

Oracle® Fusion Middleware

Domain Template Reference

12c Release 1 (12.1.2)

E28064-01

June 2013

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 Fusion Middleware Domain Template Reference, 12c Release 1 (12.1.2)

E28064-01

Copyright © 2007, 2013, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface	v
Documentation Accessibility	v
Conventions	v
1 Introduction	
1.1 Types of Templates	1-1
1.2 Location of Installed WebLogic Server Templates	1-2
1.3 Template Tools	1-2
1.4 Template Dependencies	1-3
1.5 Files Typically Included in a Template	1-4
1.6 config-groups.xml and startup-plan.xml	1-7
1.6.1 Domain Topology Profiles	1-8
1.6.2 Application Service Groups, Server Groups, and Application Service Mappings	1-9
1.6.3 Startup Groups	1-11
1.6.3.1 Managing Server Startup Configuration	1-11
2 WebLogic Server Templates	
2.1 Basic WebLogic Server Domain Template	2-2
2.1.1 Template Details	2-2
2.1.2 Resources and Services Configured in a Basic WebLogic Server Domain	2-2
2.2 WebLogic Server Starter Domain Template	2-3
2.2.1 Template Details	2-3
2.2.2 Resources and Services Configured in a WebLogic Server Starter Domain	2-4
2.3 WebLogic Advanced Web Services for JAX-RPC Extension Template	2-5
2.3.1 Template Details	2-5
2.3.2 Resources and Services Configured	2-5
2.4 WebLogic Advanced Web Services for JAX-WS Extension Template	2-6
2.4.1 Template Details	2-7
2.4.2 Resources and Services Configured	2-7
2.5 WebLogic JAX-WS SOAP/JMS Extension Template	2-9
2.5.1 Template Details	2-9
2.5.2 Resources and Services Configured	2-9
2.6 Avitek Medical Records Sample Domain Template	2-10
2.6.1 Template Details	2-10
2.6.2 Resources and Services Configured	2-10

2.7	Avitek Medical Records Sample Domain Template (Spring Version).....	2-12
2.7.1	Template Details	2-12
2.7.2	Resources and Services Configured.....	2-12
2.8	WebLogic Server Default Domain Extension Template.....	2-14
2.8.1	Template Details	2-14
2.8.2	Resources and Services Configured.....	2-15
2.9	WebLogic Server Examples Extension Template.....	2-17
2.9.1	Template Details	2-17
2.9.2	Resources and Services Configured.....	2-17

3 Fusion Middleware Product Templates

3.1	Oracle JRF and ADF Templates	3-2
3.1.1	Oracle JRF Template.....	3-2
3.1.2	Oracle JRF Asynchronous Web Services Template	3-3
3.1.3	Oracle JRF SOAP/JMS Web Services Template.....	3-3
3.1.4	Oracle OPSS Extensions for JRF	3-4
3.1.5	Oracle ADF Development Mode Logging	3-4
3.2	Enterprise Manager Templates.....	3-4
3.2.1	Oracle Enterprise Manager.....	3-5
3.3	Coherence Template	3-5
3.3.1	WebLogic Coherence Cluster Extension Template.....	3-5
3.4	Oracle Web Service Manager (OWSM) Templates	3-6
3.4.1	Oracle WSM Policy Manager Template	3-6
3.5	Oracle Virtual Assembly Builder Templates	3-6
3.5.1	Oracle Virtual Assembly Builder Template.....	3-6
3.6	Oracle Data Integrator Templates	3-7
3.6.1	Oracle Data Integrator Standalone Agent Template	3-8
3.6.2	Oracle Data Integrator Standalone Collocated Agent Template	3-8
3.6.3	Oracle Data Integrator Agent Template.....	3-8
3.6.4	Oracle Data Integrator Agent Libraries Template	3-9
3.6.5	Oracle Data Integrator Console Template.....	3-9
3.6.6	Oracle Enterprise Manager Plugin for ODI Template	3-10
3.7	Oracle HTTP Server Templates.....	3-10
3.7.1	Oracle HTTP Server (Collocated)	3-10
3.7.2	Oracle HTTP Server (Standalone)	3-11

Preface

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

Documentation Accessibility

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

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	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.

Introduction

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

This document contains the following topics:

- [Section 1.1, "Types of Templates"](#)
- [Section 1.2, "Location of Installed WebLogic Server Templates"](#)
- [Section 1.3, "Template Tools"](#)
- [Section 1.4, "Template Dependencies"](#)
- [Section 1.5, "Files Typically Included in a Template"](#)
- [Section 1.6, "config-groups.xml and startup-plan.xml"](#)

1.1 Types of Templates

The types of template include:

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

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

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

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

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

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

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

You can create a custom Managed Server template by using the `pack` command. For more information, see *Creating Templates and Domains Using the Pack and Unpack Commands*.
- Reconfiguration template**—Reconfiguration templates are automatically implemented if you are upgrading a WebLogic domain from a previous WebLogic Server version. If a currently installed product, such as WebLogic Advanced Web Services for JAX-WS, requires updates to be compatible with the domain you are upgrading, a reconfiguration template is supplied to automatically make the product compatible with the current release, such as implementing new product features. When running the reconfiguration wizard, as described in "Reconfiguring a WebLogic Domain in Graphical Mode" in *Upgrading Oracle WebLogic Server*, the wizard automatically detects all products that are installed, determines whether or not there is an available reconfiguration template for each product, and then applies the reconfiguration template to update that product.

Reconfiguration templates are also provided for all Fusion Middleware products that are configured via the Fusion Middleware Configuration Wizard, such as SOA and Web Center. This makes it easy for you to update the domains for your Fusion Middleware products when upgrading to a new release of WebLogic Server.

The JAR file name for the reconfiguration template is *base_template_name_reconfig_version*. For example the Web Services for JAX-WS template is `wls_webservices_jaxws.jar`, and the associated reconfiguration template for WebLogic Server 12.1.2 is `wls_webservices_jaxws_reconfig_12.1.2.jar`.

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
WebLogic Server Domain, Extension, and Reconfiguration	<i>WL_HOME</i> \common\templates\wls
Fusion Middleware Extension and Reconfiguration	<i>ORACLE_HOME</i> \oracle _common\common\templates\wls <i>ORACLE_HOME</i> \PRODUCT_HOME\common\templates\wls

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	<ul style="list-style-type: none"> ■ Configuration Wizard ■ WLST Offline ■ <code>unpack</code> command
Extend an existing domain	<ul style="list-style-type: none"> ■ Configuration Wizard ■ WLST Offline
Create a Managed Server domain on a remote machine	<ul style="list-style-type: none"> ■ <code>unpack</code> command ■ WLST <code>writeTemplate</code> command (online)
Create a domain template	<ul style="list-style-type: none"> ■ Domain Template Builder ■ <code>pack</code> command ■ WLST Offline
Create an extension template	Domain Template Builder
Create a Managed Server template	<code>pack</code> command
Upgrade a domain	Reconfiguration Wizard

Note: All the tools used to create or extend a domain leverage a common underlying infrastructure, which is referred to as the configuration framework.

