Oracle® Fusion Middleware

Domain Template Reference

11*g* Release 1 (10.3.3)

E14138-06

July 2010

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, 11*g* Release 1 (10.3.3)

E14138-06

Contents

3 Fusion Middleware Product Templates

Preface

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

Documentation Accessibility

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

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/support/contact.html or visit http://www.oracle.com/accessibility/support.html 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.

Convention	Meaning
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 Introduction

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

This document contains the following topics:

1.1 Types of Templates

The types of template include:

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

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

  You can 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.

  Your product installation includes several predefined extension templates. The templates that are available to you in 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.
jdbc.index	Identifies the locations of SQL scripts used to set up a database. The file lists the scripts in the order in which they must be run. If the scripts are not contained in the template, but are located in the product installation directory, that directory can be represented by a tilde (~) in the pathname for the scripts, as shown in the following example: `~/integration/common/dbscripts/oracle/reporting_runtime.sql` Specifically, the tilde represents the directory path identified by the $USER_INSTALL_DIR$ variable in the stringsubs.xml file. In a template, a jdbc.index file must be located in the _jdbc_ \dbtype\dbversion directory, where dbtype is the type of database, such as Oracle, and dbversion is the database version, such as 9i. In addition to listing the SQL files related to a data source, the jdbc.index file contains information about the categories associated with the data source. The default dbCategories that are available are: ■ 'Drop/Create P13N Database Objects' category associated with the p13nDataSource data source, which is a part of the p13n.jar domain template ■ 'Drop/Create Portal Database Objects' category associated with the "p13nDataSource" data source, which is a part of the wlp.jar domain template ■ 'Drop/Create GroupSpace Database Objects' category associated with the appsGroupSpaceDataSource data source, which is a part of the wlp_groupspacedb.jar domain template All these template jar files are located in the MW_HOME\wlserver_ <version>\common\templates\applications directory.
security.xml	Used to create user groups and roles that establish identity and access to domain resources. You can create the default Admin user only through the security.xml in a *domain* template. However, you can create user groups and roles through the security.xml included in either a domain or an extension template.

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

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

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.

Table 2–1 (Cont.) Oracle WebLogic Server and Workshop for WebLogic Templates

Template	Description
Oracle Workshop for WebLogic Extension Template	Extends the Basic WebLogic Server domain to create an Oracle Workshop for WebLogic domain.
Oracle Workshop for WebLogic 10.3 Extension Template	Extends the Basic WebLogic Server domain to create an Oracle Workshop for WebLogic domain.
Workshop for WebLogic 10.3 Sample Data Extension Template	Extends the Basic WebLogic Server domain to create an Oracle Workshop for WebLogic 10.3 domain.

Note: As of WebLogic Server 11*g* Release 1 (10.3.2), Workshop no longer ships with WebLogic Server. Oracle recommends that you use JDeveloper instead.

The Workshop templates will be present only if you are upgrading a previous installation of WebLogic Server 11*g* Release 1 (10.3.0 or 10.3.1) and you installed Workshop in that release.

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 <template name> 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 in Configuration Wizard	Basic WebLogic Server Domain
Template JAR file and location	*MW_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, and you want the Administration Server name to be different from the default name, AdminServer, you must configure the name manually. You cannot change the name later when applying an extension template.
		For information about customizing the Administration Server name while creating a domain with the Configuration Wizard, see "Creating WebLogic Domains" *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\	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\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 these 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	*MW_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, and you want the Administration Server name to be different from the default name, AdminServer, you must configure the name manually. You cannot change the name later when applying an extension template.
		For information about customizing the Administration Server name while creating a domain with the Configuration Wizard, see "Creating WebLogic Domains" *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.

Table 2–6 (Cont.) Resources Configured in a WebLogic Server Starter Domain

Resource Type	Name	Notes
Application Deployments	wls_starter	A sample web application deployed to the starter domain. check this, the name is different in admin console.

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\	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\	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.

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

Directory	File	Description
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\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 2–7 (Cont.) Output Generated from the WebLogic Server Starter Domain 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 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	*MW_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.

For this template, I also see a service, JMS System Resource (WsseJMSModule), and a second service, Work Manager (weblogic.wsee.mdb.DispatchPolicy), both targeted to the AdminServer.

There is a JMS File Store (WseeFileStore), which is the Persistent store for the JMS Server. I don't see the WseeJMSServer until I get to the Configuration Summary screen, and I select the WseeJMSServer entry on the left. I don't see the associated queues (will check the Admin Console after creating). There is also an SAF agent (ReliableWseeSAFAgent)

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."

Table 2–9 (Cont.) Resources Configured in a WebLogic Advanced Web Services Domain

Resource Type	Name	Extension Result
Security realm	myrealm	Uses the default security realm provided by the Basic WebLogic Server domain.
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\	n.a	Directory serving as a placeholder for any custom application files that you create.
user_projects\applications\base_domain\	fileRealm.properties	File containing ACLs, users, and groups that can be used for the default security realm when Compatibility security is used.
user_projects\applications\base_domain\	startWebLogic.cmd startWebLogic.sh	Scripts used to start the Administration Server on Windows and UNIX systems, respectively.
user_projects\applications\base_domain\	URLs.dat	File containing the URL for the JDBC database.
user_projects\applications\base_domain\autodeploy\	readme.txt	File providing information about the directory, which initially serves as a placeholder for automatic deployments.
user_projects\applications\base_domain\bin\	setDomainEnv.cmd setDomainEnv.sh	Scripts used to set up the development environment on Windows and UNIX systems, respectively.
user_projects\applications\base_domain\bin\	startManagedWebLogic.cmd startManagedWebLogic.sh	Scripts used to start a Managed Server on Windows and UNIX systems, respectively.
user_projects\applications\base_domain\bin\	startWebLogic.cmd startWebLogic.sh	Scripts used to start the Administration Server on Windows and UNIX systems, respectively.

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

Directory	File	Description
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."
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.

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

Directory	File	Description
user_projects\applications\base_domain\console-ext\	readme.txt	File providing information about the directory, which initially serves as a placeholder for custom extensions to the WebLogic Server Administration Console.
user_projects\applications\base_domain\init-info\	domain-info.xml	File used to identify domain creation and extension information. Such information includes the identity of the components in the domain, the location of the JDK and applications directory used by the domain, and the templates used to create and extend the domain.
user_projects\applications\base_domain\init-info\	security.xml	File used for creating user groups and roles that establish identity and access to domain resources.
user_projects\applications\base_domain\init-info\	startscript.xml	File used to create the `*.cmd` and `*.sh` files that are placed into the domain's root and bin directories.
user_projects\applications\base_domain\init-info\	tokenValue.properties	File that contains the actual values to substitute for the tokens specified in the start scripts.
user_projects\applications\base_domain\lib\	readme.txt	File providing information about the directory, which initially serves as a placeholder for the domain's libraries. The JAR files in this directory are added dynamically to the end of the server classpath at server startup.
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.

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

Directory	File	Description
user_projects\applications\base_domain\servers\AdminServer\security\	boot.properties	File containing server startup properties, including the user name and password required to start the server (in encrypted format). It is generated only when you select development startup mode. This file enables you to bypass the prompt for user name and password during a server's startup cycle. For more information, see "Provide User Credentials to Start and Stop Servers" in *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 service 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

Table 2–11 (Cont.) WebLogic Advanced Web Services for JAX-WS Extension Template

Template Detail	Information
Template JAR file and location	*MW_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."
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*

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

Resource Type	Name	Extension Result
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.
Logical store	WseeStore	Adds this logical store, which specifies `BufferingQueueJndiName` as `weblogic.wsee.BufferedRequestQueue`.
		This logical store is reserved for future use.

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 *Information Roadmap for 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	*MW_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.

