Oracle® Fusion Middleware

Domain Template Reference

12*c* Release 1 (12.1.1)

E24495-02

January 2012

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

ORACLE

Oracle Fusion Middleware Domain Template Reference, 12*c* Release 1 (12.1.1)

E24495-02

Contents

Preface

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

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

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 Fusion Middleware domain and extension templates, which are Java Archive (JAR) files that contain the files and scripts required to create or extend a WebLogic 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 WebLogic Server product installation includes a predefined Basic WebLogic Server Domain template. This template defines the core set of resources within a WebLogic domain, including an Administration Server and basic configuration information. For more information on the Basic WebLogic Server Domain template, see Section 2.1, "Basic WebLogic Server Domain Template."

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

- *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 WebLogic Server product installation includes several predefined extension templates. The templates that are available to you in the Configuration Wizard depend on the product you are installing. WebLogic Server installations include the templates described in Section 2, "WebLogic Server Templates."

  You can create a custom extension template from an existing domain or template by 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 *Creating Templates and Domains Using the Pack and Unpack Commands*.

1.2 Location of Installed WebLogic Server Templates

The following table identifies the location of the predefined template JAR files provided with the WebLogic Server 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, which is referred to as the Configuration Wizard framework.

- For information about using the Configuration Wizard, see *Creating Domains Using the Configuration Wizard*.
- For information about using the WLST Offline, see *Oracle WebLogic Scripting Tool*.
- For information about using the `pack/unpack` commands, see *Creating Templates and Domains Using the Pack and Unpack Commands*.

- For information about using the Domain Template Builder, see *Creating Domain Templates Using the Domain Template Builder*.

1.4 Template Dependencies

WebLogic Server resources must be set up in your domain before you can add resources from an extension template. This is known as a template dependency. For example, all extension templates provided with your product are dependent on, at the very least, the Administration Server and security realm resources that are configured by the Basic WebLogic Server Domain template. Other extension templates depend on resources from multiple templates. For example, to extend a domain to support the WebLogic Server Examples, the existing domain must already contain the resources from the Basic WebLogic Server Domain template and the WebLogic Server Default Domain extension template.

When you select an extension template by selecting the associated product (if listed) on the Configuration Wizard Select Domain Source or Select Extension Source screens, the Configuration Wizard automatically selects all other products that configure resources required by the product you selected.

When you select a template by browsing to and selecting the JAR file, a Dependency warning is displayed if the template depends on resources that are provided by other domain or extension templates that have not yet been configured in the domain you are extending.

1.5 Files Typically Included in a Template

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

Table 1–3 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. There is one XML file for each JDBC resource in the domain. These files are present only if the domain includes JDBC resources.
*-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. This is applicable only if the domain requires JMS resources.

Table 1–3 (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." This file is not required. When used, it must be located in the `script` directory. If it is not present, default targeting is used.
config.xml	Defines the resources that the template creates or adds to a domain. In a template, the config.xml file must be located in the config directory.
config-groups.xml	This file contains definitions of applications, services, servers, clusters, and mappings that create a relationship among these items. It enables movement of functionally related applications and services as a single operation when transitioning from one topology to another (for example, from a single server to multiple servers, or from a single server to a cluster). This ensures that all application and service dependencies are met when scaling a domain configuration. **Note:** Do not modify this file in any way. It must be used as provided in the template. An Application/Service group specifies a set of functionally related applications and services. The applications and services are grouped together on a particular server or cluster. The Domain Topology section contains definitions of servers, as well as the targeting of applications and services to a specific server, group or servers or clusters. It contains the following definitions: ■ **Server group definitions**—Specifies a server or servers that can house functionally related sets of applications and services, thereby enabling automatic server creation. ■ **Cluster group definitions**—Specifies a cluster that can house functionally related sets of applications and services, thereby enabling automatic cluster creation. ■ **Application/Service group mapping definitions**—Specifies targeting of an Application/Service group to a specific server, group of servers, or cluster, via the name of the Application/Service group.

