Oracle® Fusion Middleware

Domain Template Reference

11*g* Release 1 (10.3.1)

E14138-01

May 2009

This document provides information about templates – Java Archive (JAR) files that contain the files and scripts required to create or extend a domain.

Oracle Fusion Middleware Domain Template Reference, 11*g* Release 1 (10.3.1)

E14138-01

Contents

List of Tables

Preface

This preface describes the document accessibility features and conventions used in this guide–*Oracle Fusion Middleware Domain Template Reference*.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at `http://www.oracle.com/accessibility/`.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at `http://www.fcc.gov/cgb/consumerfacts/trs.html`, and a list of phone numbers is available at `http://www.fcc.gov/cgb/dro/trsphonebk.html`.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 Introduction

This document provides information about templates, Java Archive (JAR) files that contain the files and scripts required to create or extend a domain.

This document contains the following topics:

1.1 Types of Templates

The types of template include:

- *Domain template*—defines the full set of resources within a domain, including infrastructure components, applications, services, security options, and general environment and operating system options.

  The product installation includes a predefined Basic WebLogic Server Domain template. This template defines the core set of resources within a domain, including an Administration Server and basic configuration information. For more information on Basic WebLogic Server Domain template, see Section 2.1, "Basic WebLogic Server Domain Template."

  You can also create a custom domain template from an existing domain by using the Domain Template Builder or the `pack` command. By using the Domain Template Builder, you can also create a custom domain template from an existing template.

- *Extension template*—defines the applications and services that you can add to an existing domain, including product component functionality and resources such as JDBC or JMS.

  The product installation includes several predefined extension templates. For a summary of extension templates, see Section 1.4, "Template Summary."

  You can also create a custom extension template from an existing domain or template using the Domain Template Builder.

- *Managed Server template* – defines the subset of resources within a domain that are required to create a Managed Server domain directory on a remote machine.

  You can create a custom Managed Server template by using the `pack` command. For more information, see *Oracle Fusion Middleware Creating Templates and Domains Using the Pack and Unpack Commands*.

1.2 Location of Installed Templates

The following table identifies the location of the predefined templates provided with your product installation, where *`WL_HOME`* represents the product installation directory.

Table 1–1 Location of Templates

Type of Template	Directory Location
Domain	*`WL_HOME`*`\common\templates\domains`
Extension	*`WL_HOME`*`\common\templates\applications`

1.3 Template Tools

The following table identifies the tools with which you can create templates and the tools with which you can use templates to create or extend a domain.

Table 1–2 Template Tools

To	Use this tool
Create a domain	■ Configuration Wizard ■ WLST Offline ■ `unpack` command
Extend an existing domain	■ Configuration Wizard ■ WLST Offline
Create a managed server domain on a remote machine	`unpack` command
Create a domain template	■ Domain Template Builder ■ `pack` command ■ WLST Offline
Create an extension template	Domain Template Builder
Create a Managed Server template	`pack` command

> **Note:** All the tools used to create or extend a domain leverage a common underlying infrastructure, referred to as the Configuration Wizard framework.

- For information about using the Configuration Wizard, see *Oracle Fusion Middleware Creating Domains Using the Configuration Wizard*.
- For information about using the WLST Offline, see *Oracle Fusion Middleware Oracle WebLogic Scripting Tool*.

- For information about using the `pack/unpack` commands, see *Oracle Fusion Middleware Creating Templates and Domains Using the Pack and Unpack Commands*.
- For information about using the Domain Template Builder, see *Oracle Fusion Middleware Creating Domain Templates Using the Domain Template Builder*.

1.4 Template Summary

The following tables summarizes the predefined templates that may be provided in your product installation.

Table 1–3 Summary of Oracle WebLogic Server and Workshop for WebLogic Templates

Template	File name	Description
Basic WebLogic Server Domain Template (Domain Template)	wls.jar	Creates a base WebLogic Server domain.
WebLogic Server Starter Domain Template (Domain Template)	wls_starter.jar	Creates a WebLogic Server starter domain.
WebLogic Beehive Extension Template	weblogic-beehive.jar	Extends the base WebLogic Server domain to create a WebLogic Beehive domain. Adds required Beehive libraries to support run-time use of controls. **Note**: Resources from the WebLogic Advanced Web Services Extension template are required to create a complete WebLogic Beehive domain.
WebLogic Advanced Web Services Extension Template	wls_webservice.jar	Extends an existing WebLogic Server domain to add functionality required for advanced Web Services, including WSRM, Buffering, and JMS Transport.
Avitek Medical Records Sample Domain Template	medrec.jar	Extends the Basic WebLogic Server domain to create the Avitek Medical Records sample domain. This domain is a WebLogic Server sample application suite that demonstrates all aspects of the J2EE platform.
Avitek Medical Records Sample Domain Template (Spring Version)	medrec-spring.jar	Extends the Basic WebLogic Server domain to create the Avitek Medical Records sample domain for Spring. This domain is a WebLogic Server sample application suite that demonstrates all aspects of the J2EE platform.
Oracle Workshop for WebLogic Extension Template	workshop_wl.jar	Extends the Basic WebLogic Server domain to create Oracle Workshop for WebLogic domain.
Workshop for WebLogic 10.3 Extension Template	workshop_wl_10_3.jar	Extends the Basic WebLogic Server domain to create Workshop for WebLogic 10.3 domain.

Table 1–3 (Cont.) Summary of Oracle WebLogic Server and Workshop for WebLogic

Template	File name	Description
WebLogic Server Default Domain Extension Template	wls_default.jar	Extends the Basic WebLogic Server domain with a web application designed to guide new users through an introduction to WebLogic Server. When running the web application, users can review informative content on various topics, including highlights of WebLogic Server functionality. From the web application, users can also run several preconfigured, precompiled examples. Resources from this extension template are required for a WebLogic Server Examples domain.
WebLogic Server Examples Extension Template	wls_examples.jar	Extends the WebLogic Server domain containing resources from the base WebLogic Server domain template and the WebLogic Server Default Domain extension template to create a complete WebLogic Server Examples domain. The WebLogic Server Examples domain contains a collection of examples that illustrate best practices for coding individual J2EE and WebLogic Server APIs.
WebLogic Personalization Extension	p13n.jar	Extends an existing WebLogic Server domain to add Weblogic Personalization functionality.
WebLogic Content Extension	content.jar	Extends an existing WebLogic Server domain to add WebLogic Content Management functionality.

1.5 Relationships Between Templates

This section discusses the following topics:

- Section 1.5.1, "WebLogic Server Resources as a Prerequisite"
- Section 1.5.2, "Relationships Between Templates"

1.5.1 WebLogic Server Resources as a Prerequisite

WebLogic Server resources must be set up in your domain before you can add resources from an extension template. When you select an extension template, the Configuration Wizard framework checks to make sure the required resources are available for you.

1.5.2 Relationships Between Templates

You can create a base WebLogic domain by using the predefined Basic WebLogic Server domain template, or you can create a Basic WebLogic domain and extend it incrementally using the extension templates. The following table shows the relationships between the templates and the domains created.

Table 1–4 Relationships Between Templates

This type of domain	Requires resources from these templates
Avitek Medical Records Sample	Basic WebLogic Server Domain template, wls.jar + Avitek Medical Records Sample Domain extension template, medrec.jar
WebLogic Server (base)	Basic WebLogic Server Domain template, wls.jar
WebLogic Server (starter)	WebLogic Server Starter Domain template, wls_starter.jar

Table 1–4 (Cont.) Relationships Between Templates

This type of domain	Requires resources from these templates
WebLogic Server Default	Basic WebLogic Server Domain template, wls.jar + WebLogic Server Default Domain extension template, wls_default.jar
WebLogic Server Examples	Basic WebLogic Server Domain template, wls.jar + WebLogic Server Default Domain extension template, wls_default.jar + WebLogic Server Examples extension template, wls_examples.jar
WebLogic Advanced Web Services Extension	Basic WebLogic Server Domain, wls.jar + WebLogic Advanced Web Services Extension, wls_webservice.jar
WebLogic Beehive Extension	Basic WebLogic Server Domain, wls.jar + WebLogic Advanced Web Services Extension, wls_webservice.jar + WebLogic Beehive Extension, weblogic_beehive.jar
Oracle Workshop for WebLogic	Basic WebLogic Server Domain, wls.jar + Advanced Web Services Extension, wls_webservice.jar + Oracle Workshop for WebLogic Extension, workshop_wl.jar
Workshop for WebLogic 10.3	Basic WebLogic Server Domain, wls.jar + Advanced Web Services Extension, wls_webservice.jar + Workshop for WebLogic 10.3, workshop_wl_10_3.jar

1.6 Files Typically Included in a Template

The basic files included in any template are `config.xml` and `template-info.xml`. There are additional files in the predefined templates, and a domain is created or extended based on these files. The following table describes the files typically included in a domain or extension template.

Table 1–5 Files Included in a Template

Filename	Description
product component files	Various files used to complete the domain setup for a specific Oracle product component. Such files may provide information for security and default database settings.
*-jdbc.xml	Sets up or extends a domain with JDBC system resources required by a product component. In a template, the *-jdbc.xml files must be located in the config\jdbc directory.
*-jms.xml	Sets up or extends a domain with JMS system resources required by a product component. In a template, the *-jms.xml files must be located in the config\jms directory.

Table 1–5 (Cont.) Files Included in a Template

Filename	Description
clusters.script	Used to modify the Configuration Wizard framework's default auto-configuration of a cluster. By default, resources are targeted to the cluster. You can unassign a resource from the cluster and then assign it to another component. To specify a target, you can use the following replacement variables: ■ %AManagedServer% — Any Managed Server ■ %AllManagedServers% — Comma-separated list of all Managed Servers ■ %AdminServer% — Administration Server name ■ %Cluster% — Cluster name ■ %ProxyServer% — Proxy server name ■ %HTTPProxyApp% — http proxy application definition Note the following additional considerations: ■ You must use the name attribute of an object that is to be replaced. ■ You can use an asterisk (*) as a wildcard for "All." In a template, the clusters.script file must be located in the script directory.
config.xml	Sets up or extends the domain configuration. In a template, the config.xml file must be located in the config directory.
jdbc.index	Identifies the locations of SQL scripts used to set up a database. The file lists the scripts in the order in which they must be run. If the scripts are not contained in the template, but are located in the product installation directory, that directory can be represented by a tilde (~) in the pathname for the scripts, as shown in the following example: `~/integration/common/dbscripts/oracle/reporting_runtime.sql` Specifically, the tilde represents the directory path identified by the $USER_INSTALL_DIR$ variable in the stringsubs.xml file. In a template, a jdbc.index file must be located in the _jdbc_\dbtype\dbversion directory, where dbtype is the type of database, such as Oracle, and dbversion is the database version, such as 9i. In addition to listing the SQL files related to a data source, the jdbc.index file contains information about the categories associated with the data source. The default dbCategories that are available are: ■ 'Drop/Create P13N Database Objects' category associated with the p13nDataSource data source, which is a part of the p13n.jar domain template ■ 'Drop/Create Portal Database Objects' category associated with the "p13nDataSource" data source, which is a part of the wlp.jar domain template ■ 'Drop/Create GroupSpace Database Objects' category associated with the appsGroupSpaceDataSource data source, which is a part of the wlp_groupspacedb.jar domain template All these template jar files are located in the MW_HOME\wlserver_<version>\common\templates\applications directory.
security.xml	Used to create user groups and roles that establish identity and access to domain resources. You can create the default Admin user only through the security.xml in a *domain* template. However, you can create user groups and roles through the security.xml included in either a domain or an extension template.

Table 1–5 (Cont.) Files Included in a Template

Filename	Description
startmenu.xml	Used to create Windows start menu entries.
startscript.xml	Used to create the *.cmd and *.sh files that are placed into a domain's root and bin directories.
stringsubs.xml	Identifies string substitution values and files that will receive string substitutions during domain creation or extension. The files that will receive string substitutions must already be prepared with replacement variables. During domain creation or extension, the Configuration Wizard framework runs macros to replace variables with the appropriate string substitution, using information from WL_HOME\common\lib\macrorules.xml, where WL_HOME is the product installation directory.
template-info.xml	Provides template identification information, such as the template name, software version, type of template (domain or application), author, description, and so on.