Also a Work Manager service (weblogic.wsee.mdb.DispatchPolicy) 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.(is this true? I don't see this anywhere, I see AdminServer in the admin console) 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 Administration Server.
Application Deployments	medrec	Adds the sample medrec Enterprise application and targets it to the Administration Server.
Application Deployments	physician	Adds the sample physician Enterprise application and targets it to the Administration Server.
JDBC Data Sources	MedRecGlobalDataSourceXA	Identifies the JDBC data source as a MedRecGlobalDataSource system resource.
JMS Queues	com.bea.medrec.jms.RecordToCreateQueue com.bea.medrec.jms.PatientNotificationQueue weblogic.wsee.DefaultQueue	Adds the JMS queue to the JMS server, MedRecWseeJMSServer.
JMS Server	MedRecJMSServer	Adds the JMS server as a MedRec-jms system resource and targets it to the Administration Server.
JMS System Resources	MedRec-jms	Adds the JMS servers, connection factories, and queues to be used as JMS system resources, and targets the resources to the Administration Server.

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

Resource Type	Name	Extension Result
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 Administration Server.
Deployed library	jsf1.2@1.2.9.0	Adds the Java Server Faces Version 1.2 libraries.
Deployed library	jstl1.2.@1.2.1.0	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 Administration Server.

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 sub-directories containing library files supporting the Avitek Medical Records sample.
user_projects\applications\base_domain\modules\	Various	Includes sub-directories containing Avitek Medical Records source code including various Java, XML, JSP, HTML files, and so on.
user_projects\applications\base_domain\	build.xml	Ant build file used with corresponding scripts to set up a database for the Avitek Medical Records sample.
user_projects\domains\base_domain\	fileRealm.properties	File containing ACLs, users, and groups that can be used for the default security realm when Compatibility security is used.
user_projects\domains\base_domain\	log4j.properties	Configures Avitek Medical Records Log4j implementation including the MedRecApp.log file.
user_projects\domains\base_domain\	startWebLogic.cmd startWebLogic.sh	Scripts used to start the Administration Server on Windows and UNIX systems, respectively.

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

Directory	File	Description
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.
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).

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

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

Directory	File	Description
user_projects\domains\base_domain\security\	DefaultAuthenticatorInit.ldift DefaultAuthorizerInit.ldift	Files used for bootstrapping tasks, including authentication (user and group), authorization, and role mapping. These files contain LDAP-specific information. Note: WebLogic domains created with this release use the XACML providers, by default. These XACML security providers are compatible with policies and roles created using the WebLogic Authorization provider (DefaultAuthorizer) and WebLogic Role Mapping provider (DefaultRoleMapper). For more information, see "WebLogic Security Providers" in *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 *Information Roadmap for 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	*MW_HOME*\common\templates\applications\medrec_spring.jar

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

Template Detail	Information
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.

Two Work Managers are configured, but they are not targeted to anything. Include them in table?

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.(is this true? I don't see this anywhere except in config.xml, I see AdminServer in the admin console and Config Wizard) 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 Administration Server.
Application Deployments	medrec	Adds the sample medrec Enterprise application and targets it to the Administration Server.
Application Deployments	physician	Adds the sample physician Enterprise application and targets it to the Administration Server.
JDBC Data Sources	MedRecGlobalDataSourceXA	Identifies the JDBC data source as a MedRecGlobalDataSource system resource. Pool capacity (initial): 2 Pool capacity (maximum): 10 Protocol: Two Phase Commit

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

Resource Type	Name	Extension Result
JMS Queues	com.bea.medrec.jms.RecordToCreateQueue com.bea.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 Administration Server.
JMS System Resources	MedRec-jms	Adds the JMS servers, connection factories, and queues to be used as JMS system resources, and targets the resources to the Administration Server.
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 Administration Server. Doesn't appear to be a file store associated with this resource.
Deployed library	jsf1.2@1.2.9.0	Adds the Java Server Faces Version 1.2 library and targets it to the Administration Server.
Deployed library	jstl1.2.@1.2.1.0	Adds the Java standard tagging (JSTL) Version 1.2 library and targets it to the Administration Server.
Deployed library	weblogic-spring#10.3.2.0@10.3.2.0	Adds the WebLogic Spring Version 10.3.2 library and targets it to the Administration Server.
WLDF System Resource	MedRecWLDF	Adds this WLDF system resource, and targets it to the Administration Server. 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.

Table 2–18 (Cont.) Base Domain After Applying the Avitek Medical Records Sample Extension Template (Spring Version)

Directory	File	Description
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 *Information Roadmap for 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	*MW_HOME*\common\templates\applications\wls_webservice.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.
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

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

Resource Type	Name	Extension Result
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@1.4.0.0	Adds the HTTP PublisherSubscriber Version 1.4.0.0 library and targets it to the examplesServer Administration Server.
Deployed library	weblogic-sca#1.0@1.0.0.0	Adds the WebLogic SCA Version 1.0.0.0 library and targets it to the examplesServer Administration Server.
Deployed library	apache-xbean.jar	(need info on this one)

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

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

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

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

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

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

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

2.8 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 *Information Roadmap for 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	*MW_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.

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

Resource Type	Name	Notes
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.
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.

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

Resource Type	Name	Notes
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	(need to determine if we should include this. It is not targeted to any servers, but is available to be targeted.)

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\	empty	empty
server\	wls_samples_overview.html	File that opens the WebLogic Server examples online documentation viewer.
server\docs\	Various	Directory and files supporting the WebLogic Server examples online documentation viewer.
server\examples\build\	Various	Includes sub-directories containing various Java and XML files used to build and work with WebLogic Server examples.
server\examples\src\	Various	Includes sub-directories containing various Java, XML, and HTML files used to work with WebLogic Server examples.
user_projects\domains\base_domain\	empty	empty
empty	client2certs.pem clientkey.pem	Demo certificate and keystore files.
empty	fileRealm.properties	File containing ACLs, users, and groups that can be used for the default security realm when Compatibility security is used.
empty	setExamplesEnv.cmd setExamplesEnv.sh	Scripts that set up the environment to use the WebLogic Server Examples on Windows and UNIX systems, respectively.
empty	startWebLogic.cmd startWebLogic.sh	Scripts used to start the Administration Server on Windows and UNIX systems, respectively.

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

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

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

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

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

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

2.9 Oracle Workshop for WebLogic Extension Template

Note: As of WebLogic Server 11*g* Release 1 Patch Set 1 (10.3.2), Workshop no longer ships with WebLogic Server. Oracle recommends that you use JDeveloper instead.

The Workshop templates will be present only if you are upgrading a previous installation of WebLogic Server 11*g* Release 1 and you installed Workshop in that release.

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

Note: Using the Configuration Wizard in graphical mode, you can easily create a Workshop for WebLogic domain by checking the Workshop for WebLogic check box in the **Select Domain Source** window. The result is the same as creating a base WebLogic Server domain first and then extending that domain with the Oracle Workshop for WebLogic extension template and the Advanced Web Services extension template.

2.9.1 Template Details

The following table provides basic information about the Oracle WorkShop for WebLogic extension template.

Template Dependencies lists all templates that provide resources required by the Oracle WorkShop for WebLogic 10.3 extension template, in the order in which they must be configured in the domain.

Table 2–25 Oracle WorkShop for WebLogic 10.3 Extension Template Information

Template Detail	Information
Template type	Extension
Template name in Configuration Wizard	Workshop for WebLogic
Template JAR file and location	*WS_HOME*\common\templates\applications\workshop_wl.jar, where *WS_HOME* is the directory in which Workshop is installed.
Template Dependencies	■ Basic WebLogic Server Domain template ■ + Advanced Web Services Extension, wls_webservice.jar

2.9.2 Generated Domain Output

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

Table 2–26 Base Domain After Applying the Oracle Workshop for WebLogic Template

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

Table 2–26 (Cont.) Base Domain After Applying the Oracle Workshop for WebLogic

Directory	File	Description
user_projects\applications\base_domain\autodeploy\	readme.txt	File providing information about the directory, which initially serves as a placeholder for automatic deployments.
user_projects\applications\base_domain\bin\	setDomainEnv.cmd setDomainEnv.sh	Scripts used to set up the development environment on Windows and UNIX systems, respectively.
user_projects\applications\base_domain\bin\	startManagedWebLogic.cmd startManagedWebLogic.sh	Scripts used to start a Managed Server on Windows and UNIX systems, respectively.
user_projects\applications\base_domain\bin\	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."
user_projects\applications\base_domain\config\diagnostics\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing the system modules associated with instrumentation in the WebLogic Diagnostic Framework (WLDF).
user_projects\applications\base_domain\config\jdbc\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing global JDBC modules that can be configured directly from JMX (as opposed to JSR-88).
user_projects\applications\base_domain\config\jdbc\	cgDataSource-jdbc.xml	Global XA JDBC Data Source module for the domain configured for advanced Web services.

