config\wlidomain
   ```

   For the default location of each preconfigured domain, see "WebLogic Integration Preconfigured Domains" on page 1-3.
4. Start the server by entering:

   ```
   startWeblogic
   ```

When the messages described in "Startup Complete Messages" on page 1-19 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. Go the directory for the appropriate domain.

   For example, if WebLogic Integration is installed in the default location, and you want to start the WebLogic Integration domain, enter the following:

   ```
   cd BEA_HOME/wlintegration2.1/config/wlidomain
   ```

   For the default location of each preconfigured domain, see "WebLogic Integration Preconfigured Domains" on page 1-3.

2. Start the server by entering:

   ```
   . ./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>
<WebLogicServer> <ListenThread listening on port 7001>
<Oct 1, 2001 10:04:17 AM EDT> <Notice> <WebLogicServer>
<Started WebLogic Admin Server "myserver" for domain "domain_name"
running in Development Mode>Initialized WebLogic Integration Plugin
Framework version 1.0
Started WebLogic Integration - BPM Server version 2.1
```

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 1-20. 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

The following sections provide instructions for stopping WebLogic Integration through menus or from the command line.

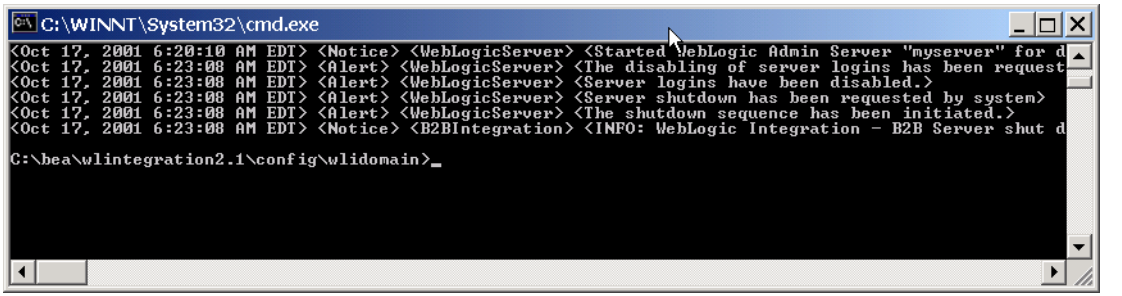
Stopping WebLogic Integration Through Menus

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

- To stop the server in a WebLogic Integration domain (`config\wlidomain`), choose Start→Programs→BEA WebLogic E-Business Platform→WebLogic Integration 2.1→Stop Server.
- To stop the server in an enterprise application integration domain (`config\eaidomain`), choose Start→Programs→BEA WebLogic E-Business Platform→WebLogic Integration 2.1→Additional Preconfigured Domain→EAI Domain→Stop Server.
- To stop the server in a BPM domain (`config\bpmdomain`), choose Start→Programs→BEA WebLogic E-Business Platform→WebLogic Integration 2.1→Additional Preconfigured Domain→BPM Domain→Stop Server.
- To stop the server in a samples domain (`config\samples`), choose Start→Programs→BEA WebLogic E-Business Platform→WebLogic Integration 2.1→Samples→Stop Server.

A command window opens briefly. It displays a message indicating that a shutdown request has been sent to the server. Shutdown progress messages are then displayed in the original command window, as shown in the following figure. When you are returned to the command prompt, shutdown is complete.

Figure 1-1 Shutdown Sequence Messages



Once WebLogic Server and all applications are terminated, you can restart WebLogic Integration, as described in "Starting WebLogic Integration on Windows" on page 1-17.

Stopping WebLogic Integration from the Command Line

To stop WebLogic Integration from the command line:

1. Choose Start→Run.
2. Enter `cmd` in the Open text box, and then click OK.
3. In the command window, go to the directory for the appropriate domain.

   For example, if WebLogic Integration is installed in the default location, and you want to start the WebLogic Integration domain, enter the following:

   ```
   cd \bea\wlintegration2.1\config\wlidomain
   ```

   For the default location for each preconfigured domain, see "WebLogic Integration Preconfigured Domains" on page 1-3.
4. Stop WebLogic Integration by entering:

   ```
   stopWeblogic
   ```

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

Stopping WebLogic Integration on UNIX

To stop WebLogic Integration on a UNIX system:

1. Go the directory for the appropriate domain.

   For example, if WebLogic Integration is installed in the default location, and you want to stop the WebLogic Integration domain, enter the following:

   ```
   cd BEA_HOME/wlintegration2.1/collaborate/config/wlidomain
   ```

   For the default location of each preconfigured domain, see "WebLogic Integration Preconfigured Domains" on page 1-3.

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 system 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 Option 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 as 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:

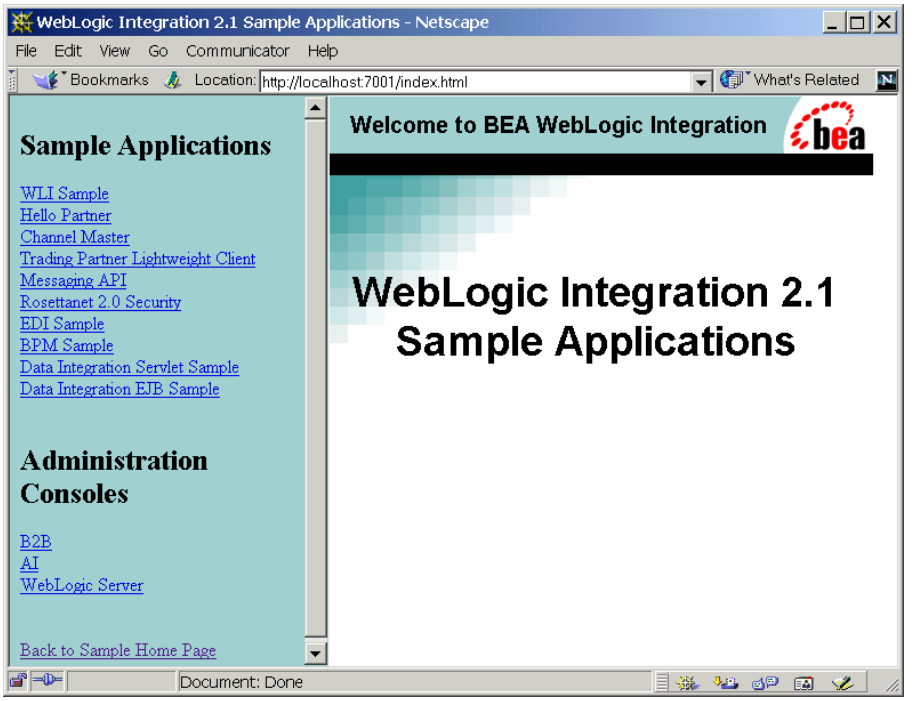
     `http://`*`host`*`:7001/index.html`

     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 domain. 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 E-Business Platform→WebLogic Integration 2.1→Samples→ Samples Launcher.

     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

Information about the sample applications is provided in the following documents:

For information about this sample . . .	See . . .
WebLogic Integration (WLI)	*Learning to Use BEA WebLogic Integration*
Business process management (BPM)	*Learning to Use BPM with WebLogic Integration*
B2B samples: ■ Hello Partner ■ Channel Master ■ Trading Partner Lightweight Client ■ Messaging API ■ RosettaNet 2.0 Security	*Running the B2B Integration Samples*
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:7001/console`

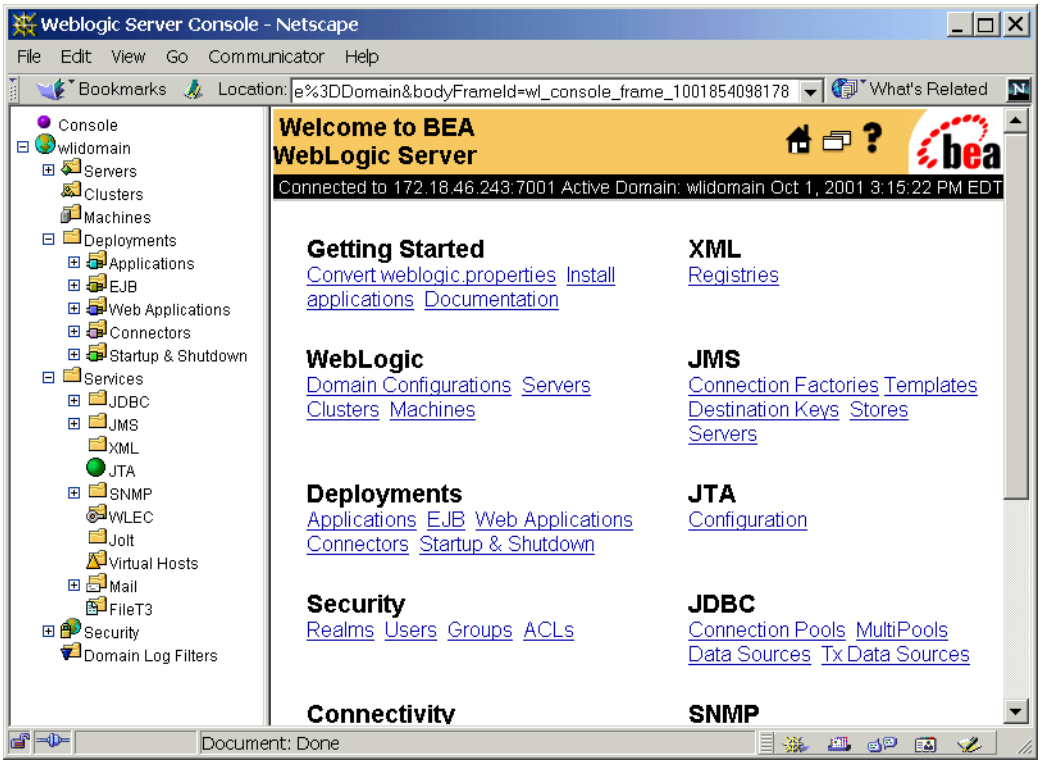
     In this URL, *host* is the computer name or IP address of the system that is running the administration server for the domain, and `7001` is the listen port configured for the domain.The server may be running on any system. If it is running on the local computer, specify `localhost` or `127.0.0.1`.
   - On a Windows system, if the administration server is running on the local computer, choose Start→Programs→BEA WebLogic E-Business Platform→WebLogic Server 6.1→Start Default Console.

     This shortcut starts the console for the administration server running on the local system.
3. When prompted, enter `system` in the user name field; then enter the system password.

   Note: The password you enter is the WebLogic Integration system password for the active domain. This password is not the default password you specified when installing WebLogic Server. If you have not yet changed the password, see "WebLogic Integration Users and Passwords" on page 1-9 for the default 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 6.1 documentation at the following URL:

`http://e-docs.bea.com/wls/docs61/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 not available in the BPM domain (`config\bpmdomain`).

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:7001/wlai`

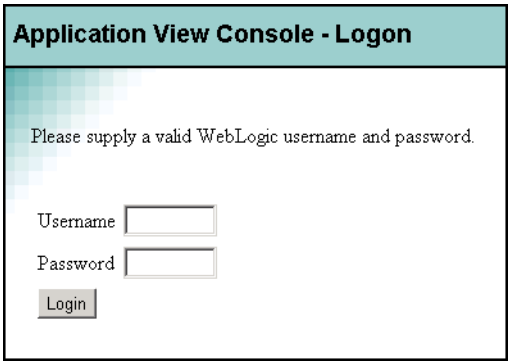
     In this URL, *`host`* is the computer name or IP address of the system that is running WebLogic Integration, and `7001` is the listen port configured for the domain. WebLogic Integration may be running on any machine. If it is running on the local computer, specify `localhost` or `127.0.0.1`.

   - On a Windows system, if WebLogic Integration is running on the local computer, choose Start→Programs→BEA WebLogic E-Business Platform→WebLogic Integration 2.1→Application View Console.

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

3. Enter `system` in the Username field and specify the system password in the password field. Click Login.

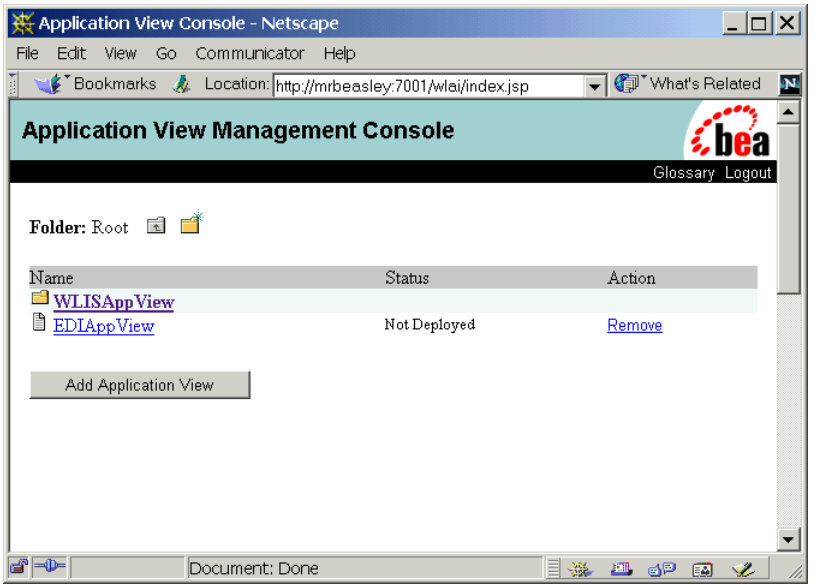
Figure 2-3 Application View Logon



Note: The password you enter is the WebLogic Integration system password for the active domain. If you have not yet changed the password, see "WebLogic Integration Users and Passwords" on page 1-9 for the default password.

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

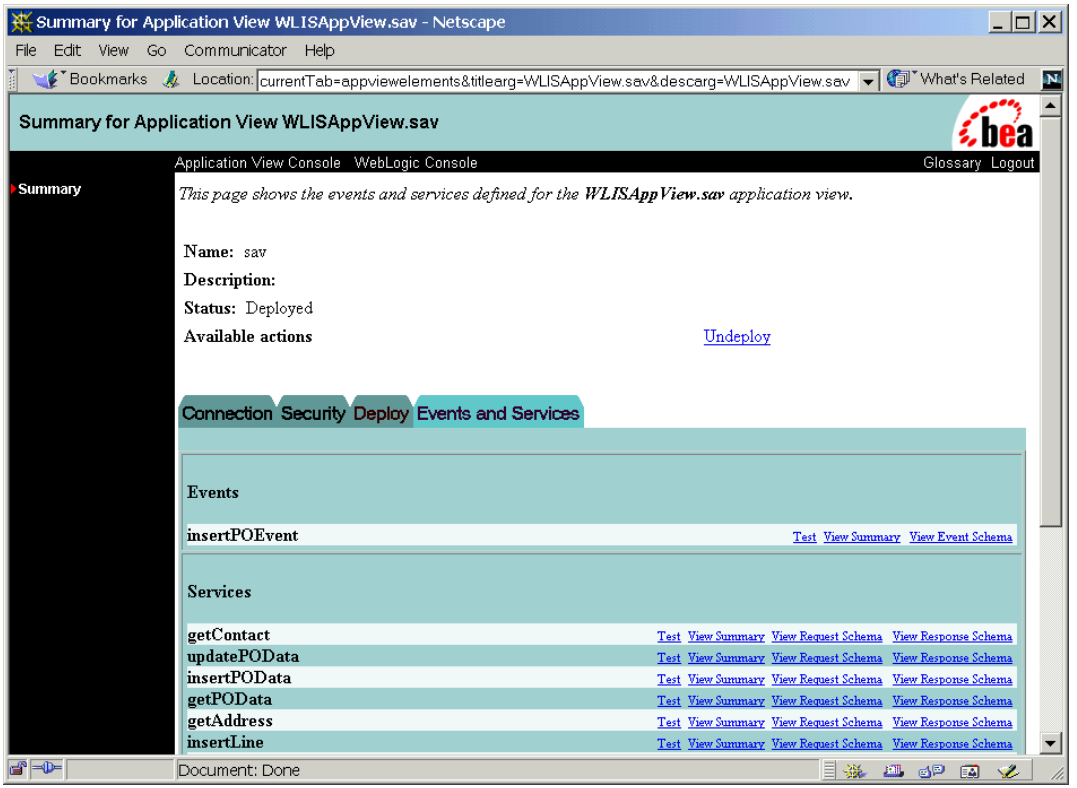
Figure 2-4 Application View Console Page for the Samples Domain



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 using 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

Note: If you installed the client applications on a system that is remote from the server, you must copy two files, `weblogic.jar` and `xmlx.jar`, from the `lib` directory of the WebLogic Server installation (for example: `C:\bea\wlserver6.1\lib`) to the `lib` directory of the client installation (for example: `C:\bea\wlintegration2.1\lib`).

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 E-Business Platform→WebLogic Integration 2.1→Studio.

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

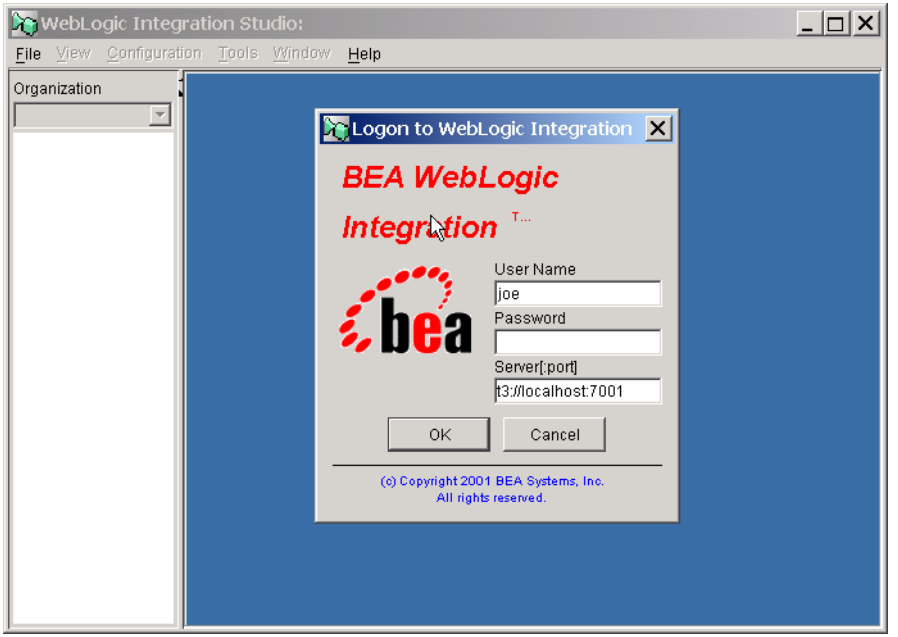
- UNIX:

  Go to the WebLogic Integration `bin` directory and run the `studio` command. For example, if WebLogic Integration is installed in the default location, enter the following:

  ```
  cd BEA_HOME/wlintegration2.1/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:

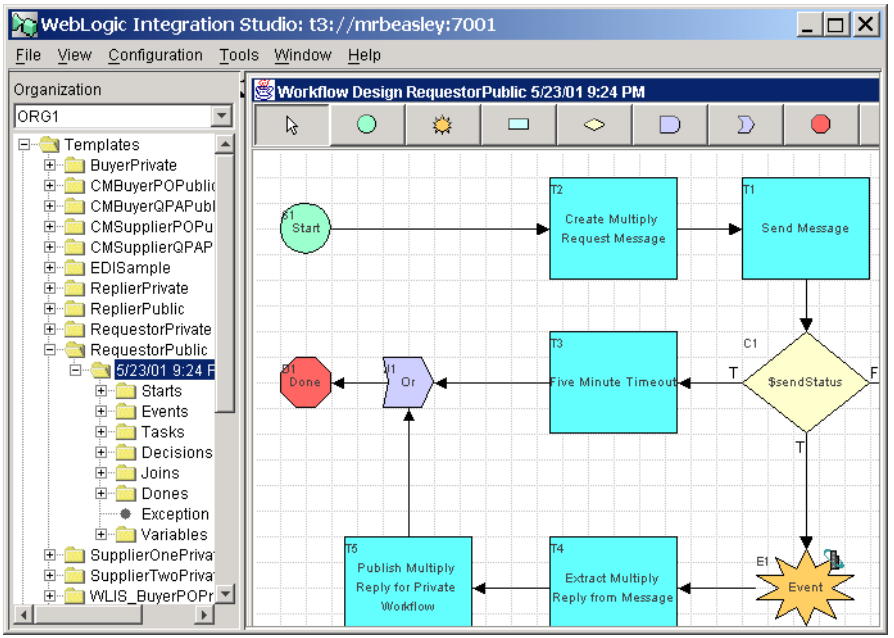
   `t3://host:7001`

   Here, *`host`* is the computer name or IP address of the system. 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. The plug-ins deployed are dependent on the configuration of the active domain. The following table summarizes the deployment of plug-ins in the preconfigured domains.

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

This plug-in . . .	Extends Studio functionality so you can . . .	It is deployed in . . .

Data integration plug-in	Perform data translation using message formats developed in the Format Builder.	All preconfigured domains:
		`config\samples`
		`config\wlidomain`
		`config\eaidomain`
		`config\bpmdomain`

Application integration plug-in	Use application view services and events in a workflow.	`config\samples`
		`config\wlidomain`
		`config\eaidomain`

B2B integration plug-in	Create workflows for use in trading partner exchanges.	`config\samples` `config\wlidomain`

To view a list of the plug-ins currently deployed, choose Start→Programs→BEA WebLogic E-Business Platform.

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 Business Process Management" 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 user task notification and response for workflow tasks that require human intervention. The following procedure describes how to start the Worklist.

Note: If you installed the client applications on a system that is remote from the server, you must copy two files, `weblogic.jar` and `xmlx.jar`, from the `lib` directory of the WebLogic Server installation (for example: `C:\bea\wlserver6.1\lib`) to the `lib` directory of the client installation (for example: `C:\bea\wlintegration2.1\lib`).

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 E-Business Platform→WebLogic Integration 2.1→Worklist.

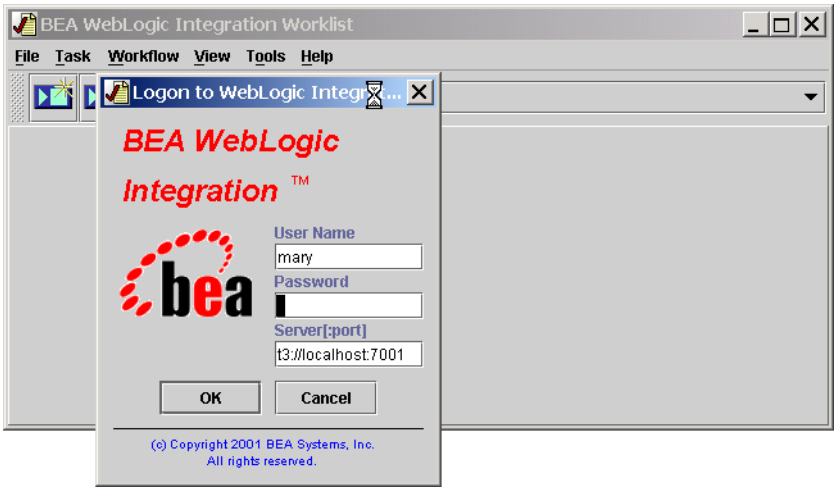
     The command window is displayed briefly as the `worklist.cmd` command is executed. Within a few moments, the Worklist application window is displayed.
   - UNIX:

     Go to the WebLogic Integration `bin` directory and run the `worklist` command. For example, if WebLogic Integration is installed in the default location, enter the following:

     ```
     cd BEA_HOME/wlintegration2.1/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:

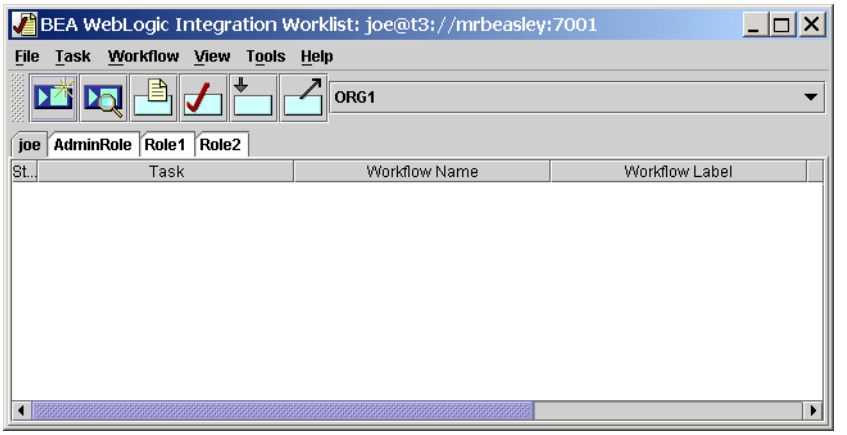
   `t3://`*`host`*`:7001`

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

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 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 actions into your workflows that send or receive data from these systems. The following procedure describes how to start the Format Builder.

Note: If you installed the client applications on a system that is remote from the server, you must copy two files, `weblogic.jar` and `xmlx.jar`, from the `lib` directory of the WebLogic Server installation (for example: `C:\bea\wlserver6.1\lib`) to the `lib` directory of the client installation (for example: `C:\bea\wlintegration2.1\lib`).

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 E-Business Platform→WebLogic Integration 2.1→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:

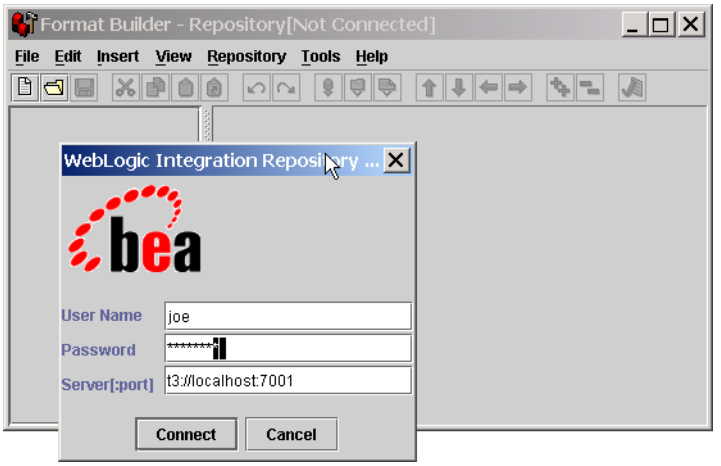
     Go to the WebLogic Integration `bin` directory and run the `fb` command. For example, if WebLogic Integration is installed in the default location, enter the following:

     ```
     cd BEA_HOME/wlintegration2.1/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:

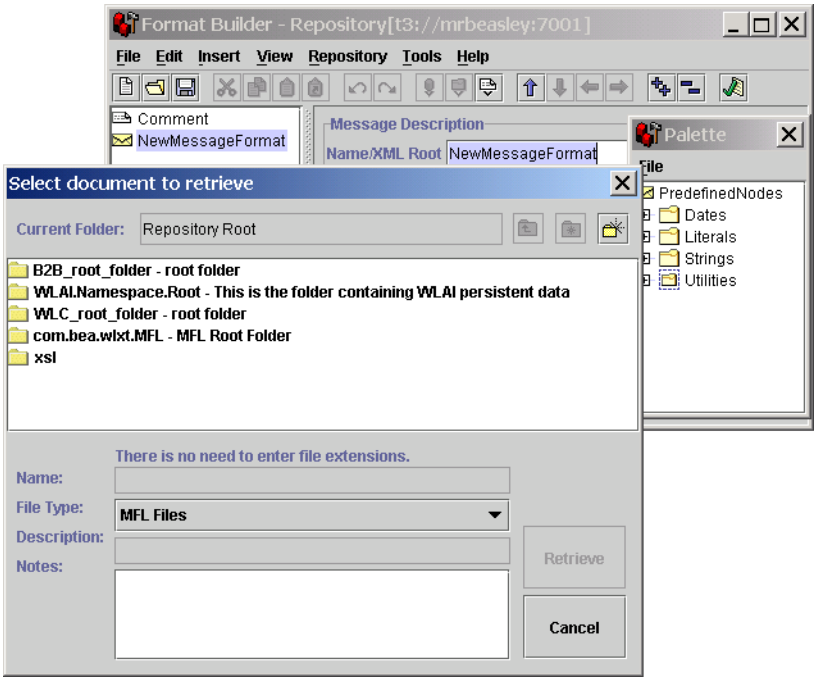
   `t3://`*`host`*`:7001`

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

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 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 and control the B2B engine, trading partner sessions, conversations, and collaboration agreements

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:7001/b2bconsole`

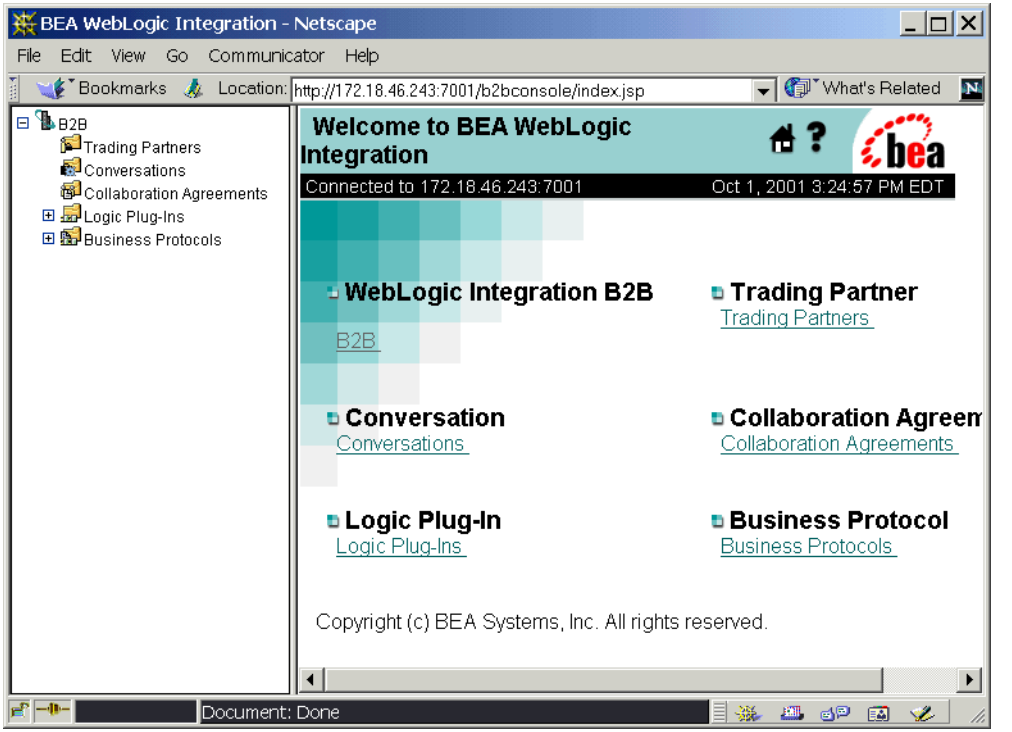
     In this URL, *host* is the computer name or IP address of the system that is running WebLogic Integration, and `7001` is the listen port configured for the domain. WebLogic Integration may be running on any system. If it is running on the local computer, specify `localhost` or `127.0.0.1`.
   - On a Windows system, if WebLogic Integration is running on the local computer, choose Start→Programs→BEA WebLogic E-Business Platform→WebLogic Integration 2.1→B2B Console.

     This shortcut starts the console for the server running on the local system.
3. When prompted, enter `system` in the user name field; then enter the system password.

   Note: The password you enter is the WebLogic Integration system password for the active domain. If you have not yet changed the password, see "WebLogic Integration Users and Passwords" on page 1-9 for the default 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



Note: Bookmark the B2B Console for quick access.

Note: Which items are displayed in the navigation tree (in the left pane) is determined by the B2B preference settings. For example, if the Hide advanced configuration controls option is selected, 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.

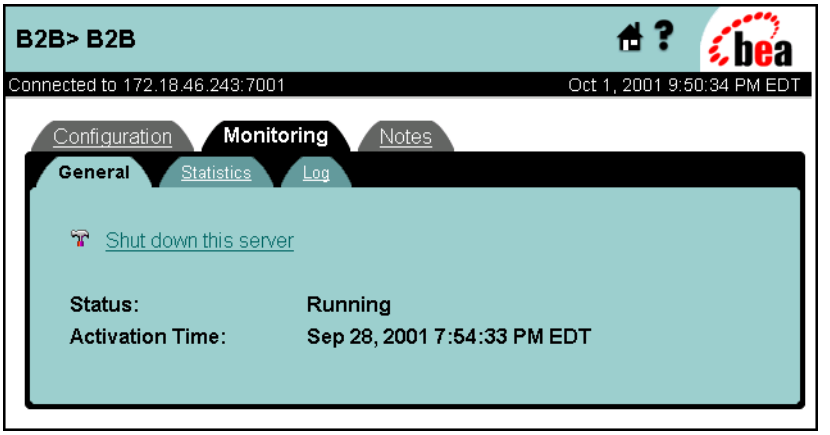
When you use the B2B Console to shut down or restart the B2B engine, only the B2B engine is affected; all other WebLogic Integration applications, including the Web application that supports the B2B Console itself, continue to run.