2 Templates

This chapter discusses the following topics:

- Section 2.1, "Basic WebLogic Server Domain Template"
- Section 2.2, "WebLogic Server Starter Domain Template"
- Section 2.4, "WebLogic Advanced Web Services Extension Template"
- Section 2.5, "Avitek Medical Records Sample Domain Template"
- Section 2.6, "Avitek Medical Records Sample Domain Template (Spring Version)"
- Section 2.7, "Oracle Workshop for WebLogic Extension Template"
- Section 2.8, "Workshop for WebLogic 10.3 Extension Template"
- Section 2.9, "WebLogic Server Default Domain Extension Template"
- Section 2.10, "WebLogic Server Examples Extension Template"

2.1 Basic WebLogic Server Domain Template

Your product installation provides one predefined Basic WebLogic Server domain template. All other predefined templates are extension templates that you may use to add resources, services, and applications to a Basic WebLogic Server domain. You can easily create or extend a domain by using these predefined templates with the Configuration Wizard or WLST.

2.1.1 Generated Domain Output

The Basic WebLogic Server Domain template enables you to create a simple WebLogic Server domain. By default, when using the Basic WebLogic Server Domain template, you generate a domain that contains only the required components: an Administration Server and a single administrative user. Any required applications must be created and configured within the domain.

The following table defines the default directory structure and files generated by the Basic WebLogic Server Domain template. Unless otherwise specified, by default, the Configuration Wizard framework creates the domain in the *`MW_HOME`*`\user_projects\domains\base_domain` directory. If you modify the default configuration settings, the output directory structure may be different from the structure described here.

Table 2–1 Output Generated from the Basic WebLogic Server Domain Template

Directory	File/s	Description
user_projects\applications\base_domain\	n.a	Directory designated as the repository for any custom application files that you create.
user_projects\domains\base_domain\	fileRealm.properties	File containing ACLs, users, and groups that can be used for the default security realm when Compatibility security is used.
user_projects\domains\base_domain\	startWebLogic.cmd startWebLogic.sh	Scripts used to start the Administration Server on Windows and UNIX systems, respectively.
user_projects\domains\base_domain\autodeploy\	readme.txt	File providing information about the directory, which initially serves as a placeholder for automatic deployments.
user_projects\domains\base_domain\bin\	setDomainEnv.cmd setDomainEnv.sh	Scripts used to set up the development environment on Windows and UNIX systems, respectively.
user_projects\domains\base_domain\bin\	startManagedWebLogic.cmd startManagedWebLogic.sh	Scripts used to start a Managed Server on Windows and UNIX systems, respectively.
user_projects\domains\base_domain\bin\	startPointBaseConsole.cmd startPointBaseConsole.sh	Scripts used to start the PointBase console on Windows and UNIX systems, respectively.
user_projects\domains\base_domain\bin\	startWebLogic.cmd startWebLogic.sh	Scripts used to start the Administration Server on Windows and UNIX systems, respectively.
user_projects\domains\base_domain\bin\	stopManagedWebLogic.cmd stopManagedWebLogic.sh	Scripts used to stop a Managed Server on Windows and UNIX systems, respectively.
user_projects\domains\base_domain\bin\	stopWebLogic.cmd stopWebLogic.sh	Scripts used to stop the Administration Server on Windows and UNIX systems, respectively.
user_projects\domains\base_domain\config\	config.xml	File containing the configuration information used by the Administration Server. For more information, see "Domain Configuration Files" in *Oracle Fusion Middleware Understanding Domain Configuration for Oracle WebLogic Server*.
user_projects\domains\base_domain\config\deployments\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for staging an application when the application's staging mode is "staged."
user_projects\domains\base_domain\config\diagnostics\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing the system modules associated with instrumentation in the WebLogic Diagnostic Framework (WLDF).

Table 2–1 (Cont.) Output Generated from the Basic WebLogic Server Domain Template

Directory	File/s	Description
user_ projects\domains\base_ domain\config\jdbc\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing global JDBC modules that can be configured directly from JMX (as opposed to JSR-88).
user_ projects\domains\base_ domain\config\jms\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing global JMS modules that can be configured directly from JMX (as opposed to JSR-88).
user_ projects\domains\base_ domain\config\lib\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing JAR files that are added to the system classpath of the server when the server's Java virtual machine starts.
user_ projects\domains\base_ domain\config\nodemanager\	nm_ password.properties	File containing Node Manager password property values.
user_ projects\domains\base_ domain\config\security\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing system modules for the security framework. The directory contains one security provider configuration extension for each type of security provider in the domain's current realm.
user_ projects\domains\base_ domain\config\startup\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing system modules that contain startup plans. Startup plans are used to generate shell scripts that can be used as part of server startup.
user_ projects\domains\base_ domain\console-ext\	readme.txt	File providing information about the directory, which initially serves as a placeholder for custom extensions to the WebLogic Server Administration Console.
user_ projects\domains\base_ domain\init-info\	domain-info.xml	File used to identify domain creation and extension information. Such information includes the identity of the components in the domain, the location of the JDK and applications directory used by the domain, and the templates used to create and extend the domain.
user_ projects\domains\base_ domain\init-info\	security.xml	File used for creating user groups and roles that establish identity and access to domain resources.
user_ projects\domains\base_ domain\init-info\	startscript.xml	File used to create the `*.cmd` and `*.sh` files that are placed into the domain's root and `bin` directories.

Table 2–1 (Cont.) Output Generated from the Basic WebLogic Server Domain Template

Directory	File/s	Description
user_ projects\domains\base_ domain\init-info\	tokenValue.properties	File that contains the actual values to substitute for the tokens specified in the start scripts.
user_ projects\domains\base_ domain\lib\	readme.txt	File providing information about the directory, which initially serves as a placeholder for the domain's libraries. The JAR files in this directory are added dynamically to the end of the server classpath at server startup.
user_ projects\domains\base_ domain\security\	DefaultAuthenticatorInit.ldift DefaultRoleMapperInit.ldift XACMLRoleMapperInit.ldift	Files used for bootstrapping tasks, including authentication (user and group), authorization, and role mapping. These files contain LDAP-specific information. Note: WebLogic domains created with this release use the XACML providers, by default. These XACML security providers are compatible with policies and roles created using the WebLogic Authorization provider (DefaultAuthorizer) and WebLogic Role Mapping provider (DefaultRoleMapper). For more information, see "WebLogic Security Providers" in *Oracle Fusion Middleware Understanding Security for Oracle WebLogic Server*.
user_ projects\domains\base_ domain\security\	SerializedSystemIni.dat	File containing encrypted security information.
user_ projects\domains\base_ domain\servers\AdminServer\security\	boot.properties	File containing server startup properties, including the user name and password required to start the server (in encrypted format). It is generated only when you select development startup mode. This file enables you to bypass the prompt for user name and password during a server's startup cycle. For more information, see "Provide User Credentials to Start and Stop Servers" in *Oracle Fusion Middleware Managing Server Startup and Shutdown for Oracle WebLogic Server*.
user_ projects\domains\base_ domain\user_staged_ config\	readme.txt	File providing information about the directory, which initially serves as a placeholder for configuration information optionally staged by an administrator to be copied to managed servers in the domain.

2.1.2 Resources and Services Configured for WebLogic Server Domain Template

The following table identifies the resources and services configured in a domain created with the Basic WebLogic Server Domain template.

Table 2–2 Resources Configured in a Basic WebLogic Server Domain

Resource Type	Name	Notes

Administration Server	AdminServer	When using the Configuration Wizard or WLST Offline to create a domain, and you want the Administration Server name to be different from the default name, `AdminServer`, you must configure the name manually. You cannot change the name later when applying an extension template.
		For information about customizing the Administration Server name while creating a domain with the Configuration Wizard, see "Creating WebLogic Domains" *Oracle Fusion Middleware Creating Domains Using the Configuration Wizard*.
		For information about customizing the Administration Server name while creating a domain with WLST Offline, see "Creating and Configuring WebLogic Domains Using WLST Offline" in *Oracle Fusion Middleware Oracle WebLogic Scripting Tool*.
		The following sample WLST Offline code snippet shows how to change the default Administration Server name, AdminServer, to MedRecServer.
		`#-----------------------------------#`
		`Read the Basic WebLogic Server Domain`
		`template`
		`readTemplate('d:/MW_HOME/wlserver_`
		`10.3/common/templates/domains/wls.jar')`
		`#Change the Administration Server name.`
		`cd('Servers/AdminServer')`
		`set('Name', 'MedRecServer')`
		`#----------------------------------`

Security realm	myrealm	n.a.

2.2 WebLogic Server Starter Domain Template

Your product installation also provides one predefined WebLogic Server domain template. This template contains the default domain configuration settings and an application that provides a welcome page to help you get started. You can easily create or extend a domain by using these predefined template with the Configuration Wizard or WLST.

2.2.1 Generated Domain Output

The Starter WebLogic Server Domain template enables you to create a simple WebLogic Server domain. By default, when using the Basic WebLogic Server Domain template, you generate a domain that contains only the required components: an Administration Server and a single administrative user. Any required applications must be created and configured within the domain.

The following table defines the default directory structure and files generated by the Starter WebLogic Server Domain template. Unless otherwise specified, by default, the Configuration Wizard framework creates the domain in the `MW_HOME\user_projects\domains\base_domain` directory. If you modify the default configuration settings, the output directory structure may be different from the structure described here.

Table 2–3 Output Generated from the WebLogic Server Starter Domain Template

Directory	File	Description
user_ projects\applications\base_domain\	n.a	Directory designated as the repository for any custom application files that you create.
user_ projects\applications\target\wl_starter\	wls_starter.war	The web application files deployed to the starter domain.
user_ projects\domains\base_ domain\	fileRealm.properties	File containing ACLs, users, and groups that can be used for the default security realm when Compatibility security is used.
user_ projects\domains\base_ domain\	startWebLogic.cmd startWebLogic.sh	Scripts used to start the Administration Server on Windows and UNIX systems, respectively.
user_ projects\domains\base_ domain\autodeploy\	readme.txt	File providing information about the directory, which initially serves as a placeholder for automatic deployments.
user_ projects\domains\base_ domain\bin\	setDomainEnv.cmd setDomainEnv.sh	Scripts used to set up the development environment on Windows and UNIX systems, respectively.
user_ projects\domains\base_ domain\bin\	startManagedWebLogic.cmd startManagedWebLogic.sh	Scripts used to start a Managed Server on Windows and UNIX systems, respectively.
user_ projects\domains\base_ domain\bin\	startPointBaseConsole.cmd startPointBaseConsole.sh	Scripts used to start the PointBase console on Windows and UNIX systems, respectively.
user_ projects\domains\base_ domain\bin\	startWebLogic.cmd startWebLogic.sh	Scripts used to start the Administration Server on Windows and UNIX systems, respectively.
user_ projects\domains\base_ domain\bin\	stopManagedWebLogic.cmd stopManagedWebLogic.sh	Scripts used to stop a Managed Server on Windows and UNIX systems, respectively.
user_ projects\domains\base_ domain\bin\	stopWebLogic.cmd stopWebLogic.sh	Scripts used to stop the Administration Server on Windows and UNIX systems, respectively.
user_ projects\domains\base_ domain\config\	config.xml	File containing the configuration information used by the Administration Server. For more information, see "Domain Configuration Files" in *Oracle Fusion Middleware Understanding Domain Configuration for Oracle WebLogic Server*.
user_ projects\domains\base_ domain\config\deployments\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for staging an application when the application's staging mode is "staged."

Table 2–3 (Cont.) Output Generated from the WebLogic Server Starter Domain Template

Directory	File	Description
user_projects\domains\base_domain\config\diagnostics\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing the system modules associated with instrumentation in the WebLogic Diagnostic Framework (WLDF).
user_projects\domains\base_domain\config\jdbc\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing global JDBC modules that can be configured directly from JMX (as opposed to JSR-88).
user_projects\domains\base_domain\config\jms\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing global JMS modules that can be configured directly from JMX (as opposed to JSR-88).
user_projects\domains\base_domain\config\lib\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing JAR files that are added to the system classpath of the server when the server's Java virtual machine starts.
user_projects\domains\base_domain\config\nodemanager\	nm_password.properties	File containing Node Manager password property values.
user_projects\domains\base_domain\config\security\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing system modules for the security framework. The directory contains one security provider configuration extension for each type of security provider in the domain's current realm.
user_projects\domains\base_domain\config\startup\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing system modules that contain startup plans. Startup plans are used to generate shell scripts that can be used as part of server startup.
user_projects\domains\base_domain\console-ext\	readme.txt	File providing information about the directory, which initially serves as a placeholder for custom extensions to the WebLogic Server Administration Console.
user_projects\domains\base_domain\init-info\	domain-info.xml	File used to identify domain creation and extension information. Such information includes the identity of the components in the domain, the location of the JDK and applications directory used by the domain, and the templates used to create and extend the domain.