Table 1–3 (Cont.) Files Included in a Template

Filename	Description
database.xml	This file is included only in Fusion Middleware product templates that require JDBC data source definitions. It groups data sources into component schemas that are required to configure and load data into database objects via the Oracle Repository Creation Utility (RCU). It also contains the eligible database vendors and drivers, eliminating the possibility of selecting an unsupported database in the Fusion Middleware Configuration Wizard. **Note:** Do not modify this file in any way. It must be used as provided in the template.
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 *WL_HOME*\common\templates\applications directory.
jvm-config.xml	This file is specific to FMW product installations in a WebSphere environment, and can be ignored in a WebLogic Server environment.
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 file in a *domain* template. However, you can create user groups and roles through the security.xml file included in either a domain or an extension template.
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 WebLogic Server installation directory.

Table 1–3 (Cont.) Files Included in a Template

Filename	Description
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. This file also includes template dependency information (if applicable).
was-variable.xml	This file is specific to FMW product installations in a WebSphere environment, and is ignored in a WebLogic Server environment.

2

WebLogic Server Templates

This chapter describes the following WebLogic domain templates that are provided with your WebLogic Server installation. You can create or extend domains by selecting these templates on the Select Domain Source or Select Extension Source screens of the Oracle Fusion Middleware Configuration Wizard.

Table 2–1 Oracle WebLogic Server and Workshop for WebLogic Templates

Template	Description
Basic WebLogic Server Domain Template	Creates a base WebLogic Server domain.
WebLogic Server Starter Domain Template	Creates a WebLogic Server starter domain.
WebLogic Advanced Web Services Extension Template	Extends an existing WebLogic Server domain to add functionality required for advanced Web services, including WSRM, buffering, and JMS transport.
WebLogic Advanced Web Services for JAX-WS Extension Template	Extends a domain to add functional required for advanced Web services, including asynchronous messaging, Web services reliable messaging, message buffering, Web services atomic transactions, and security using WS-SecureConversation.
Avitek Medical Records Sample Domain Template	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)	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.
WebLogic Server Default Domain Extension Template	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	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.

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 Template Details

The following table provides basic information about the Basic WebLogic Server Domain template. Template name is the name of the template as it is shown in the product list on the Configuration Wizard Select Domain Source and Select Extension Source screens.

Table 2–2 Basic WebLogic Server Domain Template Information

Template Detail	Information
Template type	Domain
Template name	Basic WebLogic Server Domain
Template JAR file and location	*WL_HOME*\common\templates\domain\wls.jar
Template Dependencies	None

2.1.2 Resources and Services Configured in a Basic WebLogic Server Domain

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

Table 2–3 Resources Configured in a Basic WebLogic Server Domain

Resource Type	Name	Notes

Administration Server	AdminServer	The default configuration for the Administration Server is as follows:
		■ Listen address: All Local Addresses
		■ Listen port: 7001
		■ SSL is disabled
		When using the Configuration Wizard or WLST Offline to create a domain, if 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" *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 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	This is the default (active) WebLogic Server security realm. The administration user account, `weblogic`, is configured in this security realm.

2.1.3 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 2–4 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\	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 *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\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 2–4 (Cont.) Output Generated from the Basic WebLogic Server Domain Template

Directory	File/s	Description
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\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.
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 2–4 (Cont.) Output Generated from the Basic WebLogic Server Domain Template

Directory	File/s	Description
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 *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 *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.

2.2 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 this predefined template with the Configuration Wizard or WLST.

2.2.1 Template Details

The following table provides basic information about the WebLogic Server Starter Domain template.

Table 2–5 WebLogic Server Starter Domain Template Information

Template Detail	Information
Template type	Domain

Table 2–5 (Cont.) WebLogic Server Starter Domain Template Information

Template Detail	Information
Template name in Configuration Wizard	This template is not available from the list of products in the Configuration Wizard. You can select it only by using the **Browse** option on the Select Domain Source or Select Extension Source screens, and navigating to the JAR file location.
Template JAR file and location	*WL_HOME*\common\templates\domain\wls_starter.jar
Template Dependencies	None

2.2.2 Resources and Services Configured in a WebLogic Server Starter Domain

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

Table 2–6 Resources Configured in a WebLogic Server Starter Domain

Resource Type	Name	Notes

Administration Server	AdminServer	The default configuration for the Administration Server is as follows:
		■ Listen address: All Local Addresses
		■ Listen port: 7001
		■ SSL is disabled
		When using the Configuration Wizard or WLST Offline to create a domain, if 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" *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 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	The default (active) WebLogic Server security realm.
Application Deployments	wls_starter	A sample Web application deployed to the starter domain.