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.

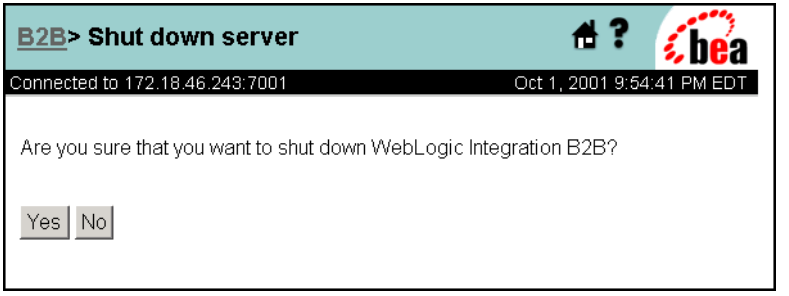
Figure 2-13 General Tab for B2B Monitoring



3. Select Shut down this server.

   The shutdown confirmation page is displayed.

Figure 2-14 Shutdown Confirmation Page



4. Click Yes to confirm that you want to shut down the B2B engine.

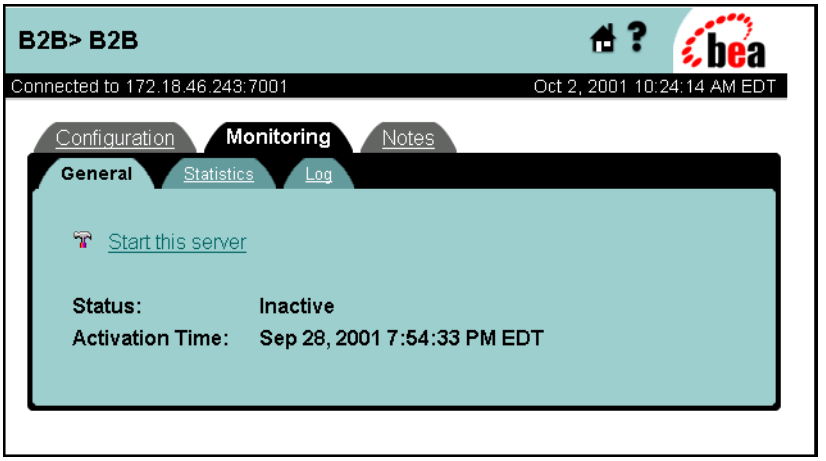
   The B2B engine is shut down. WebLogic Server and all other WebLogic Integration applications and resources remain active. When you have shut down the B2B engine from the B2B Console, you can restart it, as described in the following procedure.

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.

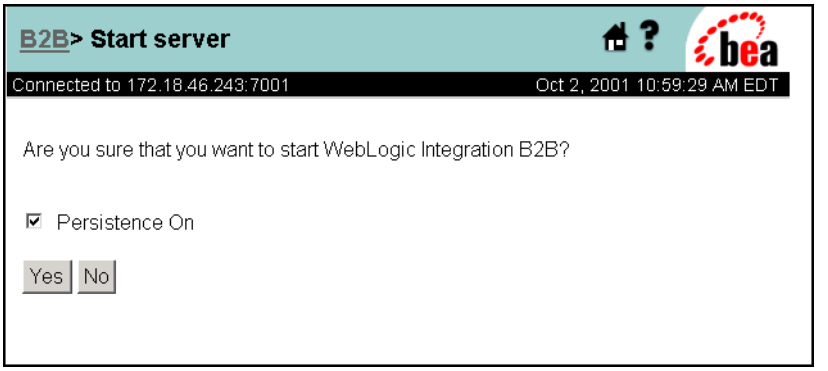
Figure 2-15 General Tab for B2B Monitoring



2. Select Start this server.

   The startup confirmation page is displayed. This page allows you to turn persistence on or off, as shown in the following figure.

Figure 2-16 Start Up Confirmation Page



3. Do one of the following:
   - Click Yes to start with persistence on.
   - Deselect Persistence On, and then Click Yes to start with persistence off.

   For information about persistence in B2B integration, see "Configuring Persistence and Recovery" in *Administering B2B Integration*.

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

3 Customizing WebLogic Integration

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

- WebLogic Integration Commands and Supporting Files
- Initializing the Database for a Domain
- Using the Database Configuration Wizard
- Updating the Database Configuration from the WebLogic Server Administration Console
- 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
- Changing the Root Directory for a Domain
- Using an Alternate Character Set

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 WebLogic Integration database configuration wizard, and the `RunSamples` script described in "Domain Configuration Requirements" on page 1-10. Although such commands are not normally invoked in isolation, there are situations where 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 has been provided in Appendix B, "WebLogic Integration Commands."

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

Specifying a New Database for a Domain

The method you use to update the database for a domain depends on:

- Whether the database you are replacing is for the samples domain, or for a preconfigured production domain
- 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 . . .
The samples domain	Do not want to preserve repository data in the existing database	1. Start the WebLogic Integration database configuration 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.
The samples domain	Want to preserve repository data in the existing database	1. Export the information required. 2. Start the WebLogic Integration database configuration 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. See **Note**
A preconfigured production domain	Do not want to preserve repository data in the existing database	1. Start the WebLogic Integration database configuration wizard for the domain. 2. Select the Create Database option. This option prompts for the new database access information, and initializes the database.
A preconfigured production domain	Want to preserve repository data in the existing database	1. Export the information required. 2. Start the WebLogic Integration database configuration 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**

Table 3-1 Database Update Methods (Continued)

If you are updating the database for . . .	And you . . .	Then . . .
Note You also have the option of preserving the entire installation. To do so, install WebLogic Server and WebLogic Integration in a new BEA Home directory. When prompted during installation, specify the information required to connect to the new database. You can then export workflow packages or B2B configuration elements, as required, for import to the new installation. When it is no longer needed, you can uninstall the obsolete installation		

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 WebLogic Integration database configuration wizard	"Using the Database Configuration Wizard" on page 3-6
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

The method you use to initialize the database for a domain depends on whether the database is for the samples domain, or for a preconfigured production domain:

- 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 one of the preconfigured production databases, use the WebLogic Integration database configuration wizard, as described in the following section. Select the Create Database option to initialize the database.

For additional information about the tasks performed by the `RunSamples` command and the WebLogic Integration database configuration wizard, see "Domain Configuration Requirements" on page 1-10.

Using the Database Configuration Wizard

As described in "Configuring the Database for a Domain" on page 1-13, the WebLogic Integration database configuration wizard is provided to automate database configuration tasks. The wizard operates on the domain specified when you execute the `wliconfig.exe` command. On Windows systems, domain-specific shortcuts are provided.

The following sections provide the information you need to use the wizard to initialize the database for a domain, specify a new database for a domain, or migrate a WebLogic Integration 2.0 database to WebLogic Integration 2.1:

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

Database Connection Information

The following table summarizes the information required to connect to each supported database type. The WebLogic Integration database configuration 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
Cloudscape	None	*Cloudscape is available for evaluation and testing on Windows systems.* A cloudscape database is created for the domain in the `config\`*`domain_name`*`\dbInfo\cloudscape\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 Configuration Options

The WebLogic Integration database configuration 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.

- *Migrate Database*
  Select this option to update a specified WebLogic Integration 2.0 database as required for use with WebLogic Integration 2.1. For information about migrating from WebLogic Integration 2.0, or a previous release, see *Migrating to BEA WebLogic Integration Release 2.1*.

How the Database Configuration Wizard Works

The WebLogic Integration database configuration 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 configuration 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 Wizard in Graphical Mode
- Using the Wizard in Console Mode

Using the Wizard in Graphical Mode

The following procedure describes how to update your database configuration using the WebLogic Integration database configuration 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 the . . .	**Choose . . .**
Samples domain (`config\samples`)	Start→Programs→BEA WebLogic E-Business Platform→WebLogic Integration 2.1→Samples→Configure
WebLogic Integration domain (`config\wlidomain`)	Start→Programs→BEA WebLogic E-Business Platform→WebLogic Integration 2.1→Configure
Enterprise application integration domain (`config\eaidomain`)	Start→Programs→BEA WebLogic E-Business Platform→WebLogic Integration 2.1→Additional Preconfigured Domain→EAI Domain→Configure
Business process management (BPM) domain (`config\bpmdomain`)	Start→Programs→BEA WebLogic E-Business Platform→WebLogic Integration 2.1→Additional Preconfigured Domain→BPM Domain→Configure

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

2. On a UNIX platform, do the following:

   a. Execute the following commands:

      ```
      cd WLI_HOME/bin
      wliconfig
      ```

      The Choose BEA Home Directory dialog box is displayed.

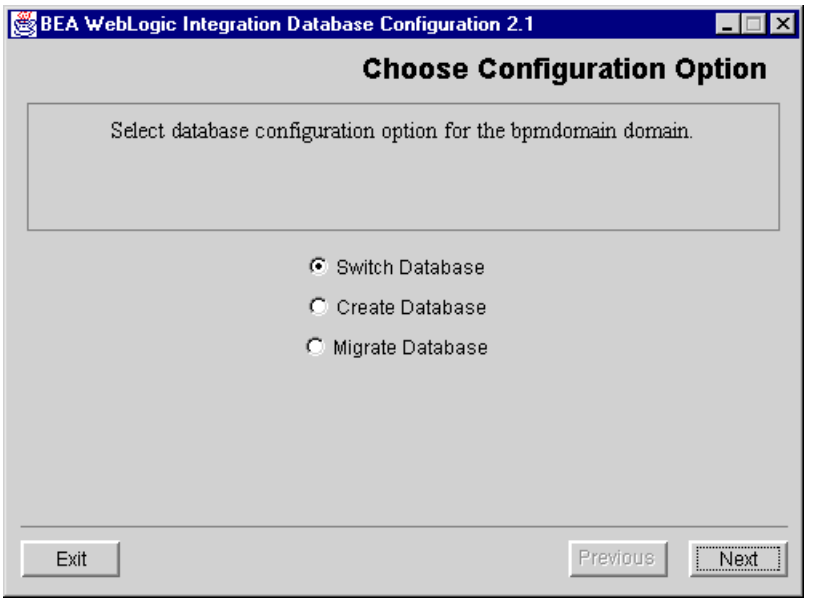
   b. Select an existing BEA Home directory, and then click Next.

      The Choose Domain to Configure dialog box is displayed.

c. Select a domain, and then click Next.

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

Figure 3-1 Choose Configuration Option Dialog Box

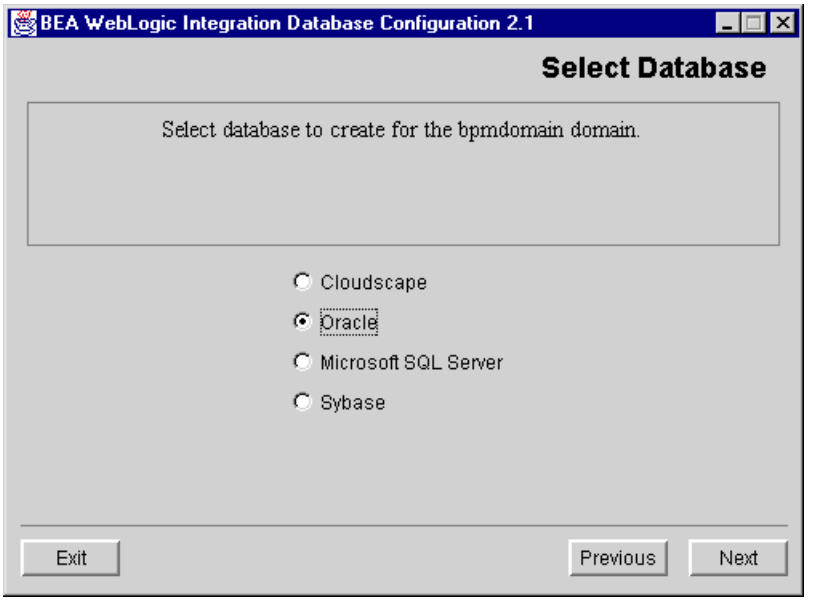


Note: If you are on a UNIX system, and are connecting to a Microft SQL Server database, you will be unable use the wizard to initialize the database because there is no Microsoft SQL client for UNIX. Instead, you will need to initialize the database from a Windows system.

3. Select the appropriate option. See "Database Configuration Options" on page 3-8.
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 Oracle is selected for the domain during installation, that option is selected now, as shown in the following figure.

Figure 3-2 Database Selection Dialog Box

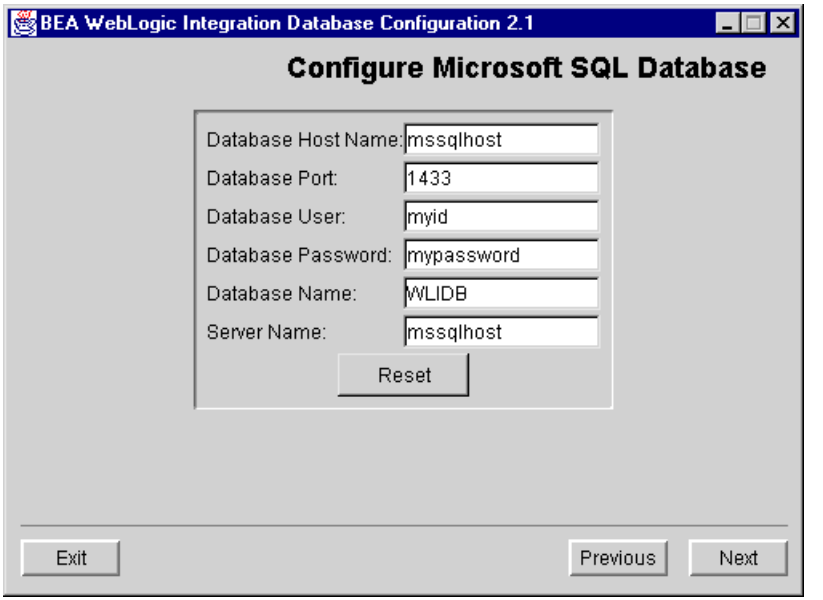


Note: On UNIX systems, Cloudscape is not an option.

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 results are as follows:
   - If you selected Cloudscape, 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 Oracle, Microsoft SQL Server, or Sybase, the Database Configuration dialog box is displayed, as shown in the following figure. This dialog box reflects information that was provided when you installed the product, or when you last used the wizard to specify a new database.

Figure 3-3 Database Configuration Dialog Box



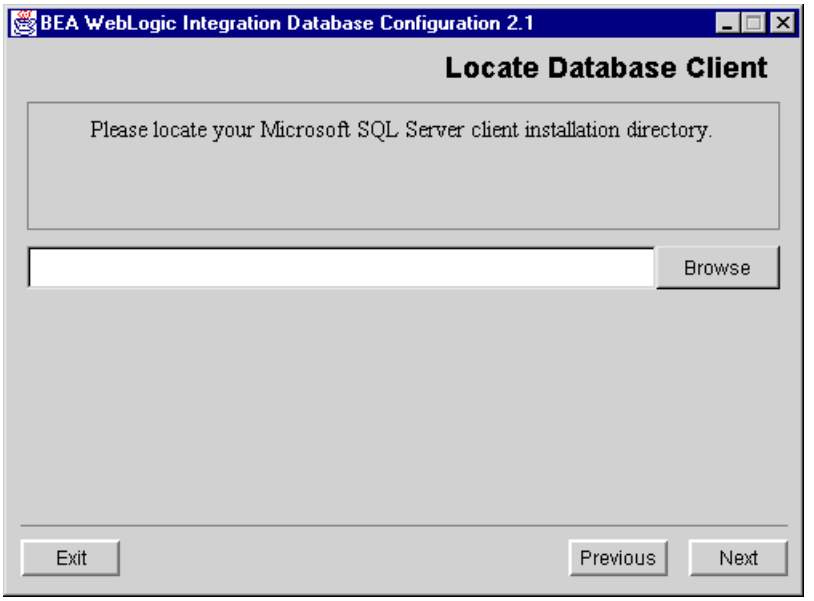
6. Do one of the following:

   - 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 3-6.