- For information about using the Configuration Wizard, see *Creating Domains Using the Configuration Wizard*.
- For information about using the WLST Offline, see *Understanding the 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*.
- For information about using the Reconfiguration Wizard, see *Upgrading Oracle WebLogic Server*.

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.

The `template-info.xml` file in a template JAR defines the template dependencies for a given template. Dependencies are chained. For example:

- Template A defines a dependency on Template B and Template C in its `template-info.xml` file.
- Template B defines a dependency on Template D and Template E in its `template-info.xml` file.
- Template C defines a dependency on Template F in its `template-info.xml` file.

In this example, if you select Template A on the Configuration Wizard's Templates screen, templates B, C, D, E, and F are automatically included in the domain. If any of these templates are displayed on the Templates screen, you will see the check boxes for those template automatically selected. This ensures that when you select a product template on the Configuration Wizard Templates screen, the Configuration Wizard automatically includes in the domain all other product templates that configure resources required by the product you selected.

Similarly, if you specify a template JAR in a WLST script, all other templates that are dependencies of that template (either directly or indirectly) are included in the domain. Using the above example, if you specify Template A in a WLST script, templates B, C, D, E, and F are also included in the domain without you having to explicitly specify them in the script.

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 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 <code>config\jdbc</code> 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 <code>config\jms</code> directory. This is applicable only if the domain requires JMS resources.

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

Filename	Description
clusters.script	<p>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:</p> <ul style="list-style-type: none"> ■ %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 <p>Note the following additional considerations:</p> <ul style="list-style-type: none"> ■ You must use the name attribute of an object that is to be replaced. ■ You can use an asterisk (*) as a wildcard for "All." <p>This file is not required. When used, it must be located in the <code>script</code> directory. If it is not present, default targeting is used.</p>
config.xml	<p>Defines the resources that the template creates or adds to a domain. In a template, the <code>config.xml</code> file must be located in the <code>config</code> directory.</p>
config-groups.xml	<p>This file contains definitions of applications, services, servers, clusters, and mappings that create a relationship among these items. It enables movement of functionally related applications and services as a single operation when transitioning from one topology to another (for example, from a single server to multiple servers, or from a single server to a cluster). This ensures that all application and service dependencies are met when scaling a domain configuration.</p> <p>Note: Do not modify this file in any way. It must be used as provided in the template.</p> <p>An Application/Service group specifies a set of functionally related applications and services. The applications and services are grouped together on a particular server or cluster.</p> <p>The Domain Topology section contains definitions of servers, as well as the targeting of applications and services to a specific server, group or servers or clusters. It contains the following definitions:</p> <ul style="list-style-type: none"> ■ Server group definitions—Specifies a server or servers that can house functionally related sets of applications and services, thereby enabling automatic server creation. ■ Cluster group definitions—Specifies a cluster that can house functionally related sets of applications and services, thereby enabling automatic cluster creation. ■ Application/Service group mapping definitions—Specifies targeting of an Application/Service group to a specific server, group of servers, or cluster, via the name of the Application/Service group. <p>As of WebLogic Server 12.1.2, the domain topology section may define separate server groups, startup groups, and application/service group mapping definitions for compact and expanded domain profiles. For more information about domain profiles, see Section 1.6, "config-groups.xml and startup-plan.xml."</p>
config-mapping.xml	<p>This file is used to dynamically assign values to custom variables that are defined in a deployment plan, using name/value pairs.</p>

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

Filename	Description
database.xml	<p>This file is included only in Fusion Middleware product templates that require JDBC data source definitions. It groups data sources into component schemas that are required to configure and load data into database objects via the Oracle Repository Creation Utility (RCU). It also contains the eligible database vendors and drivers, eliminating the possibility of selecting an unsupported database in the Fusion Middleware Configuration Wizard.</p> <p>Note: Do not modify this file in any way. It must be used as provided in the template.</p>
file-definition.xml	<p>Applies only to Fusion Middleware product templates. It defines file copy and string substitution operations that are done during domain creation or extension. String substitution operations are supported only for WebLogic Server and not supported in a WebSphere environment.</p>
jdbc.index	<p>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:</p> <pre>~/integration/common/dbscripts/oracle/reporting_runtime.sql</pre> <p>Specifically, the tilde represents the directory path identified by the \$USER_INSTALL_DIR\$ variable in the stringsubs.xml file.</p> <p>In a template, a jdbc.index file must be located in the <code>_jdbc_ _dbtype\dbversion</code> directory, where <i>dbtype</i> is the type of database, such as Oracle, and <i>dbversion</i> is the database version, such as 9i.</p> <p>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:</p> <ul style="list-style-type: none"> ▪ '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. <p>All these template jar files are located in the <code>WL_ HOME\common\templates\applications</code> directory.</p>
jvm-config.xml	<p>This file is specific to FMW product installations in a WebSphere environment, and can be ignored in a WebLogic Server environment.</p>
security.xml	<p>Used to create user groups and roles that establish identity and access to domain resources. You can create the default Admin user only through the security.xml file in a <i>domain</i> template. However, you can create user groups and roles through the security.xml file included in either a domain or an extension template.</p>
startscript.xml	<p>Used to create the *.cmd and *.sh files that are placed into a domain's root and bin directories.</p>

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

Filename	Description
startup-plan.xml	<p>Defines startup parameters for WebLogic Server instances, at the domain level or server group level. One or more of the following startup parameters may be defined in this file:</p> <ul style="list-style-type: none"> ■ Environmental variables ■ Java system properties ■ Java protocol handlers ■ WLS PRE_CLASSPATH ■ WLS POST_CLASSPATH ■ Java library path ■ WLS JVM initial heap size ■ WLS JVM maximum heap size ■ WLS JVM perm size ■ WLS JVM maximum perm size ■ Other java arguments <p>If the template defines both compact and expanded domain profiles, separate groups of startup parameter definitions are defined, one group for each profile type. The startup parameters used for your domain may differ depending on whether you create a compact or expanded domain. For more information about domain profiles and startup groups, see Section 1.6, "config-groups.xml and startup-plan.xml."</p>
stringsubs.xml	<p>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 <code>WL_HOME\common\lib\macrorules.xml</code>, where <code>WL_HOME</code> is the WebLogic Server installation directory.</p>
template-info.xml	<p>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).</p>
was-variable.xml	<p>This file is specific to FMW product installations in a WebSphere environment, and is ignored in a WebLogic Server environment.</p>

1.6 config-groups.xml and startup-plan.xml

Many templates contain a config-groups.xml file. If present, it defines one or more of the following items:

- domain topology profile
- one or more server groups
- application service groups
- one or more startup groups

Some templates also contain a startup-plan.xml file, which defines server startup parameters at a global (domain-wide) level or server group level.