Table 2–26 (Cont.) Base Domain After Applying the Oracle Workshop for WebLogic

Directory	File	Description
user_projects\applications\base_domain\config\jdbc\	cgDataSource-nonXA-jdbc.xml	Global non-XA JDBC Data Source module for the domain configured for advanced Web services.
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.
user_projects\applications\base_domain\init-info\	tokenValue.properties	File that contains the actual values to substitute for the tokens specified in the start scripts.
user_projects\applications\base_domain\lib\	readme.txt	File providing information about the directory, which initially serves as a placeholder for the domain's libraries. The JAR files in this directory are added dynamically to the end of the server classpath at server startup.

Table 2–26 (Cont.) Base Domain After Applying the Oracle Workshop for WebLogic

Directory	File	Description
user_projects\applications\base_domain\security\	DefaultAuthenticatorInit.ldift DefaultRoleMapperInit.ldift XACMLRoleMapperInit.ldift	Files used for bootstrapping tasks, including authentication (user and group), authorization, and role mapping. These files contain LDAP-specific information. Note: WebLogic domains created with this release use the XACML providers, by default. These XACML security providers are compatible with policies and roles created using the WebLogic Authorization provider (DefaultAuthorizer) and WebLogic Role Mapping provider (DefaultRoleMapper). For more information, see "WebLogic Security Providers" in *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.

2.9.3 Resources and Services Configured

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

Table 2–27 Resources Configured in a Oracle Workshop for WebLogic Domain

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

Table 2–27 (Cont.) Resources Configured in a Oracle Workshop for WebLogic Domain

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

2.10 Oracle Workshop for WebLogic 10.3 Extension Template

Note: As of WebLogic Server 11*g* Release 1 (10.3.2), Workshop no longer ships with WebLogic Server. Oracle recommends that you use JDeveloper instead.

The Workshop templates will be present only if you are upgrading a previous installation of WebLogic Server 11*g* Release 1 and you installed Workshop in that release.

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

Note: Using the Configuration Wizard in graphical mode, you can easily create a Workshop for WebLogic domain by checking the Workshop for WebLogic check box in the Select Domain Source window. The result is the same as creating a base WebLogic Server domain first and then extending that domain with the Oracle Workshop for WebLogic extension template and the Advanced Web Services extension template.

2.10.1 Template Details

The following table provides basic information about the Oracle WorkShop for WebLogic extension template.

Template Dependencies lists all templates that provide resources required by the Oracle WorkShop for WebLogic 10.3 extension template, in the order in which they must be configured in the domain.

Table 2–28 Oracle WorkShop for WebLogic 10.3 Extension Template Information

Template Detail	Information
Template type	Extension
Template name in Configuration Wizard	Workshop for WebLogic
Template JAR file and location	*WS_HOME*\common\templates\applications\workshop_wl.jar, where *WS_HOME* is the directory in which Workshop is installed (default is *MW_HOME*/workshop_10.3).
Template Dependencies	▪ Basic WebLogic Server Domain template ▪ + Advanced Web Services Extension, wls_webservice.jar

2.10.2 Resources and Services Configured

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

Table 2–29 Resources Configured in a Oracle Workshop for WebLogic 10.3 Domain

Resource Type	Name	Extension Result
Administration Server	AdminServer	Uses the Administration Server provided in the base WebLogic Server domain. The default name is AdminServer, unless changed during domain creation. The Administration Server referenced in the extension template is cgServer. For information about naming the Administration Server during domain creation, see Section 2.1.2, "Resources and Services Configured in a Basic WebLogic Server Domain."
Security realm	myrealm	Uses the security realm provided by the base WebLogic Server domain.
JDBC Data Source	cgDataSource	Defines an XA JDBC data source including its associated JDBC connection pool. The data source is named cgDataSource.
JDBC Data Source	cgDataSource-nonXA	Includes the JDBC data source and connection pool setups defined as cgDataSource in the domain and targets them to the correct server(s).
JDBC Store	cgJMSStore	Uses the JDBC store provided by the Oracle Workshop for WebLogic extension template. The JDBC store is to be used with the JDBC data source, cgDataSource-nonXA, and the JMS server, WseeJmsServer, as a persistent store, and is targeted to the Administration Server, AdminServer.
JDBC System Resources	cgDataSource cgDataSource-nonXA	Identifies the JDBC data source and connection pool setups to be used for JDBC system.
JMS Server	WseeJmsServer	Uses the JMS server provided by the Workshop for WebLogic extension template. Identifies the JMS server as a system resource and targets it to the Administration Server, AdminServer.
Commons-Logging Bridge	wls-commonslogging-bridge#1.0@1.0	Hooks commons-logging into the WLS logging mechanism.
Libraries Deployed	beehive-netui-1.0#1.0@1.0	Adds the Apache Beehive NetUI Version 1.0 libraries. These libraries support pageflow development, and depend upon the libraries contained in struts-1.1.war and weblogic-beehive-1.0.ear.
Libraries Deployed	jstl#1.1@1.1.2	Adds the Java standard tagging (JSTL) Version 1.1 libraries.
Libraries Deployed	jsf-ri#1.1@1.1.1	Adds the Java Server Faces Reference Implementation libraries.
Libraries Deployed	jsf-myfaces#1.1@1.1.1	Adds the Apache MyFaces libraries.
Libraries Deployed	struts-1.1#1.1@1.0	Adds the Apache Struts Version 1.1 libraries.
Libraries Deployed	struts-1.2#1.2@1.0	Adds the Apache Struts Version 1.2 libraries.

Table 2–29 (Cont.) Resources Configured in a Oracle Workshop for WebLogic 10.3

Resource Type	Name	Extension Result
Libraries Deployed	weblogic-controls-10.0#10.0@10.0	Adds the Oracle Workshop for WebLogic controls extensions, including additional system controls (such as service control and timer control) and support for adding transactions, security, and message buffering to existing controls. Packaged for EARs.
Libraries Deployed	weblogic-controls-10.0-war#10.0@10.0	Adds the Oracle Workshop for WebLogic controls extensions including additional system controls (such as service control) and support for adding transactions, security, and message buffering to existing controls. Excludes those features which require EAR support such as timer control. Packaged for WARs.
Libraries Deployed	beehive-controls-1.0#1.0@1.0	Adds the Apache Beehive Controls 1.0.1 libraries to the domain. This includes the control run time and the Beehive system controls - JdbcControl, JMSControl, and EJBControl.

2.10.3 Generated Domain Output

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

Table 2–30 Base Domain After Applying the Oracle Workshop for WebLogic Template

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

Table 2–30 (Cont.) Base Domain After Applying the Oracle Workshop for WebLogic

Directory	File	Description
user_projects\applications\base_domain\bin\	startManagedWebLogic.cmd startManagedWebLogic.sh	Scripts used to start a Managed Server on Windows and UNIX systems, respectively.
user_projects\applications\base_domain\bin\	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."
user_projects\applications\base_domain\config\diagnostics\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing the system modules associated with instrumentation in the WebLogic Diagnostic Framework (WLDF).
user_projects\applications\base_domain\config\jdbc\	readme.txt	File providing information about the directory, which initially serves as a placeholder, and is later used for storing global JDBC modules that can be configured directly from JMX (as opposed to JSR-88).
user_projects\applications\base_domain\config\jdbc\	cgDataSource-jdbc.xml	Global XA JDBC Data Source module for the domain configured for advanced Web services.
user_projects\applications\base_domain\config\jdbc\	cgDataSource-nonXA-jdbc.xml	Global non-XA JDBC Data Source module for the domain configured for advanced Web services.
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.