Table 2–3 (Cont.) Output Generated from the WebLogic Server Starter Domain Template

Directory	File	Description
user_projects\domains\base_domain\init-info\	security.xml	File used for creating user groups and roles that establish identity and access to domain resources.
user_projects\domains\base_domain\init-info\	startscript.xml	File used to create the `*.cmd` and `*.sh` files that are placed into the domain's root and `bin` directories.
user_projects\domains\base_domain\init-info\	tokenValue.properties	File that contains the actual values to substitute for the tokens specified in the start scripts.
user_projects\domains\base_domain\lib\	readme.txt	File providing information about the directory, which initially serves as a placeholder for the domain's libraries. The JAR files in this directory are added dynamically to the end of the server classpath at server startup.
user_projects\domains\base_domain\security\	DefaultAuthenticatorInit.ldift DefaultRoleMapperInit.ldift XACMLRoleMapperInit.ldift	Files used for bootstrapping tasks, including authentication (user and group), authorization, and role mapping. These files contain LDAP-specific information. Note: WebLogic domains created with this release use the XACML providers, by default. These XACML security providers are compatible with policies and roles created using the WebLogic Authorization provider (DefaultAuthorizer) and WebLogic Role Mapping provider (DefaultRoleMapper). For more information, see "WebLogic Security Providers" in *Oracle Fusion Middleware Understanding Security for Oracle WebLogic Server*.
user_projects\domains\base_domain\security\	SerializedSystemIni.dat	File containing encrypted security information.
user_projects\domains\base_domain\servers\AdminServer\security\	boot.properties	File containing server startup properties, including the user name and password required to start the server (in encrypted format). It is generated only when you select development startup mode. This file enables you to bypass the prompt for user name and password during a server's startup cycle. For more information, see "Provide User Credentials to Start and Stop Servers" in *Oracle Fusion Middleware Managing Server Startup and Shutdown for Oracle WebLogic Server*.

2.2.2 Resources and Services Configured for WebLogic Server Starter Domain Template

The following table identifies the resources and services configured in a domain created with the Basic WebLogic Server Starter Domain template.

Table 2–4 Resources Configured in a WebLogic Server Starter Domain

Resource Type	Name	Notes

Administration Server	AdminServer	When using the Configuration Wizard or WLST Offline to create a domain, and you want the Administration Server name to be different from the default name, `AdminServer`, you must configure the name manually. You cannot change the name later when applying an extension template.
		For information about customizing the Administration Server name while creating a domain with the Configuration Wizard, see "Creating WebLogic Domains" *Oracle Fusion Middleware Creating Domains Using the Configuration Wizard*.
		For information about customizing the Administration Server name while creating a domain with WLST Offline, see "Creating WebLogic Domains Using WLST Offline" in *Oracle Fusion Middleware Oracle WebLogic Scripting Tool*.
		The following sample WLST Offline code snippet shows how to change the default Administration Server name, AdminServer, to MedRecServer.
		`#-----------------------------------#`
		`Read the Basic WebLogic Server Domain template`
		`readTemplate('d:/MW_HOME/wlserver_10.3/common/templates/domains/wls.jar')`
		`#Change the Administration Server name.`
		`cd('Servers/AdminServer')`
		`set('Name', 'MedRecServer')`
		`#-----------------------------------`

Application Deployments	wls_starter	The web application deployed to the starter domain.
Security realm	myrealm	n.a.

2.3 WebLogic Beehive Extension Template

By using the Configuration Wizard or WLST, you can easily extend a base WebLogic Server domain to include the resources required for using WebLogic Beehive. You accomplish this by adding the resources and services provided in the WebLogic Beehive and WebLogic Advanced Web Services extension templates to a base WebLogic Server domain.

2.3.1 Generated Domain Output

The following table defines the default directory structure and files generated after applying the WebLogic Beehive and WebLogic Advanced Web Services extension templates to a base WebLogic Server domain. Unless otherwise specified, by default, the Configuration Wizard creates the domain in the `MW_HOME\user_projects\domains\base_domain` directory. If you modify the default configuration settings, the output directory structure may be different from the structure described here.

Table 2–5 Base Domain After Applying the WebLogic Beehive and WebLogic Advanced Web Services Extension Templates

Directory	File	Description
user_projects\applications\base_domain\	empty	empty
empty	n.a.	Directory serving as a placeholder for any custom application files that you create.
user_projects\domains\base_domain\	empty	empty
empty	fileRealm.properties	File containing ACLs, users, and groups that can be used for the default security realm when Compatibility security is used.
empty	pointbase.ini	File containing initialization information for a PointBase JDBC database.
empty	startWebLogic.cmd startWebLogic.sh	Scripts used to start the Administration Server on Windows and UNIX systems, respectively.
empty	URLs.dat	File containing the URL for the JDBC database.
autodeploy\	readme.txt	File providing information about the directory, which initially serves as a placeholder for automatic deployments.
bin\	setDomainEnv.cmd setDomainEnv.sh	Scripts used to set up the development environment on Windows and UNIX systems, respectively.
bin\	startManagedWebLogic.cmd startManagedWebLogic.sh	Scripts used to start a Managed Server on Windows and UNIX systems, respectively.
bin\	startPointBaseConsole.cmd startPointBaseConsole.sh	Scripts used to start the PointBase console on Windows and UNIX systems, respectively.
bin\	startWebLogic.cmd startWebLogic.sh	Scripts used to start the Administration Server on Windows and UNIX systems, respectively.
bin\	stopManagedWebLogic.cmd stopManagedWebLogic.sh	Scripts used to stop a Managed Server on Windows and UNIX systems, respectively.
bin\	stopWebLogic.cmd stopWebLogic.sh	Scripts used to stop the Administration Server on Windows and UNIX systems, respectively.
config\	config.xml	File containing the configuration information used by the Administration Server. For more information, see "Domain Configuration Files" in *Oracle Fusion Middleware Understanding Domain Configuration for Oracle WebLogic Server*.
config\deployments\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for staging an application when the application's staging mode is "staged."

Table 2–5 (Cont.) Base Domain After Applying the WebLogic Beehive and WebLogic Advanced Web Services Extension Templates

Directory	File	Description
config\diagnostics\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing the system modules associated with instrumentation in the WebLogic Diagnostic Framework (WLDF).
config\jdbc\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing global JDBC modules that can be configured directly from JMX (as opposed to JSR-88).
config\jdbc\	cgDataSource-jdbc.xml	Global XA JDBC Data Source module for the domain configured for conversational Web services.
config\jdbc\	cgDataSource-nonXA-jdbc.xml	Global non-XA JDBC Data Source module for the domain configured for conversational Web services.
config\jms\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing global JMS modules that can be configured directly from JMX (as opposed to JSR-88).
config\jms\	conversational-jms.xml	Global JMS module for the domain configured for conversational Web services.
config\lib\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing JAR files that are added to the system classpath of the server when the server's Java virtual machine starts.
config\nodemanager\	nm_password.properties	File containing Node Manager password property values.
config\security\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing system modules for the security framework. The directory contains one security provider configuration extension for each type of security provider in the domain's current realm.
config\startup\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing system modules that contain startup plans. Startup plans are used to generate shell scripts that can be used as part of server startup.
console-ext\	readme.txt	File providing information about the directory, which initially serves as a placeholder for custom extensions to the WebLogic Server Administration Console.
init-info\	domain-info.xml	File used to identify domain creation and extension information. Such information includes the identity of the components in the domain, the location of the JDK and applications directory used by the domain, and the templates used to create and extend the domain.

Table 2–5 (Cont.) Base Domain After Applying the WebLogic Beehive and WebLogic Advanced Web Services Extension Templates

Directory	File	Description
init-info\	security.xml	File used for creating user groups and roles that establish identity and access to domain resources.
init-info\	startscript.xml	File used to create the *.cmd and *.sh files that are placed into the domain's root and bin directories.
init-info\	tokenValue.properties	File that contains the actual values to substitute for the tokens specified in the start scripts.
lib\	readme.txt	File providing information about the directory, which initially serves as a placeholder for the domain's libraries. The JAR files in this directory are added dynamically to the end of the server classpath at server startup.
security\	DefaultAuthenticatorInit.ldift DefaultRoleMapperInit.ldift XACMLRoleMapperInit.ldift	Files used for bootstrapping tasks, including authentication (user and group), authorization, and role mapping. These files contain LDAP-specific information. Note: WebLogic domains created with this release use the XACML providers by default. These XACML security providers are compatible with policies and roles created using the WebLogic Authorization provider (DefaultAuthorizer) and WebLogic Role Mapping provider (DefaultRoleMapper). For more information, see WebLogic Security Providers in Understanding WebLogic Security at http://e-docs.bea.com/wls/docs103/secintro/archtect.html#archtect_0111.
security\	SerializedSystemIni.dat	File containing encrypted security information.
servers\AdminServer\security\	boot.properties	File containing server startup properties, including the user name and password required to start the server (in encrypted format). It is generated only when you select development startup mode. This file enables you to bypass the prompt for user name and password during a server's startup cycle. For more information, see "Provide User Credentials to Start and Stop Servers" in Starting and Stopping Servers in Managing Server Startup and Shutdown at http://edocs.bea.com/wls/docs103/server_start/overview.html.
user_staged_config\	readme.txt	File providing information about the directory, which initially serves as a placeholder for configuration information optionally staged by an administrator to be copied to managed servers in the domain.

2.3.2 Resources and Services Configured

The following table identifies the resources and services configured in a domain extended with the WebLogic Beehive and WebLogic Advanced Web Services extension templates.

Table 2–6 Resources Configured in a WebLogic Beehive Domain

Resource Type	Name	Extension Result
Administration Server	AdminServer	Uses the Administration Server provided in the base WebLogic Server domain. The default name is AdminServer, unless changed during domain creation. The Administration Server referenced in the extension template is cgServer. For information about naming the Administration Server during domain creation, see Section 2.1.2, "Resources and Services Configured for WebLogic Server Domain Template."
Security realm	myrealm	Uses the security realm provided by the base WebLogic Server domain.
Libraries Deployed	beehive-netui-1.0#1.0@1.0	Adds the Apache Beehive NetUI Version 1.0 libraries provided by the WebLogic Beehive extension template and targets them to the Administration Server, AdminServer. These libraries support pageflow development, and depend on the libraries contained in struts-1.1.war and weblogic-beehive-1.0.ear.
Libraries Deployed	struts-1.1#1.1@1.0	Adds the Apache Struts Version 1.1 libraries provided by the WebLogic Beehive extension template and targets them to the Administration Server, AdminServer.
Libraries Deployed	struts-1.2#1.2@1.0	Adds the Apache Struts Version 1.2 libraries provided by the WebLogic Beehive extension template and targets them to the Administration Server, AdminServer.

2.4 WebLogic Advanced Web Services Extension Template

By using the Configuration Wizard or WLST, you can easily extend a base WebLogic Server domain to include the resources required for advanced Web services. You accomplish this by adding the resources and services provided in the WebLogic Advanced Web Services extension template to a base WebLogic Server domain.

2.4.1 Generated Domain Output

The following table defines the default directory structure and files generated after applying the WebLogic Advanced Web Services extension template to a base WebLogic Server domain. Unless otherwise specified, by default, the Configuration Wizard creates the domain in the `*MW_HOME*\user_projects\domains\base_domain` directory. If you modify the default configuration settings, the output directory structure may be different from the structure described here.