When you create a domain using multiple templates, the config-group.xml files from all templates included in the domain are used to create the config-groups.xml file for the domain.

Similarly, the startup-plan.xml files from all templates included in the domain are merged to create the startup-plan.xml file for the domain. At domain creation, the merged startup plan is used to generate the appropriate scripts for the domain.

The merged config-groups.xml and startup-plan.xml files are stored in the domain's `init-info` directory.

Note: Do not manually edit either the config-groups.xml or startup-plan.xml files in the `init-info` directory.

The following sections describe each of these items in detail and how they work together in a domain.

- [Section 1.6.1, "Domain Topology Profiles"](#)
- [Section 1.6.2, "Application Service Groups, Server Groups, and Application Service Mappings"](#)
- [Section 1.6.3, "Startup Groups"](#)

1.6.1 Domain Topology Profiles

When creating a domain, you specify the domain topology profile (Compact or Expanded) to use for that domain. Domain profiles are defined only in some Fusion Middleware product templates, and do not apply to domains in which no Fusion Middleware products are installed with WebLogic Server.

Note: You can explicitly specify the domain topology profile only if you use WLST to create the domain. Graphical interfaces automatically configure the appropriate profile. For example, when creating a domain using the JDeveloper domain creation utility, the domain is created with a Compact profile; when creating a domain using the Fusion Middleware Configuration Wizard, the domain is created with an Expanded profile.

The config-groups.xml file in a Fusion Middleware product template may define the domain profiles for the domain, in the `profile` attribute of the `<domain-topology>` element. The domain profile can be either Compact (or single-instance) or Expanded (clustered). If the `<domain-topology>` does not define separate Custom and Expanded profiles, the configuration defined in `<domain-topology>` is used for both types of domains.

The domain profile defines:

- The server groups for the profile.
- Which server group is the Administration Server server group. The Administration Server group is not user-expandable.
- Whether or not a server group is user-expandable, that is, whether or not you can assign Managed Servers to it. Note that if a server group is not user-expandable, you can still assign Managed Servers to it by cloning an existing server that is already assigned to the server group.

Note: Use the WLST offline command `listServerGroups()` to list all user-expandable server groups in the domain.

- The application service groups that are targeted to each defined server group. All servers that are assigned to a server group inherit its targets.
- Whether or not a prefix is assigned to the name of any servers that are added to the server group. For example, if a prefix of 'xyz' is defined for the server group, and you add a server called 'server2' to the group, the server name is registered as 'xyz_server2'.

Table 1–4 describes the differences between Expanded and Compact domain profiles.

Table 1–4 Differences Between Expanded and Compact Domains

Expanded Domain	Compact Domain
Also known as a clustered domain.	Also known as a single-instance domain.
Contains an Administration Server. Contains Managed Servers for the Fusion Middleware product(s) in the domain. Fusion Middleware Managed Servers can be assigned to clusters.	Contains only an Administration Server, with no Managed Servers or clusters for the Fusion Middleware product(s) in the domain. This domain type is used primarily for development purposes.
Defines an Administration Server server group and one or more Managed Server server groups.	Defines an Administration Server server group. Although a Managed Server server group may be defined in config-groups.xml, it is not used.
All application service groups defined in config-groups.xml are targeted. Some application service groups are targeted to the Administration Server server group, other application service groups are targeted to the Managed Server server group(s).	All or a subset of the defined application service groups are targeted to the Administration Server server group. Some application service groups may not be targeted.

There are two ways to select the domain profile for a new domain:

- When creating a domain using the FMW Configuration Wizard, you select the profile to use on the Configuration Type screen. See "Configuration Type" in *Creating Domains Using the Configuration Wizard*.
- When creating a domain using WLST, you specify the profile to use in the `readTemplate` command. See "readTemplate" in *WebLogic Scripting Tool Command Reference*. The default profile type is Expanded.

1.6.2 Application Service Groups, Server Groups, and Application Service Mappings

The config-groups.xml file in Fusion Middleware templates may define application service groups, server groups, and application service mappings. This automates the assignment of applications and services to the appropriate servers in the domain.

- **Application service group**—These are defined in the `<app-svc-groups>` element. Each `<group>` element defines a unique application service group, which contain various applications and services that will be included in the domain, such as application deployments, work managers, JMS system resources, libraries, and other items that are needed in the product domain. Application service groups are always the same for each domain profile, although some application service groups may not be used in a Compact domain. Application service groups may be mapped to multiple server groups.

- **Server Group**—A named server group. Typically, there is at least one Administration Server group and at least one Managed Server group defined in a domain's config-groups.xml file. These are defined by a <server-group> element in the <domain-topology> element. Note that although a Compact profile may define both Administration Server and Managed Server server groups, only the Administration Server server group is used in a Compact domain.

If a server group is defined as user-expandable, you can add Managed Servers to the server group. User-expandable servers are listed in the **Server Groups** drop-down list of the Managed Server screen of the Fusion Middleware Configuration Wizard. In WLST, you can determine which server groups are user-expandable by using the `listServerGroups` command.

- **Application Service Mappings**—Define which application service groups are mapped to each defined server group. These mappings differ depending on the domain profile. They are defined in the <app-svc-group-mapping> elements in the <domain-topology> element of config-groups.xml.

Server groups target Fusion Middleware applications and services to one or more servers by mapping defined application service groups to each defined server group. A given application service group may be mapped to multiple server groups if needed. Any application services that are mapped to a given server group are automatically targeted to all servers that are assigned to that group.