Table 2–30 (Cont.) Base Domain After Applying the Oracle Workshop for WebLogic

Directory	File	Description
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.
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*.

Table 2–30 (Cont.) Base Domain After Applying the Oracle Workshop for WebLogic

Directory	File	Description
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.

2.11 Workshop for WebLogic 10.3 Sample Data Extension Template

Note: As of WebLogic Server 11*g* Release 1 (10.3.2), Workshop no longer ships with WebLogic Server. Oracle recommends that you use JDeveloper instead.

The Workshop templates will be present only if you are upgrading a previous installation of WebLogic Server 11*g* Release 1 and you installed Workshop in that release.

Using the Configuration Wizard or WLST, you can easily extend a Workshop for WebLogic 10.3 domain to include a sample Pointbase data source. You accomplish this by adding the resources and services provided in the Workshop for WebLogic 10.3 template to a base WebLogic Server domain, then adding the resources provided by the Workshop for WebLogic 10.3 Sample Data extension template.

Note: Using the Configuration Wizard in graphical mode, you can easily create an Oracle Workshop for WebLogic 10.3 domain by checking the Workshop for WebLogic 10.3 check box on the Select Domain Source screen. The result is the same as creating a base WebLogic Server domain first and then extending that domain with both the Workshop for WebLogic 10.3 extension template.

2.11.1 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 2–31 Workshop for WebLogic 10.3 Sample Data 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	*WS_HOME*\common\templates\applications\workshop_samples.jar, where *WS_HOME* is the directory in which Workshop is installed (default is *MW_HOME*/workshop_10.3).
Template Dependencies	■ Basic WebLogic Server Domain template ■ + Advanced Web Services Extension, wls_webservice.jar ■ + Oracle WorkShop for WebLogic Extension, workshop_wl.jar

2.11.2 Resources and Services Configured

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

Table 2–32 Resources Configured in a Workshop for WebLogic 10.3 template

Resource Type	Name	Extension Result
Administration Server	AdminServer	Uses the Administration Server provided in the base WebLogic Server domain. The default name is AdminServer, unless changed during domain creation. The Administration Server referenced in the extension template is cgServer. For information about naming the Administration Server during domain creation, see Section 2.1.2, "Resources and Services Configured in a Basic WebLogic Server Domain."
Security realm	myrealm	Uses the security realm provided by the base WebLogic Server domain.
JDBC System Resources	SamplesDataSource	Identifies the JDBC data source and connection pool setups to be used for the JDBC system.
JDBC Data Source	samplesDataSource	Identifies the JDBC data source as a samplesDataSource system resource. Pool capacity (initial): 5 Pool capacity (maximum): 20 Protocol: Emulated Two Phase Commit

3

Fusion Middleware Product Templates

This chapter describes the WebLogic domain and extension templates that are used to configure WebLogic domains for various Fusion Middleware product installations. For most of these templates, you can create or extend domains by selecting the template on the Select Domain Source or Select Extension Source screens of the Oracle Fusion Middleware Configuration Wizard. You can also use the WebLogic Scripting Tool (WLST) in offline mode to create or extend domains using these templates. See Section 1.3, "Template Tools," for more information.

Your product installation includes only those templates that are relevant to your product. Some templates in this chapter, such as the Oracle JRF template, apply to domains for multiple Fusion Middleware products. Other templates apply only to domains for a single Fusion Middleware product, and are therefore not included with any other Fusion Middleware product installations.

This chapter contains the following sections:

Each template section contains the following template information:

- **Template type**—A template can be either a *domain* or an *extension* template.

  A domain template defines the full set of resources within a WebLogic domain, including the Administration Server, clusters, servers, applications, services, security options, and other options. A given product installation is based on one domain template.

  An extension template adds resources, such as applications, libraries, services, and other options, to an existing domain. A given product installation may require one or more extension templates to complete the domain.

A group template is an XML file that is a collection of template dependencies. It simplifies the process of pulling in multiple dependent templates by the template that requires them.

When creating your product domain using the Configuration Wizard, you can select the domain and extension templates concurrently for the products you want to install.

- **Template name and version**—The template (product) name and template version. If the template is selectable on the Select Domain Source or Select Extension Source screen of the Configuration Wizard, this is the name as it is listed in the Configuration Wizard.
- **Template dependencies**—Prerequisite templates that provide resources required by the template being described in a given section, in the order in which they must be added to the domain. When you select a template from the Product List on the Configuration Wizard Select Domain Source or Select Extension Source screen, the Configuration Wizard automatically selects all prerequisite templates.
- **Template JAR file and location**—The JAR filename and its location in the Middleware home directory. You need this information only if you plan to use WLST scripts to set up a product domain.

3.1 Enterprise Manager Templates

This section describes the Enterprise Manager templates that are used to add Enterprise Manager resources to the WebLogic domains for various Fusion Middleware products. There are currently three Enterprise Manager JAR files, all of which are installed when you select the product on the Select Domain Source screen of the Fusion Middleware Configuration Wizard.

3.1.1 Oracle Enterprise Manager

This template adds Oracle Enterprise Manager Fusion Middleware Control to your domain. Fusion Middleware Control is a Web browser-based, graphical user interface that you can use to monitor and administer Oracle Fusion Middleware.

Together with the Oracle WebLogic Server Administration Console, Fusion Middleware Control provides tools to help you manage the Oracle Fusion Middleware environment, including the Oracle WebLogic Server domain, the Oracle Fusion Middleware components you have installed and configured, and the applications you deploy.

For more complete information about Oracle Enterprise Manager Fusion Middleware Control, see "Getting Started Using Oracle Enterprise Manager Fusion Middleware Control" in the *Oracle Fusion Middleware Administrator's Guide*.

Note: [This template must be used as provided. Do not modify them in any way. Doing so can cause issues in your domain.]]

The following table provides key information about this template.

Table 3–1 Oracle Enterprise Manager Template Details

Template Detail	Information
Template type	Extension

Table 3–1 (Cont.) Oracle Enterprise Manager Template Details

Template Detail	Information
Template name and version	Oracle Enterprise Manager - 11.1.1.0
Template Dependencies	■ Basic WebLogic Server Domain ■ + Oracle JRF - 11.1.1.0
Template JAR file and location	This template consists of multiple template JAR files. These JAR files are located in the following directory after you install Oracle Fusion Middleware: *ORACLE_COMMON_HOME*/common/applications/templates The template consists of the following JAR files in the templates directory: ■ oracle.em_11_1_1_0_0_template.jar ■ oracle.emai_template_11.1.1.jar ■ oracle.emas_template_11.1.1.jar

3.2 Enterprise Content Management Templates

This section describes the Enterprise Content Management (ECM) templates that are used to extend (add resources to) WebLogic Server domains in which ECM products are installed. Templates for other products may also be required for an ECM domain. These are described elsewhere in this chapter.

> **Note:** [The templates described in this section must be used as provided. Do not modify them in any way. Doing so can cause issues in your domain.]]

The following templates are described in this section:

- Oracle Universal Content Management Core Template
- Oracle Universal Content Management - Content Server Template
- Oracle Universal Records Management Template
- Oracle Information Rights Management (IRM) Template
- Oracle Imaging and Process Management Template
- Oracle Universal Content Management - Inbound Refinery Template

3.2.1 Oracle Universal Content Management Core Template

The Oracle Universal Content Management (UCM) Core Template is responsible for deploying UCM help, UCM Web services, UCM Native Web services, and Dav service. Along with these deployments, it also defines the ReportPublisher library and `oracle.xdo.runtime` shared libraries that are used by Records Management in both the UCM and ECM products.