2.2.3 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 2–7 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\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\	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 *Understanding Domain Configuration for Oracle WebLogic Server*.

Table 2–7 (Cont.) Output Generated from the WebLogic Server Starter Domain Template

Directory	File	Description
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\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\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.

Table 2–7 (Cont.) Output Generated from the WebLogic Server Starter Domain Template

Directory	File	Description
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 *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 *Managing Server Startup and Shutdown for Oracle WebLogic Server*.

2.3 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.

2.3.1 Template Details

The following table provides basic information about the WebLogic Advanced Web Services extension template.

Table 2–8 WebLogic Advanced Web Services Extension Template Information

Template Detail	Information
Template type	Extension
Template name in Configuration Wizard	WebLogic Advanced Web Services Extension
Template JAR file and location	*WL_HOME*\common\templates\applications\wls_webservice.jar
Template Dependencies	Basic WebLogic Server Domain template

2.3.2 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 2–9 Resources Configured in a WebLogic Advanced Web Services Domain

Resource Type	Name	Extension Result
Administration Server	AdminServer	Uses the Administration Server provided in the Basic WebLogic Server domain. The default name is AdminServer, unless changed during domain creation. The Administration Server referenced in this extension template is cgServer. The default configuration for the Administration Server is as follows: ■ Listen address: All Local Addresses ■ Listen port: 7001 ■ SSL is enabled ■ SSL listen port: 7002 For information about naming the Administration Server during domain creation, see Section 2.1.2, "Resources and Services Configured in a Basic WebLogic Server Domain."
Security realm	myrealm	Uses the default security realm provided by the Basic WebLogic Server domain.
File Store	WseeFileStore	Adds the file store to be used as the persistent store for the WseeJmsServer JMS server. This file store is targeted to the Administration Server.
SAF Agent	ReliableWseeSAFAgent	Adds this store-and-forward agent, which uses the WseeFileStore, and targets it to the Administration Server. The SAF agent controls receipt and handling of reliable messages.
JMS Queues	WseeMessageQueue	Adds the JMS queue to the JMS server, WseeJmsServer. Queues are under JMSModules/module name, these are under WseeJMSModule
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.

2.3.3 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 2–10 Base Domain After Applying the WebLogic Advanced Web Services Extension Template

Directory	File	Description
user_projects\applications\base_domain\	Not applicable	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\	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 domain 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\	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 *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."

Table 2–10 (Cont.) Base Domain After Applying the WebLogic Advanced Web Services Extension Template

Directory	File	Description
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\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.

Table 2–10 (Cont.) Base Domain After Applying the WebLogic Advanced Web Services Extension Template

Directory	File	Description
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 *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 *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 WebLogic Advanced Web Services for JAX-WS Extension Template

The WebLogic Advanced Web Services for JAX-WS extension template automatically configures the resources required to support the following advanced Web services features:

- Web services atomic transactions

- Security using WS-SecureConversation

> **Note:** Each of the two Advanced Web Services templates can be used individually or together in a domain. If, however, you apply this template to the same domain to which you applied the WebLogic Advanced Web Services extension template, you must apply the Advanced Web Services template before applying the Advanced Web Services for JAX-WS template.

For more information, see "Configuring Your Domain for Advanced Web Services Features" in *Getting Started With JAX-WS Web Services for Oracle WebLogic Server*.

2.4.1 Template Details

The following table provides basic information about the WebLogic Advanced Web Services for JAX-WS extension template.

Table 2–11 WebLogic Advanced Web Services for JAX-WS Extension Template Details

Template Detail	Information
Template type	Extension
Template name in Configuration Wizard	WebLogic Advanced Web Services for JAX-WS Extension - 10.3.3.0
Template JAR file and location	*WL_HOME*\common\templates\applications\wls_webservice_jaxws.jar
Template Dependencies	Basic WebLogic Server Domain template

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 for JAX-WS extension template.