   The results are as follows

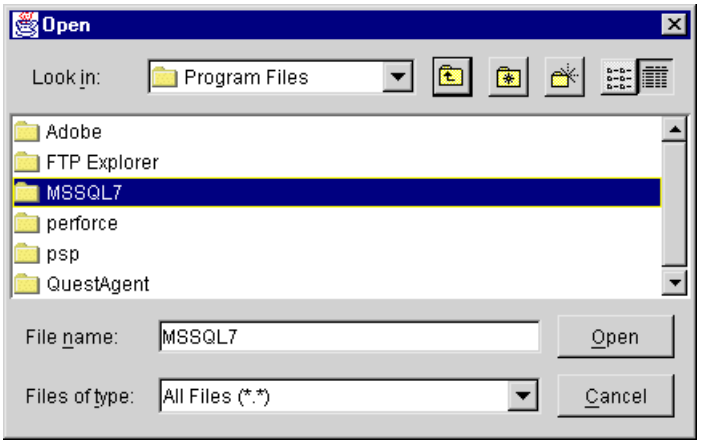
   - If you selected Switch Database, the selected action proceeds. When the changes are complete, the Changes Successful dialog box is displayed, as shown in Figure 3-7. Go to step 12.
   - If you selected Create Database or Migrate Database, you are prompted to locate the client installation directory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

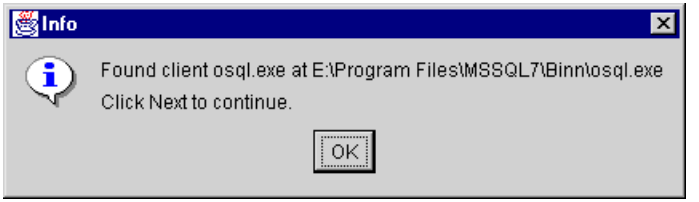
Figure 3-5 Open Dialog Box



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



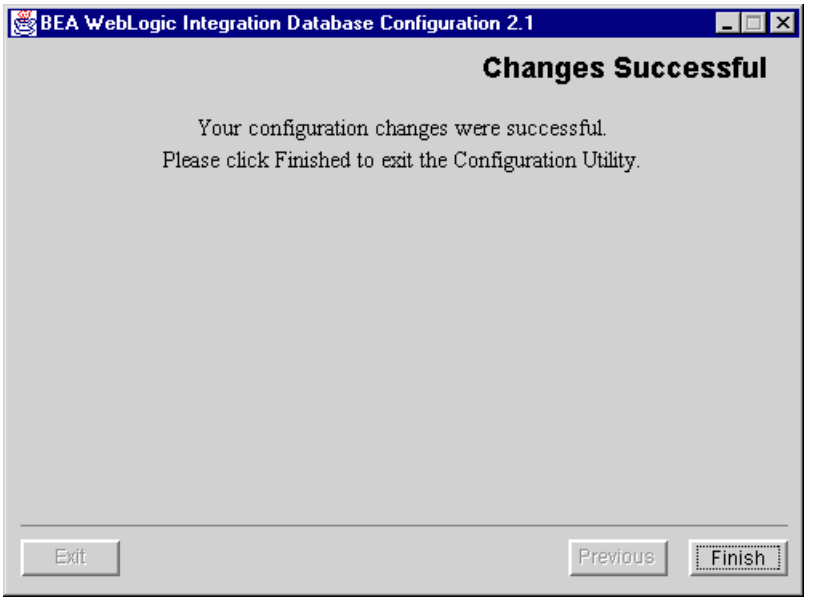
Note: If an error message is displayed indicating that the client application was not found, click OK to dismiss the message, and then Click Browse to locate the correct directory.

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 is 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 will indicate 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 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 configuration wizard in console mode, enter the following commands at the prompt:

```
cd WLI_HOME/bin
wliconfig
```

The following listing shows the console-mode prompts and responses for the Create Database option described in "Database Configuration Options" on page 3-8. (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 Configuration Procedure in Console Mode

```
================================================================================
BEA WebLogic Integration Database Configuration 1.0.0
--------------------------------------------------------------------------------


================================================================================
Choose BEA Home Directory
-------------------------------------------------

        ->1- C:\bea
          2- C:\bea

Existing BEA Home(1): 1

================================================================================
Choose Domain
-------------------------------------------------

          1- bpmdomain
          2- eaidomain
          3- samples
          4- wlidomain

Please Select the Domain to Configure: 2

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

          1- Switch Database
          2- Create Database
          3- Migrate 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 configuration wizard should be used to update the database configuration for any of the preconfigured domains, 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 `config/domain_name/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 3-21.
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

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 Cloudscape driver, enter: `COM.cloudscape.core.JDBCDriver` ■ 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

The JDBC connection pool URL includes the following:

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

The following list 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`

Cloudscape driver	`jdbc:cloudscape:`*`database`*
	Or
	`jdbc:cloudscape:`*`full_path_to_db_directory`*
	
	For example:
	`jdbc:cloudscape:db`
	Or
	`jdbc:cloudscape:c:/bea/wli/config/wlidomain/dbInfo/cloudscape/db`

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`.
   - `BEA_HOME`
     Location of your BEA Home directory. For example: `C:\bea`.
   - `WL_HOME`
     Location of your WebLogic Server 6.1 installation. For example: `C:\bea\wlserver6.1`.
   - `WLI_HOME`
     Location of your WebLogic Integration installation. For example: `C:\bea\wlintegration2.1`.

   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 6.1 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→JMSServer-0→Destinations, and create a new JMSQueue, specifying a name and JNDI name for the queue. Accept the default settings for the other fields, or see your WebLogic Server 6.1 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→JMSServer-0→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 arguments for the `mdbgenerator` command are listed in the following table.

Table 3-5 mdbgenerator Command-Line Arguments

Argument	Description
`-queue` *`queue_name`*	Name of the custom queue for which you want to generate the MDB. This argument 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 argument 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.

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.

   Note: A sample of the `config.xml` file for the for the WebLogic Integration domain (`config\wlidomain`) is provided in Appendix A, "WebLogic Integration Sample Configuration Files."
6. Locate the following XML Tag:

   ```
   <Application Name="WLI" Path="WLI_HOME/lib>
   ```

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

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

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

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

`http://e-docs.bea.com/wls/docs61/ejb/deploy.html`

3. Add the EJB to the WLI application by manually updating the configuration file, `config.xml` located in *`WLI_HOME`*`/config/`*`domain_name`*, where *`WLI_HOME`* is the directory under which WebLogic Integration is installed (typically *`BEA_HOME`*`/wlintegration2.1`).

   This step is described in "Adding an EJB to the WLI Application" on page 3-27. The initial WLI application configuration is shown in Figure A-4 in 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 3-29.

Adding an EJB to the WLI Application

To add an EJB to the WLI application, you must update the configuration file, `config.xml`, located in *`WLI_HOME`*`/config/`*`domain_name`*, to specify the associated deployment descriptor files as part of the WebLogic Integration application. You must then restart the server, as described in "Getting Started" on page 1-1.

To specify the EJB descriptor files, use the `<EJBComponent>` element. You can control the order in which the EJB JAR files are deployed using the `DeploymentOrder` attribute. In general, if EJB A is dependent upon EJB B, then EJB B must be deployed first.

The following code listing is excerpted from the samples domain `config.xml` showing the information required to deploy an EJB called `MyEJB.jar`. This file is located in the `WLI_HOME/config/samples` directory. Notable lines of code are shown in **bold**.

Listing 3-2 Adding an EJB to the WLI Application

```
      .
      .
      .
<Application Deployed="true" Name="WLI" Path="E:\bea\wlintegration2.1\lib">
   <EJBComponent DeploymentOrder="0" Name="repository-ejb.jar"
      Targets="myserver" URI="repository-ejb.jar"/>
   <WebAppComponent Name="XTPlugin" Targets="myserver" URI="wlxtpi.war"/>
   <WebAppComponent Name="wlai" ServletReloadCheckSecs="1"
      Targets="myserver" URI="wlai.war"/>
   <EJBComponent DeploymentOrder="2" Name="wlpi-master-ejb.jar"
      Targets="myserver" URI="wlpi-master-ejb.jar"/>
   <EJBComponent DeploymentOrder="1" Name="wlpi-ejb.jar"
      Targets="myserver" URI="wlpi-ejb.jar"/>
   <EJBComponent DeploymentOrder="4" Name="wlc-wlpi-plugin.jar"
      Targets="myserver" URI="wlc-wlpi-plugin.jar"/>
   <EJBComponent DeploymentOrder="8" Name="wlai-admin-ejb"
      Targets="myserver" URI="wlai-admin-ejb.jar"/>
   <EJBComponent DeploymentOrder="5" Name="pobean.jar"
      Targets="myserver" URI="pobean.jar"/>
   <WebAppComponent Name="b2bconsole" ServletReloadCheckSecs="1"
      Targets="myserver" URI="b2bconsole.war"/>
   <EJBComponent DeploymentOrder="3" Name="wlpi-mdb-ejb.jar"
      Targets="myserver" URI="wlpi-mdb-ejb.jar"/>
   <EJBComponent DeploymentOrder="7" Name="wlai-ejb-server"
      Targets="myserver" URI="wlai-ejb-server.jar"/>
   <EJBComponent DeploymentOrder="6" Name="wlxtpi.jar"
      Targets="myserver" URI="wlxtpi.jar"/>
   <EJBComponent DeploymentOrder="9" Name="wlaiplugin-ejb.jar"
       Targets="myserver" URI="wlaiplugin-ejb.jar"/>
   <WebAppComponent Name="WLAIPlugin" Targets="myserver" URI="wlai-plugin.war"/>
   <EJBComponent DeploymentOrder="10" Name="sampleplugin-ejb.jar"
       Targets="myserver" URI="sampleplugin-ejb.jar"/>
   <WebAppComponent Name="com.bea.wlpi.SamplePlugin"
      Targets="myserver" URI="sampleplugin.war"/>
   <EJBComponent DeploymentOrder="11" Name="MyEJB.jar"
       Targets="myserver" URI="MyEJB.jar"/>
</Application>
      .
      .
      .
```

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/docs61/config_xml/index.html`

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 all the preconfigured domains; 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`*`DomainName`*`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, `$a * NULL` causes an exception.
- Null values cannot be used in logical expression. For example, `$a XOR NULL` will cause 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 before if they rely, in any way, on functionality that is affected. You must reexamine any legacy workflows and initialize appropriate workflow variable values, otherwise 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 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.

When you install WebLogic Integration, the software is configured by default 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 located in the domain directory.

A default RDBMSRealm is also configured, but is not initially enabled. If you chose to enable this realm, then users and groups are maintained in the database configured for the WebLogic Integration repository. When you use the WebLogic Integration database configuration wizard to specify a new database, the configuration for both the JDBC connection pool and the RDMBSRealm are updated.

The BPM-defined ACL objects are managed differently from users and groups. Even if you enable the RDBMSRealm, the 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/docs61/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 3-36.

The procedure for enabling the RDBMSRealm is provided in "Enabling the RDBMS Security Realm" on page 3-37.

The procedure for creating a custom security realm is provided in "Configuring a Custom Security Realm" on page 3-38. 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, the default users described in "WebLogic Integration Users and Passwords" on page 1-9, are organized into user groups. There are 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 default groups and assigned members are summarized in the following tables.

Table 3-6 Permission Groups and Members

Permission Group	Members
AdministerUser	admin, joe, mary, wlcsystem, wlpisystem
ConfigureComponents	admin, joe, mary, wlcsystem, wlpisystem
ConfigureSystem	admin, joe, mary, wlcsystem, wlpisystem
CreateTemplate	admin, joe, mary, wlcsystem, wlpisystem
DeleteTemplate	admin, joe, mary, wlcsystem, wlpisystem
ExecuteTemplate	admin, joe, mary, wlcsystem, wlpisystem
MonitorInstance	admin, joe, mary, wlcsystem, wlpisystem
UpdateTemplate	admin, joe, mary, wlcsystem, wlpisystem

Table 3-7 System Groups and Members

System Group	Members
everyone	admin, joe, system, mary, wlcsystem, wlpisystem
wlpiAdministrators	admin, joe, system, mary, wlpisystem
wlpiUsers	admin, joe, system, mary, wlcsystem, wlpisystem

Table 3-8 Role Groups and Members

Role Group	Members
AccountingCDE	admin, joe
CustomerServiceCDE	admin

Table 3-8 Role Groups and Members (Continued)

Role Group	Members
Role1Org1	admin, joe, mary
Role2Org1	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 `wlpisystem`.
- You must create the following WebLogic Server groups:
  - All permission groups, as listed in Table 3-6
  - `wlpiUsers` group
- The `wlpisystem` 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 on creating an alternate security realm, see your WebLogic Server 6.1 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*.

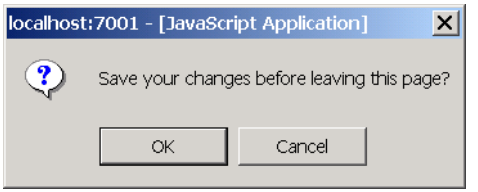
Enabling the RDBMS Security Realm

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

1. 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, select Security.
4. Select the Filerealm tab.

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

   Figure 3-8 Leaving the Page? Confirmation



5. From the Caching Realm drop-down list, select `wlpiCachingRealm`.
6. Click Apply.
7. Shut down WebLogic Integration, as described in "Stopping WebLogic Integration" on page 1-20.
8. Go to the `config/`*`domain_name`* directory for the domain you are updating.
9. Back up the file `fileRealm.properties` by copying and renaming it.
10. Open the original `fileRealm.properties` file in a text editor.
11. Delete all entries beginning with `user` and `group`, and save the file.

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

12. 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 procedures:

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

The following sections describe each procedure.

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 Security→Realms.
4. Click the 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. Click Create.

Configuring the Caching Realm

To configure the caching realm:

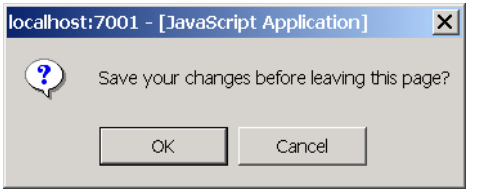
1. In the navigation tree, choose Security→Caching Realms→wlpiCachingRealm.
2. Select the realm you created from the Basic Realm drop-down list.
3. Deselect the Case Sensitive Cache option.
4. Click Apply.

Enabling the WLPI Caching Realm

1. In the navigation tree, choose 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

This is an optional step. To remove the RDMBS realm:

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

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 `wlcsystem` user name and password are used in the WebLogic Integration run-time environment and are stored in the WebLogic Integration repository. This password *must* be updated using the B2B Administration Console. *Do not* use the WebLogic Server Administration Console to update this password.

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.

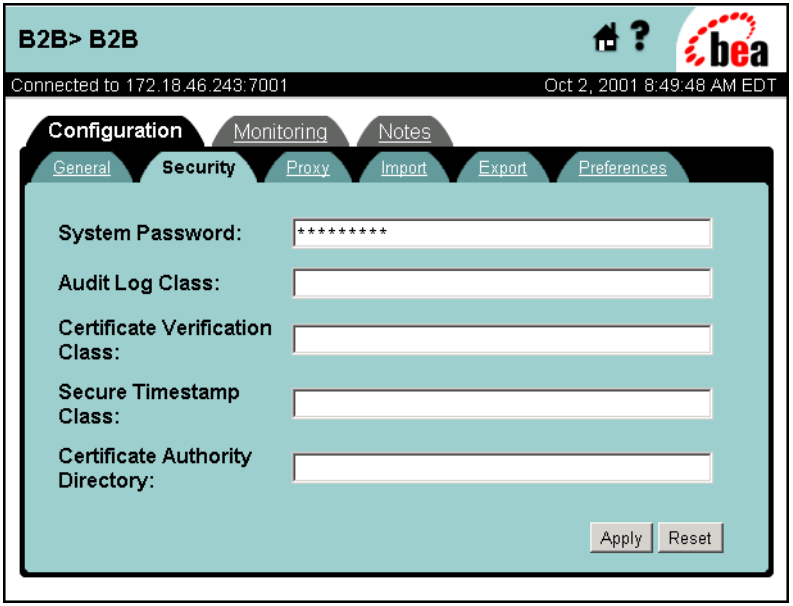
Updating the wlcsystem Password

The password for the `wlcsystem` 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.

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 `wlcsystem` user.

Figure 3-10 B2B Configuration Security Tab



4. To update the `wlcsystem` 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 `wlcsystem` password.

Customizing Mail Session Properties

The mail session properties you specify during installation are used to set up a basic mail session configuration for each of the preconfigured domains. 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 3-9 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.

Changing the Root Directory for a Domain

Some components of WebLogic Integration expect to find files, such as the XML DTD, in the root directory. By default, the root directory for a domain is the parent directory of the `config/`*`domain_name`* domain directory.

For example, for the WebLogic Integration domain, *`BEA_HOME`*`/wlintegration2.1/config/wlidomain`, the root directory (also known as the runtime directory) is *`BEA_HOME`*`/wlintegration2.1`.

Figure 3-12 Domain Root (Runtime) Directory

Because the server root directory is the installation directory (*WLI_HOME*), many files can accumulate in this directory, particularly if you are working with more than one of the preconfigured domains. To work around this problem, you can designate the domain directory as the current directory (`config/`*domain_name*) by changing the value of the `weblogic.RootDirectory` property in the `startWeblogic` command script for the domain.

If you change the root directory for the samples domain, there are special considerations. Sample code that requires access to a file must specify the exact location of the file. One possible approach is to make all files relative to the server root directory and prepend the name of the server root directory to the name of every file. The following listing shows how to implement this workaround.

Listing 3-3 Prepending Root to Filenames