The core template also updates the Managed Server's `setDomainEnv` script with the definition for the `UCM_ORACLE_HOME` environment variable, updates the `CLASSPATH` entry defined by this script to include `idcloader.jar` and `nativeosutils.jar`, and adds a line to source the *`ucm_oracle_home`*`/ucm/idc/bin/idcCommEnv` script. The core template is also responsible for defining the dependency on the JRF and Enterprise Manager templates.

Note: The Oracle Enterprise Manager Plugin for UCM template, which is listed as a dependency for this template, is an internal template. It is automatically called when you select the Universal Content Management (UCM) Core template, or specify the UCM Core template in a WLST script. Therefore, it is not displayed as a selectable component in Configuration Wizard, and you do not have to specify it in your WLST scripts.

The following table provides key information about this template.

Table 3–2 Oracle UCM Core Template Details

Template Detail	Information
Template type	Extension Template
Template name and version	Oracle UCM Core - 11.1.1.0
Template Dependencies	■ Basic WebLogic Server Domain ■ + Oracle JRF - 11.1.1.0 ■ + Oracle Enterprise Manager - 11.1.1.0 ■ + Oracle Enterprise Manager Plugin for UCM
Template JAR file and location	*ECM_HOME*\common\templates\applications\oracle.ucm.core_template_11.1.1.jar

3.2.2 Oracle Universal Content Management - Content Server Template

The Oracle Universal Content Management - Content Server template depends upon the core template and extends it by defining the default managed server (UCM_server1) and port (16200). It also defines the CSDS data source and associates it with the default managed server.

Note: The Oracle Enterprise Manager Plugin for UCM template, which is listed as a dependency for this template, is an internal template. It is automatically called when you select the Content Server template, or specify the Content Server template in a WLST script. Therefore, it is not displayed as a selectable component in Configuration Wizard, and you do not have to specify it in your WLST scripts.

The following table provides key information about this template.

Table 3–3 Oracle Universal Content Management - Content Server Template Details

Template Detail	Information
Template type	Extension Template
Template name and version	Oracle Universal Content Management - Content Server 11.1.1.0

Table 3–3 (Cont.) Oracle Universal Content Management - Content Server Template

Template Detail	Information
Template Dependencies	■ Basic WebLogic Server Domain ■ + Oracle JRF - 11.1.1.0 ■ + Oracle Enterprise Manager - 11.1.1.0 ■ + Oracle Enterprise Manager Plugin for UCM ■ + Oracle UCM Core - 11.1.1
Template JAR file and location	*ECM_HOME*\common\templates\applications\oracle.ucm.cs_template_11.1.1.jar

3.2.3 Oracle Universal Records Management Template

This template adds the Oracle Universal Records Management application to an existing domain.

Note: The Oracle Enterprise Manager Plugin for UCM template, which is listed as a dependency for this template, is an internal template. It is automatically called when you select the Universal Content Management (UCM) Core Template, or specify the UCM Core template in a WLST script. Therefore, it is not displayed as a selectable component in Configuration Wizard, and you do not have to specify it in your WLST scripts.

The following table provides key information about this template.

Table 3–4 Oracle Universal Records Management Template Details

Template Detail	Information
Template type	Extension Template
Template name and version	Oracle Universal Records Management - 11.1.1.0
Template Dependencies	■ Basic WebLogic Server Domain ■ + Oracle JRF - 11.1.1.0 ■ + Oracle Enterprise Manager - 11.1.1.0 ■ + Oracle Enterprise Manager Plugin for UCM ■ + Oracle UCM Core Template - 11.1.1.0
Template JAR file and location	*ECM_HOME*\common\templates\applications\oracle.ucm.urm_template_11.1.1.jar

3.2.4 Oracle Information Rights Management (IRM) Template

The Oracle Information Rights Management template adds the Oracle Information Rights server to an existing domain.

Note: The Oracle Enterprise Manager Plugin for IRM template, which is listed as a dependency for this template, is an internal template. It is automatically called when you select the Information Rights Management template, or specify the Information Rights Management template in a WLST script. Therefore, it is not displayed as a selectable component in Configuration Wizard, and you do not have to specify it in your WLST scripts.

The following table provides key information about this template.

Table 3–5 Oracle Information Rights Management Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle Information Rights Management - 11.1.1.0
Template dependencies	■ Basic WebLogic Server Domain ■ + Oracle JRF - 11.1.1.0 ■ + Oracle Enterprise Manager - 11.1.1.0 ■ + Oracle Enterprise Manager Plugin for IRM - 11.1.1.0
Template JAR file and location	*ECM_HOME*\common\templates\applications\oracle.irm_template_11.1.1.jar

3.2.5 Oracle Imaging and Process Management Template

The Oracle Imaging and Process Management template provides the ability to create or extend domains with the Oracle Imaging and Process Management product.

The following table provides key information about this template.

Table 3–6 Oracle Imaging and Process Management Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle Imaging and Process Management - 11.1.1.0
Template Dependencies	■ Basic WebLogic Server Domain ■ + Oracle JRF - 11.1.1.0 ■ + Oracle Enterprise Manager - 11.1.1.0
Template JAR file and location	*ECM_HOME*/common/templates/applications/oracle.ipm_template_11.1.1.jar

3.2.6 Oracle Universal Content Management - Inbound Refinery Template

The Oracle Universal Content Management - Inbound Refinery template adds the Inbound Refinery server and UCM Inbound Refinery application to an existing domain.

Note: The Oracle Enterprise Manager Plugin for UCM template, which is listed as a dependency for this template, is an internal template. It is automatically called when you select the Inbound Refinery template, or specify the Inbound Refinery template in a WLST script. Therefore, it is not displayed as a selectable component in Configuration Wizard, and you do not have to specify it in your WLST scripts.

The following table provides key information about this template.

Table 3–7 Oracle Universal Content Management - Inbound Refinery Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle Imaging and Process Management - 11.1.1.0
Template Dependencies	■ Basic WebLogic Server Domain ■ + Oracle JRF - 11.1.1.0 ■ + Oracle Enterprise Manager 11.1.1.0 ■ + Oracle Enterprise Manager Plugin for UCM ■ + Oracle UCM Core - 11.1.1.0
Template JAR file and location	*ECM_HOME*/common/templates/applications/oracle.ucm.ibr_template_11.1.1.jar

3.3 Oracle SOA Suite Templates

This section describes the following Oracle SOA Suite templates:

- Oracle SOA Suite Template
- Oracle Business Activity Monitor Template
- Oracle Business Rules Extension Template

3.3.1 Oracle SOA Suite Template

The Oracle SOA Suite template configures the data sources, Java Messaging Service (JMS), applications (SOA Infrastructure, Oracle BPM Worklist, and the Oracle B2B user interface), and JCA adapters (file, FTP, socket, database, Advanced Queuing (AQ), JMS, Oracle Applications, and MQ Series).

The following table provides basic information about the Oracle SOA Suite template.

Table 3–8 Oracle SOA Suite Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle SOA Suite - 11.1.1.0

Table 3–8 (Cont.) Oracle SOA Suite Template Details

Template Detail	Information
Template Dependencies	■ Oracle JRF - 11.1.1.0 ■ + Oracle Business Rules Extension - 11.1.1 ■ + Oracle User Messaging Service for SOA - 11.1.1.0 ■ + Oracle WSM Policy Manager - 11.1.1.0
Template JAR file and location	*ORACLE_HOME*\common\templates\applications\oracle.soa_11.1.1_template.jar

3.3.2 Oracle Business Activity Monitor Template

The Oracle BAM template configures the Oracle BAM Server and Oracle BAM Web Applications tier.

The following table provides basic information about the Oracle BAM template.