Table 2–7 Base Domain After Applying the WebLogic Advanced Web Services Extension Template

Directory	File	Description
user_ projects\applications\base_domain\	n.a	Directory serving as a placeholder for any custom application files that you create.
user_ projects\applications\base_domain\	fileRealm.properties	File containing ACLs, users, and groups that can be used for the default security realm when Compatibility security is used.
user_ projects\applications\base_domain\	pointbase.ini	File containing initialization information for a PointBase JDBC database.
user_ projects\applications\base_domain\	startWebLogic.cmd startWebLogic.sh	Scripts used to start the Administration Server on Windows and UNIX systems, respectively.
user_ projects\applications\base_domain\	URLs.dat	File containing the URL for the JDBC database.
user_ projects\applications\base_domain\autodeploy\	readme.txt	File providing information about the directory, which initially serves as a placeholder for automatic deployments.
user_ projects\applications\base_domain\bin\	setDomainEnv.cmd setDomainEnv.sh	Scripts used to set up the development environment on Windows and UNIX systems, respectively.
user_ projects\applications\base_domain\bin\	startManagedWebLogic.cmd startManagedWebLogic.sh	Scripts used to start a Managed Server on Windows and UNIX systems, respectively.
user_ projects\applications\base_domain\bin\	startPointBaseConsole.cmd startPointBaseConsole.sh	Scripts used to start the PointBase console on Windows and UNIX systems, respectively.
user_ projects\applications\base_domain\bin\	startWebLogic.cmd startWebLogic.sh	Scripts used to start the Administration Server on Windows and UNIX systems, respectively.
user_ projects\applications\base_domain\bin\	stopManagedWebLogic.cmd stopManagedWebLogic.sh	Scripts used to stop a Managed Server on Windows and UNIX systems, respectively.
user_ projects\applications\base_domain\bin\	stopWebLogic.cmd stopWebLogic.sh	Scripts used to stop the Administration Server on Windows and UNIX systems, respectively.
user_ projects\applications\base_domain\config\	config.xml	File containing the configuration information used by the Administration Server. For more information, see "Domain Configuration Files" in *Oracle Fusion Middleware Understanding Domain Configuration for Oracle WebLogic Server*.

Table 2–7 (Cont.) Base Domain After Applying the WebLogic Advanced Web Services Extension Template

Directory	File	Description
user_ projects\applications\base_ domain\config\deployments\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for staging an application when the application's staging mode is "staged."
user_ projects\applications\base_ domain\config\diagnostics\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing the system modules associated with instrumentation in the WebLogic Diagnostic Framework (WLDF).
`user_ projects\applications\base_domain\config\jms`	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing global JMS modules that can be configured directly from JMX (as opposed to JSR-88).
`user_ projects\applications\base_domain\config\jms`	wseejmsmodule-jms.xml	Global JMS module for the domain configured for advanced Web Services.
user_ projects\applications\base_domain\config\lib\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing JAR files that are added to the system classpath of the server when the server's Java virtual machine starts.
user_ projects\applications\base_ domain\config\nodemanager\	nm_password.properties	File containing Node Manager password property values.
user_ projects\applications\base_ domain\config\security\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing system modules for the security framework. The directory contains one security provider configuration extension for each type of security provider in the domain's current realm.
user_ projects\applications\base_ domain\config\startup\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing system modules that contain startup plans. Startup plans are used to generate shell scripts that can be used as part of server startup.
user_ projects\applications\base_domain\console-ext\	readme.txt	File providing information about the directory, which initially serves as a placeholder for custom extensions to the WebLogic Server Administration Console.

Table 2–7 (Cont.) Base Domain After Applying the WebLogic Advanced Web Services Extension Template

Directory	File	Description
user_projects\applications\base_domain\init-info\	domain-info.xml	File used to identify domain creation and extension information. Such information includes the identity of the components in the domain, the location of the JDK and applications directory used by the domain, and the templates used to create and extend the domain.
user_projects\applications\base_domain\init-info\	security.xml	File used for creating user groups and roles that establish identity and access to domain resources.
user_projects\applications\base_domain\init-info\	startscript.xml	File used to create the *.cmd and *.sh files that are placed into the domain's root and bin directories.
user_projects\applications\base_domain\init-info\	tokenValue.properties	File that contains the actual values to substitute for the tokens specified in the start scripts.
user_projects\applications\base_domain\lib\	readme.txt	File providing information about the directory, which initially serves as a placeholder for the domain's libraries. The JAR files in this directory are added dynamically to the end of the server classpath at server startup.
user_projects\applications\base_domain\security\	DefaultAuthenticatorInit.ldift DefaultRoleMapperInit.ldift XACMLRoleMapperInit.ldift	Files used for bootstrapping tasks, including authentication (user and group), authorization, and role mapping. These files contain LDAP-specific information. Note: WebLogic domains created with this release use the XACML providers by default. These XACML security providers are compatible with policies and roles created using the WebLogic Authorization provider (DefaultAuthorizer) and WebLogic Role Mapping provider (DefaultRoleMapper). For more information, see "WebLogic Security Providers" in *Oracle Fusion Middleware Understanding Security for Oracle WebLogic Server*.
user_projects\applications\base_domain\security\	SerializedSystemIni.dat	File containing encrypted security information.

Table 2–7 (Cont.) Base Domain After Applying the WebLogic Advanced Web Services Extension Template

Directory	File	Description
user_projects\applications\base_domain\servers\AdminServer\security\	boot.properties	File containing server startup properties, including the user name and password required to start the server (in encrypted format). It is generated only when you select development startup mode. This file enables you to bypass the prompt for user name and password during a server's startup cycle. For more information, see "Provide User Credentials to Start and Stop Servers" in *Oracle Fusion Middleware Managing Server Startup and Shutdown for Oracle WebLogic Server*.
user_projects\applications\base_domain\user_staged_config\	readme.txt	File providing information about the directory, which initially serves as a placeholder for configuration information optionally staged by an administrator to be copied to managed servers in the domain.

2.4.2 Resources and Services Configured

The following table identifies the resources and services configured in a domain extended with the WebLogic Advanced Web Services extension template.

Table 2–8 Resources Configured in a WebLogic Advanced Web Services Domain

Resource Type	Name	Extension Result
Administration Server	AdminServer	Uses the Administration Server provided in the base WebLogic Server domain. The default name is AdminServer, unless changed during domain creation. The Administration Server referenced in the extension template is cgServer. For information about naming the Administration Server during domain creation, see Section 2.1.2, "Resources and Services Configured for WebLogic Server Domain Template."
JMS Queues	WseeMessageQueue	Adds the JMS queue to the JMS server, WseeJmsServer.
JMS Queues	WseeCallbackQueue	Adds the JMS queue to the JMS server, WseeJmsServer.
JMS Server	WseeJmsServer	Adds the JMS server as a system resource and targets it to the Administration Server, AdminServer.
Security realm	myrealm	Uses the security realm provided by the base WebLogic Server domain.

2.5 Avitek Medical Records Sample Domain Template

By using the Configuration Wizard or WLST, you can easily extend a base WebLogic Server domain to create an Avitek Medical Records Sample domain. You accomplish this by adding the resources and services provided in the Avitek Medical Records Sample domain extension template to a base WebLogic Server domain.

For more information about the Avitek Medical Records sample application, see *Oracle Fusion Middleware Avitek Medical Records Development Tutorials for Oracle WebLogic Server*.

2.5.1 Generated Domain Output

The following table defines the default directory structure and files generated after applying the Avitek Medical Records Sample Domain extension template to a base WebLogic Server domain. Unless otherwise specified, by default, the Configuration Wizard creates the domain in the `*MW_HOME*\user_projects\domains\base_domain` directory. If you modify the default configuration settings, the output directory structure may be different from the structure described here.

Table 2–9 Base Domain After Applying the Avitek Medical Records Sample Extension Template

Directory	File	Description
user_projects\applications\base_domain\dist\	Various	Includes sub-directories containing various distributions of the Avitek Medical Records applications.
user_projects\applications\base_domain\doc\	Various	Directory and files containing the Avitek Medical Records online documentation.
user_projects\applications\base_domain\lib\	Various	Includes sub-directories containing library files supporting the Avitek Medical Records sample.
user_projects\applications\base_domain\modules\	Various	Includes sub-directories containing Avitek Medical Records source code including various Java, XML, JSP, HTML files, and so on.
user_projects\applications\base_domain\	build.xml	Ant build file used with corresponding scripts to set up a database for the Avitek Medical Records sample.
`user_projects\domains\base_domain\`	fileRealm.properties	File containing ACLs, users, and groups that can be used for the default security realm when Compatibility security is used.
`user_projects\domains\base_domain\`	log4j.properties	Configures Avitek Medical Records Log4j implementation including the MedRecApp.log file.
`user_projects\domains\base_domain\`	pointbase.ini	File containing initialization information for a PointBase JDBC database.
`user_projects\domains\base_domain\`	startWebLogic.cmd startWebLogic.sh	Scripts used to start the Administration Server on Windows and UNIX systems, respectively.
user_projects\domains\base_domain\autodeploy\	readme.txt	File providing information about the directory, which initially serves as a placeholder for automatic deployments.
user_projects\domains\base_domain\bin\	setDomainEnv.cmd setDomainEnv.sh	Scripts used to set up the development environment on Windows and UNIX systems, respectively.
user_projects\domains\base_domain\bin\	startManagedWebLogic.cmd startManagedWebLogic.sh	Scripts used to start a Managed Server on Windows and UNIX systems, respectively.

Table 2–9 (Cont.) Base Domain After Applying the Avitek Medical Records Sample Extension Template

Directory	File	Description
user_ projects\domains\base_ domain\bin\	startPointBaseConsole.cmd startPointBaseConsole.sh	Scripts used to start the PointBase console on Windows and UNIX systems, respectively.
user_ projects\domains\base_ domain\bin\	startWebLogic.cmd startWebLogic.sh	Scripts used to start the Administration Server on Windows and UNIX systems, respectively.
user_ projects\domains\base_ domain\bin\	stopManagedWebLogic.cmd stopManagedWebLogic.sh	Scripts used to stop a Managed Server on Windows and UNIX systems, respectively.
user_ projects\domains\base_ domain\bin\	stopWebLogic.cmd stopWebLogic.sh	Scripts used to stop the Administration Server on Windows and UNIX systems, respectively.
user_ projects\domains\base_ domain\config\	config.xml	File containing the configuration information used by the Administration Server. For more information, see "Domain Configuration Files" in *Oracle Fusion Middleware Understanding Domain Configuration for Oracle WebLogic Server*.
user_ projects\domains\base_ domain\config\deployments\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for staging an application when the application's staging mode is staged."
user_ projects\domains\base_ domain\config\diagnostics\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing the system modules associated with instrumentation in the WebLogic Diagnostic Framework (WLDF).
user_ projects\domains\base_ domain\config\diagnostics\	MedRecWLDF.xml	Diagnostic descriptor information for the Avitek Medical Records diagnostics instrumentation.
user_ projects\domains\base_ domain\config\jdbc\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing global JDBC modules that can be configured directly from JMX (as opposed to JSR-88).
user_ projects\domains\base_ domain\config\jdbc\	MedRec-jdbc.xml	Global XA JDBC Data Source module for the Avitek Medical Records domain.
user_ projects\domains\base_ domain\config\jms\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing global JMS modules that can be configured directly from JMX (as opposed to JSR-88).
user_ projects\domains\base_ domain\config\jms\	MedRec-jms.xml	Global JMS module for the Avitek Medical Records domain.

Table 2–9 (Cont.) Base Domain After Applying the Avitek Medical Records Sample Extension Template

Directory	File	Description
user_ projects\domains\base_ domain\config\lib\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing JAR files that are added to the system classpath of the server when the server's Java virtual machine starts.
user_ projects\domains\base_ domain\config\nodemanager\	nm_ password.properties	File containing Node Manager password property values.
user_ projects\domains\base_ domain\config\security\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing system modules for the security framework. The directory contains one security provider configuration extension for each type of security provider in the domain's current realm.
user_ projects\domains\base_ domain\config\startup\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing system modules that contain startup plans. Startup plans are used to generate shell scripts that can be used as part of server startup.
user_ projects\domains\base_ domain\console-ext\	readme.txt	File providing information about the directory, which initially serves as a placeholder for custom extensions to the WebLogic Server Administration Console.
user_ projects\domains\base_ domain\console-ext\	diagnostics-console-extension.jar	File used to demonstrate an extension to the WebLogic Server Administration Console that shows diagnostics features.
user_ projects\domains\base_ domain\init-info\	domain-info.xml	File used to identify domain creation and extension information. Such information includes the identity of the components in the domain, the location of the JDK and applications directory used by the domain, and the templates used to create and extend the domain.
user_ projects\domains\base_ domain\init-info\	security.xml	File used for creating user groups and roles that establish identity and access to domain resources.
user_ projects\domains\base_ domain\init-info\	startscript.xml	File used to create the *.cmd and *.sh files that are placed into the domain's root and bin directories.
user_ projects\domains\base_ domain\init-info\	tokenValue.properties	File that contains the actual values to substitute for the tokens specified in the start scripts.
user_ projects\domains\base_ domain\lib\	readme.txt	File providing information about the directory, which initially serves as a placeholder for the domain's libraries. The JAR files in this directory are added dynamically to the end of the server classpath at server startup.