Table 2–12 Resources Configured in a WebLogic Advanced Web Services for JAX-WS Domain

Resource Type	Name	Extension Result
Administration Server	AdminServer	Uses the Administration Server provided in the Basic WebLogic Server domain. The default name is AdminServer, unless changed during domain creation. The Administration Server referenced in this extension template is cgServer. The default configuration for the Administration Server is as follows: ■ Listen address: All Local Addresses ■ Listen port: 7001 ■ SSL is enabled ■ SSL listen port: 7002 For information about naming the Administration Server during domain creation, see Section 2.1.2, "Resources and Services Configured in a Basic WebLogic Server Domain."

Table 2–12 (Cont.) Resources Configured in a WebLogic Advanced Web Services for JAX-WS Domain

Resource Type	Name	Extension Result
Security realm	myrealm	Uses the default security realm provided by the Basic WebLogic Server domain.
JMS Server	WseeJaxwsJmsServer	Adds the JMS server as a system resource and targets it to the Administration Server.
File Store	WseeJaxwsFileStore	Adds the file store to be used as the persistent store for the WseeJaxwsJmsServer JMS server. This file store is targeted to the Administration Server.
JMS System Resource	WseeJaxwsJmsModule	Defines a distributed destination for the cluster. All associated targets will be used to support JAX-WS Web services. The subdeployment name is WseeJaxwsJmsServerSub. **Note**: By default, a weighted distributed destination (WDD) is configured. In a clustered environment, Oracle strongly recommends that you upgrade the destination to a uniform distributed destination (UDD). For information about how to change the distributed destination type using Configuration Wizard, see "Select JMS Distributed Destination Type" in *Creating Domains Using the Configuration Wizard*
JMS Queues	weblogic.wsee.BufferedRequest Queue	Adds these JMS queues to the JMS server, and targets them to WseeJaxwsJmsServer.
	weblogic.wsee.BufferedRequestErrorQueue	These queues are reserved for future use.
	weblogic.wsee.BufferedResponseQueue	
	weblogic.wsee.BufferedResponseErrorQueue	
SAF Agent	ReliableWseeJaxwsSAFAgent	Adds this store-and-forward agent, which uses the WseeJaxwsFileStore, and targets it to the Administration Server. The SAF agent controls receipt and handling of reliable messages.
Work Manager	weblogic.wsee.jaxws.mdb.DispatchPolicy	Adds this Work Manager and targets it to the Administration Server. The Work Manager defines the thread pool resources.

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 "Sample Application and Code Examples" in *Understanding Oracle WebLogic Server*.

2.5.1 Template Details

The following table provides basic information about the Avitek Medical Records Sample domain extension template.

Table 2–13 Avitek Medical Records Sample Domain Information

Template Detail	Information
Template type	Extension
Template name in Configuration Wizard	This template is not available from the list of products in the Configuration Wizard. You can select it only by using the **Browse** option on the Select Domain Source or Select Extension Source screens, and navigating to the JAR file location.
Template JAR file and location	*WL_HOME*\common\templates\applications\medrec.jar
Template Dependencies	Basic WebLogic Server Domain template

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.

A Work Manager service (weblogic.wsee.mdb.DispatchPolicy) is also available, but it is not targeted to the Administration Server.

Table 2–14 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. The default configuration for the Administration Server is as follows: ■ Listen address: All Local Addresses ■ Listen port: 7001 ■ SSL is enabled ■ SSL port: 7002 For information about naming the Administration Server during domain creation, see Section 2.1.2, "Resources and Services Configured in a Basic WebLogic Server Domain."
Security realm	myrealm	Uses the default security realm provided in the Basic WebLogic Server domain.
Application Deployments	browser-starter	Adds the browser-starter Web application and targets it to the MedRecServer.
Application Deployments	medrec	Adds the sample medrec Enterprise application and targets it to the MedRecServer.
Application Deployments	physician	Adds the sample physician Enterprise application and targets it to the MedRecServer.
JDBC Data Sources	MedRecGlobalDataSourceXA	Identifies the JDBC data source as a MedRecGlobalDataSourceXA system resource.