For example, suppose that the following items are defined in config-groups.xml:

- Server group ADMIN-SVR (the server group for the Administration Server)
- Server group MGD-SVRS (the server group for Managed Servers)
- Application service group ADMIN-APPS, which defines the application services that will run only on the Administration Server
- Application service group MAIN-APPS, which defines applications that will run on Managed Servers
- Application service group MAIN-LIBS, which defines libraries that need to be targeted to the Administration Server and Managed Servers
- An application service mapping that maps MAIN-APPS to the MGD-SVRS server group
- An application service mapping that maps ADMIN-APPS to the ADMIN-SVR server group
- An application service mapping that maps MAIN-LIBS to the ADMIN-SVR server group
- An application service mapping that maps MAIN-LIBS to the MGD-SVRS server group

In this example, all applications and other resources that are defined in ADMIN-APPS are targeted to the Administration Server. All applications and other resources that are defined in MAIN-APPS are targeted to all Managed Servers. All libraries that are defined in MAIN-LIBS are targeted to the Administration Server and all Managed Servers.

Add a Server To or Remove a Server From a Server Group

You can use the WLST `setServerGroups()` command to add a server to any user-expandable server group or any server group that you created. You can also remove a server from any server group. The following examples demonstrate this.


```
# add a server to a server group
setServerGroups('my_server4', 'XYZ-MAN-SRVS', '180000')

# remove a server from a server group by setting the group to null
serverGroup = []
setServerGroups('my_server3', serverGroup)
```

1.6.3 Startup Groups

The startup-plan.xml file in a template defines the startup groups, which allow different startup parameters to be defined for different servers or groups of servers in a domain. A domain template may contain:

- a global startup definition, which defines the domain-wide startup settings for all servers in the domain.
- one or more server startup groups, which are associated with a server group. If present, these define the startup settings for all servers assigned to the server group. If a server startup group defines a setting that is already defined at the global level, the server-level setting takes precedence.

Different startup settings may be defined for Expanded and Compact domain profiles. In addition, when creating a domain, multiple templates may be applied to a domain. Therefore, all possible startup settings and startup groups are combined into a single startup-plan.xml file in the domain's `/init-info` directory. When you start a server, this file is referenced to determine:

- The startup group (if any) to which a server is assigned, based on the server group to which the server is assigned and the startup group to which the server group is assigned.
- The startup settings to use for the server, based on the startup group to which the server's server group is assigned.

If a server is not assigned to a server group, it is started using the global settings that are defined in startup-plan.xml.

[startup-plan.xml](#) in [Table 1-3](#) lists the startup parameters that may be configured in this file.

1.6.3.1 Managing Server Startup Configuration

Although the merged startup-plan.xml file for a domain defines the startup parameters for the servers in the domain, in some situations, you may want to use offline WebLogic Scripting Tool (WLST) to:

- Create your own startup groups to define unique startup parameters for one or more servers in the domain.
- Adjust the startup parameters for a server group.
- Add a server to or remove a server from a startup group. Note that although you can remove a server from a server group that is not user-modifiable, you can only add a server to either a user-modifiable server group or a server group that you have created.

The startup-plan.xml file for a domain is automatically updated with any changes you make to the startup configuration.

Creating and Modifying a Startup Group

You can create a new startup group from any existing server group in the domain. The new startup group inherits *only* the startup parameters from the server group you used to create the startup group. You can then change the startup parameter settings for the new startup group and assign individual servers to it.

There are two ways you can determine the server groups in a domain:

- Enter the WLST command `listServerGroups()`. This displays only user-expandable server groups.
- Open the domain's `init-info/config-groups.xml` file. Server group names are defined by the `name` attribute of each `<server group>` element in this file.

The following WLST example shows how to create a new startup group called XYZ-MGD-SVRS based on server group JRF-MAN-SVR, add a server to the group, and view and adjust the settings for the new group.

Example 1–1 Creating and Modifying a Startup Group

```
# Create a new startup group called XYZ-MGD-SVRS based on the startup settings
# for server group JRF-MAN-SVR
addStartupGroup('XYZ-MGD-SRVS', 'JRF-MAN-SVR')

# Set the startup group for my_server1 to XYZ-MGD-SRVS
setStartupGroup('my_server1', 'XYZ-MGD-SRVS')
# select the XYZ-MGD-SRVS startup group for modification
cd('/StartupGroupConfig/XYZ-MGD-SRVS')

# display the setting for MaxHeapSize
get('MaxHeapSize')
'1024'
# change the setting for MaxHeapSize
set('MaxHeapSize', '1536')

# get Java system properties for a startup group as a Python dictionary
dictionary = get('SystemProperties')

# set Java system properties for a startup group
dictionary['key.1'] = 'value.1'
dictionary['key.2'] = 'value.2'
set('SystemProperties',dictionary)

# get Java environment settings for a startup group as a Python dictionary
dictionary = get('EnvVars')

# set Java system properties for a startup group
dictionary['env.1'] = 'value.1'
dictionary['env.2'] = 'value.2'
set('EnvVars',dictionary)
```

Note: Calling `set('EnvVars', {})` resets all customizations, and environment variables for the startup group will revert back to the settings derived from the server groups associated with the startup group.

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 for JAX-RPC Extension Template	Extends an existing WebLogic Server domain to add functionality required for advanced JAX-RPC Web services, including Web services reliable messaging, 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.
WebLogic JAX-WS SOAP/JMS Extension Template	Extends a domain to include the resources required to use SOAP over JMS transport for JAX-WS Web services.
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.

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

Template	Description
WebLogic Server Examples Extension Template	Extends the WebLogic Server domain containing resources from the base WebLogic Server domain template and the WebLogic Server Default Domain extension template to create a complete WebLogic Server Examples domain. The WebLogic Server Examples domain contains a collection of examples that illustrate best practices for coding individual J2EE and WebLogic Server APIs.

2.1 Basic WebLogic Server Domain Template

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

2.1.1 Template Details

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

Table 2–2 Basic WebLogic Server Domain Template Information

Template Detail	Information
Template type	Domain
Template name	Basic WebLogic Server Domain
Template JAR file and location	WL_HOME\common\templates\wls\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	<p>The default configuration for the Administration Server is as follows:</p> <ul style="list-style-type: none"> ■ Listen address: All Local Addresses ■ Listen port: 7001 ■ SSL is disabled <p>When using the Configuration Wizard or WLST Offline to create a domain, if you want the Administration Server name to be different from the default name, AdminServer, you must configure the name manually. You cannot change the name later when applying an extension template.</p> <p>For information about customizing the Administration Server name while creating a domain with the Configuration Wizard, see "Creating WebLogic Domains" <i>Creating Domains Using the Configuration Wizard</i>.</p> <p>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 <i>Understanding the WebLogic Scripting Tool</i>.</p> <p>The following sample WLST Offline code snippet shows how to change the default Administration Server name, AdminServer, to MedRecServer.</p> <pre>#-----# Read the Basic WebLogic Server Domain template readTemplate('d:/MW_ HOME/wlserver/common/templates/wls/wls.jar') #Change the Administration Server name. cd('Servers/AdminServer') set('Name', 'MedRecServer') #-----#</pre>
Security realm	myrealm	<p>This is the default (active) WebLogic Server security realm. The administration user account, <code>weblogic</code>, is configured in this security realm, as well as all default groups and roles.</p>