```
import java.io.*;

public class Sample {
   static File rootDir;

   // Get the server root directory
   static {
     String rootPath = System.getProperty("weblogic.RootDirectory");
     if (rootPath != null){
       rootDir = new File(rootPath);
     }
   }

   // Helper method that returns a properly rooted File
   public static File setRoot(String fileName){
     File f = new File(fileName);

     if (rootDir == null || f.isAbsolute()){
       return f;
}

     return new File(rootDir, fileName);
  }

   // One method that uses the helper
   public void do(){
     String file = "config/samples/data/PO.xml";

     try {
       FileInputStream fis = new FileInputStream(setRoot(file));
...
   } catch (IOException e){
...
   }
  }
 }
```

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 set. 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*.

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
- wlai.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

By default, the BEA WebLogic Integration installation sets up the domains described in "WebLogic Integration Preconfigured Domains" on page 1-3. The configuration for each domain is defined in eXtensible Markup Language (XML). Persistent storage for a domain configuration is provided by the `config.xml` file located in *`WLI_HOME`*`/config/`*`domain_name`*, where *`WLI_HOME`* is the directory under which WebLogic Integration is installed (typically *`BEA_HOME`*`/wlintegration2.1`).

This section presents and describes the contents of the WebLogic Integration domain `config.xml` file (`config/wlidomain/config.xml`).

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.

Note: You will need to edit the `config.xml` file directly when you are configuring a custom JMS queue or adding an EJB to the WLI application. For more information, see "Configuring a Custom Java Message Service Queue" on page 3-24 or "Adding an EJB to the WLI Application" on page 3-27, respectively.

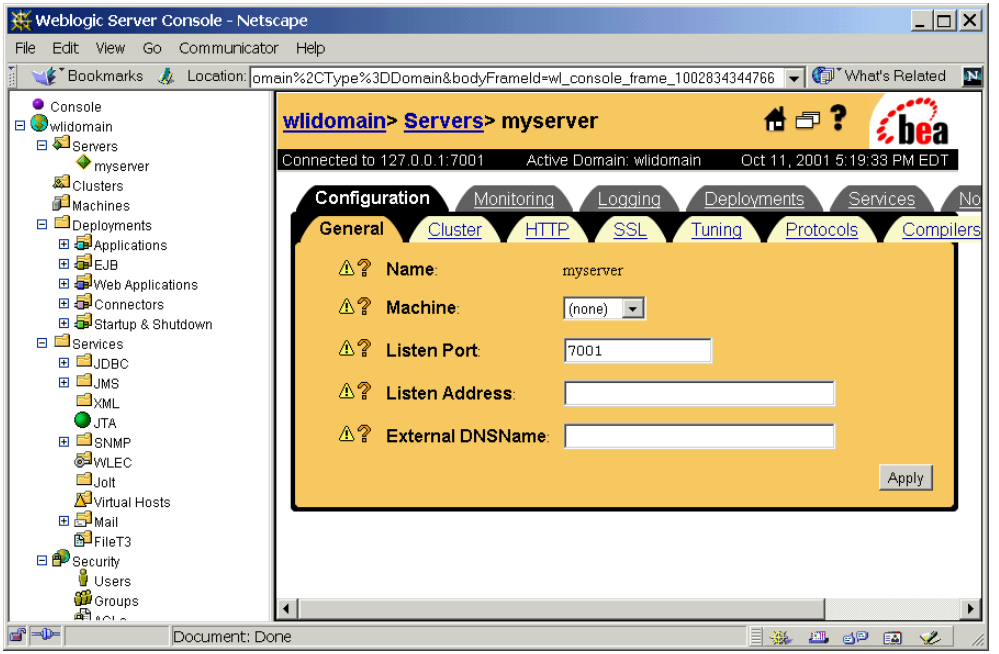
The root element of the `config.xml` file is `Domain`. The first entries in the domain are the domain log file and administrative server configurations, as shown in the following figure.

Figure A-1 config.xml File: Part 1

```
<?xml version="1.0"?>

<Domain
   Name="wlidomain">

   <!-- Domain log -->
   <Log
      Name="wlidomain"
      FileName="C:\bea\wlintegration2.1/config/wlidomain/logs/wlidomain.log"/>

   <Server
      Name="myserver"
      ListenPort="7001"
      TransactionLogFilePrefix="C:\bea\wlintegration2.1/config/wlidomain/logs/"
      NativeIOEnabled="true">

      <!--  Include followings in Server to see log info on screen
      StdoutDebugEnabled="true"
      StdoutEnabled="true"
      StdoutSeverityLevel="64"
      -->

      <!-- Server log -->
      <Log
         Name="myserver"
         FileName="C:\bea\wlintegration2.1/config/wlidomain/logs/myserver.log"/>

      <WebServer
         Name="myserver"
         DefaultWebApp="DefaultWebApp_myserver"
         HttpsKeepAliveSecs="120"
         KeepAliveSecs="60"
         LogFileName="C:\bea\wlintegration2.1/config/wlidomain/logs/access.log"
         LoggingEnabled="true"/>

      <KernelDebug
         Name="myserver"/>
      <ServerDebug
         Name="myserver"/>
      <SSL
         Name="myserver"/>
   </Server>
```

The `Domain` and `Server` elements and attributes define the basic configuration. The settings here are reflected 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-2 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/docs61`

As shown in the following figure, the server element is followed by the specification of the Java Transaction API (JTA), security related information, the Mbean for the application manager, and the default Web application.

Figure A-3 config.xml File: Part 2

```
<JTA
   Name="wlidomain"
   TimeoutSeconds="3600"/>

<ApplicationManager
   Name="wlidomain"/>

<Security
   Name="wlidomain"
   GuestDisabled="false"
   PasswordPolicy="mypasswordpolicy"
   Realm="myRealm"/>

<PasswordPolicy
   Name="mypasswordpolicy"
   MinimumPasswordLength="4"/>

<Realm
   Name="myRealm"
   CachingRealm=""
   FileRealm="myFileRealm"/>

<FileRealm
   Name="myFileRealm"/>

<Application
   Name="DefaultWebApp_myserver"
   Deployed="true"
   Path="C:\bea\wlintegration2.1/config/wlidomain/applications">
   <WebAppComponent
      Name="DefaultWebApp_myserver"
      Targets="myserver"
      URI="DefaultWebApp_myserver"
      WebServers="myserver"
      IndexDirectoryEnabled="true"/>
</Application>
```

As shown in the following two figures, the next section deploys the EJBs and web applications required for the WebLogic Integration application. The deployment order is controlled by the `DeploymentOrder` setting. You can use this setting, as required, to control deployment order for your custom applications and EJBs.

Figure A-4 config.xml File: Part 3

```
<Application
    Name="WLI"
    Path="C:\bea\wlintegration2.1/lib">
    <EJBComponent
       Name="repository-ejb.jar"
       Targets="myserver"
       URI="repository-ejb.jar"
       DeploymentOrder="0"/>
    <EJBComponent
       Name="wlpi-ejb.jar"
       Targets="myserver"
       URI="wlpi-ejb.jar"
       DeploymentOrder="1"/>
    <EJBComponent
       Name="wlpi-master-ejb.jar"
       Targets="myserver"
       URI="wlpi-master-ejb.jar"
       DeploymentOrder="2"/>
    <EJBComponent
       Name="wlpi-mdb-ejb.jar"
       Targets="myserver"
       URI="wlpi-mdb-ejb.jar"
       DeploymentOrder="3"/>
    <EJBComponent
       Name="wlc-wlpi-plugin.jar"
       Targets="myserver"
       URI="wlc-wlpi-plugin.jar"
       DeploymentOrder="4"/>
    <EJBComponent
       Name="pobean.jar"
       Targets="myserver"
       URI="pobean.jar"
       DeploymentOrder="5"/>
    <EJBComponent
       Name="wlxtpi.jar"
       Targets="myserver"
       URI="wlxtpi.jar"
       DeploymentOrder="6"/>
    <EJBComponent
       Name="wlai-ejb-server"
       Targets="myserver"
       URI="wlai-ejb-server.jar"
       DeploymentOrder="7"/>
    <EJBComponent
       Name="wlai-admin-ejb"
       Targets="myserver"
       URI="wlai-admin-ejb.jar"
       DeploymentOrder="8"/>
    <EJBComponent
       Name="wlaiplugin-ejb.jar"
       Targets="myserver"
       URI="wlaiplugin-ejb.jar"
       DeploymentOrder="9"/>
```

Controls deployment order

Figure A-5 config.xml File: Part 4

```
    <WebAppComponent
       Name="b2bconsole"
       ServletReloadCheckSecs="1"
       Targets="myserver"
       URI="b2bconsole.war"
       DeploymentOrder="10"/>
    <WebAppComponent
       Name="wlai"
       ServletReloadCheckSecs="1"
       Targets="myserver"
       URI="wlai.war"
       DeploymentOrder="11"/>
    <WebAppComponent
       Name="WLAIPlugin"
       Targets="myserver"
       URI="wlai-plugin.war"
       DeploymentOrder="12"/>
    <WebAppComponent
       Name="XTPlugin"
       Targets="myserver"
       URI="wlxtpi.war"
       DeploymentOrder="13"/>
 </Application>
```

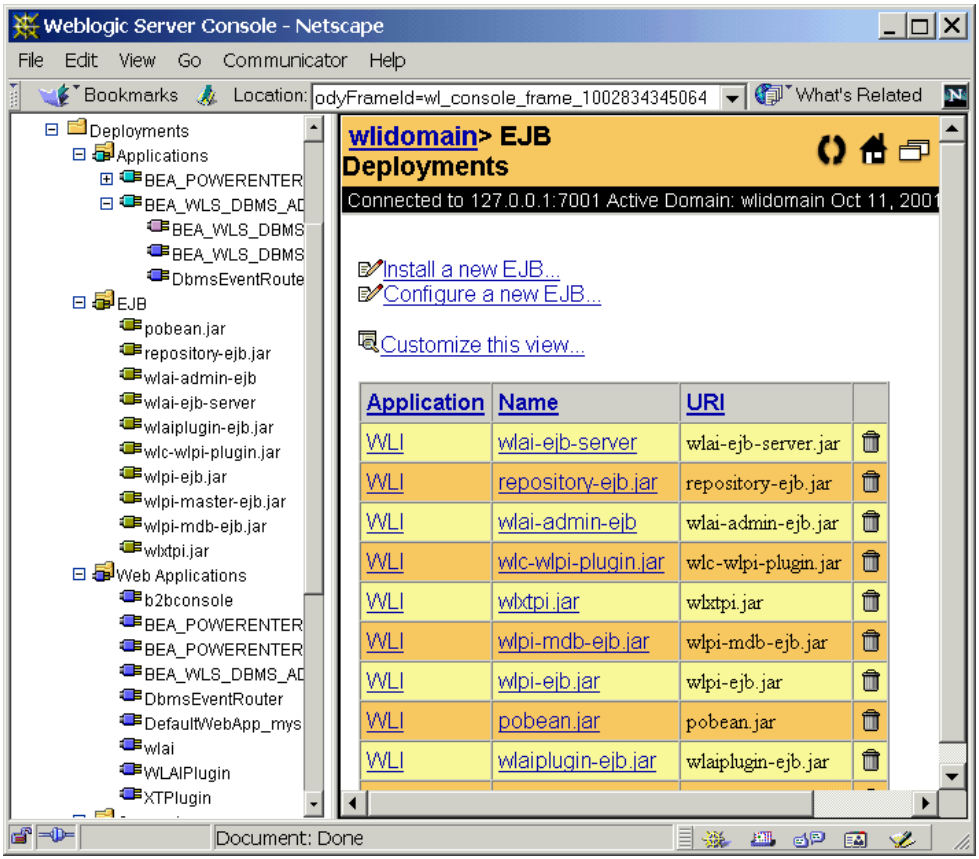
As shown in the following figure, the next section of the `config.xml` file deploys the application integration DBMS sample adapter and the BEA WebLogic Adapter for Peregrine Power.Enterprise!. The arrow indicates the continuation of text that appears on a single line in the file.

Figure A-6 config.xml File: Part 5

```
<Application
   Name="BEA_WLS_DBMS_ADK"
   Deployed="true"
   Path="C:\bea\wlintegration2.1/adapters/dbms/lib/BEA_WLS_DBMS_ADK.ear">
   <ConnectorComponent
      Name="BEA_WLS_DBMS_ADK"
      Targets="myserver" URI="BEA_WLS_DBMS_ADK.rar"/>
   <WebAppComponent
      Name="DbmsEventRouter"
      Targets="myserver"
      URI="BEA_WLS_DBMS_ADK_EventRouter.war"/>
   <WebAppComponent
      Name="BEA_WLS_DBMS_ADK_Web"
      Targets="myserver"
      URI="BEA_WLS_DBMS_ADK_Web.war"/>
</Application>

<!-- BEA WebLogic Adapter for Peregrine Power.Enterprise! -->
<Application
   Name="BEA_POWERENTERPRISE_3_0"
   Deployed="true"
   Path="C:\bea\wlintegration2.1/adapters/powerenterprise/lib/BEA_POWER
                                                        ENTERPRISE_3_0_EAR.ear">
  <ConnectorComponent
      Name="BEA_POWERENTERPRISE_3_0"
      Description="J2EE CA adapter for PowerEnterprise!"
      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>
```

You can view and update parameters for the applications, Web applications, and EJBs deployed by selecting Deployments from the WebLogic Server Administration Console navigation tree, then selecting the appropriate deployment type. The EJB deployments are shown in the following figure.

Figure A-7 Deployments



As shown in the following figure, the next section of the `config.xml` file includes the startup and shutdown classes for the B2B engine, process engine, and application integration. The arrows indicate the continuation of text that appears on a single line in the file.

Figure A-8 config.xml File: Part 6

```
<StartupClass
   Name="WLAI_StartupClass"
   ClassName="com.bea.wlai.WLAIStartup"
   FailureIsFatal="true"
   Targets="myserver"/>

<StartupClass
   Name="WLPIInit"
   ClassName="com.bea.wlpi.server.wlpiinit.WLPIInit"
   Arguments="poolName=wliPool,user=wlpisystem,pwd=wlpisystem,
              expireInterval=30000,deliverInterval=86400000,
              checkInterval=600000"
   Targets="myserver"/>

<StartupClass
   Name="TimeProcessor"
   ClassName="com.bea.wlpi.server.timeprocessor.TimeProcessor"
   Arguments="poolName=wliPool,user=wlpisystem,pwd=wlpisystem,
              connectionFactory=com.bea.wlpi.TopicConnectionFactory,
              topic=com.bea.wlpi.TimerTopic"
   Targets="myserver"/>

<StartupClass
   Name="WLCStartup"
   ClassName="com.bea.b2b.server.Startup"
   Arguments="PERSISTENCE=OFF"
   Targets="myserver"/>

<ShutdownClass
   Name="WLCShutdown"
   ClassName="com.bea.b2b.server.Shutdown"
   Arguments="mode=TERMINATE"
   Targets="myserver"/>