Table 2–14 (Cont.) Resources Configured in an Avitek Medical Records Domain

Resource Type	Name	Extension Result
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 Server	MedRecJMSServer	Adds the JMS server as a MedRec-jms system resource and targets it to the MedRecServer
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 MedRecServer.
Mail Session	mail/MedRecMailSession	Adds this mail session.
SAF Agent	WsrmAgent	Adds this store-and-forward agent, which uses the MedRecWseeFileStorfile store, and targets it to the MedRecServer.
Deployed library	jsf1.2@1.2.9.0	Adds the Java Server Faces Version 1.2 libraries.
Deployed library	jstl1.2.@1.2.0.1	Adds the Java standard tagging (JSTL) Version 1.2 libraries.
WLDF System Resource	MedRecWLDF	Adds the WLDF system resource and defined WLDF instrumentation monitors for dye injection, and targets them to the MedRecServer.

2.5.3 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 2–15 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 subdirectories containing library files supporting the Avitek Medical Records sample.
user_projects\applications\base_domain\modules\	Various	Includes subdirectories containing Avitek Medical Records source code including various Java, XML, JSP, HTML files, and so on.

Table 2–15 (Cont.) Base Domain After Applying the Avitek Medical Records Sample Extension Template

Directory	File	Description
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\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\	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 *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.

Table 2–15 (Cont.) Base Domain After Applying the Avitek Medical Records Sample Extension Template

Directory	File	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\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.
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\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.

Table 2–15 (Cont.) Base Domain After Applying the Avitek Medical Records Sample Extension Template

Directory	File	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 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 *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 *Managing Server Startup and Shutdown for Oracle WebLogic Server*.

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 "Sample Application and Code Examples" in *Understanding Oracle WebLogic Server*.

2.6.1 Template Details

The following table provides basic information about the WebLogic Advanced Web Services Extension template. Template name is the name of the template as it is shown in the product list on the Configuration Wizard Select Domain Source and Select Extension Source screens.

Table 2–16 Avitek Medical Records Sample Domain (Spring) Information

Template Detail	Information
Template type	Extension
Template name	This template is not available from the list of products in the Configuration Wizard. You can select it only by using the **Browse** option on the Select Domain Source or Select Extension Source screens, and navigating to the JAR file location.
Template JAR file and location	*WL_HOME*\common\templates\applications\medrec_spring.jar
Template Dependencies	Basic WebLogic Server Domain template

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 2–17 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. The default configuration for the Administration Server is as follows: ■ Listen address: All Local Addresses ■ Listen port: 7001 ■ SSL is enabled ■ SSL port: 7002 For information about naming the Administration Server during domain creation, see Section 2.1.2, "Resources and Services Configured in a Basic WebLogic Server Domain."
Security realm	myrealm	Uses the security realm provided in the base WebLogic Server domain.
Application Deployments	browser-starter	Adds the browser-starter Web application and targets it to the MedRecServer.
Application Deployments	medrec	Adds the sample medrec Enterprise application and targets it to the MedRecServer.
Application Deployments	physician	Adds the sample physician Enterprise application and targets it to the MedRecServer.

Table 2–17 (Cont.) Resources Configured in an Avitek Medical Records Domain for

Resource Type	Name	Extension Result
JDBC Data Sources	MedRecGlobalDataSourceXA	Identifies the JDBC data source as a MedRecGlobalDataSourceXA system resource. Pool capacity (initial): 2 Pool capacity (maximum): 10 Protocol: Two Phase Commit
JMS Queues	com.oracle.medrec.jms.RecordToCreateQueue com.oracle.medrec.jms.PatientNotificationQueue weblogic.wsee.DefaultQueue	Adds these JMS queues to the JMS server, MedRecWseeJMSServer.
JMS Servers	MedRecJMSServer	Adds the JMS server as a MedRec-jms system resource and targets it to the MedRecServer.
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 MedRecServer.
Mail Session	mail/MedRecMailSession	Adds this mail session.
SAF Agent	WsrmAgent	Adds this store-and-forward agent, which uses the file store, MedRecWseeFileStore, and targets it to the MedRecServer.
Deployed library	jsf1.2@1.2.9.0	Adds the Java Server Faces Version 1.2 library and targets it to the MedRecServer.
Deployed library	jstl1.2.@1.2.0.1	Adds the Java standard tagging (JSTL) Version 1.2 library and targets it to the MedRecServer.
Deployed library	weblogic-spring#10.3.6.0@10.3.6.0	Adds the WebLogic Spring Version 10.3.6 library and targets it to the MedRecServer.
WLDF System Resource	MedRecWLDF	Adds this WLDF system resource, and targets it to the MedRecServer. The WLDF resource defines an instrumentation monitor for dye injection, and a harvester metric (com.oracle.medrec.admin.AdminReport).