Table 3–9 Oracle Business Activity Monitor Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle Business Activity Monitor - 11.1.1.0
Template Dependencies	■ Oracle JRF - 11.1.1.0 ■ + Oracle User Messaging Service - 11.1.1.0 ■ + Oracle Business Rules Extension - 11.1.1.0
Template JAR file and location	*ORACLE_HOME*\common\templates\applications\oracle.bam_11.1.1_template.jar

3.3.3 Oracle Business Rules Extension Template

The Oracle Business Rules Extension template configures Oracle Business Rules components and the `oracle.rules` library.

The following table provides basic information about the Oracle Business Rules Extension template.

Table 3–10 Oracle Business Rules Extension Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle Business Rules Extension - 11.1.1.0
Template Dependencies	Oracle JRF - 11.1.1.0
Template JAR file and location	*ORACLE_HOME*\common\templates\applications\oracle.rules_template_11.1.1.jar

3.3.4 Oracle User Messaging Service for SOA Template

The Oracle User Messaging Service for SOA template is a group template that combines the Oracle User Messaging Service and Oracle User Messaging Service Drivers templates into one simplified dependency for SOA.

The following table provides key information about this template.

Table 3–11 Oracle User Messaging Service for SOA Template Details

Template Detail	Information
Template type	Group template
Template name and version	Oracle User Messaging Service for SOA - 11.1.1.0
Template Dependencies	■ Oracle User Messaging Service - 11.1.1.0 ■ Oracle User Messaging Service Drivers - 11.1.1.0
Template group file and location	*ORACLE_HOME*\common\templates\groups\oracle.ums.soa_group_11.1.1.xml

3.4 Oracle Web Service Manager (OWSM) Templates

This section describes the OWSM template that is provided for adding OWSM resources to various Fusion Middleware product domains. There is currently one OWSM template available, as described in Section 3.4.1, "Oracle WSM Policy Manager Template."

Note: [The templates described in this section must be used as provided. Do not modify them in any way. Doing so can cause issues in your domain.]]

3.4.1 Oracle WSM Policy Manager Template

The WSM PM template is used to deploy and configure the WSM Policy Manager JEE application.

The following table provides key information about this template.

Table 3–12 Oracle WSM Policy Manager Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle WSM Policy Manager - 11.1.1.0
Template Dependencies	■ Basic WebLogic Server Domain ■ + Oracle JRF - 11.1.1.0
Template JAR file and location	*ORACLE_COMMON_HOME*/common/templates/applications/oracle.wsmpm_template_11.1.1.jar

3.5 Oracle User Messaging Service Templates

This section describes the following Oracle User Messaging Service templates:

- Oracle User Messaging Service Template
- Oracle User Messaging Service Drivers Template
- Oracle User Messaging Service Worklist Driver Template
- Oracle WebLogic Communications Service Client Library Extension Template

Notes: Oracle User Messaging Service templates are not visible or selectable from the list of products on the Configuration Wizard Select Domain Source and Select Extension Source screens.

The templates described in this section must be used as provided. Do not modify them in any way. Doing so can cause issues in your domain.

3.5.1 Oracle User Messaging Service Template

The Oracle User Messaging Service template configures Oracle User Messaging Service. This component is part of the Oracle SOA Suite and provides services to send/receive alerts and notifications to/from end users using messaging channels such as Email, IM, SMS, and Voice.

The following table provides key information about this template.

Table 3–13 Oracle User Messaging Service Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle User Messaging Service - 11.1.1.0
Template Dependencies	■ Oracle JRF - 11.1.1.0 ■ + Oracle Business Rules Extension - 11.1.1.0 ■ + Oracle WSM Policy Manager - 11.1.1.0
Template JAR file and location	*ORACLE_HOME*\common\templates\applications\oracle.ums_template_11.1.1.jar

3.5.2 Oracle User Messaging Service Drivers Template

The Oracle User Messaging Service Drivers template configures the Oracle User Messaging Service drivers for IM (XMPP), SMS (SMPP), and Voice (VoiceXML).

The following table provides key information about this template.

Table 3–14 Oracle User Messaging Service Drivers Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle User Messaging Service Drivers - 11.1.1.0
Template Dependencies	■ Oracle JRF - 11.1.1.0 ■ + Oracle User Messaging Service - 11.1.1.0
Template JAR file and location	*ORACLE_HOME*\common\templates\applications\oracle.ums.drivers_template_11.1.1.jar

3.5.3 Oracle User Messaging Service Worklist Driver Template

The Oracle User Messaging Service Driver template configures the Oracle User Messaging Service Worklist driver.

The following table provides key information about this template.

Table 3–15 Oracle User Messaging Service Worklist Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle User Messaging Service Worklist Driver - 11.1.1.0
Template Dependencies	■ Oracle JRF - 11.1.1.0 ■ Oracle User Messaging Service - 11.1.1.0
Template JAR file and location	*ORACLE_HOME*\common\templates\applications\oracle.ums.driver.worklist_template_11.1.1.jar

3.5.4 Oracle WebLogic Communications Service Client Library Extension Template

The Oracle WebLogic Communications Service Client Library Extension template configures the Oracle WebLogic Communications Service Client Library. This library provides Web Service clients for Oracle WebLogic Communications Service's presence, messaging, contact management and third party call services.

The following table provides key information about this template.

Table 3–16 Oracle WebLogic Communications Service Client Library Extension Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle WebLogic Communications Service Client Library Extension - 11.1.1.0
Template Dependencies	Oracle JRF - 11.1.1.0
Template JAR file and location	*ORACLE_HOME*\common\templates\applications\oracle.communications.client_template_11.1.1.jar

3.6 Oracle WebLogic Communication Services Templates

This section describes the following Oracle WebLogic Communication Services templates:

- Oracle WebLogic Communication Services Template
- Oracle WebLogic Communication Services, All in One Administration Server Topology Template

Note: [The templates described in this section must be used as provided. Do not modify them in any way. Doing so can cause issues in your domain.]]

3.6.1 Oracle WebLogic Communication Services Template

The Oracle WebLogic Communication Services template configures Oracle WebLogic Communication Services and dependent components in a single managed server topology.

The following table provides key information about this template.

Table 3–17 Oracle WebLogic Communication Services ***Template Details***

Template Detail	Information
Template type	Extension
Template name and version	Oracle WebLogic Communication Services - 11.1.1.0
Template Dependencies	■ Basic WebLogic Server Domain ■ + Oracle JRF - 1.1.1.0 ■ + Oracle WSM Policy Manager - 11.1.1.0 through 11.1.1.2 ■ + Oracle User Messaging Service - 11.1.1.0 ■ + Oracle User Messaging Service Drivers - 11.1.1.0
Template JAR file and location	*ORACLE_HOME*/common/templates/applications/oracle.communications.allinonemanaged_template_11.1.1.jar

3.6.2 Oracle WebLogic Communication Services, All in One Administration Server Topology Template

The Oracle WebLogic Communication Services All in One Administration Server template configures the Oracle Weblogic Communications Services in an All in One Administration Server topology.

The following table provides key information about this template.

Table 3–18 Oracle WebLogic Communication Services ***All in One Administration Server Template Details***

Template Detail	Information
Template type	Extension
Template name and version	Oracle WebLogic Communication Services All in One Administration Server - 11.1.1.0
Template Dependencies	■ Basic WebLogic Server Domain ■ + Oracle JRF - 11.1.1.0 ■ + Oracle WSM Policy Manager - 11.1.1.0 through 11.1.1.2 ■ + Oracle User Messaging Service - 11.1.1.0 ■ + Oracle User Messaging Service Drivers - 11.1.1.0
Template JAR file and location	*ORACLE_HOME*/common/templates/applications/oracle.communications.allinoneadmin_template_11.1.1.jar

3.7 Oracle WebLogic SIP Server Domain Templates

This section describes the following Oracle WebLogic SIP Server domain templates. These domain templates are installed with WebLogic Server. You can use them to create various SIP domains.