<ShutdownClass
   Name="WLAI_ShutdownClass"
   ClassName="com.bea.wlai.WLAIStartup"
   Targets="myserver"/>
```

As shown in the following figure, the next section of the `config.xml` file includes the JDBC connection pool information, transaction data sources, and the JMS connection factories used by BPM JMS clients to create JMS connections. The bold text indicates settings that are updated based on the information you provide during installation.

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

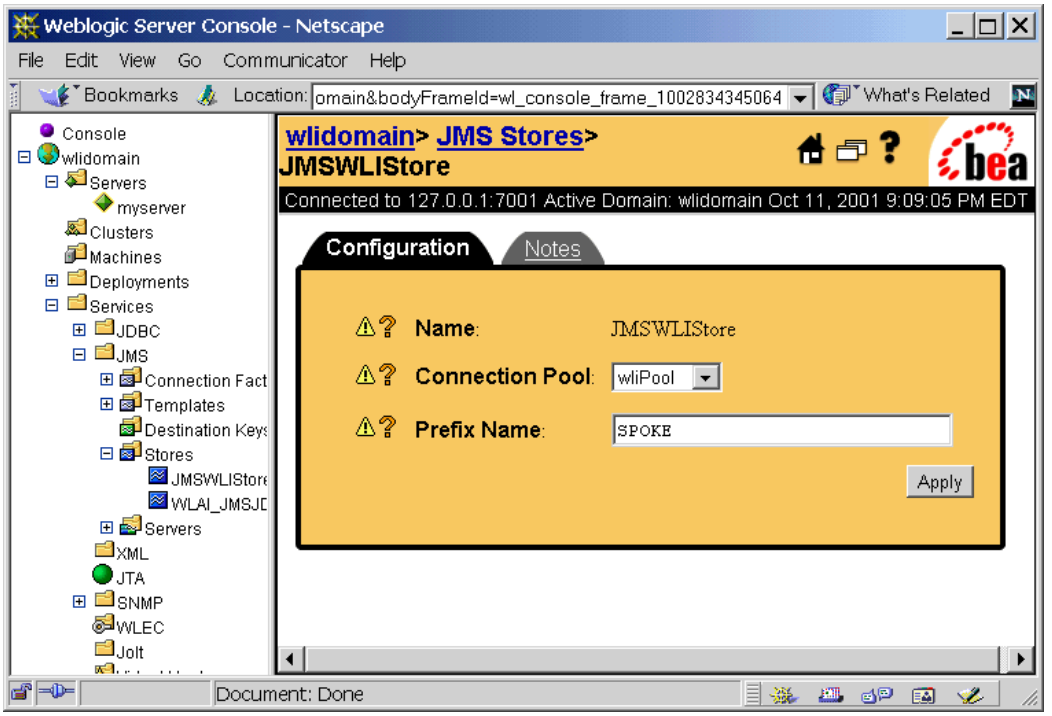
Figure A-9 config.xml File: Part 7


```
<JDBCConnectionPool
    Name="wliPool"
    CapacityIncrement="2"
    DriverName="weblogic.jdbc.mssqlserver4.Driver"
    InitialCapacity="8"
    LoginDelaySeconds="1"
    MaxCapacity="36"
    Properties="user=myuser;password=mypassword"
    RefreshMinutes="0"
    ShrinkPeriodMinutes="15"
    ShrinkingEnabled="true"
    Targets="myserver"
    URL="jdbc:weblogic:mssqlserver4:WLIDB@mssqlhost:1433"/>

<JMSJDBCStore
    Name="JMSWLIStore"
    ConnectionPool="wliPool"
    PrefixName="SPOKE" />

 <JDBCTxDataSource
    Name="WLCHub.DS"
    JNDIName="WLCHub.DS"
    PoolName="wliPool"
    EnableTwoPhaseCommit="true"
    Targets="myserver"/>

 <JDBCTxDataSource
    Name="TXDataSource"
    EnableTwoPhaseCommit="true"
    JNDIName="com.bea.wlpi.TXDataSource"
    PoolName="wliPool"
    Targets="myserver"/>

 <JMSConnectionFactory
    Name="wlpiQueueFactory"
    AllowCloseInOnMessage="true"
    JNDIName="com.bea.wlpi.QueueConnectionFactory"
    Targets="myserver"
    UserTransactionsEnabled="true"/>

<JMSConnectionFactory
    Name="wlpiFactory"
    AllowCloseInOnMessage="true"
    JNDIName="com.bea.wlpi.
    TopicConnectionFactory"
    Targets="myserver"
    UserTransactionsEnabled="true"/>
```

To optimize performance, the setting for the JMS JDBC store prefix name should be customized for your database. The syntax for a prefix name is `[[`*`catalog`*`.]`*`schema`*`.]`*`prefix`*`]`.

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

Figure A-10 JMS JDBC Store Page



As shown in the following figure, the next section of the `config.xml` file defines the JMS server and mail session properties for the BPM framework.

Figure A-11 config.xml File: Part 8

```
<JMSServer
    Name="WLIJMSServer"
    Targets="myserver"
   TemporaryTemplate="TemporaryTemplate"
    Store="JMSWLIStore">
   <JMSTemplate
      Name="TemporaryTemplate"/>
   <JMSTopic
      Name="wlpiTimer"
      JNDIName="com.bea.wlpi.TimerTopic"/>
   <JMSQueue
      Name="eventQueue"
      JNDIName="com.bea.wlpi.EventQueue"/>
   <JMSQueue
      Name="eventQueueExt"
      JNDIName="com.bea.wlpi.EventQueueExt"/>
   <JMSQueue
      Name="wlpiValidatingEvent"
      JNDIName="com.bea.wlpi.ValidatingEventQueue"/>
   <JMSTopic
      Name="wlpiError"
      JNDIName="com.bea.wlpi.ErrorTopic"/>
   <JMSTopic
      Name="Timer
      JNDIName="com.bea.wlpi.Timer""
      StoreEnabled="default"/>
   <JMSTopic
      Name="wlpiAudit"
      JNDIName="com.bea.wlpi.AuditTopic"/>
   <JMSQueue
      Name="initQueue"/>
      JNDIName="com.bea.wlpi.InitQueue"
   <JMSTopic
      Name="wlpiEvent"
      JNDIName="com.bea.wlpi.EventTopic"/>
   <JMSTopic
      Name="wlpiNotify"
      JNDIName="com.bea.wlpi.NotifyTopic"/>
</JMSServer>

<MailSession
     Name="wlpiMailSession"
     JNDIName="com.bea.wlpi.MailSession"
     Properties="mail.sender=WebLogic Integration;mail.from=user@domain.
                 com;mail.host=mail.domain.com"
     Targets="myserver"/>
```

As shown in the following figure, the next section of the `config.xml` file includes the data sources, data stores, JMS server, and JMS connection factory used by application integration JMS clients to create JMS connections.

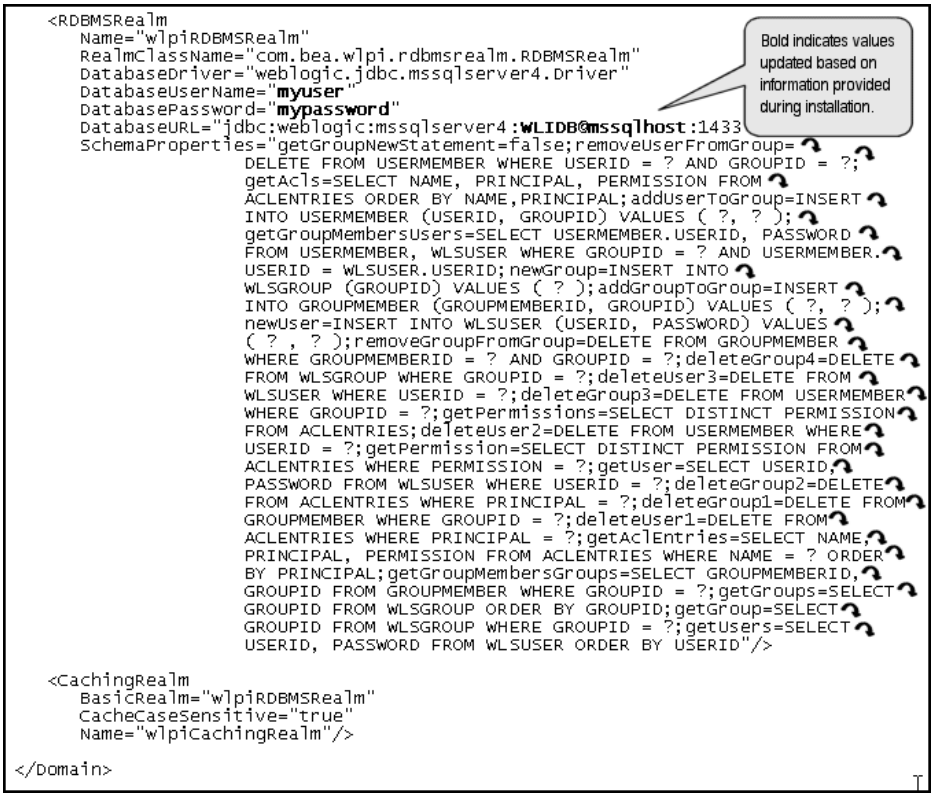
Figure A-12 config.xml File: Part 9

```
 <JDBCDataSource
    Name="WLAI_DataSource"
    JNDIName="WLAI_DataSource"
    PoolName="wliPool"
    Targets="myserver"/>

<JMSJDBCStore
    ConnectionPool="wliPool"
    Name="WLAI_JMSJDBCStore"/>

 <JMSConnectionFactory
    Name="WLAI_JMSConnectionFactory"
    AllowCloseInOnMessage="false"
    DefaultDeliveryMode="Persistent"
    DefaultPriority="4"
    DefaultTimeToLive="0"
    JNDIName="com.bea.wlai.JMSConnectionFactory"
    MessagesMaximum="10"
    OverrunPolicy="KeepOld"
    Targets="myserver"
    UserTransactionsEnabled="true"/>

<JMSServer
    Name="WLAI_JMSServer"
    Store="WLAI_JMSJDBCStore"
    Targets="myserver"
    TemporaryTemplate="WLAI_TemporaryTopicTemplate">
    <JMSTemplate
       Name="WLAI_TemporaryTopicTemplate"
       DeliveryModeOverride="Persistent"/>
    <JMSQueue
       Name="WLAI_APP_VIEW_DEPLOYMENTS_QUEUE"
       JNDIName="wlai.APP_VIEW_DEPLOYMENTS_QUEUE"/>
    <JMSQueue
       Name="WLAI_ASYNC_RESPONSE"
       JNDIName="wlai.myserver.ASYNC_RESPONSE"/>
    <JMSQueue
       Name="WLAI_ASYNC_REQUEST"
       JNDIName="wlai.myserver.ASYNC_REQUEST"/>
 </JMSServer>
```

The last section of the file configures the mail session properties and sets up the RDBMS and caching security realms. The arrows indicate the continuation of text that appears on a single line in the file.

For information about configuring alternate security realms, see “Understanding the BPM Security Model” on page 3-33.

Figure A-13 config.xml File: Part 10




```
   <RDBMSRealm
      Name="wlpiRDBMSRealm"
      RealmClassName="com.bea.wlpi.rdbmsrealm.RDBMSRealm"
      DatabaseDriver="weblogic.jdbc.mssqlserver4.Driver"
      DatabaseUserName="myuser"
      DatabasePassword="mypassword"
      DatabaseURL="jdbc:weblogic:mssqlserver4:WLIDB@mssqlhost:1433"
      SchemaProperties="getGroupNewStatement=false;removeUserFromGroup=
                        DELETE FROM USERMEMBER WHERE USERID = ? AND GROUPID = ?;
                        getAcls=SELECT NAME, PRINCIPAL, PERMISSION FROM
                        ACLENTRIES ORDER BY NAME,PRINCIPAL;addUserToGroup=INSERT
                        INTO USERMEMBER (USERID, GROUPID) VALUES ( ?, ? );
                        getGroupMembersUsers=SELECT USERMEMBER.USERID, PASSWORD
                        FROM USERMEMBER, WLSUSER WHERE GROUPID = ? AND USERMEMBER.
                        USERID = WLSUSER.USERID;newGroup=INSERT INTO
                        WLSGROUP (GROUPID) VALUES ( ? );addGroupToGroup=INSERT
                        INTO GROUPMEMBER (GROUPMEMBERID, GROUPID) VALUES ( ?, ? );
                        newUser=INSERT INTO WLSUSER (USERID, PASSWORD) VALUES
                        ( ? , ? );removeGroupFromGroup=DELETE FROM GROUPMEMBER
                        WHERE GROUPMEMBERID = ? AND GROUPID = ?;deleteGroup4=DELETE
                        FROM WLSGROUP WHERE GROUPID = ?;deleteUser3=DELETE FROM
                        WLSUSER WHERE USERID = ?;deleteGroup3=DELETE FROM USERMEMBER
                        WHERE GROUPID = ?;getPermissions=SELECT DISTINCT PERMISSION
                        FROM ACLENTRIES;deleteUser2=DELETE FROM USERMEMBER WHERE
                        USERID = ?;getPermission=SELECT DISTINCT PERMISSION FROM
                        ACLENTRIES WHERE PERMISSION = ?;getUser=SELECT USERID,
                        PASSWORD FROM WLSUSER WHERE USERID = ?;deleteGroup2=DELETE
                        FROM ACLENTRIES WHERE PRINCIPAL = ?;deleteGroup1=DELETE FROM
                        GROUPMEMBER WHERE GROUPID = ?;deleteUser1=DELETE FROM
                        ACLENTRIES WHERE PRINCIPAL = ?;getAclEntries=SELECT NAME,
                        PRINCIPAL, PERMISSION FROM ACLENTRIES WHERE NAME = ? ORDER
                        BY PRINCIPAL;getGroupMembersGroups=SELECT GROUPMEMBERID,
                        GROUPID FROM GROUPMEMBER WHERE GROUPID = ?;getGroups=SELECT
                        GROUPID FROM WLSGROUP ORDER BY GROUPID;getGroup=SELECT
                        GROUPID FROM WLSGROUP WHERE GROUPID = ?;getUsers=SELECT
                        USERID, PASSWORD FROM WLSUSER ORDER BY USERID"/>

   <CachingRealm
      BasicRealm="wlpiRDBMSRealm"
      CacheCaseSensitive="true"
      Name="wlpiCachingRealm"/>

</Domain>
```

setEnv

This executable file is used to set the environment variables for WebLogic Integration. A version 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 the WebLogic Integration distribution.

The beginning of the `setEnv.cmd` (Windows) or `setEnv.sh` (UNIX) file for the `config/wlidomain` domain is shown in the following figure. The values in bold are set during installation based on your installation location.

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

Figure A-14 setEnv Command

```
@echo off

set JAVA_HOME=C:\bea\jdk131
set BEA_HOME=C:\bea
set WL_HOME=C:\bea\wlserver6.1
set WLI_HOME=C:\bea\wlintegration2.1

: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

for %%a in (
   lib\weblogic.jar,
   lib\xmlx.jar,
   samples\eval\cloudscape\lib\cloudscape.jar
) do call :cmnwls %%a

set WLICOMMONCP=%WLICOMMONCP%;%BEA_HOME%;%WLI_HOME%\ext

REM Set the WLI server Classpath
set WLISERVERCP=%WLICOMMONCP%

for %%a in (
   wliserver.jar
) do call :wlisvrlib %%a

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

.
.
.
```

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 preconfigured domain directory.

The `startWeblogic.cmd` (Windows) or `startWeblogic` (UNIX) file for the `config/wlidomain` domain is shown in the following figure. The arrows indicate the continuation of a line that appears as a single line in the file.

Figure A-15 startWeblogic Command

```
@echo off

setlocal

call %WLI_HOME%\config\wlidomain\SetWliDomainData || goto :EOF

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

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

REM WLIS deployment directory
if not exist %WLI_DOMAIN_HOME%\wlai\deploy mkdir %WLI_DOMAIN_HOME%\wlai\deploy || goto finish

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

REM Start weblogic
%JAVA_HOME%\bin\java %DB_JVMARGS% -Xmx256m -classpath %SVRCP% -Dbea.home=%BEA_HOME%
-Dweblogic.home=%WL_HOME% -Dweblogic.system.home=%WLI_HOME%
-Dwli.bpm.server.evaluator.supportsNull=false -Dweblogic.Domain=wlidomain
-Dweblogic.RootDirectory=%WLI_HOME% -Djava.security.policy=%WL_HOME%\lib\weblogic.policy
-Dweblogic.management.discover=false weblogic.Server
-Dweblogic.management.password=security -Dweblogic.Name=myserver

endlocal
```

The Password is provided with this statement. If you change the password you must update accordingly.
If you delete this statement, you will be prompted to provide a password to boot the server.

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 `config/wlidomain` domain is shown in the following figure.

Figure A-16 fileRealm.properties File: Part 1