2.6.3 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 2–18 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.

2.7 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 more information about the samples that are supported in the WebLogic Server Examples domain, see "Sample Application and Code Examples" in *Understanding Oracle WebLogic Server*.

2.7.1 Template Details

The following table provides basic information about the WebLogic Server Default Domain Extension template.

Template Dependencies lists all templates that provide resources required by the WebLogic Server Default Domain extension template.

Table 2–19 WebLogic Server Default Domain Information

Template Detail	Information
Template type	Extension
Template name in Configuration Wizard	This template is not available from the list of products in the Configuration Wizard. You can select it only by using the **Browse** option on the Select Domain Source or Select Extension Source screens, and navigating to the JAR file location.
Template JAR file and location	*WL_HOME*\common\templates\applications\wls_default.jar
Template Dependencies	Basic WebLogic Server Domain template

2.7.2 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 2–20 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. The default configuration for the Administration Server is as follows: ■ Listen address: All Local Addresses ■ Listen port: 7001 ■ SSL is disabled For information about naming the Administration Server during domain creation, see Section 2.1.2, "Resources and Services Configured in a Basic WebLogic Server Domain."
Security realm	myrealm	Uses the security realm provided by the base WebLogic Server domain.
Application Deployment	mainWebApp	Adds the mainWebApp application and targets it to the examplesServer Administration Server.
Application Deployment	examplesWebApp	Adds the examplesWebApp application and targets it to the examplesServer Administration Server.
Application Deployment	ejb20BeanMgedEar	Adds the ejb20BeanMgedEar application and targets it to the examplesServer Administration Server.
Application Deployment	ejb30	Adds the ejb30 application and targets it to the examplesServer Administration Server.
Application Deployment	stockEar	Adds the stockear application and targets it to the examplesServer Administration Server.
Application Deployment	asyncServletEar	Adds the asyncServletEar Enterprise application and targets it to the examplesServer Administration Server.
Application Deployment	extServletAnnotationsEar	Adds the extServletAnnotationsEar application and targets it to the examplesServer Administration Server.
Application Deployment	jdbcRowSetsEar	Adds the jdbcRowSetsEar application and targets it to the examplesServer Administration Server.
Application Deployment	jspSimpleTagEar	Adds the jspSimpleTagEar application and targets it to the examplesServer Administration Server.
Application Deployment	webservicesJwsSimpleEar	Adds the webservicesJwsSimpleEar application and targets it to the examplesServer Administration Server.
Application Deployment	xmlBeanEar	Adds the xmlBeanEar application and targets it to the examplesServer Administration Server.

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

Resource Type	Name	Extension Result
JDBC System Resource	examples-demo	Identifies this JDBC data source, which has the following configuration: ■ JNDI name: examples-dataSource-demoPool ■ Global transaction protocol: Two Phase Commit The connection pool settings are: ■ Initial capacity: 1 ■ Maximum capacity: 10
JDBC System Resource	examples-demoXA	Identifies this JDBC data source, which has the following configuration: ■ JNDI Name: examples-dataSource-demoXAPool ■ Global transaction protocol: Two Phase Commit The connection pool settings are: ■ Initial capacity: 2 ■ Maximum capacity: 10
Deployed library	pubsub#1.0@1.6.0.0	Adds the HTTP PublisherSubscriber Version 1.6.0.0 library and targets it to the examplesServer Administration Server.
Deployed library	weblogic-sca#1.0@1.1.0.0	Adds the WebLogic SCA Version 1.1.0.0 library and targets it to the examplesServer Administration Server.
Deployed library	apache-xbean.jar	Adds the apache-xbean.jar library dependency to this domain.