Table 2–9 (Cont.) Base Domain After Applying the Avitek Medical Records Sample Extension Template

Directory	File	Description
user_projects\domains\base_domain\security\	DefaultAuthenticatorInit.ldift DefaultAuthorizerInit.ldift	Files used for bootstrapping tasks, including authentication (user and group), authorization, and role mapping. These files contain LDAP-specific information. Note: WebLogic domains created with this release use the XACML providers, by default. These XACML security providers are compatible with policies and roles created using the WebLogic Authorization provider (DefaultAuthorizer) and WebLogic Role Mapping provider (DefaultRoleMapper). For more information, see "WebLogic Security Providers" in *Oracle Fusion Middleware Understanding Security for Oracle WebLogic Server*.
user_projects\domains\base_domain\servers\AdminServer\security\	boot.properties	File containing server startup properties, including the user name and password required to start the server (in encrypted format). It is generated only when you select development startup mode. This file enables you to bypass the prompt for user name and password during a server's startup cycle. For more information, see "Provide User Credentials to Start and Stop Servers" in *Oracle Fusion Middleware Managing Server Startup and Shutdown for Oracle WebLogic Server*.

2.5.2 Resources and Services Configured

The following table identifies the resources and services configured in a domain extended with the Avitek Medical Records Sample extension template.

Table 2–10 Resources Configured in an Avitek Medical Records Domain

Resource Type	Name	Extension Result
Administration Server	AdminServer	Uses the Administration Server provided in the base WebLogic Server domain. The default name is `AdminServer`, unless changed during domain creation. The Administration Server referenced in the extension template is MedRecServer. For information about naming the Administration Server during domain creation, see Section 2.1.2, "Resources and Services Configured for WebLogic Server Domain Template."
Application Deployments	browser-starter	Adds the browser-starter application and targets it to the Administration Server, AdminServer.
Application Deployments	medrec	Adds the medrec application and targets it to the Administration Server, AdminServer.
Application Deployments	physician	Adds the physician application and targets it to the Administration Server, AdminServer.

Table 2–10 (Cont.) Resources Configured in an Avitek Medical Records Domain

Resource Type	Name	Extension Result
JDBC Data Sources	MedRecGlobalDataSourceXA	Identifies the JDBC data source as a MedRecGlobalDataSource system resource.
JMS Queues	com.bea.medrec.jms.RecordToCreateQueue com.bea.medrec.jms.PatientNotificationQueue weblogic.wsee.DefaultQueue	Adds the JMS queue to the JMS server, MedRecWseeJMSServer.
JMS Servers	MedRecJMSServer	Adds the JMS server as a MedRec-jms system resource and targets it to the Administration Server, AdminServer.
JMS System Resources	MedRec-jms	Adds the JMS servers, connection factories, and queues to be used as JMS system resources, and targets the resources to the Administration Server, AdminServer.
Mail Session	mail/MedRecMailSession	Adds the mail session.
SAF Agent	WsrmAgent	Adds this store-and-forward agent, which uses the file store, MedRecWseeFileStore, and targets it to the Administration Server, AdminServer.
Security realm	myrealm	Uses the security realm provided in the base WebLogic Server domain.
WLDF System Resource	MedRecWLDF	Adds the WLDF system resource and defined WLDF instrumentation monitors for dye injection, and targets them to the Administration Server, AdminServer.

2.6 Avitek Medical Records Sample Domain Template (Spring Version)

By using the Configuration Wizard or WLST, you can easily extend a base WebLogic Server domain to create an Avitek Medical Records Sample domain in Spring. You accomplish this by adding the resources and services provided in the Avitek Medical Records Sample domain extension template to a base WebLogic Server domain.

For more information about the Avitek Medical Records sample application, see *Oracle Fusion Middleware Avitek Medical Records Development Tutorials for Oracle WebLogic Server*.

2.6.1 Generated Domain Output

The following table defines the default directory structure and files generated after applying the Avitek Medical Records Sample Domain extension template for Spring to a base WebLogic Server domain. Unless otherwise specified, by default, the Configuration Wizard creates the domain in the `MW_HOME\user_projects\domains\base_domain` directory. If you modify the default configuration settings, the output directory structure may be different from the structure described here.

Table 2–11 Base Domain After Applying the Avitek Medical Records Sample Extension Template (Spring Version)

Directory	File	Description
user_projects\applications\base_domain\dist\	Various	Includes sub-directories containing various distributions of the Avitek Medical Records applications.
user_projects\applications\base_domain\doc\	Various	Directory and files containing the Avitek Medical Records online documentation.
user_projects\applications\base_domain\lib\	Various	Includes sub-directories containing library files supporting the Avitek Medical Records sample.
user_projects\applications\base_domain\modules\	Various	Includes sub-directories containing Avitek Medical Records source code including various Java, XML, JSP, HTML files, and so on.

2.6.2 Resources and Services Configured

The following table identifies the resources and services configured in a domain extended with the Avitek Medical Records Sample extension template for Spring.

Table 2–12 Resources Configured in an Avitek Medical Records Domain for Spring

Resource Type	Name	Extension Result
Administration Server	AdminServer	Uses the Administration Server provided in the base WebLogic Server domain. The default name is `AdminServer`, unless changed during domain creation. The Administration Server referenced in the extension template is MedRecServer. For information about naming the Administration Server during domain creation, see Section 2.1.2, "Resources and Services Configured for WebLogic Server Domain Template."
Application Deployments	browser-starter	Adds the browser-starter application and targets it to the Administration Server, AdminServer.
Application Deployments	medrec	Adds the medrec application and targets it to the Administration Server, AdminServer.
Application Deployments	physician	Adds the physician application and targets it to the Administration Server, AdminServer.
JDBC Data Sources	MedRecGlobalDataSourceXA	Identifies the JDBC data source as a MedRecGlobalDataSource system resource.
JMS Queues	com.bea.medrec.jms.RecordToCreateQueue com.bea.medrec.jms.PatientNotificationQueue weblogic.wsee.DefaultQueue	Adds the JMS queue to the JMS server, MedRecWseeJMSServer.
JMS Servers	MedRecJMSServer	Adds the JMS server as a MedRec-jms system resource and targets it to the Administration Server, AdminServer.

Table 2–12 (Cont.) Resources Configured in an Avitek Medical Records Domain for

Resource Type	Name	Extension Result
JMS System Resources	MedRec-jms	Adds the JMS servers, connection factories, and queues to be used as JMS system resources, and targets the resources to the Administration Server, AdminServer.
Mail Session	mail/MedRecMailSession	Adds the mail session.
SAF Agent	WsrmAgent	Adds this store-and-forward agent, which uses the file store, MedRecWseeFileStore, and targets it to the Administration Server, AdminServer.
Security realm	myrealm	Uses the security realm provided in the base WebLogic Server domain.
WLDF System Resource	MedRecWLDF	Adds the WLDF system resource and defined WLDF instrumentation monitors for dye injection, and targets them to the Administration Server, AdminServer.
Administration Server	AdminServer	Uses the Administration Server provided in the base WebLogic Server domain. The default name is `AdminServer`, unless changed during domain creation. The Administration Server referenced in the extension template is MedRecServer. For information about naming the Administration Server during domain creation, see Section 2.1.2, "Resources and Services Configured for WebLogic Server Domain Template."
Application Deployments	browser-starter	Adds the browser-starter application and targets it to the Administration Server, AdminServer.
Application Deployments	medrec	Adds the medrec application and targets it to the Administration Server, AdminServer.
Application Deployments	physician	Adds the physician application and targets it to the Administration Server, AdminServer.
JDBC Data Sources	MedRecGlobalDataSourceXA	Identifies the JDBC data source as a MedRecGlobalDataSource system resource.
JMS Queues	com.bea.medrec.jms.RecordToCreateQueue com.bea.medrec.jms.PatientNotificationQueue weblogic.wsee.DefaultQueue	Adds the JMS queue to the JMS server, MedRecWseeJMSServer.
JMS Servers	MedRecJMSServer	Adds the JMS server as a MedRec-jms system resource and targets it to the Administration Server, AdminServer.
JMS System Resources	MedRec-jms	Adds the JMS servers, connection factories, and queues to be used as JMS system resources, and targets the resources to the Administration Server, AdminServer.

2.7 Oracle Workshop for WebLogic Extension Template

Using the Configuration Wizard or WLST, you can easily extend a base WebLogic Server domain to include the resources required for using Workshop for WebLogic. You accomplish this by adding the resources and services provided in the Workshop for WebLogic template to a base WebLogic Server domain.

> **Note:** Using the Configuration Wizard in graphical mode, you can easily create a Workshop for WebLogic domain by checking the Workshop for WebLogic check box in the **Select Domain Source** window. The result is the same as creating a base WebLogic Server domain first and then extending that domain with both the Oracle Workshop for WebLogic extension template. For more information about the templates required to create a Oracle Workshop for WebLogic domain, see Section 1.5, "Relationships Between Templates."

2.7.1 Generated Domain Output

The following table defines the default directory structure and files generated after applying the Workshop for WebLogic template to a base WebLogic Server domain. Unless otherwise specified, by default, the Configuration Wizard creates the domain in the *MW_HOME*\user_projects\domains\base_domain directory. If you modify the default configuration settings, the output directory structure may be different from the structure described here.

Table 2–13 Base Domain After Applying the Oracle Workshop for WebLogic Template

Directory	File	Description
user_ projects\applications\base_domain\	n.a.	Directory serving as a placeholder for any custom application files that you create.
user_ projects\applications\base_domain\	fileRealm.properties	File containing ACLs, users, and groups that can be used for the default security realm when Compatibility security is used.
user_ projects\applications\base_domain\	pointbase.ini	File containing initialization information for a PointBase JDBC database.
user_ projects\applications\base_domain\	startWebLogic.cmd startWebLogic.sh	Scripts used to start the Administration Server on Windows and UNIX systems, respectively.
user_ projects\applications\base_domain\	URLs.dat	File containing the URL for the JDBC database.
user_ projects\applications\base_ domain\autodeploy\	readme.txt	File providing information about the directory, which initially serves as a placeholder for automatic deployments.
user_ projects\applications\base_ domain\bin\	setDomainEnv.cmd setDomainEnv.sh	Scripts used to set up the development environment on Windows and UNIX systems, respectively.

Table 2–13 (Cont.) Base Domain After Applying the Oracle Workshop for WebLogic

Directory	File	Description
user_projects\applications\base_domain\bin\	startManagedWebLogic.cmd startManagedWebLogic.sh	Scripts used to start a Managed Server on Windows and UNIX systems, respectively.
user_projects\applications\base_domain\bin\	startPointBaseConsole.cmd startPointBaseConsole.sh	Scripts used to start the PointBase console on Windows and UNIX systems, respectively.
user_projects\applications\base_domain\bin\	startWebLogic.cmd startWebLogic.sh	Scripts used to start the Administration Server on Windows and UNIX systems, respectively.
user_projects\applications\base_domain\bin\	stopManagedWebLogic.cmd stopManagedWebLogic.sh	Scripts used to stop a Managed Server on Windows and UNIX systems, respectively.
user_projects\applications\base_domain\bin\	stopWebLogic.cmd stopWebLogic.sh	Scripts used to stop the Administration Server on Windows and UNIX systems, respectively.
user_projects\applications\base_domain\config\	config.xml	File containing the configuration information used by the Administration Server. For more information, see "Domain Configuration Files" in *Oracle Fusion Middleware Understanding Domain Configuration for Oracle WebLogic Server*.
user_projects\applications\base_domain\config\deployments\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for staging an application when the application's staging mode is "staged."
user_projects\applications\base_domain\config\diagnostics\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing the system modules associated with instrumentation in the WebLogic Diagnostic Framework (WLDF).
user_projects\applications\base_domain\config\jdbc\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing global JDBC modules that can be configured directly from JMX (as opposed to JSR-88).
user_projects\applications\base_domain\config\jdbc\	cgDataSource-jdbc.xml	Global XA JDBC Data Source module for the domain configured for advanced Web services.
user_projects\applications\base_domain\config\jdbc\	cgDataSource-nonXA-jdbc.xml	Global non-XA JDBC Data Source module for the domain configured for advanced Web services.