```
#Thu Oct 11 23:15:43 PDT 2001
acl.reset.weblogic.jdbc.connectionPool.wliPool=wlcSamplesUser,admin,guest
acl.reset.weblogic.jdbc.connectionPool.wlpiPool=admin,guest
user.admin=0xab3a488db0652704287970cdf97854812feea77b
group.CreateTemplate=admin,joe,mary,guest,wlcsystem,wlpisystem
acl.list.weblogic.jndi.weblogic=system,everyone
acl.lockServer.weblogic.admin=system
acl.list.weblogic.jndi.weblogic.ejb=system,everyone
group.wlpiUsers=admin,joe,system,mary,wlcsystem,guest,wlpisystem
acl.list.weblogic.jndi.weblogic.rmi=everyone
acl.reserve.weblogic.jdbc.connectionPool.wliPool=everyone
acl.execute.weblogic.servlet.AdminThreads=system
acl.hubconfig.WLCAdmin=admin
acl.write.weblogic.workspace=system,everyone
user.mary=0xa078cb45e6f6c4eefdd1f14495ff739b5536904c
user.wlcSamplesUser=0x6f53333430ac49627feb4d439eb3a6108660673d
acl.shrink.weblogic.jdbc.connectionPool.wlpiPool=admin,guest
user.wlpisystem=0xfd511836a8e67fa799348c8635de15185677aff2
acl.execute.weblogic.servlet.T3AdminMain=system
acl.unlockServer.weblogic.admin=system
group.ConfigureComponents=admin,joe,mary,guest,wlcsystem,wlpisystem
acl.execute.weblogic.servlet.AdminJDBC=system
acl.shrink.weblogic.jdbc.connectionPool.examplesPool=everyone
acl.execute.weblogic.servlet.AdminRealm=system
user.wlcsystem=0x461c9174f9a5c37c02872cf1adaa019dc8f82740
acl.modify.weblogic.jndi.weblogic=system,everyone,guest
acl.execute.weblogic.servlet.ConsoleHelp=everyone
acl.execute.weblogic.servlet.Certificate=system
group.wlcSamplesGroup=wlcSamplesUser,guest,wlpisystem
acl.modify.weblogic.jndi.weblogic.ejb=system,everyone,guest
acl.reset.weblogic.jdbc.connectionPool.examplesPool=wlcSamplesUser,admin,guest
acl.modify.weblogic.jndi.weblogic.rmi=everyone
acl.execute.weblogic.servlet.classes=everyone
group.ExecuteTemplate=admin,joe,mary,guest,wlcsystem,wlpisystem
acl.execute.weblogic.servlet=system,everyone
acl.modify.weblogic.admin.acl=system
acl.lookup.weblogic.jndi.weblogic.ejb=system,everyone,guest
acl.execute.weblogic.servlet.AdminClients=system
acl.read.managedObject=system
group.wlpiAdministrators=admin,wlcSamplesUser,joe,system,mary,guest,wlpisystem
acl.lookup.weblogic.jndi.weblogic.rmi=everyone
acl.lookup.weblogic.admin.mbean.MBeanHome=wlcSamplesUser,everyone,guest
acl.list.weblogic.jndi.weblogic.fileSystem=everyone
```

Figure A-17 fileRealm.properties File: Part 2

```
group.Role2Org2=admin,joe,mary
group.Role2Org1=admin,joe,mary
group.ConfigureSystem=admin,joe,mary,guest,wlcsystem,wlpisystem
acl.lookup.weblogic.jndi.weblogic.fileSystem=everyone
acl.enablermonitor.WLCAdmin=admin
user.joe=0xa078cb45e6f6c4eefdd1f14495ff739b5536904c
group.DeleteTemplate=admin,joe,mary,guest,wlcsystem,wlpisystem
group.AdministerUser=admin,joe,mary,guest,wlcsystem,wlpisystem
acl.create.weblogic.jms.ServerSessionPool=everyone
acl.execute.weblogic.servlet.AdminEvents=system
group.UpdateTemplate=admin,joe,mary,guest,wlcsystem,wlpisystem
acl.execute.weblogic.servlet.AdminVersion=system
user.hub=0x1f1e1abf32b840eee4a0f35efa7a44bba3afbc5c
group.adminGroup=admin,joe,mary
group.Role1Org2=admin,joe,mary
group.Role1Org1=admin,joe,mary
acl.shrink.weblogic.jdbc.connectionPool.wliPool=wlcSamplesUser,admin,guest
group.MonitorInstance=admin,joe,mary,guest,wlcsystem,wlpisystem
acl.execute.weblogic.servlet.containerManaged=everyone
user.system=0xab3a488db0652704287970cdf97854812feea77b
acl.write.managedObject=system
acl.execute.weblogic.servlet.AdminMain=system
acl.execute.weblogic.servlet.AdminLicense=system
acl.execute.weblogic.servlet.AdminProps=system
acl.modify.weblogic.jndi.weblogic.fileSystem=everyone
acl.lookup.weblogic.jndi.weblogic=everyone
acl.shutdown.weblogic.admin=system
acl.reserve.weblogic.jdbc.connectionPool=system
acl.execute.weblogic.servlet.AdminConnections=system
acl.reset.weblogic.jdbc.connectionPool=system
acl.read.weblogic.workspace=system,everyone
acl.boot.weblogic.server=system,everyone
acl.access.weblogic.admin.mbean.MBeanHome=wlcSamplesUser,everyone,guest
group.adapter=joe,system,guest,hub,wlpisystem,mary,wlcsystem,wlcSamplesUser,admin
group.AccountingCDE=admin,joe
group.CustomerServiceCDE=admin
group.ShippingCDE=admin,mary
```

wlai.properties

The `wlai.properties` file sets the global properties used by WebLogic Integration for application integration. The following table provides a description of each property.

Application Integration Properties
`wlai.appView.deploy=`*`appview_name`*`[,`*`appview_name2`*`,...]`
Specifies the (one or more) persistent application views to be deployed. *Example:* `wlai.appView.deploy=WLISAppView.sav`
`wlai.deploymentRepositoryRootPath=`*`directory`*
Specifies the J2EE CA connection factory deployment repository directory. This property is required. *Example:* `wlai.deploymentRepositoryRootPath=D:/wli/src/staging_nt/config/wlidomain/wlai/deploy`
`wlai.hostNameAndPort=`*`host`*`:`*`port`* `wlai.hostPassword=`*`password`* `wlai.hostUserID=`*`user`*

Specifies the localhost server access information. These properties are required.
Example:
`wlai.hostNameAndPort=localhost:7001`
`wlai.hostPassword=security`
`wlai.hostUserID=system`

`wlai.admin.hostNameAndPort=<host:port>`
`wlai.admin.hostPassword=<password>`
`wlai.admin.hostUserID=<user>`

Specifies the administration server access information. These properties are required only if the localhost is a managed server.
Example:
`wlai.admin.hostNameAndPort=203.198.68.3:7001`
`wlai.admin.hostPassword=security`
`wlai.admin.hostUserID=system`

Application Integration Properties

`wlai.jms.connectionFactoryJNDIName=`*`connection_factory_JNDI_name`*

Specifies the JMS connection factory JNDI name. This property is optional.

Example:

```
wlai.jms.connectionFactoryJNDIName=com.bea.wlai.JMSConnectionFactory
```

`wlai.jms.serverName=`*`JMS_server_name`*

Specifies the JMS server name. This property is optional.

Example:

```
wlai.jms.serverName=WLAI_JMSServer
```

`wlai.jms.asyncServiceTransFlag=`*`transaction_flag`*

Specifies the JMS Asynchronous Service Transactions flag.

`wlai.numAsyncServiceRequestProcessors=`*`integer`*

Specifies the number of asynchronous service request processors. This property is optional.

Example:

```
wlai.numAsyncServiceRequestProcessors=4
```

`wlai.forceGCIntervalSeconds=`*`seconds`*

Forces garbage collection at the specified interval (in seconds). This property is optional.

Example:

```
wlai.forceGCIntervalSeconds=60
```

`wlai.repositoryDatasourceName=`*`datasource_name`*
`wlai.repositoryPassword=`*`password`*
`wlai.repositoryUsername=`*`username`*

Specifies the database repository to be used by application integration.

Example:

```
wlai.repositoryDatasourceName=WLAI_DataSource
wlai.repositoryPassword=security
wlai.repositoryUsername=system
```

`wlai.logLevel=`*`valid_logging_level`*

Specifies the logging level for the Log Manager. This property is optional.

Example:

```
wlai.logLevel=warning
```

The following figure shows a sample of the `wlai.properties` file that is included in the `wlidomain` domain.

Figure A-18 wlai.properties File

```
wlai.appView.deploy=WLISAppView.sav
wlai.deploymentRepositoryRootPath=C:/bea/wlintegration2.1/config/wlidomain/wlai/deploy
wlai.hostNameAndPort=localhost:7001
wlai.hostPassword=security
wlai.hostUserID=system
wlai.jms.connectionFactoryJNDIName=com.bea.wlai.JMSConnectionFactory
wlai.jms.serverName=WLAI_JMSServer
wlai.repositoryDatasourceName=WLAI_DataSource
wlai.repositoryPassword=security
wlai.repositoryUsername=system
```

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 . . .
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 FormatBuilder	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 2.0 to 2.1
PIMigrator	Migrate a workflow template
runJavadocs	Open the API specification home page
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
SetEaiDomainData SetSampleData SetWliDomainData	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
startcloudview	Start the Cloudscape 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
worklist	Start the WebLogic Integration Worklist
wliconfig	Launch the WebLogic Integration database configuration wizard

Environment Variables

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

Table B-2 WebLogic Integration Environment Variables

Variable	Description
CLASSPATH	The classpath invoked by the `mdbgenerator` command.
CP	The classpath set by the `Set`*`DomainName`*`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`*`DomainName`*`Data` command. For example, in the `SetSampleData` command, `CMNCP` is set as follows: `CMNCP=$WLI_SAMPLES_HOME/lib/wlissample.jar:$WLI_SAMPLES_HOME/lib/edisampledeploy.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 `setdbdata` and `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=$DBINFO/oracle See the description of the DBINFO environment variable (in this table) and the setdbdata command.
DB_DRIVER	The JDBC driver required to connect to the configured database. For example, if the current domain is configured to use an Oracle database, then DB_DRIVER is set as follows: DB_DRIVER="oracle.jdbc.driver.OracleDriver" See the bulkloader, migratedb, setdbdata, and setDBVars commands.
DB_JVMARGS	When the domain is configured to use a Cloudscape database, this variable is set as follows: DB_JVMARGS="-Djava.compiler=NONE -Dcloudscape.system.home=$CLOUDSCAPE_HOME" If a Microsoft SQL, Sybase, or Oracle database is configured, DB_JVMARGS is not set. See the bulkloader, migratedb, startWeblogic, stopWeblogic, setdbdata, setDBVars, setupAppView, and setupEDIAppView commands.
DB_PASSWORD	The password required to connect to the database configured for the current domain. See the bulkloader, migratedb, setdbdata, 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 Cloudscape database the following setting is used: DB_SCHEMA="APP" 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, setdbdata, and setDBVars commands.
DB_SERVERNAME	The Microsoft SQL server name. See the %WLI_DOMAIN\dbInfo\mssql\setDBVars command.
DB_TYPE	The type of database for the current domain. Valid values are: oracle, mssql, sybase, and cloudscape. **Note:** cloudscape is not a supported option on UNIX systems. See the createdb, 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 configuration 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, setdbdata, and setDBVars commands.
DB_USER	The user name required to connect to the database configured for the current domain. See the bulkloader, migratedb, setdbdata, and setDBVars commands.
DBINFO	The location of the dbInfo directory for the current domain. This variable is set as follows: On UNIX: DBINFO=$WLI_DOMAIN/dbInfo On Windows DBINFO=%WLI_DOMAIN%\dbInfo See the setdbdata, setdomain, switchdb, setDBVars, and setDBVarsExt commands.
EVENROUTER_URL	The URL for the event router for the domain. For example: EVENTROUTER_URL="http://$WLIS_HOST_AND_PORT/DbmsEventRouter/EventRouter" See the Set*DomainName*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 setenv, bulkmigrator, and PIMigrator commands.
JAVA_BIN	The JDK bin directory. For example, if you install WebLogic Server in the default location on a Solaris system, JAVA_BIN is set as follows: JAVA_BIN=/home/mydir/bea/jdk131/bin See the setenv command.

Table B-2 WebLogic Integration Environment Variables (Continued)

Variable	Description
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 bulkloader, fb, mdbgenerator, migratedb, setenv, startcloudview, studio, switchdb, wliconfig, worklist, startWeblogic, stopWeblogic, setupAppView, setupEDIAppView, and RunSamples commands.
JAVA_OPTIONS	Operating system specific Java options. For example, on a Solaris system, JAVA_OPTIONS is set as follows: JAVA_OPTIONS="-hotspot -Xms64m -Xmx256m" See the setenv, startWeblogic, bulkloader, bulkmigrator, migratedb, and PIMigrator commands.
JAVAC	The location of the java compiler. For example, if you install WebLogic Server in the default location on a Solaris system, JAVAC is set as follows: JAVA=$JAVA_BIN/javac See the setenv command.
JAVACLASSPATH	The classpath variable invoked by the bulkmigrator and PIMigrator commands.
JDBC_DATASOURCE_JNDI_NAME	The JDBC data source JNDI name for the domain. For example: JDBC_DATASOURCE_JNDI_NAME=WLAI_DataSource See the Set*DomainName*Data and setupAppView commands.
SQLARGS	The set of arguments required by a Microsoft SQL or Sybase database. For example, if a Sybase database has been configured for the current domain, SQLARGS is set as follows: SQLARGS="-b -e -n -S *servername* -U *username* -P \"*password*\"" See the setDBVars, createmssql, and createsybase commands.
SVRCP	Server classpath set by the Set*DomainName*Data command. For example, when the SetSampleData command is run, SVRCP is set as follows: SVRCP=$WLISERVERCP:$CMNCP

Table B-2 WebLogic Integration Environment Variables (Continued)

Variable	Description
WL_HOME	The WebLogic Server installation directory. For example, if you install the product in the default location on a Windows system, WL_HOME is set as follows: WL_HOME=C:\bea\wlserver6.1 See the mdbgenerator, setenv, startcloudview, startWeblogic, stopWeblogic, setDBVars, and RunSamples commands.
WLI_DOMAIN	The currently set domain. See the checkdomain, createdb, migratedb, setdbdata, setdomain, domainname, showdomain, switchdb, Set*DomainName*Data, and startWeblogic commands.
WLI_HOME	The WebLogic Integration installation directory. For example, if you install the product in the default location on a Windows system, WLI_HOME is set as follows: WLI_HOME=C:\bea\wlintegration2.1. See the mdbgenerator, setenv, startcloudview, startWeblogic, stopWeblogic, setDBVars, setupAppView, setupEDIAppView, and RunSamples commands.
WLI_SAMPLES_HOME	The root of the domain. For example: WLI_SAMPLES_HOME=$WLI_HOME/config/samples See the Set*DomainName*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, studio, switchdb, worklist, Set*DomainName*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*DomainName*Data, setupAppView, and setupEDIAppView commands.
WLIS_PASSWORD	The WebLogic Server password for the domain. For example: WLIS_USER_NAME=system See the Set*DomainName*Data, setupAppView, and setupEDIAppView commands.

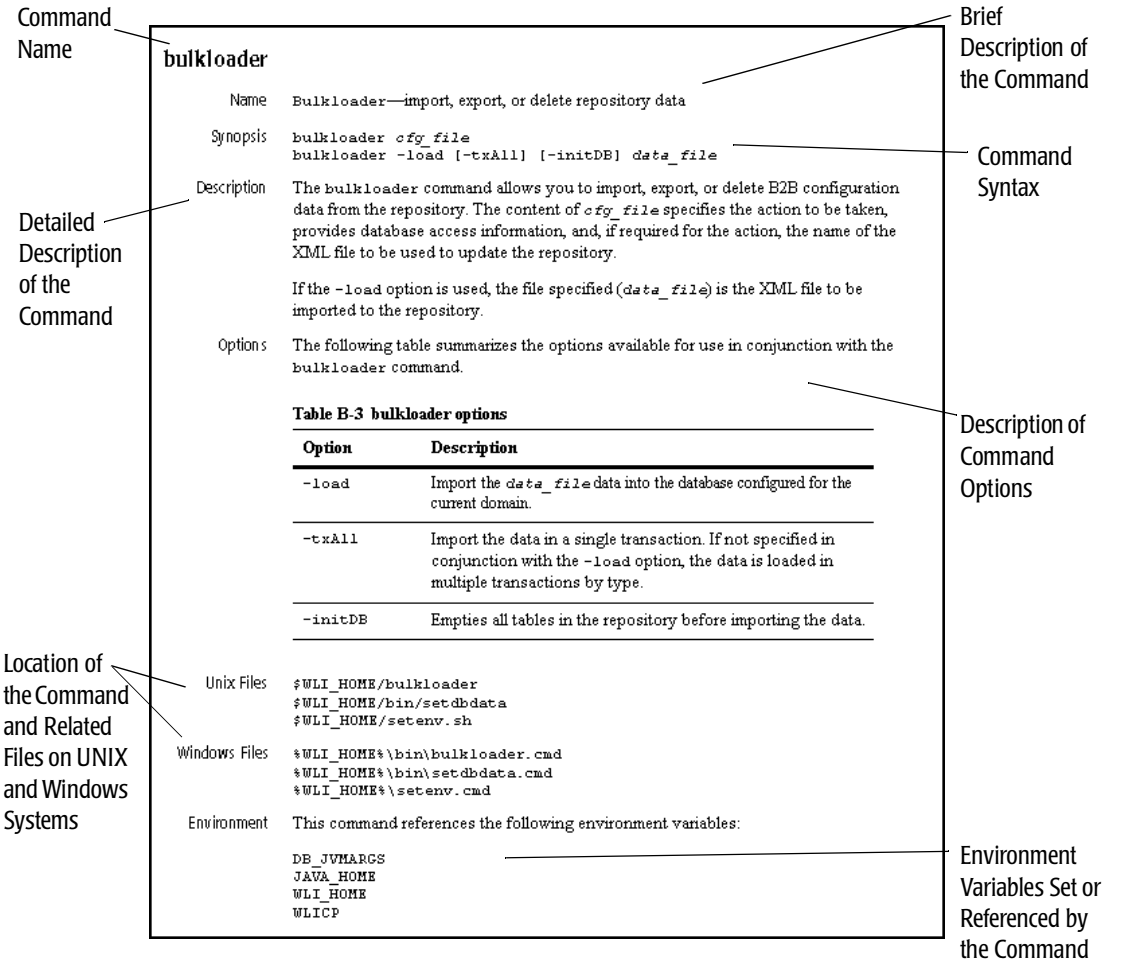
Table B-2 WebLogic Integration Environment Variables (Continued)

Variable	Description
WLIS_USER_NAME	The WebLogic Server username for the domain. For example: WLIS_PASSWORD=security See the Set*DomainName*Data, setupAppView, and setupEDIAppView commands.
WLISERVERCP	WebLogic Integration server classpath. See the setenv, Set*DomainName*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



bulkloader

Name

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

Synopsis

```
bulkloader cfg_file
bulkloader -load [-txAll] [-initDB] 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*.