2.7.3 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 2–21 Base Domain After Applying the WebLogic Server Default Domain Extension Template

Directory	File	Description
user_projects\applications\base_domain\server\docs\	Various	Includes sub-directories containing the style sheet and graphics files to support the online documentation.
user_projects\applications\base_domain\server\examples\build\	Various	Includes WebLogic Server examples deployments.

Table 2–21 (Cont.) Base Domain After Applying the WebLogic Server Default Domain Extension Template

Directory	File	Description
user_projects\applications\base_domain\server\examples\src\	Various	Includes source code and instructions for WebLogic Server examples.
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\	setExamplesEnv.cmd setExamplesEnv.sh	Scripts that set up the environment to use the WebLogic Server Examples on Windows and UNIX systems, respectively.
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\	startWebLogicEx.cmd startWebLogicEx.sh	Scripts used to start the Administration Server for the WebLogic Server Examples domain 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\	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 *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 2–21 (Cont.) Base Domain After Applying the WebLogic Server Default Domain Extension 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\jdbc\	examples-demo-jdbc.xml	Global non-XA JDBC Data Source module for the WebLogic Server default domain.
user_projects\domains\base_domain\config\jdbc\	examples-demoXA-jdbc.xml	Global XA JDBC Data Source module for the WebLogic Server default 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\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\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 2–21 (Cont.) Base Domain After Applying the WebLogic Server Default Domain Extension Template

Directory	File	Description
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.
user_projects\domains\base_domain\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 *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 *Managing Server Startup and Shutdown for Oracle WebLogic Server*.

Table 2–21 (Cont.) Base Domain After Applying the WebLogic Server Default Domain Extension Template

Directory	File	Description
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.

2.8 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 and Code Examples" in *Understanding Oracle WebLogic Server*.

2.8.1 Template Details

The following table provides basic information about the WebLogic Server Default Domain Extension template.

Template Dependencies lists all templates that provide resources required by the WebLogic Server Examples extension template, in the order in which they must be configured in the domain.

Table 2–22 WebLogic Server Examples Extension Information

Template Detail	Information
Template type	Extension
Template name in Configuration Wizard	This template is not available from the list of products in the Configuration Wizard. You can select it only by using the **Browse** option on the Select Domain Source or Select Extension Source screens, and navigating to the JAR file location.
Template JAR file and location	*WL_HOME*\common\templates\applications\wls_examples.jar
Template Dependencies	■ Basic WebLogic Server Domain template ■ + WebLogic Server Default Domain Extension template

2.8.2 Resources and Services Configured

In addition to the resources configured by the WebLogic Server Default Domain extension template (see Table 2–20), the WebLogic Server Examples extension template configures the resources and services listed in the following table.

Table 2–23 Additional Resources Configured by the WebLogic Server Examples Domain

Resource Type	Name	Notes
Administration Server	AdminServer	Uses the Administration Server provided in the Basic WebLogic Server domain. The default name is AdminServer, unless changed during domain creation. The Administration Server referenced in the extension template is examplesServer. The default configuration for the Administration Server is as follows: ■ Listen address: All Local Addresses ■ Listen port: 7001 ■ SSL is disabled For information about naming the Administration Server during domain creation, see Section 2.1.2, "Resources and Services Configured in a Basic WebLogic Server Domain."
Security realm	myrealm	Uses the security realm provided by the base WebLogic Server domain.
Application Deployment	SamplesSearchWebApp	Adds the application and targets it to the Administration Server.
JMS Server	examplesJMSServer	Adds this JMS server as an examples-jms system resource and targets it to the Administration Server. ■ Persistent Store: exampleJDBCStore ■ JMS Message log file: examplesJMSServer
JMS Server	WseeJMSServer	Adds this JMS server as an examples-jms system resource and targets it to the Administration Server. This server is configured to use the WSeeFileStore persistent store.
File Store	WseeFileStore	Adds the file store to be used as the persistent store for the WSEEJMSServer JMS server and the ReliableWseeSAFAgent SAF Agent. This file store is targeted to the examplesServer Administration Server.
JDBC Store	exampleJDBCStore	Adds the JDBC store to be used as the persistent store for the examples-demo JDBC data source and the examplesJMSServer JMS server, and targets the store to the examplesServer Administration Server.
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 weblogic.examples.ejb30.QueueConnectionFactory	Adds these connection factories as examples-jms system resources and targets them to the examplesServer server.