Table 2–13 (Cont.) Base Domain After Applying the Oracle Workshop for WebLogic

Directory	File	Description
user_projects\applications\base_domain\config\lib\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing JAR files that are added to the system classpath of the server when the server's Java virtual machine starts.
user_projects\applications\base_domain\config\nodemanager\	nm_password.properties	File containing Node Manager password property values.
user_projects\applications\base_domain\config\security\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing system modules for the security framework. The directory contains one security provider configuration extension for each type of security provider in the domain's current realm.
user_projects\applications\base_domain\config\startup\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing system modules that contain startup plans. Startup plans are used to generate shell scripts that can be used as part of server startup.
user_projects\applications\base_domain\console-ext\	readme.txt	File providing information about the directory, which initially serves as a placeholder for custom extensions to the WebLogic Server Administration Console.
user_projects\applications\base_domain\init-info\	domain-info.xml	File used to identify domain creation and extension information. Such information includes the identity of the components in the domain, the location of the JDK and applications directory used by the domain, and the templates used to create and extend the domain.
user_projects\applications\base_domain\init-info\	security.xml	File used for creating user groups and roles that establish identity and access to domain resources.
user_projects\applications\base_domain\init-info\	startscript.xml	File used to create the *.cmd and *.sh files that are placed into the domain's root and bin directories.
user_projects\applications\base_domain\init-info\	tokenValue.properties	File that contains the actual values to substitute for the tokens specified in the start scripts.
user_projects\applications\base_domain\lib\	readme.txt	File providing information about the directory, which initially serves as a placeholder for the domain's libraries. The JAR files in this directory are added dynamically to the end of the server classpath at server startup.

Table 2–13 (Cont.) Base Domain After Applying the Oracle Workshop for WebLogic

Directory	File	Description
user_projects\applications\base_domain\security\	DefaultAuthenticatorInit.ldift DefaultRoleMapperInit.ldift XACMLRoleMapperInit.ldift	Files used for bootstrapping tasks, including authentication (user and group), authorization, and role mapping. These files contain LDAP-specific information. Note: WebLogic domains created with this release use the XACML providers, by default. These XACML security providers are compatible with policies and roles created using the WebLogic Authorization provider (DefaultAuthorizer) and WebLogic Role Mapping provider (DefaultRoleMapper). For more information, see "WebLogic Security Providers" in *Oracle Fusion Middleware Understanding Security for Oracle WebLogic Server*.
user_projects\applications\base_domain\security\	SerializedSystemIni.dat	File containing encrypted security information.
user_projects\applications\base_domain\servers\AdminServer\security\	boot.properties	File containing server startup properties, including the user name and password required to start the server (in encrypted format). It is generated only when you select development startup mode. This file enables you to bypass the prompt for user name and password during a server's startup cycle. For more information, see "Provide User Credentials to Start and Stop Servers" in *Oracle Fusion Middleware Managing Server Startup and Shutdown for Oracle WebLogic Server*.
user_projects\applications\base_domain\user_staged_config\	readme.txt	File providing information about the directory, which initially serves as a placeholder for configuration information optionally staged by an administrator to be copied to managed servers in the domain.

2.7.2 Resources and Services Configured

The following table identifies the resources and services configured in a domain extended with the Oracle Workshop for WebLogic template.

Table 2–14 Resources Configured in a Oracle Workshop for WebLogic Domain

Resource Type	Name	Extension Result
Administration Server	AdminServer	Uses the Administration Server provided in the base WebLogic Server domain. The default name is AdminServer, unless changed during domain creation. The Administration Server referenced in the extension template is cgServer. For information about naming the Administration Server during domain creation, see Section 2.1.2, "Resources and Services Configured for WebLogic Server Domain Template."

Table 2–14 (Cont.) Resources Configured in a Oracle Workshop for WebLogic Domain

Resource Type	Name	Extension Result
JDBC Data Source	cgDataSource	Defines an XA JDBC data source including its associated jdbc connection pool. The data source is named cgDataSource.
JDBC Data Source	cgDataSource-nonXA	Includes the JDBC data source and connection pool setups defined as cgDataSource in the domain and targets them to the correct server(s).
JDBC Store	cgJMSStore	Uses the JDBC store provided by the Oracle Workshop for WebLogic extension template. The JDBC store is to be used with the JDBC data source, cgDataSource-nonXA, and the JMS server, WseeJmsServer, as a persistent store, and is targeted to the Administration Server, AdminServer.
JDBC System Resources	cgDataSource cgDataSource-nonXA	Identifies the JDBC data source and connection pool setups to be used for JDBC system.
JMS Server	WseeJmsServer	Uses the JMS server provided by the Workshop for WebLogic extension template. Identifies the JMS server as a system resource and targets it to the Administration Server, AdminServer.
Security realm	myrealm	Uses the security realm provided by the base WebLogic Server domain.
Commons-Logging Bridge	wls-commonslogging-bridge#1.0@1.0	Hooks commons-logging into the WLS logging mechanism.
Libraries Deployed	beehive-netui-1.0#1.0@1.0	Adds the Apache Beehive NetUI Version 1.0 libraries. These libraries support pageflow development, and depend upon the libraries contained in struts-1.1.war and weblogic-beehive-1.0.ear.
Libraries Deployed	jstl#1.1@1.1.2	Adds the Java standard tagging (JSTL) Version 1.1 libraries.
Libraries Deployed	jsf-ri#1.1@1.1.1	Adds the Java Server Faces Reference Implementation libraries.
Libraries Deployed	jsf-myfaces#1.1@1.1.1	Adds the Apache MyFaces libraries.
Libraries Deployed	struts-1.1#1.1@1.0	Adds the Apache Struts Version 1.1 libraries.
Libraries Deployed	struts-1.2#1.2@1.0	Adds the Apache Struts Version 1.2 libraries.
Libraries Deployed	weblogic-controls-10.0#10.0@10.0	Adds the Oracle Workshop for WebLogic controls extensions, including additional system controls (such as service control and timer control) and support for adding transactions, security, and message buffering to existing controls. Packaged for EARs.
Libraries Deployed	weblogic-controls-10.0-war#10.0@10.0	Adds the Oracle Workshop for WebLogic controls extensions including additional system controls (such as service control) and support for adding transactions, security, and message buffering to existing controls. Excludes those features which require EAR support such as timer control. Packaged for WARs.

Table 2–14 (Cont.) Resources Configured in a Oracle Workshop for WebLogic Domain

Resource Type	Name	Extension Result
Libraries Deployed	beehive-controls-1.0#1.0@1.0	Adds the Apache Beehive Controls 1.0.1 libraries to the domain. This includes the control run time and the Beehive system controls - JdbcControl, JMSControl, and EJBControl.

2.8 Workshop for WebLogic 10.3 Extension Template

Using the Configuration Wizard or WLST, you can easily extend a base WebLogic Server domain to include the resources required for using Workshop for WebLogic 10.3. You accomplish this by adding the resources and services provided in the Workshop for WebLogic 10.3 template to a base WebLogic Server domain.

> **Note:** Using the Configuration Wizard in graphical mode, you can easily create an Oracle Workshop for WebLogic domain by checking the Workshop for WebLogic 10.3 check box in the **Select Domain Source** window. The result is the same as creating a base WebLogic Server domain first and then extending that domain with both the Workshop for WebLogic 10.3 extension template. For more information about the templates required to create a Workshop for WebLogic 10.3 domain, see Section 1.5, "Relationships Between Templates."

2.8.1 Generated Domain Output

The following table defines the default directory structure and files generated after applying the Workshop for WebLogic 10.3 template to a base WebLogic Server domain. Unless otherwise specified, by default, the Configuration Wizard creates the domain in the *MW_HOME*`\user_projects\domains\base_domain` directory. If you modify the default configuration settings, the output directory structure may be different from the structure described here.

Table 2–15 Base Domain After Applying the Workshop for WebLogic 10.3 Template

Directory	File	Description
user_projects\applications\base_domain\	empty	empty
empty	n.a.	Directory serving as a placeholder for any custom application files that you create.
user_projects\domains\base_domain\	empty	empty
empty	fileRealm.properties	File containing ACLs, users, and groups that can be used for the default security realm when Compatibility security is used.
empty	pointbase.ini	File containing initialization information for a PointBase JDBC database.
empty	startWebLogic.cmd startWebLogic.sh	Scripts used to start the Administration Server on Windows and UNIX systems, respectively.

Table 2–15 (Cont.) Base Domain After Applying the Workshop for WebLogic 10.3

Directory	File	Description
empty	URLs.dat	File containing the URL for the JDBC database.
autodeploy\	readme.txt	File providing information about the directory, which initially serves as a placeholder for automatic deployments.
bin\	setDomainEnv.cmd setDomainEnv.sh	Scripts used to set up the development environment on Windows and UNIX systems, respectively.
bin	startManagedWebLogic.cmd startManagedWebLogic.sh	Scripts used to start a Managed Server on Windows and UNIX systems, respectively.
bin	startPointBaseConsole.cmd startPointBaseConsole.sh	Scripts used to start the PointBase console on Windows and UNIX systems, respectively.
bin	startWebLogic.cmd startWebLogic.sh	Scripts used to start the Administration Server on Windows and UNIX systems, respectively.
bin	stopManagedWebLogic.cmd stopManagedWebLogic.sh	Scripts used to stop a Managed Server on Windows and UNIX systems, respectively.
bin	stopWebLogic.cmd stopWebLogic.sh	Scripts used to stop the Administration Server on Windows and UNIX systems, respectively.
config\	config.xml	File containing the configuration information used by the Administration Server. For more information, see "Domain Configuration Files" in *Oracle Fusion Middleware Understanding Domain Configuration for Oracle WebLogic Server*.
config\deployments\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for staging an application when the application's staging mode is "staged."
config\diagnostics\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing the system modules associated with instrumentation in the WebLogic Diagnostic Framework (WLDF).
config\jdbc\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing global JDBC modules that can be configured directly from JMX (as opposed to JSR-88).
config\jdbc\	cgDataSource-jdbc.xml	Global XA JDBC Data Source module for the domain configured for advanced web services.
config\jdbc\	cgDataSource-nonXA-jdbc.xml	Global non-XA JDBC Data Source module for the domain configured for advanced web services.

Table 2–15 (Cont.) Base Domain After Applying the Workshop for WebLogic 10.3

Directory	File	Description
config\lib\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing JAR files that are added to the system classpath of the server when the server's Java virtual machine starts.
config\nodemanager\	nm_password.properties	File containing Node Manager password property values.
config\security\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing system modules for the security framework. The directory contains one security provider configuration extension for each type of security provider in the domain's current realm.
config\startup\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing system modules that contain startup plans. Startup plans are used to generate shell scripts that can be used as part of server startup.
console-ext\	readme.txt	File providing information about the directory, which initially serves as a placeholder for custom extensions to the WebLogic Server Administration Console.
init-info\	domain-info.xml	File used to identify domain creation and extension information. Such information includes the identity of the components in the domain, the location of the JDK and applications directory used by the domain, and the templates used to create and extend the domain.
init-info\	security.xml	File used for creating user groups and roles that establish identity and access to domain resources.
init-info\	startscript.xml	File used to create the *.cmd and *.sh files that are placed into the domain's root and bin directories.
init-info\	tokenValue.properties	File that contains the actual values to substitute for the tokens specified in the start scripts.
lib\	readme.txt	File providing information about the directory, which initially serves as a placeholder for the domain's libraries. The JAR files in this directory are added dynamically to the end of the server classpath at server startup.