2.2 WebLogic Server Starter Domain Template

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

2.2.1 Template Details

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

Table 2–4 WebLogic Server Starter Domain Template Information

Template Detail	Information
Template type	Domain
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	<code>WL_HOME\common\templates\wls\wls_starter.jar</code>
Template Dependencies	Base WebLogic Server domain template

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 2–5 Resources Configured in a WebLogic Server Starter Domain

Resource Type	Name	Notes
Administration Server	AdminServer	<p>The default configuration for the Administration Server is as follows:</p> <ul style="list-style-type: none"> ■ Listen address: All Local Addresses ■ Listen port: 7001 ■ SSL is disabled <p>When using the Configuration Wizard or WLST Offline to create a domain, if you want the Administration Server name to be different from the default name, AdminServer, you must configure the name manually. You cannot change the name later when applying an extension template.</p> <p>For information about customizing the Administration Server name while creating a domain with the Configuration Wizard, see "Creating WebLogic Domains" <i>Creating Domains Using the Configuration Wizard</i>.</p> <p>For information about customizing the Administration Server name while creating a domain with WLST Offline, see "Creating WebLogic Domains Using WLST Offline" in <i>Understanding the WebLogic Scripting Tool</i>.</p> <p>The following sample WLST Offline code snippet shows how to change the default Administration Server name, AdminServer, to MedRecServer.</p> <pre>#-----# Read the Basic WebLogic Server Domain template readTemplate('d:/MW_ HOME/wlserver/common/templates/wls/wls.jar') #Change the Administration Server name. cd('Servers/AdminServer') set('Name', 'MedRecServer') #-----#</pre>
Security realm	myrealm	The default (active) WebLogic Server security realm.

Table 2–5 (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.

2.3 WebLogic Advanced Web Services for JAX-RPC 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 JAX-RPC Web services. You accomplish this by adding the resources and services provided in the WebLogic Advanced Web Services for JAX-RPC 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 for JAX-RPC extension template.

Table 2–6 WebLogic Advanced Web Services for JAX-RPC Extension Template Information

Template Detail	Information
Template type	Extension
Category	Uncategorized
Template name and version	WebLogic Advanced Web Services for JAX-RPC Extension—12.1.2.0
Template Dependencies	Base WebLogic Server domain template
Template JAR file and location	<i>WL_HOME</i> \common\templates\wls\wls_webservice.jar
Reconfiguration template JAR file and location	<i>WL_HOME</i> \common\templates\wls\wls_webservice_reconfig_ <i>version</i> .jar
Domain topology profiles	None defined. The defined application service group is included for both Expanded and Compact domains.

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 for JAX-RPC extension template.

Table 2–7 Resources Configured in a WebLogic Advanced Web Services for JAX-RPC Domain

Resource Type	Name	Extension Result
Administration Server	AdminServer	<p>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.</p> <p>The default configuration for the Administration Server is as follows:</p> <ul style="list-style-type: none"> ■ Listen address: All Local Addresses ■ Listen port: 7001 ■ SSL is enabled ■ SSL listen port: 7002 <p>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."</p>
Security realm	myrealm	Uses the default security realm provided by the Basic WebLogic Server domain.
File Store	WseeFileStore	Adds the file store to be used as the persistent store for the WseeJmsServer JMS server. This file store is targeted to the Administration Server.
SAF Agent	ReliableWseeSAFAgent	Adds this store-and-forward agent, which uses the WseeFileStore, and targets it to the Administration Server. The SAF agent controls receipt and handling of reliable messages.
JMS Queues	WseeMessageQueue	<p>Adds the JMS queue to the JMS server, WseeJmsServer.</p> <p>Queues are under JMSModules/module name, these are under WseeJMSModule</p>
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.
Work Manager	weblogic.wsee.mdb.DispatchPolicy	Add this Work Manager and targets it to the Administration Server, AdminServer.

2.4 WebLogic Advanced Web Services for JAX-WS Extension Template

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

- Web services atomic transactions
- Security using WS-SecureConversation

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

For more information, see "Configuring Your Domain for Advanced Web Services Features" in *Developing 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 2–8 WebLogic Advanced Web Services for JAX-WS Extension Template Details

Template Detail	Information
Template type	Extension
Category	Uncategorized
Template name and version	WebLogic Advanced Web Services for JAX-WS Extension - 12.1.2.0
Template Dependencies	Base WebLogic Server domain template
Template JAR file and location	<i>WL_HOME</i> \common\templates\wls\wls_webservice_jaxws.jar
Reconfiguration template JAR file and location	<i>WL_HOME</i> \common\templates\wls\wls_webservice_jaxws_reconfig_ <i>version</i> .jar
Domain topology profiles	None defined. The defined application service group is included for both Expanded and Compact domains.

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 2–9 Resources Configured in a WebLogic Advanced Web Services for JAX-WS Domain

Resource Type	Name	Extension Result
Administration Server	AdminServer	<p>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.</p> <p>The default configuration for the Administration Server is as follows:</p> <ul style="list-style-type: none"> ■ Listen address: All Local Addresses ■ Listen port: 7001 ■ SSL is enabled ■ SSL listen port: 7002 <p>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."</p>
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	<p>Defines a distributed destination for the cluster. All associated targets will be used to support JAX-WS Web services. The subdeployment name is WseeJaxwsJmsServerSub.</p> <p>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).</p>
JMS Queues	weblogic.wsee.BufferedRequest Queue weblogic.wsee.BufferedRequestErrorQueue weblogic.wsee.BufferedResponseQueue weblogic.wsee.BufferedResponseErrorQueue	<p>Adds these JMS queues to the JMS server, and targets them to WseeJaxwsJmsServer.</p> <p>These queues are reserved for future use.</p>
SAF Agent	ReliableWseeJaxwsSAF Agent	Adds this store-and-forward agent, which uses the WseeJaxwsFileStore, and targets it to the Administration Server. The SAF agent controls receipt and handling of reliable messages.
Work Manager	weblogic.wsee.jaxws.mdb.DispatchPolicy	Adds this Work Manager and targets it to the Administration Server. The Work Manager defines the thread pool resources.

2.5 WebLogic JAX-WS SOAP/JMS Extension Template

By using the Configuration Wizard or WLST, you can easily extend a base WebLogic Server domain to include the resources required to use SOAP over JMS transport for JAX-WS 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.