Table 2–23 (Cont.) Additional Resources Configured by the WebLogic Server Examples

Resource Type	Name	Notes
JMS Queues	exampleQueue jms/MULTIDATASOURCE_MDB_QUEUE weblogic.examples.ejb30.ExampleQueue	Adds these JMS queues to the examplesJMSServer JMS server.
JMS Queue	weblogic.wsee.wseeExamplesDestinationQueue	Adds this JMS queue to the WseeJMSServer JMS server.
JMS Topics	exampleTopic quotes stockTopic	Adds these JMS topics and targets them to the examplesJMSServer JMS server.
JDBC System Resource	examples-oracleXA	Identifies this JDBC data source, which has the following configuration: ■ JNDI name: examples-dataSource-oracleXAPool ■ Global transaction protocol: Two Phase Commit The database driver is configured as oracle.jdbc.xa.client.OracleXADataSource. This data source is mapped to the examples-multiDataSource-demoXAPool multi data source.
JDBC System Resource	examples-demoXA-2	Identifies this JDBC data source, which is targeted to the Administration Server and has the following configuration: ■ Connection pool maximum: 100 ■ Global transaction protocol: Two Phase Commit This data source is mapped to the examples-multiDataSource-demoXAPool multi data source.
JDBC System Resource	examples-multiDataSource-demoXAPool	Identifies this JDBC multi data source, which is targeted to the Administration Server. It is configured for failover, and maps to the examples-oracleXA and examples-demo-XA-2 data sources.
SAF Agent	ReliableWseeSAFAgent	Adds this store-and-forward agent, which uses the WseeFileStore file store, and targets it to the Administration Server.
Work Manager	weblogic.wsee.mdb.DispatchPolicy	Adds this Work Manager, but does not target it to any servers.

2.8.3 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.

Table 2–24 Base Domain After Applying the WebLogic Server Default and WebLogic Server Examples Extension Templates

Directory	File	Description
user_projects\applications\base_domain\server\	wls_samples_overview.html	File that opens the WebLogic Server examples online documentation viewer.
user_projects\applications\base_domain\server\docs\	Various	Directory and files supporting the WebLogic Server examples online documentation viewer.
user_projects\applications\base_domain\server\examples\build\	Various	Includes sub-directories containing various Java and XML files used to build and work with WebLogic Server examples.
user_projects\applications\base_domain\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\	client2certs.pem clientkey.pem	Demo certificate and keystore files.
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\	setExamplesEnv.cmd setExamplesEnv.sh	Scripts that set up the environment to use the WebLogic Server Examples on Windows and UNIX systems, respectively.
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\	startWebLogicEx.cmd startWebLogicEx.sh	Scripts used to start the Administration Server for the WebLogic Server Examples domain 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\	startWebLogic.cmd startWebLogic.sh	Scripts used to start the Administration Server on Windows and UNIX systems, respectively.

Table 2–24 (Cont.) Base Domain After Applying the WebLogic Server Default and WebLogic Server Examples Extension Templates

Directory	File	Description
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 *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\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\	examples-demo-jdbc.xml	Global non-XA JDBC Data Source module for the WebLogic Server Examples domain.
user_projects\domains\base_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.
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\	examples-jms.xml	Global JMS module for the WebLogic Server Examples domain.

Table 2–24 (Cont.) Base Domain After Applying the WebLogic Server Default and WebLogic Server Examples Extension Templates

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\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.
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 2–24 (Cont.) Base Domain After Applying the WebLogic Server Default and WebLogic Server Examples Extension Templates

Directory	File	Description
user_projects\domains\base_domain\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 *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 *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.
user_projects\domains\base_domain\WseeFileStore\	Not applicable.	Directory to be used for the file store for system resources.