Options

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

Table B-3 bulkloader Options

Option	Description
`-load`	Import the *`data_file`* data into the database configured for the current domain.
`-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.

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 2.1.

For detailed instructions, see "Migrating from WebLogic Collaborate 1.0/1.0.1" in *Migrating to BEA WebLogic Integration Release 2.1*.

Options None.

UNIX Files `$WLI_HOME/bin/bulkmigrator`

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

Environment This command references the following environment variables:

```
JAVA
JAVA_OPTIONS
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
$WLI_HOME/config/eaidomain/checkdomain
$WLI_HOME/config/wlidomain/checkdomain
$WLI_HOME/bin/domainname
$WLI_HOME/setenv.sh
```

Windows Files

```
%WLI_HOME%\bin\checkdomain.cmd
%WLI_HOME%\config\eaidomain\checkdomain.cmd
%WLI_HOME%\config\wlidomain\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.

The tables created are dependent on the requirements of the current domain. (For a description of the required tables, see Table 1-4 in "Domain Configuration Requirements" on page 1-10.) For example, if the current domain is the `config/bpmdomain` domain, then the B2B tables are not created, and the B2B system data is not loaded. If the current domain is the `config/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-4 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
$WLI_HOME/dbscripts/$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
$WLI_HOME/samples/wlis/dbscripts/$DB_TYPE/WLIS_SCHEMA.sql
$WLI_HOME/adapters/dbms/src/sql/CreateCloudscapeEventTables.sql
$WLI_HOME/adapters/dbms/src/sql/CreateSybaseEventTables.sql
$WLI_HOME/adapters/dbms/src/sql/CreateOracleEventTables.sql
$WLI_HOME/adapters/dbms/src/sql/CreateMSSQLServerEventTables.sql
$WLI_HOME/adapters/dbms/src/sql/CreateCloudscapeCustomer_Table.sql
```

```
$WLI_HOME/adapters/dbms/src/sql/CreateSybaseCustomer_Table.sql
$WLI_HOME/adapters/dbms/src/sql/CreateOracleCustomer_Table.sql
$WLI_HOME/adapters/dbms/src/sql/CreateMSSQLServerCustomer_Table.sql
$WLI_HOME/dbscripts/SystemRepData.xml
$WLI_HOME/setenv.sh
```

Windows Files

```
%WLI_HOME%\bin\createDB.cmd
%WLI_HOME%\bin\setdbdata.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
%WLI_HOME%\samples\wlis\dbscripts\%DB_TYPE%\WLIS_SCHEMA.sql
%WLI_HOME%\adapters\dbms\src\sql\CreateCloudscapeEventTables.sql
%WLI_HOME%\adapters\dbms\src\sql\CreateSybaseEventTables.sql
%WLI_HOME%\adapters\dbms\src\sql\CreateOracleEventTables.sql
%WLI_HOME%\adapters\dbms\src\sql\CreateMSSQLServerEventTables.sql
%WLI_HOME%\adapters\dbms\src\sql\CreateCloudscapeCustomer_Table.sql
%WLI_HOME%\adapters\dbms\src\sql\CreateSybaseCustomer_Table.sql
%WLI_HOME%\adapters\dbms\src\sql\CreateOracleCustomer_Table.sql
%WLI_HOME%\adapters\dbms\src\sql\CreateMSSQLServerCustomer_Table.sql
%WLI_HOME%\dbscripts\SystemRepData.xml
%WLI_HOME%\setenv.cmd
```

Environment

The `createdb` command references the following environment variables:

```
WLI_HOME
DB_TYPE
WLI_DOMAIN
DB_CREATE
DB_SCRIPT_DIR
DB_COMMAND
```

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:

```
eval %DB_CREATE%%DB_SCRIPT_DIR%/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
FormatBuilder

Name

`fb`—start Format Builder
`FormatBuilder`—start Format Builder

Synopsis

```
fb
FormatBuilder
```

Description

This command starts the Format Builder client. A shortcut to the `FormatBuilder.exe` 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%\bin\FormatBuilder.exe
%WLI_HOME%\setenv.cmd
```

Environment

The `fb` command sets the `CP` environment variable as follows:

```
CP=$WLICP:$WLI_HOME/samples/di/fml:$WLI_HOME/lib/wlxtfb.jar:$WLI_
HOME/lib/jamdmd42.jar:$WLI_HOME/ext
```

The fb command references the following environment variables:

```
CP
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 `$WLI_HOME/samples/bin/launchSamplesPage`

Windows Files `%WLI_HOME%\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 `createdb` command.

Table B-5 mdbgenerator Options

Option	Description
`-queue` *`queue_name`*	Name of the custom queue for which you want to generate the MDB. This argument 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 argument 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:

```
WLPI_HOME
WL_HOME
JAVA_HOME
```

This command attempts to execute commands that do not exist in WebLogic Integration 2.1. To address this, comment out the following:

```
. ../config/bpmdomain/setEnv.sh
. ../config/bpmdomain/checkEnv.sh
```

The `WLPI_Home` environment variable is a legacy from the WebLogic Integration 2.0 release and may not be set. Before you execute the `mdbgenerator` command, execute the `setenv` command and set `WLPI_HOME=$WLI_HOME`.

migratedb

Name `migratedb`—migrate a database from WebLogic Integration Release 2.0 to Release 2.1

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 2.0 database tables and schema as required for compatibility with WebLogic Integration 2.1.

This command is invoked when you run the WebLogic Integration database configuration wizard (the `wliconfig` command) and select the migrate option.

For additional information see *Migrating to BEA WebLogic Integration Release 2.1*.

Options None.

UNIX Files

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

Windows Files

```
%WLI_HOME%\bin\migratedb.cmd
%WLI_HOME%\setenv.cmd
```

Environment This command references the following environment variables:

```
DB_TYPE
JAVA_HOME
WLI_HOME
```

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
JAVA_OPTIONS
WLC_HOME
JAVACLASSPATH
WLCCLASSPATH
```

runJavadocs

Name `runJavadocs`—open the API specification home page

Synopsis `runJavadocs`

Description This command opens the WebLogic Integration 2.1 API specification home page (`%WLI_HOME%\docs\apidocs\index.html`) in your default browser.

Options None.

UNIX Files Not available on UNIX.

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

Environment No environment variables are referenced or set.

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 1-4 in "Domain Configuration Requirements" on page 1-10.
- 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-6 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-6 RunSamples Options (Continued)

Option	Description
`StartServer`	Start the sample server

UNIX Files

```
$WLI_HOME/samples/bin/RunSamples
$WLI_HOME/bin/setdomain
$WLI_HOME/samples/bin/SetSampleData
```

Windows Files

```
%WLI_HOME%\samples\bin\RunSamples.cmd
%WLI_HOME%\bin\setdomain.cmd
%WLI_HOME%\samples\bin\SetSampleData.cmd
```

Environment

This command references the following environment variables:

```
$WLI_HOME
$WLICP
$WL_HOME
$JAVA_HOME
```

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 install the product, execute the `switchdb` command, or use the WebLogic Integration database configuration 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/config/eaidomain/setdbdata
$WLI_HOME/config/wlidomain/setdbdata
$WLI_HOME/bin/checkdomain
$WLI_HOME/config/eaidomain/checkdomain
$WLI_HOME/config/wlidomain/checkdomain
$WLI_HOME/config/wlidomain/dbInfo/$DB_TYPE/setDBVars
$WLI_HOME/config/wlidomain/dbInfo/$DB_TYPE/setDBVarsExt
$WLI_HOME/config/wlidomain/dbInfo/setdbtype
```

Windows Files

```
%WLI_HOME%\bin\setdbdata.cmd
%WLI_HOME%\config\eaidomain\setdbdata.cmd
%WLI_HOME%\config\wlidomain\setdbdata.cmd
%WLI_HOME%\bin\checkdomain.cmd
%WLI_HOME%\config\eaidomain\checkdomain.cmd
%WLI_HOME%\config\wlidomain\checkdomain.cmd
%WLI_HOME%\config\wlidomain\dbInfo\%DB_TYPE%\setDBVars
%WLI_HOME%\config\wlidomain\dbInfo\%DB_TYPE%\setDBVarsExt
%WLI_HOME%\config\wlidomain\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 install the product, execute the `switchdb` command, or use the WebLogic Integration database configuration wizard (the `wliconfig` command) to update the database for a domain.

Options None.

UNIX Files
```
$WLI_HOME/config/eaidomain/dbInfo/setdbtype
$WLI_HOME/config/samples/dbInfo/setdbtype
$WLI_HOME/config/wlidomain/dbInfo/setdbtype
```

Windows Files
```
%WLI_HOME%\config\eaidomain\dbInfo\setdbtype.cmd
%WLI_HOME%\config\samples\dbInfo\setdbtype.cmd
%WLI_HOME%\config\wlidomain\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 install the product, execute the `switchdb` command, or use the WebLogic Integration database configuration 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 Cloudscape only.

setdomain

Name setdomain—set the current domain

Synopsis setdomain *domain*

Here, *domain* 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 $WLI_HOME directory to set the current domain to wlidomain, enter the following:

```
setdomain config/wlidomain
```

If you execute the command from the domain directory (for example, from the $WLI_HOME/config/wlidomain directory), you need not specify *domain*.

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:

```
JAVA
JAVAC
JAVA_BIN
JAVA_HOME
JAVA_OPTIONS
WL_HOME
WLI_HOME
WLICOMMONCP
WLICP
WLISERVERCP
```

The command also adds `WLI_HOME` and `JAVA_BIN` to the existing `PATH` environment variable.

SetEaiDomainData
SetSampleData
SetWliDomainData

Name

`SetEaiDomainData`—set environment variables for the `eaidomain` domain
`SetSampleData`—set environment variables for the `samples` domain
`SetWliDomainData`—set environment variables for the `wlidomain` domain

Synopsis

```
SetEaiDomainData
SetSampleData
SetWliDomainData
```

Description

The `Set`*`DomainName`*`Data` command sets environment variables for the domain.

Options

None.

UNIX Files

```
$WLI_HOME/config/eaidomain/SetEaiDomainData
$WLI_HOME/config/wlidomain/SetWliDomainData
$WLI_HOME/samples/bin/SetSampleData
$WLI_HOME/setenv.sh
$WLI_HOME/bin/setdbdata
```

Windows Files

```
%WLI_HOME%\config\eaidomain\SetEaiDomain.cmd
%WLI_HOME%\config\wlidomain\SetWliDomain.cmd
%WLI_HOME%\samples\bin\SetSampleData.cmd
%WLI_HOME%\setenv.cmd
%WLI_HOME%\setdbdata.cmd
```

Environment

The `Set`*`DomainName`*`Data` command sets the following environment variables for the `samples`, `eaidomain`, and `wlidomain` domains:

```
WLI_SAMPLES_HOME
WLIS_HOST_AND_PORT
WLIS_USER_NAME
WLIS_PASSWORD
JDBC_DATASOURCE_JNDI_NAME
EVENROUTER_URL
CMNCP
CP
SVRCP
```

The `Set`*`DomainName`*`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

```
$WLI_HOME/samples/bin/setupAppView
$WLI_HOME/samples/bin/setupEDIAppView
$WLI_HOME/samples/bin/SetSampleData
$WLI_HOME/setenv.sh
```

Windows Files

```
%WLI_HOME%\samples\bin\setupAppView.cmd
%WLI_HOME%\samples\bin\setupEDIAppView.cmd
%WLI_HOME%\setenv.cmd
```

Environment

The `setupAppView` and the `setupEDIAppView` commands reference the following environment variables:

```
WLI_HOME
JAVA_HOME
DB_JVMARGS
CP
WLIS_HOST_AND_PORT
WLIS_PASSWORD
WLIS_USER_NAME
```

In addition to the above, 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.

startcloudview

Name `startcloudview`—start the Cloudscape administrative tool

Synopsis

```
startcloudview
```

Description The `startcloudview` command launches Cloudview, the Cloudscape administrative tool.

Options None.

UNIX Files

```
$WLI_HOME/bin/startcloudview
```

Windows Files

```
%WLI_HOME%\bin\startcloudview.cmd
```

Environment This command references the following environment variables:

```
JAVA_HOME
WL_HOME
```

You must execute the `setenv` command before executing the `startcloudview` command.

startWeblogic

Name `startWeblogic`—start WebLogic Server and deploy WebLogic Integration resources

Synopsis `startWeblogic`

Description The `startWeblogic` command:

- Executes the `setenv` and `Set`*`DomainName`*`Data` commands to set the domain environment
- Removes old log files.
- When executed from the `config/wlidomain` or the `config/samples` domain directory, this command creates the application integration deployment (*`domain_name`*`/wlai`) and WebLogic Integration sample data (*`domain_name`*`/data`) directories, if they do 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

```
$WLI_HOME/config/bpmdomain/startWeblogic
$WLI_HOME/config/eaidomain/startWeblogic
$WLI_HOME/config/samples/startWeblogic
$WLI_HOME/config/wlidomain/startWeblogic
$WLI_HOME/config/wlidomain/SetWliDomainData
$WLI_HOME/config/eaidomain/SetEaiDomainData
$WLI_HOME/samples/bin/SetSampleData
$WLI_HOME/setenv.sh
```

Windows Files

```
%WLI_HOME%\config\bpmdomain\startWeblogic.cmd
%WLI_HOME%\config\eaidomain\startWeblogic.cmd
%WLI_HOME%\config\samples\startWeblogic.cmd
%WLI_HOME%\config\wlidomain\startWeblogic.cmd
%WLI_HOME%\config\wlidomain\SetWliDomainData.cmd
%WLI_HOME%\config\eaidomain\SetEaiDomainData.cmd
%WLI_HOME%\samples\bin\SetSampleData.cmd
%WLI_HOME%\setenv.cmd
```

Environment

The `startWeblogic` command references the following environment variables:

```
BEA_HOME
DB_JVMARGS
JAVA_HOME
JAVA_OPTIONS
WL_HOME
WLI_DOMAIN_HOME
WLI_HOME
SVRCP
```

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

```
$WLI_HOME/config/bpmdomain/stopWeblogic
$WLI_HOME/config/eaidomain/stopWeblogic
$WLI_HOME/config/samples/stopWeblogic
$WLI_HOME/config/wlidomain/stopWeblogic
$WLI_HOME/setenv.sh
```

Windows Files

```
%WLI_HOME%\config\bpmdomain\stopWeblogic.cmd
%WLI_HOME%\config\eaidomain\stopWeblogic.cmd
%WLI_HOME%\config\samples\stopWeblogic.cmd
%WLI_HOME%\config\wlidomain\stopWeblogic.cmd
%WLI_HOME%\setenv.cmd
```

Environment The `stopWeblogic` command references the following environment variables:

```
BEA_HOME
DB_JVMARGS
JAVA_HOME
WLISERVERCP
WL_HOME
WLI_HOME
```

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|cloudscape|sybase
```

Note: `cloudscape` is not a supported option on UNIX systems.

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 and RDBMS 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 configuration 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.

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 configuration wizard

Synopsis

```
wliconfig
```

Description This command launches the WebLogic Integration database configuration wizard. For additional information, see "Using the Database Configuration Wizard" on page 3-6.

Options No options are currently supported. The wizard prompts for the required information.

UNIX Files

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

Windows Files

```
%WLI_HOME%\bin\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

T

U

V

W

X