2.5.1 Template Details

The following table provides basic information about the WebLogic JAX-WS SOAP/JMS extension template.

Table 2–10 WebLogic JAX-WS SOAP/JMS Extension Template Details

Template Detail	Information
Template type	Extension
Category	Uncategorized
Template name and version	WebLogic JAX-WS SOAP/JMS Extension—12.1.2.0
Template Dependencies	Base WebLogic Server domain template
Template JAR file and location	<code>WL_HOME\common\templates\wls\wls_webservice_soapjms.jar</code>
Reconfiguration template JAR file and location	<code>WL_HOME\common\templates\wls\wls_webservice_soapjms_reconfig_<version>.jar</code>
Domain topology profiles	None defined. The defined application service group is included for both Expanded and Compact domains.

2.5.2 Resources and Services Configured

The following table identifies the resources and services configured in a domain extended with the WebLogic JAX-WS SOAP/JMS extension template.

Table 2–11 Resources Configured in a WebLogic JAX-WS SOAP/JMS Domain

Resource Type	Name	Extension Result
JMS server	<code>WseeSoapjmsJmsServer</code>	JMS server management container.
File store	<code>WseeSoapjmsFileStore</code>	File store, or physical store, used by the WebLogic Server to handle the I/O operations to save and retrieve data from the physical storage (such as file, DBMS, and so on).
JMS Module	<code>WseeSoapjmsJmsModule</code>	JMS module that defines the JMS resources needed for SOAP over JMS transport.
JMS subdeployment	<code>WseeSoapjmsJmsServerSub</code>	JMS subdeployment for targeting the JMS resources to the <code>WseeSoapJmsServer</code> .
JMS Connection Factory	<code>com.oracle.webservices.jms.ConnectionFactory</code>	Default JMS connection factory used to create connections for SOAP over JMS transport.
JMS Queue	<code>com.oracle.webservices.api.jms.RequestQueue</code>	Default JMS request queue.

Table 2–11 (Cont.) Resources Configured in a WebLogic JAX-WS SOAP/JMS Domain

Resource Type	Name	Extension Result
JMS Queue	com.oracle.webservices. api.jms.ResponseQueue	Default JMS response queue.

2.6 Avitek Medical Records Sample Domain Template

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

For more information about the Avitek Medical Records sample application, see "Sample Application and Code Examples" in *Understanding Oracle WebLogic Server*.

2.6.1 Template Details

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

Table 2–12 Avitek Medical Records Sample Domain Information

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

2.6.2 Resources and Services Configured

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

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

Table 2–13 Resources Configured in an Avitek Medical Records Domain

Resource Type	Name	Extension Result
Administration Server	AdminServer	<p>Uses the Administration Server provided in the Base WebLogic Server domain. The default name is AdminServer, unless changed during domain creation. The Administration Server is referenced in the template as @SERVER_NAME, and appears as AdminServer in the Administration Console.</p> <p>The default configuration for the Administration Server is as follows:</p> <ul style="list-style-type: none"> ■ Listen address: All Local Addresses ■ Listen port: 7001 ■ SSL is enabled ■ SSL port: 7002 <p>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."</p>
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.
Application Deployment	SamplesSearchWebApp	Adds a sample search application and targets it to the Administration Server.
JDBC Data Sources	MedRecGlobalDataSourceXA	Identifies the JDBC data source as a MedRecGlobalDataSourceXA system resource.
JMS Queues	com.oracle.medrec.jms.RecordToCreateQueue com.oracle.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.
Mail Session	mail/MedRecMailSession	Adds this mail session.

Table 2–13 (Cont.) Resources Configured in an Avitek Medical Records Domain

Resource Type	Name	Extension Result
SAF Agent	WsrnAgent	Adds this store-and-forward agent, which uses the MedRecWseeFileStorfile store, and targets it to the MedRecServer.
Deployed library	jsf1.2@1.2.9.0	Adds the Java Server Faces Version 1.2 libraries.
Deployed library	jstl1.2.@1.2.0.2	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.

2.7 Avitek Medical Records Sample Domain Template (Spring Version)

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

For more information about the Avitek Medical Records sample application, see "Sample Application and Code Examples" in *Understanding Oracle WebLogic Server*.

2.7.1 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 2–14 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	<code>WL_HOME\common\templates\wls\medrec_spring.jar</code>
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 Avitek Medical Records Sample extension template for Spring.

Table 2–15 Resources Configured in an Avitek Medical Records Domain for Spring

Resource Type	Name	Extension Result
Administration Server	AdminServer	<p>Uses the Administration Server provided in the base WebLogic Server domain. The default name is AdminServer, unless changed during domain creation. The Administration Server is referenced in the template as @SERVER_NAME, and appears as AdminServer in the Administration Console.</p> <p>The default configuration for the Administration Server is as follows:</p> <ul style="list-style-type: none"> ■ Listen address: All Local Addresses ■ Listen port: 7001 ■ SSL is enabled ■ SSL port: 7002 <p>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."</p>
Security realm	myrealm	Uses the security realm provided in the base WebLogic Server domain.
Application Deployments	browser-starter	Adds the browser-starter Web application and targets it to the MedRecServer.
Application Deployments	medrec	Adds the sample medrec Enterprise application and targets it to the MedRecServer.
Application Deployments	physician	Adds the sample physician Enterprise application and targets it to the MedRecServer.
Application Deployment	SamplesSearchWebApp	Adds a sample search application and targets it to the Administration Server.
JDBC Data Sources	MedRecGlobalDataSourceXA	<p>Identifies the JDBC data source as a MedRecGlobalDataSourceXA system resource.</p> <p>Pool capacity (initial): 2</p> <p>Pool capacity (maximum): 10</p> <p>Protocol: Two Phase Commit</p>
JMS Queues	com.oracle.medrec.jms.RecordToCreateQueue com.oracle.medrec.jms.PatientNotificationQueue weblogic.wsee.DefaultQueue	Adds these JMS queues to the JMS server, MedRecWseeJMSServer.
JMS Servers	MedRecJMSServer	Adds the JMS server as a MedRec-jms system resource and targets it to the MedRecServer.
JMS System Resources	MedRec-jms	Adds the JMS servers, connection factories, and queues to be used as JMS system resources, and targets the resources to the MedRecServer.