> **Notes:** The Oracle WebLogic SIP server domain templates are not visible or selectable from the list of products on the Configuration Wizard Select Domain Source and Select Extension Source screens.

- Oracle WebLogic SIP Server Domain - Geo Topology Site 1 Template

- Oracle WebLogic SIP Server Domain - Geo Topology Site 2 Template
- Oracle WebLogic SIP Server Domain - Replicated Topology
- Basic Oracle WebLogic SIP Server Domain Template
- Oracle WebLogic SIP Server Domain - Diameter Topology Template

3.7.1 Oracle WebLogic SIP Server Domain - Geo Topology Site 1 Template

The Oracle WebLogic SIP Server Domain - Geo Topology Site 1 template is used together with the Geo2 Topology to set up a geographically-redundant SIP domain. For more information about geographical-redundancy, see *Oracle WebLogic SIP Server Administrator's Guide*.

The following table provides key information about this template.

Table 3–19 Oracle WebLogic SIP Server Domain - Geo Topology Site 1 Template Details

Template Detail	Information
Template type	Domain
Template name and version	WebLogic SIP Server Domain - Geo Topology Site 1, 10.3.3.0
Template Dependencies	None
Template JAR file and location	*WL_HOME*/common/templates/applications/geo1domain.jar

3.7.2 Oracle WebLogic SIP Server Domain - Geo Topology Site 2 Template

The Oracle WebLogic SIP Server Domain - Geo Topology Site 2 template is used together with the Geo1 Topology to set up a geographically-redundant SIP domain. For more information about geographical-redundancy, see *Oracle WebLogic SIP Server Administrator's Guide*.

The following table provides key information about this template.

Table 3–20 Oracle WebLogic SIP Server Domain - Geo Topology Site 2 Template Details

Template Detail	Information
Template type	Domain
Template name and version	WebLogic SIP Server Domain - Geo Topology Site 2, 10.3.3.0
Template Dependencies	None
Template JAR file and location	WLS_HOME/common/templates/applications/geo2domain.jar

3.7.3 Oracle WebLogic SIP Server Domain - Replicated Topology

The Oracle WebLogic SIP Server Domain - Replicated Topology template sets up a replicated SIP cluster with two engines and two replicas.

The following table provides key information about this template.

Table 3–21 Oracle WebLogic SIP Server Domain - Replicated Topology Template Details

Template Detail	Information
Template type	Domain
Template name and version	WebLogic SIP Server Domain - Replicated Topology - 10.3.3.0
Template Dependencies	None
Template JAR file and location	*WL_HOME*/common/templates/applications/replicateddomain.jar

3.7.4 Basic Oracle WebLogic SIP Server Domain Template

The Basic Oracle WebLogic SIP Server Domain template is a basic SIP domain intended for development, consisting of only an Administration Server.

The following table provides key information about this template.

Table 3–22 Oracle WebLogic Communication Services Domain - AdminServer Topology Template Details

Template Detail	Information
Template type	Domain
Template name and version	Basic WebLogic SIP Server Domain - 10.3.3.0
Template Dependencies	None
Template JAR file and location	*WL_HOME*/common/templates/applications/geo1domain.jar

3.7.5 Oracle WebLogic SIP Server Domain - Diameter Topology Template

The Oracle WebLogic SIP Server Domain - Diameter Topology template is used when setting up an IMS-like system. For more information, see *Oracle WebLogic SIP Server Administrator's Guide*.

The following table provides key information about this template.

Table 3–23 Oracle WebLogic Communication Services Domain - Diameter Topology Template Details

Template Detail	Information
Template type	Domain
Template name and version	WebLogic SIP Server Domain - Diameter Topology - 10.3.3.0
Template Dependencies	None
Template JAR file and location	*WL_HOME*/common/templates/applications/diameterdomain.jar

3.8 Oracle WebCenter Templates

This section describes the following Oracle WebCenter templates:

- Oracle WebCenter Spaces

- Oracle Portlet Producers
- Oracle Wiki and Blog Server
- Oracle WebCenter Discussion Server

3.8.1 Oracle WebCenter Spaces

The Oracle WebCenter Spaces template adds the `WLS_Spaces` managed server, and deploys the Oracle WebCenter Spaces application and required libraries. It also deploys the WebCenter Help application and adds JDBC data sources for accessing MDS and the WebCenter schema.

The following table provides key information about this template.

Table 3–24 Oracle WebCenter Spaces Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle WebCenter Spaces - 11.1.1.0
Template Dependencies	■ Oracle WebCenter Framework ■ + Oracle Enterprise Manager - 11.1.1.0 ■ + Oracle WSM Policy Manager - 11.1.1.0 ■ + Oracle JRF - 11.1.1.0 **Note:** The Oracle WebCenter Framework template is an internal template that should not be used directly.
Template JAR file and location	*WC_ORACLE_HOME*/common/templates/applications/oracle.wc_spaces_template_11.1.1.jar

3.8.2 Oracle Portlet Producers

The Oracle Portlet Producers template adds the `WLS_Portlet` managed server, and deploys the Oracle WebCenter portlet producer applications (Portal Tools and WSRP Tools) and required libraries. It also adds a JDBC data source for accessing the Portlet schema.

The following table provides key information about this template.

Table 3–25 Oracle Portlet Producers Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle Portlet Producers - 11.1.1.0
Template Dependencies	■ Oracle JRF - 11.1.1.0 ■ + Oracle WSM Policy Manager - 11.1.1.0
Template JAR file and location	*WC_ORACLE_HOME*/common/templates/applications/oracle.producer_apps_template_11.1.1.jar

3.8.3 Oracle Wiki and Blog Server

The Oracle Wiki and Blog Server template adds the `WLS_Services` managed server and deploys the Oracle WebCenter Wiki application. It also adds a JDBC data source for accessing the Wiki schema.

The following table provides key information about this template.

Table 3–26 Oracle Wiki and Blog Server Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle Wiki and Blog Server - 11.1.1.0
Template Dependencies	Oracle JRF - 11.1.1.0
Template JAR file and location	*WC_ORACLE_HOME*/common/templates/applications/oracle.wc_wiki_11.1.1.jar

3.8.4 Oracle WebCenter Discussion Server

The Oracle WebCenter Discussion Server template adds the `WLS_Services` managed server and deploys the Oracle WebCenter Discussion application. It also adds a JDBC data source for accessing the Discussions schema.

The following table provides key information about this template.

Table 3–27 Oracle WebCenter Discussion Server Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle WebCenter Discussion Server - 11.1.1.0
Template Dependencies	Oracle JRF - 11.1.1.0
Template JAR file and location	*WC_ORACLE_HOME*/common/templates/applications/oracle.wc_discussions_template_11.1.1.jar

3.9 Oracle Identity Management Templates

This section describes the following Oracle Identity Management (IDM) templates:

- Oracle Adaptive Access Manager Admin Server Template
- Oracle Adaptive Access Manager - Server Template
- Oracle Access Manager Template
- Oracle Access Manager with Database Policy Store Template
- Oracle Identity Manager Template
- Oracle Authorization Policy Manager Core Template
- Oracle Authorization Policy Manager Template
- Oracle Identity Navigator Template

Note: The Oracle IDM Common template, which is listed as a dependency for other IDM templates, is an internal template. It is automatically called when you select other IDM templates in Configuration Wizard, or specify an IDM template in a WLST script. Therefore, it is not displayed as a selectable component in Configuration Wizard, and you do not have to specify it in your WLST scripts.

3.9.1 Oracle IDM Common Template

This template provides IDM Shell and other common IDM-wide infrastructure components that are not present in JRF.

The following table provides key information about this template.

Table 3–28

Template Detail	Information
Template type	Extension
Template name and version	Oracle IDM Common Template - 11.1.1.2.0
Template Dependencies	Oracle JRF - 11.1.1.0
Template JAR file and location	*ORACLE_HOME*/common/templates/applications/idm_common_template_11.1.1.2.0.jar

3.9.2 Oracle Adaptive Access Manager Admin Server Template

The Oracle Adaptive Access Manager Admin Server template adds the Oracle Adaptive Access Manager (OAAM) Administration Server application to an existing domain.