Table 2–15 (Cont.) Base Domain After Applying the Workshop for WebLogic 10.3

Directory	File	Description
security\	DefaultAuthenticatorInit.ldift DefaultRoleMapperInit.ldift XACMLRoleMapperInit.ldift	Files used for bootstrapping tasks, including authentication (user and group), authorization, and role mapping. These files contain LDAP-specific information. Note: WebLogic domains created with this release use the XACML providers, by default. These XACML security providers are compatible with policies and roles created using the WebLogic Authorization provider (DefaultAuthorizer) and WebLogic Role Mapping provider (DefaultRoleMapper). For more information, see "WebLogic Security Providers" in *Oracle Fusion Middleware Understanding Security for Oracle WebLogic Server*.
security\	SerializedSystemIni.dat	File containing encrypted security information.
servers\AdminServer\security\	boot.properties	File containing server startup properties, including the user name and password required to start the server (in encrypted format). It is generated only when you select development startup mode. This file enables you to bypass the prompt for user name and password during a server's startup cycle. For more information, see "Provide User Credentials to Start and Stop Servers" in *Oracle Fusion Middleware Managing Server Startup and Shutdown for Oracle WebLogic Server*.
user_staged_config\	readme.txt	File providing information about the directory, which initially serves as a placeholder for configuration information optionally staged by an administrator to be copied to managed servers in the domain.

2.8.2 Resources and Services Configured

The following table identifies the resources and services configured in a domain extended with the Workshop for WebLogic 10.3 template.

Table 2–16 Resources Configured in a Workshop for WebLogic 10.3 template

Resource Type	Name	Extension Result
Administration Server	AdminServer	Uses the Administration Server provided in the base WebLogic Server domain. The default name is AdminServer, unless changed during domain creation. The Administration Server referenced in the extension template is cgServer. For information about naming the Administration Server during domain creation, see Section 2.1.2, "Resources and Services Configured for WebLogic Server Domain Template."
JDBC Data Source	cgDataSource	Defines an XA JDBC data source including its associated jdbc connection pool. The data source is named cgDataSource.

Table 2–16 (Cont.) Resources Configured in a Workshop for WebLogic 10.3 template

Resource Type	Name	Extension Result
JDBC Data Source	cgDataSource-nonXA	Includes the JDBC data source and connection pool setups defined as cgDataSource in the domain and targets them to the correct server(s).
JDBC Store	cgJMSStore	Uses the JDBC store provided by the Workshop for WebLogic 10.3 extension template. The JDBC store is to be used with the JDBC data source, cgDataSource-nonXA, and the JMS server, WseeJmsServer, as a persistent store, and is targeted to the Administration Server, AdminServer.
JDBC System Resources	cgDataSource cgDataSource-nonXA	Identifies the JDBC data source and connection pool setups to be used for JDBC system.
JMS Server	WseeJmsServer	Uses the JMS server provided by the Workshop for WebLogic 10.3 extension template. Identifies the JMS server as a system resource and targets it to the Administration Server, AdminServer.
Security realm	myrealm	Uses the security realm provided by the base WebLogic Server domain.
Commons-Logging Bridge	wls-commonslogging-bridge#1.0@1.0	Hooks commons-logging into the WLS logging mechanism.
Libraries Deployed	beehive-netui-1.0#1.0@1.0	Adds the Apache Beehive NetUI Version 1.0 libraries. These libraries support pageflow development, and depend upon the libraries contained in struts-1.1.war and weblogic-beehive-1.0.ear.
Libraries Deployed	jstl#1.1@1.1.2	Adds the Java standard tagging (JSTL) Version 1.1 libraries.
Libraries Deployed	jsf-ri#1.1@1.1.1	Adds the Java Server Faces Reference Implementation libraries.
Libraries Deployed	jsf-myfaces#1.1@1.1.1	Adds the Apache MyFaces libraries.
Libraries Deployed	struts-1.1#1.1@1.0	Adds the Apache Struts Version 1.1 libraries.
Libraries Deployed	struts-1.2#1.2@1.0	Adds the Apache Struts Version 1.2 libraries.
Libraries Deployed	weblogic-controls-10.0#10.0@10.0	Adds the Workshop for WebLogic 10.3 controls extensions, including additional system controls (such as service control and timer control) and support for adding transactions, security, and message buffering to existing controls. Packaged for EARs.
Libraries Deployed	weblogic-controls-10.0-war#10.0@10.0	Adds the Workshop for WebLogic 10.3 controls extensions including additional system controls (such as service control) and support for adding transactions, security, and message buffering to existing controls. Excludes those features which require EAR support such as timer control. Packaged for WARs.
Libraries Deployed	beehive-controls-1.0#1.0@1.0	Adds the Apache Beehive Controls 1.0.1 libraries to the domain. This includes the control run time and the Beehive system controls - JdbcControl, JMSControl, and EJBControl.

2.9 WebLogic Server Default Domain Extension Template

Using the Configuration Wizard or WLST, you can easily extend a base WebLogic Server domain to include resources required for a default WebLogic Server domain. You accomplish this by adding the resources and services provided in the WebLogic Server Default Domain extension template to a base WebLogic Server domain.

> **Note:** Applying the WebLogic Server Default Domain extension template to a base WebLogic domain is a prerequisite to using the WebLogic Server Examples extension template. For information about the relationship between templates, see Section 1.5, "Relationships Between Templates."

For more information about the samples that are supported in the WebLogic Server Examples domain, see Sample Application Examples and Tutorials for WebLogic Server 10.3.

2.9.1 Generated Domain Output

The following table defines the default directory structure and files generated after applying the WebLogic Server Default Domain extension template to a base WebLogic Server domain. Unless otherwise specified, by default, the Configuration Wizard creates the domain in the *MW_HOME*`\user_projects\domains\base_domain` directory. If you modify the default configuration settings, the output directory structure may be different from the structure described here.

Table 2–17 Base Domain After Applying the WebLogic Server Default Domain Extension Template

Directory	File	Description
user_projects\applications\base_domain\	empty	empty
server\docs\	Various	Includes sub-directories containing style sheet and graphics files to support the online documentation.
server\examples\build\	Various	Includes WebLogic Server examples deployments.
server\examples\src\	Various	Includes source code and instructions for WebLogic Server examples.
user_projects\domains\base_domain\	empty	empty
empty	fileRealm.properties	File containing ACLs, users, and groups that can be used for the default security realm when Compatibility security is used.
empty	pointbase.ini	File containing initialization information for a PointBase JDBC database.
empty	setExamplesEnv.cmd setExamplesEnv.sh	Scripts that set up the environment to use the WebLogic Server Examples on Windows and UNIX systems, respectively.

Table 2–17 (Cont.) Base Domain After Applying the WebLogic Server Default Domain Extension Template

Directory	File	Description
empty	startWebLogic.cmd startWebLogic.sh	Scripts used to start the Administration Server on Windows and UNIX systems, respectively.
empty	startWebLogicEx.cmd startWebLogicEx.sh	Scripts used to start the Administration Server for the WebLogic Server Examples domain on Windows and UNIX systems, respectively.
autodeploy\	readme.txt	File providing information about the directory, which initially serves as a placeholder for automatic deployments.
bin\	setDomainEnv.cmd setDomainEnv.sh	Scripts used to set up the development environment on Windows and UNIX systems, respectively.
bin\	startManagedWebLogic.cmd startManagedWebLogic.sh	Scripts used to start a Managed Server on Windows and UNIX systems, respectively.
bin\	startPointBaseConsole.cmd startPointBaseConsole.sh	Scripts used to start the PointBase console on Windows and UNIX systems, respectively.
bin\	startWebLogic.cmd startWebLogic.sh	Scripts used to start the Administration Server on Windows and UNIX systems, respectively.
bin\	stopManagedWebLogic.cmd stopManagedWebLogic.sh	Scripts used to stop a Managed Server on Windows and UNIX systems, respectively.
bin\	stopWebLogic.cmd stopWebLogic.sh	Scripts used to stop the Administration Server on Windows and UNIX systems, respectively.
config\	config.xml	File containing the configuration information used by the Administration Server. For more information, see "Domain Configuration Files" in *Oracle Fusion Middleware Understanding Domain Configuration for Oracle WebLogic Server*.
config\deployments\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for staging an application when the application's staging mode is "staged."
config\diagnostics\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing the system modules associated with instrumentation in the WebLogic Diagnostic Framework (WLDF).
config\jdbc\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing global JDBC modules that can be configured directly from JMX (as opposed to JSR-88).

Table 2–17 (Cont.) Base Domain After Applying the WebLogic Server Default Domain Extension Template

Directory	File	Description
config\jdbc\	examples-demo-jdbc.xml	Global non-XA JDBC Data Source module for the WebLogic Server default domain.
config\jdbc\	examples-demoXA-jdbc.xml	Global XA JDBC Data Source module for the WebLogic Server default domain.
config\jms\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing global JMS modules that can be configured directly from JMX (as opposed to JSR-88).
config\lib\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing JAR files that are added to the system classpath of the server when the server's Java virtual machine starts.
config\nodemanager\	nm_password.properties	File containing Node Manager password property values.
config\security\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing system modules for the security framework. The directory contains one security provider configuration extension for each type of security provider in the domain's current realm.
config\startup\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing system modules that contain startup plans. Startup plans are used to generate shell scripts that can be used as part of server startup.
console-ext\	readme.txt	File providing information about the directory, which initially serves as a placeholder for custom extensions to the WebLogic Server Administration Console.
init-info\	domain-info.xml	File used to identify domain creation and extension information. Such information includes the identity of the components in the domain, the location of the JDK and applications directory used by the domain, and the templates used to create and extend the domain.
init-info\	security.xml	File used for creating user groups and roles that establish identity and access to domain resources.
init-info\	startscript.xml	File used to create the *.cmd and *.sh files that are placed into the domain's root and bin directories.
init-info\	tokenValue.properties	File that contains the actual values to substitute for the tokens specified in the start scripts.

Table 2–17 (Cont.) Base Domain After Applying the WebLogic Server Default Domain Extension Template

Directory	File	Description
lib\	readme.txt	File providing information about the directory, which initially serves as a placeholder for the domain's libraries. The JAR files in this directory are added dynamically to the end of the server classpath at server startup.
security\	DefaultAuthenticatorInit.ldift DefaultAuthorizerInit.ldift DefaultRoleMapperInit.ldift XACMLAuthorizerInit.ldift XACMLRoleMapperInit.ldift	Files used for bootstrapping tasks, including authentication (user and group), authorization, and role mapping. These files contain LDAP-specific information. Note: WebLogic domains created with this release use the XACML providers by default. These XACML security providers are compatible with policies and roles created using the WebLogic Authorization provider (DefaultAuthorizer) and WebLogic Role Mapping provider (DefaultRoleMapper). For more information, see "WebLogic Security Providers" in *Oracle Fusion Middleware Understanding Security for Oracle WebLogic Server*.
security\	SerializedSystemIni.dat	File containing encrypted security information.
servers\AdminServer\security\	boot.properties	File containing server startup properties, including the user name and password required to start the server (in encrypted format). It is generated only when you select development startup mode. This file enables you to bypass the prompt for user name and password during a server's startup cycle. For more information, see "Provide User Credentials to Start and Stop Servers" in *Oracle Fusion Middleware Managing Server Startup and Shutdown for Oracle WebLogic Server*.
user_staged_config\	readme.txt	File providing information about the directory, which initially serves as a placeholder for configuration information optionally staged by an administrator to be copied to managed servers in the domain.

2.9.2 Resources and Services Configured

The following table identifies the resources and services configured in a domain extended with the WebLogic Server Default Domain extension template.

Table 2–18 Resources Configured in a WebLogic Server Default Domain

Resource Type	Name	Extension Result
Administration Server	AdminServer	Uses the Administration Server provided in the base WebLogic Server domain. The default name is AdminServer, unless changed during domain creation. The Administration Server referenced in the extension template is examplesServer. For information about naming the Administration Server during domain creation, see Section 2.1.2, "Resources and Services Configured for WebLogic Server Domain Template."
Application Deployments	ejb20BeanMgedEar	Adds the application and targets it to the Administration Server, AdminServer.
Application Deployments	examplesWebApp	Adds the application and targets it to the Administration Server, AdminServer.
Application Deployments	jdbcRowSetsEar	Adds the application and targets it to the Administration Server, AdminServer.
Application Deployments	jspSimpleTagEar	Adds the application and targets it to the Administration Server, AdminServer.
Application Deployments	mainWebApp	Adds the application and targets it to the Administration Server, AdminServer.
Application Deployments	webappCachingEar	Adds the application and targets it to the Administration Server, AdminServer.
Application Deployments	webservicesJwsSimpleEar	Adds the application and targets it to the Administration Server, AdminServer.
Application Deployments	xmlBeanEar	Adds the application and targets it to the Administration Server, AdminServer.
JDBC Data Sources	examples-demo	Identifies the JDBC data source as an examples-demo system resource.
JDBC Data Sources	examples-demoXA	Identifies the JDBC data source as an examples-demoXA system resource.
JDBC System Resources	examples-demo examples-demoXA	Identifies the JDBC data source and connection pool setups to be used for non-XA and XA JDBC system resources and targets them to the Administration Server, AdminServer.
Security realm	myrealm	Uses the security realm provided by the base WebLogic Server domain.