Table 2–15 (Cont.) Resources Configured in an Avitek Medical Records Domain for

Resource Type	Name	Extension Result
Mail Session	mail/MedRecMailSession	Adds this mail session.
SAF Agent	WsrnAgent	Adds this store-and-forward agent, which uses the file store, MedRecWseeFileStore, and targets it to the MedRecServer.
Deployed library	jsf1.2@1.2.9.0	Adds the Java Server Faces Version 1.2 library and targets it to the MedRecServer.
Deployed library	jstl1.2.@1.2.0.1	Adds the Java standard tagging (JSTL) Version 1.2 library and targets it to the MedRecServer.
Deployed library	weblogic-spring#10.3.6.0@10.3.6.0	Adds the WebLogic Spring Version 10.3.6 library and targets it to the MedRecServer.
WLDF System Resource	MedRecWLDF	Adds this WLDF system resource, and targets it to the MedRecServer. The WLDF resource defines an instrumentation monitor for dye injection, and a harvester metric (com.oracle.medrec.admin.AdminReport).

2.8 WebLogic Server Default Domain Extension Template

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

Note: Applying the WebLogic Server Default Domain extension template to a base WebLogic domain is a prerequisite to using the WebLogic Server Examples extension template.

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

2.8.1 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 Default Domain extension template.

Table 2–16 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.

Table 2–16 (Cont.) WebLogic Server Default Domain Information

Template Detail	Information
Template JAR file and location	WL_HOME\common\templates\wls\wls_default.jar
Template Dependencies	Basic WebLogic Server Domain template

2.8.2 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 2–17 Resources Configured in a WebLogic Server Default Domain

Resource Type	Name	Extension Result
Administration Server	AdminServer	<p>Uses the Administration Server provided in the base WebLogic Server domain. The default name is AdminServer, unless changed during domain creation. The Administration Server is referenced in the template as @SERVER_NAME, and appears as AdminServer in the Administration Console.</p> <p>The default configuration for the Administration Server is as follows:</p> <ul style="list-style-type: none"> ■ Listen address: All Local Addresses ■ Listen port: 7001 ■ SSL is disabled <p>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."</p>
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	entityBeanValidation	Adds the entityBeanValidation application and targets it to the examplesServer Administration Server.
Application Deployment	jsfBeanValidation	Adds the jsfBeanValidation application and targets it to the examplesServer Administration Server.
Application Deployment	cdi	Adds the cdi application and targets it to the examplesServer Administration Server.
Application Deployment	asyncMethodOfEJB	Adds the asyncMethodOfEJB Enterprise application and targets it to the examplesServer Administration Server.
Application Deployment	calendarStyledTimer	Adds the calendarStyledTimer application and targets it to the examplesServer Administration Server.
Application Deployment	noInterfaceViewInWAR	Adds the noInterfaceViewInWAR application and targets it to the examplesServer Administration Server.

Table 2–17 (Cont.) Resources Configured in a WebLogic Server Default Domain

Resource Type	Name	Extension Result
Application Deployment	portableGlobalJNDIName	Adds the portableGlobalJNDIName application and targets it to the examplesServer Administration Server.
Application Deployment	singletonBean	Adds the singletonBean application and targets it to the examplesServer Administration Server.
Application Deployment	jaxrs	Adds the jaxrs application and targets it to the examplesServer Administration Server.
Application Deployment	stockAdapter	Adds the stockAdapter application and targets it to the examplesServer Administration Server.
Application Deployment	stockFrontEnd	Adds the stockFrontEnd application and targets it to the examplesServer Administration Server.
Application Deployment	stockBackEnd	Adds the stockBackEnd application and targets it to the examplesServer Administration Server.
Application Deployment	jdbcDataSource.war	Adds the jdbcDataSource.war application and targets it to the examplesServer Administration Server.
Application Deployment	criteriaQuery	Adds the criteriaQuery application and targets it to the examplesServer Administration Server.
Application Deployment	elementCollection	Adds the elementCollection application and targets it to the examplesServer Administration Server.
Application Deployment	ajaxJSF	Adds the ajaxJSF application and targets it to the examplesServer Administration Server.
Application Deployment	bookmarkingJSF	Adds the bookmarkingJSF application and targets it to the examplesServer Administration Server.
Application Deployment	faceletsJSF	Adds the faceletsJSF application and targets it to the examplesServer Administration Server.
Application Deployment	annotation	Adds the annotation application and targets it to the examplesServer Administration Server.
Application Deployment	asyncServlet30	Adds the asyncServlet30 application and targets it to the examplesServer Administration Server.
Application Deployment	multipartFileHandling	Adds the multipartFileHandling application and targets it to the examplesServer Administration Server.
Application Deployment	apache_xbean.jar	Adds the apache_xbean.jar application and targets it to the examplesServer Administration Server.
Application Deployment	annotation	Adds the annotation application and targets it to the examplesServer Administration Server.
JDBC System Resource	examples-demo	<p>Identifies this JDBC data source, which has the following configuration:</p> <ul style="list-style-type: none"> ■ JNDI name: examples-dataSource-demoPool ■ Global transaction protocol: Two Phase Commit <p>The connection pool settings are:</p> <ul style="list-style-type: none"> ■ Initial capacity: 1 ■ Maximum capacity: 10

Table 2–17 (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: <ul style="list-style-type: none"> ▪ JNDI Name: examples-dataSource-demoXAPool ▪ Global transaction protocol: Two Phase Commit The connection pool settings are: <ul style="list-style-type: none"> ▪ Initial capacity: 2 ▪ Maximum capacity: 10
Deployed library	apache-xbean.jar	Adds the apache-xbean.jar library dependency to this domain.

2.9 WebLogic Server Examples Extension Template

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

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

2.9.1 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 2–18 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	<code>WL_HOME\common\templates\wls\wls_examples.jar</code>
Template Dependencies	<ul style="list-style-type: none"> ▪ Basic WebLogic Server Domain template ▪ + WebLogic Server Default Domain Extension template

2.9.2 Resources and Services Configured

In addition to the resources configured by the WebLogic Server Default Domain extension template (see [Table 2–17](#)), the WebLogic Server Examples extension template configures the resources and services listed in the following table.