The following table provides key information about this template.

Table 3–29 Oracle Adaptive Access Manager Admin Server Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle Adaptive Access Manager Admin Server - 11.1.1.3.0
Template Dependencies	■ Oracle JRF - 11.1.1.0 ■ + Oracle Identity Navigator - 11.1.1.2.0 ■ + Oracle IDM Common Template - 11.1.1.2.0 ■ + Oracle Enterprise Manager - 11.1.1.0
Template JAR file and location	*ORACLE_HOME*/common/templates/applications/oracle.oaam_admin_11.1.1.3.0_template.jar

3.9.3 Oracle Adaptive Access Manager - Server Template

The Oracle Adaptive Access Manager - Server template adds the Oracle Adaptive Access Manager (OAAM) Server application to an existing domain.

The following table provides key information about this template.

Table 3–30 Oracle Adaptive Access Manager - Server Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle Adaptive Access Manager - Server 11.1.1.3.0
Template Dependencies	■ Oracle JRF - 11.1.1.0 ■ + Oracle Identity Navigator - 11.1.1.3.0 ■ + Oracle IDM Common Template - 11.1.1.2.0 ■ + Oracle WSM Policy Manager - 11.1.1.0 ■ + Oracle Enterprise Manager - 11.1.1.0
Template JAR file and location	*ORACLE_HOME*/common/templates/applications/oracle.oaam_server_11.1.1.3.0_template.jar

3.9.4 Oracle Access Manager Template

The Oracle Access Manager template adds an Oracle Access Manager (OAM) server to an existing domain. It also adds OAM applications and authentication providers to the domain.

The following table provides key information about this template.

Table 3–31 Oracle Access Manager Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle Access Manager - 11.1.1.3.0
Template Dependencies	■ Oracle JRF - 11.1.1.0 ■ + Oracle IDM Common - 11.1.1.2.0
Template JAR file and location	*ORACLE_HOME*/common/templates/applications/oracle.oam_11.1.1.3.0_template.jar

3.9.5 Oracle Access Manager with Database Policy Store Template

The Oracle Access Manager with Database Policy Store template is the same as the Oracle Access Manager template, except that it includes a JDBC data source and the associated JDBC component schema.

The following table provides key information about this template.

Table 3–32 Oracle Access Manager with Database Policy Store Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle Access Manager with File Policy Store 11.1.1.3.0
Template Dependencies	■ Oracle JRF - 11.1.1.0 ■ + Oracle IDM Common - 11.1.1.2.0 ■ + Oracle Access Manager - 11.1.1.3.0

Table 3–32 (Cont.) Oracle Access Manager with Database Policy Store Template Details

Template Detail	Information
Template JAR file and location	*ORACLE_HOME*/common/templates/applications/oracle.oam_ds_11.1.1.3.0_template.jar

3.9.6 Oracle Identity Manager Template

The Oracle Identity Manager template adds Oracle Identity Management (OIM) applications, as well as required JDBC and JMS resources, to an existing domain.

The following table provides key information about this template.

Table 3–33 Oracle Identity Manager Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle Identity Manager - 11.1.1.3.0
Template Dependencies	■ Oracle JRF - 11.1.1.0 ■ + Oracle IDM Common - 11.1.1.2.0 ■ + Oracle SOA Suite - 11.1.1.0 ■ + Oracle WSM Policy Manager - 11.1.1.0 ■ + Oracle Enterprise Manager - 11.1.1.0
Template JAR file and location	*ORACLE_HOME*/common/templates/applications/oracle.oim_11.1.1.3.0_template.jar

3.9.7 Oracle Authorization Policy Manager Core Template

The Oracle Authorization Policy Manager Core template deploys and configures Core Oracle Authorization Policy Manager (APM) components.

The following table provides key information about this template.

Table 3–34 Oracle Authorization Policy Manager Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle Authorization Policy Manager - Core - 11.1.1.3.0
Template Dependencies	■ Oracle JRF - 11.1.1.0 ■ + Oracle IDM Common - 11.1.1.2.0
Template JAR file and location	*ORACLE_HOME*/common/templates/applications/oracle.apm_core_11.1.1.3.0_template.jar

3.9.8 Oracle Authorization Policy Manager Template

The Oracle Authorization Policy Manager template deploys and configures Oracle Authorization Policy Manager (APM).

The following table provides key information about this template.

Table 3–35 Oracle Authorization Policy Manager Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle Authorization Policy Manager - 11.1.1.3.0
Template Dependencies	■ Oracle JRF - 11.1.1.0 ■ + Oracle IDM Common - 11.1.1.2.0 ■ + Oracle Authorization Policy Manager - Core - 11.1.1.3.0
Template JAR file and location	*ORACLE_HOME*/common/templates/applications/oracle.apm_11.1.1.3.0_template.jar

3.9.9 Oracle Identity Navigator Template

The Oracle Identity Navigator template adds the Oracle Identity Navigator application to an existing domain.

The following table provides key information about this template.

Table 3–36 Oracle Identity Navigator Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle Identity Navigator - 11.1.1.3.0
Template Dependencies	■ Oracle JRF - 11.1.1.0 ■ + Oracle IDM Common - 11.1.1.2.0
Template JAR file and location	*ORACLE_HOME*/common/templates/applications/oracle.oinav_11.1.1.3.0_template.jar

3.10 Miscellaneous Templates

This section describes templates that add necessary resources to many of the Fusion Middleware product domains. The following templates are described here:

- Oracle JRF Template
- Oracle JRF Asynchronous Web Services Template
- JSFDomain Template

Note: [The templates described in this section must be used as provided. Do not modify them in any way. Doing so can cause issues in your domain.]]

3.10.1 Oracle JRF Template

The Oracle Java Required Files (JRF) template configures components that not included in the WebLogic Server installation. These components provide common functionality for Oracle business applications and application frameworks. The SOA Suite and WebCenter are examples of applications and frameworks that depend on the JRF template.

The JRF template is also used independently to configure domains that contain applications that are developed using Oracle ADF and other core components.

The following table provides key information about this template.

Table 3–37 Oracle JRF Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle JRF - 11.1.1.0
Template Dependencies	None
Template JAR file and location	*ORACLE_COMMON_HOME*/common/templates/applications/jrf_template_11.1.1.jar

3.10.2 Oracle JRF Asynchronous Web Services Template

The Oracle JRF Asynchronous Web Services template creates default JMS resources that are required for JRF Asynchronous Web Services running on a WebLogic Server domain. This template does not create the JMS UDDs (Uniform Distributed Destinations) required for cluster(s) in the domain. To create the default JMS UDDs, a separate WLST script, `jrfws-async-createUDDs.py`, is provided.

This template must be targeted to non-clustered servers in the domain.

The following table provides key information about this template.

Table 3–38 Oracle JRF Asynchronous Web Services Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle JRF Web Services Asynchronous Services - 11.1.1.0
Template Dependencies	Oracle JRF - 11.1.1.0
Template JAR file and location	*ORACLE_COMMON_HOME*/common/templates/applications/oracle.jrf.ws.async_template_11.1.1.jar

3.10.3 JSFDomain Template

If you start JDeveloper in the Java EE role, the JSFDomain template provides the shared libraries that are needed to run JSF applications in Java EE role. If JDeveloper is in the Default role (studio edition), JRF provides the shared libraries to run ADFFaces Richclient, and the JSFDomain Template is not needed.

The following table provides key information about this template.

Table 3–39 JSF Domain Template Details

Template Detail	Information
Template type	Domain
Template name and version	JSFDomain 9.0.0.0

Table 3–39 (Cont.) JSF Domain Template Details

Template Detail	Information
Template Dependencies	None
Template JAR file and location	*ORACLE_HOME*\jdeveloper\common\templates\domains\jsf_template_1.2.9.0.jar