2.10 WebLogic Server Examples Extension Template

Using the Configuration Wizard or WLST, you can easily extend a base WebLogic Server domain to create a WebLogic Server Examples domain. You accomplish this by adding the resources and services provided in both the WebLogic Server Default and WebLogic Server Examples extension templates to a base WebLogic Server domain.

For more information about the samples that are supported in the WebLogic Server Examples domain, see Sample Application Examples and Tutorials for WebLogic Server 10.3.

2.10.1 Generated Domain Output

The WebLogic Server Examples domain contains a collection of examples that illustrate best practices for coding individual J2EE APIs, and a set of scripts to run those examples. Once the WebLogic Server Default extension template has been applied to a base domain, applying the WebLogic Server Examples extension template enables you to create the WebLogic Server Examples domain. See Section 1.5, "Relationships Between Templates" for more details.

Table 2–19 Base Domain After Applying the WebLogic Server Default and WebLogic Server Examples Extension Templates

Directory	File	Description
user_projects\applications\base_domain\	empty	empty
server\	wls_samples_overview.html	File that opens the WebLogic Server examples online documentation viewer.
server\docs\	Various	Directory and files supporting the WebLogic Server examples online documentation viewer.
server\examples\build\	Various	Includes sub-directories containing various Java and XML files used to build and work with WebLogic Server examples.
server\examples\src\	Various	Includes sub-directories containing various Java, XML, and HTML files used to work with WebLogic Server examples.
user_projects\domains\base_domain\	empty	empty
empty	client2certs.pem clientkey.pem	Demo certificate and keystore files.
empty	fileRealm.properties	File containing ACLs, users, and groups that can be used for the default security realm when Compatibility security is used.
empty	pointbase.ini	File containing initialization information for a PointBase JDBC database.
empty	setExamplesEnv.cmd setExamplesEnv.sh	Scripts that set up the environment to use the WebLogic Server Examples on Windows and UNIX systems, respectively.
empty	startWebLogic.cmd startWebLogic.sh	Scripts used to start the Administration Server on Windows and UNIX systems, respectively.
empty	startWebLogicEx.cmd startWebLogicEx.sh	Scripts used to start the Administration Server for the WebLogic Server Examples domain on Windows and UNIX systems, respectively.
autodeploy\	readme.txt	File providing information about the directory, which initially serves as a placeholder for automatic deployments.
bin\	setDomainEnv.cmd setDomainEnv.sh	Scripts used to set up the development environment on Windows and UNIX systems, respectively.
bin\	startManagedWebLogic.cmd startManagedWebLogic.sh	Scripts used to start a Managed Server on Windows and UNIX systems, respectively.

Table 2–19 (Cont.) Base Domain After Applying the WebLogic Server Default and WebLogic Server Examples Extension Templates

Directory	File	Description
bin\	startPointBaseConsole.cmd startPointBaseConsole.sh	Scripts used to start the PointBase console on Windows and UNIX systems, respectively.
bin\	startWebLogic.cmd startWebLogic.sh	Scripts used to start the Administration Server on Windows and UNIX systems, respectively.
bin\	stopManagedWebLogic.cmd stopManagedWebLogic.sh	Scripts used to stop a Managed Server on Windows and UNIX systems, respectively.
bin\	stopWebLogic.cmd stopWebLogic.sh	Scripts used to stop the Administration Server on Windows and UNIX systems, respectively.
config\	config.xml	File containing the configuration information used by the Administration Server. For more information, see "Domain Configuration Files" in *Oracle Fusion Middleware Understanding Domain Configuration for Oracle WebLogic Server*.
config\deployments\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for staging an application when the application's staging mode is "staged."
config\diagnostics\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing the system modules associated with instrumentation in the WebLogic Diagnostic Framework (WLDF).
config\jdbc\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing global JDBC modules that can be configured directly from JMX (as opposed to JSR-88).
config\jdbc\	examples-demo-jdbc.xml	Global non-XA JDBC Data Source module for the WebLogic Server Examples domain.
config\jdbc\	examples-demoXA-2-jdbc.xml examples-demoXA-jdbc.xml examples-multiDataSource-demoXAPool-jdbc.xml examples-oracleXA-jdbc.xml	Global XA JDBC Data Source modules for the WebLogic Server Examples domain.
config\jms\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing global JMS modules that can be configured directly from JMX (as opposed to JSR-88).
config\jms\	examples-jms.xml	Global JMS module for the WebLogic Server Examples domain.

Table 2–19 (Cont.) Base Domain After Applying the WebLogic Server Default and WebLogic Server Examples Extension Templates

Directory	File	Description
config\lib\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing JAR files that are added to the system classpath of the server when the server's Java Virtual Machine starts.
config\nodemanager\	nm_password.properties	File containing Node Manager password property values.
config\security\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing system modules for the security framework. The directory contains one security provider configuration extension for each type of security provider in the domain's current realm.
config\startup\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing system modules that contain startup plans. Startup plans are used to generate shell scripts that can be used as part of server startup.
console-ext\	readme.txt	File providing information about the directory, which initially serves as a placeholder for custom extensions to the WebLogic Server Administration Console.
init-info\	domain-info.xml	File used to identify domain creation and extension information. Such information includes the identity of the components in the domain, the location of the JDK and applications directory used by the domain, and the templates used to create and extend the domain.
init-info\	security.xml	File used for creating user groups and roles that establish identity and access to domain resources.
init-info\	startscript.xml	File used to create the *.cmd and *.sh files that are placed into the domain's root and bin directories.
init-info\	tokenValue.properties	File that contains the actual values to substitute for the tokens specified in the start scripts.
lib\	readme.txt	File providing information about the directory, which initially serves as a placeholder for the domain's libraries. The JAR files in this directory are added dynamically to the end of the server classpath at server startup.

Table 2–19 (Cont.) Base Domain After Applying the WebLogic Server Default and WebLogic Server Examples Extension Templates

Directory	File	Description
security\	DefaultAuthenticatorInit.ldift DefaultAuthorizerInit.ldift DefaultRoleMapperInit.ldift XACMLAuthorizerInit.ldift XACMLRoleMapperInit.ldift	Files used for bootstrapping tasks, including authentication (user and group), authorization, and role mapping. These files contain LDAP-specific information. Note: WebLogic domains created with this release use the XACML providers, by default. These XACML security providers are compatible with policies and roles created using the WebLogic Authorization provider (DefaultAuthorizer) and WebLogic Role Mapping provider (DefaultRoleMapper). For more information, see "WebLogic Security Providers" in *Oracle Fusion Middleware Understanding Security for Oracle WebLogic Server*.
security\	SerializedSystemIni.dat	File containing encrypted security information.
servers\AdminServer\security\	boot.properties	File containing server startup properties, including the user name and password required to start the server (in encrypted format). It is generated only when you select development startup mode. This file enables you to bypass the prompt for user name and password during a server's startup cycle. For more information, see "Provide User Credentials to Start and Stop Servers" in *Oracle Fusion Middleware Managing Server Startup and Shutdown for Oracle WebLogic Server*.
user_staged_config\	readme.txt	File providing information about the directory, which initially serves as a placeholder for configuration information optionally staged by an administrator to be copied to managed servers in the domain.
WseeFileStore\	n.a.	Directory to be used for the file store for system resources.

2.10.2 Resources and Services Configured

The following table identifies the resources and services configured in a domain extended with the WebLogic Server Examples extension template.

Table 2–20 Resources Configured in a WebLogic Server Examples Domain

Resource Type	Name	Extension Result
Administration Server	AdminServer	Uses the Administration Server provided in the base WebLogic Server domain. The default name is AdminServer, unless changed during domain creation. The Administration Server referenced in the extension template is examplesServer. For information about naming the Administration Server during domain creation, see Section 2.1.2, "Resources and Services Configured for WebLogic Server Domain Template."
Application Deployments	ejb20BeanMgedEar	Uses the application provided by the WebLogic Server Default extension template applied to the base WebLogic Server domain.
Application Deployments	examplesWebApp	Uses the application provided by the WebLogic Server Default extension template applied to the base WebLogic Server domain.
Application Deployments	jdbcRowSetsEar	Uses the application provided by the WebLogic Server Default extension template applied to the base WebLogic Server domain.
Application Deployments	jspSimpleTagEar	Uses the application provided by the WebLogic Server Default extension template applied to the base WebLogic Server domain.
Application Deployments	mainWebApp	Uses the application provided by the WebLogic Server Default extension template applied to the base WebLogic Server domain.
Application Deployments	SamplesSearchWebApp	Adds the application and targets it to the Administration Server, AdminServer.
Application Deployments	webappCachingEar	Uses the application provided by the WebLogic Server Default extension template applied to the base WebLogic Server domain.
Application Deployments	webservicesJwsSimpleEar	Uses the application provided by the WebLogic Server Default extension template applied to the base WebLogic Server domain.
Application Deployments	xmlBeanEar	Uses the application provided by the WebLogic Server Default extension template applied to the base WebLogic Server domain.
File Store	WseeFileStore	Adds the file store to be used as the persistent store for the JMS server, WseeJMSServer, and the SAF Agent, ReliableWseeSAFAgent, and targets the store to the Administration Server, AdminServer.
JDBC Data Sources	examples-demo examples-demoXA	Uses the non-XA and XA JDBC data sources provided by the WebLogic Server Default extension template applied to the base WebLogic Server domain.
JDBC Data Sources	examples-oracleXA	Adds the XA JDBC data source and targets it to the Administration Server, AdminServer.
JDBC Data Sources	examples-demoXA-2	Adds the XA JDBC data source and targets it to the Administration Server, AdminServer.

Table 2–20 (Cont.) Resources Configured in a WebLogic Server Examples Domain

Resource Type	Name	Extension Result
JDBC Data Sources	examples-multiDataSource-demoXAPool	Adds the XA JDBC multi data source and targets it to the Administration Server, AdminServer. Maps to examples-demoXA and examples-demoXA-2 data sources.
JDBC Store	exampleJDBCStore	Adds the JDBC store to be used as the persistent store for the JDBC data source, examples-demo, and the JMS server, examplesJMSServer, and targets the store to the Administration Server, AdminServer.
JDBC System Resources	examples-demo examples-demoXA	Uses the JDBC data source and connection pool setups provided by the WebLogic Server Default extension template applied to the base WebLogic Server domain.
JDBC System Resources	examples-demoXA-2 examples-oracleXA examples-multiDataSource-demoXAPool	Adds the JDBC data source and connection pool setups and targets them to the Administration Server, AdminServer.
JMS System Resources	examples-jms	Identifies the JMS servers, connection factories, queues, and topics to be used for JMS system resources.
JMS Connection Factories	exampleTopic exampleTrader weblogic.examples.jms.QueueConnectionFactory	Adds the JMS connection factories as examples-jms system resources and targets them to the Administration Server, AdminServer.
JMS Servers	examplesJMSServer	Adds the JMS server as an examples-jms system resource and targets it to the Administration Server, AdminServer.
JMS Servers	WseeJMSServer	Adds the JMS server as an examples-jms system resource and targets it to the Administration Server, AdminServer.
JMS Queues	exampleQueue	Adds the JMS queue to the JMS server, examplesJMSServer.
JMS Queues	jms/MULTIDATASOURCE_MDB_QUEUE	Adds the JMS queue to the JMS server, examplesJMSServer.
JMS Queues	weblogic.wsee.wseeExamplesDestinationQueue	Adds the JMS queue to the JMS server, WseeJMSServer.
JMS Topics	exampleTopic	Adds the JMS topic to the JMS server, examplesJMSServer.
JMS Topics	quotes	Adds the JMS topic to the JMS server, examplesJMSServer.
SAF Agent	ReliableWseeSAFAgent	Adds the SAF agent and targets it to the Administration Server, AdminServer.
Security realm	myrealm	Uses the security realm provided by the base WebLogic Server domain.