Table 2–19 Additional Resources Configured by the WebLogic Server Examples Domain

Resource Type	Name	Notes
Administration Server	AdminServer	<p>Uses the Administration Server provided in the Basic WebLogic Server domain. The default name is AdminServer, unless changed during domain creation. The Administration Server is referenced in the template as @SERVER_NAME, and appears as AdminServer in the Administration Console.</p> <p>The default configuration for the Administration Server is as follows:</p> <ul style="list-style-type: none"> ■ Listen address: All Local Addresses ■ Listen port: 7001 ■ SSL is disabled <p>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."</p>
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	<p>Adds this JMS server as an examples-jms system resource and targets it to the Administration Server.</p> <ul style="list-style-type: none"> ■ Persistent Store: exampleJDBCStore ■ JMS Message log file: examplesJMSServer
JMS Server	WseeJMSServer	<p>Adds this JMS server as an examples-jms system resource and targets it to the Administration Server.</p> <p>This server is configured to use the WseeFileStore persistent store.</p>
File Store	WseeFileStore	Adds the file store to be used as the persistent store for the WSEEJMSServer JMS server and the ReliableWseeSAFAgent SAF Agent. This file store is targeted to the examplesServer Administration Server.
JDBC Store	exampleJDBCStore	Adds the JDBC store to be used as the persistent store for the examples-demo JDBC data source and the examplesJMSServer JMS server, and targets the store to the examplesServer Administration Server.
JMS System Resources	examples-jms	Identifies the JMS servers, connection factories, queues, and topics to be used for JMS system resources.
JMS Connection Factories	exampleTopic exampleTrader weblogic.examples.jms.QueueConnectionFactory weblogic.examples.ejb30.QueueConnectionFactory	Adds these connection factories as examples-jms system resources and targets them to the examplesServer server.

Table 2–19 (Cont.) Additional Resources Configured by the WebLogic Server Examples

Resource Type	Name	Notes
JMS Queues	exampleQueue jms/MULTIDATASOURCE_MDB_QUEUE weblogic.examples.ejb30.ExampleQueue	Adds these JMS queues to the examplesJMSserver JMS server.
JMS Queue	weblogic.wsee.wseeExamplesDestinationQueue	Adds this JMS queue to the WseeJMSserver JMS server.
JMS Topics	exampleTopic quotes stockTopic	Adds these JMS topics and targets them to the examplesJMSserver JMS server.
JDBC System Resource	examples-oracleXA	Identifies this JDBC data source, which has the following configuration: <ul style="list-style-type: none"> ■ JNDI name: examples-dataSource-oracleXAPool ■ Global transaction protocol: Two Phase Commit <p>The database driver is configured as oracle.jdbc.xa.client.OracleXADataSource.</p> <p>This data source is mapped to the examples-multiDataSource-demoXAPool multi data source.</p>
JDBC System Resource	examples-demoXA-2	Identifies this JDBC data source, which is targeted to the Administration Server and has the following configuration: <ul style="list-style-type: none"> ■ Connection pool maximum: 100 ■ Global transaction protocol: Two Phase Commit <p>This data source is mapped to the examples-multiDataSource-demoXAPool multi data source.</p>
JDBC System Resource	examples-multiDataSource-demoXAPool	Identifies this JDBC multi data source, which is targeted to the Administration Server. It is configured for failover, and maps to the examples-oracleXA and examples-demo-XA-2 data sources.
SAF Agent	ReliableWseeSAFAgent	Adds this store-and-forward agent, which uses the WseeFileStore file store, and targets it to the Administration Server.
Work Manager	weblogic.wsee.mdb.DispatchPolicy	Adds this Work Manager, but does not target it to any servers.

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 Templates screen 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:

- [Section 3.1, "Oracle JRF and ADF Templates"](#)
- [Section 3.2, "Enterprise Manager Templates"](#)
- [Section 3.3, "Coherence Template"](#)
- [Section 3.4, "Oracle Web Service Manager \(OWSM\) Templates"](#)
- [Section 3.5, "Oracle Virtual Assembly Builder Templates"](#)
- [Section 3.6, "Oracle Data Integrator Templates"](#)
- [Section 3.7, "Oracle HTTP Server Templates"](#)

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 a special 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.

- **Category**—The category in which the template is listed on the Templates screen of the Configuration Wizard.
- **Template name and version**—The template (product) name and template version. If the template is selectable on the Templates 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. For more information on template dependencies, see [Section 1.4, "Template Dependencies."](#)
- **Template JAR file and location**—The JAR filename and its location in the Oracle home directory. You need this information only if you plan to use WLST scripts to set up a product domain.

3.1 Oracle JRF and ADF 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](#)
- [Oracle JRF SOAP/JMS Web Services Template](#)
- [Oracle ADF Development Mode Logging](#)

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.

3.1.1 Oracle JRF Template

The Oracle Java Required Files (JRF) template configures components that are 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 3–1 Oracle JRF Template Details

Template Detail	Information
Template type	Extension
Category	Oracle
Template name and version	Oracle JRF - 12.1.2.0

Table 3–1 (Cont.) Oracle JRF Template Details

Template Detail	Information
Template Dependencies	<ul style="list-style-type: none"> ■ Oracle Click History - 12.1.2 ■ WebLogic Coherence Cluster Extension - 12.1.2.0 ■ Oracle WSM Agent - 12.1.2.0 ■ Oracle ADF and MDS Extensions for JRF - 12.1.2.0 ■ Oracle OPSS Metadata for JRF - 12.1.2.0 ■ Oracle User Messaging Service Client Extensions for JRF - 12.1.2.0 ■ Common Infrastructure Engineering Runtime - 12.1.2.0
Template JAR file and location	<code>ORACLE_HOME/oracle_common/common/templates/wls/oracle.jrf_template_12.1.2.jar</code>

3.1.2 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 clusters 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 3–2 Oracle JRF Asynchronous Web Services Template Details

Template Detail	Information
Template type	Extension
Category	Oracle
Template name and version	Oracle JRF Web Services Asynchronous Services - 12.1.2.0
Template Dependencies	Oracle JRF - 12.1.2.0
Template JAR file and location	<code>ORACLE_HOME/oracle_common/common/templates/wls/oracle.jrf.ws.async_template_12.1.2.jar</code>

3.1.3 Oracle JRF SOAP/JMS Web Services Template

This template creates JMS queues for JRF SOAP/JMS Web services.

Table 3–3 Oracle JRF Asynchronous Web Services Template Details

Template Detail	Information
Template type	Extension
Category	Oracle
Template name and version	Oracle JRF SOAP/JMS Web Services - 12.1.2.0
Template Dependencies	Oracle JRF - 12.1.2.0

Table 3–3 (Cont.) Oracle JRF Asynchronous Web Services Template Details

Template Detail	Information
Template JAR file and location	<code>ORACLE_HOME/oracle_common/common/templates/wls/oracle.jrf.ws.soapjms_template_12.1.2.jar</code>

3.1.4 Oracle OPSS Extensions for JRF

This template is required for Fusion Middleware components and applications that were built using Oracle Application Development Framework.

The following table provides key information about this template.

Table 3–4 Oracle OPSS Extensions for JRF Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle OPSS Extensions for JRF - 12.1.2.0
Template Dependencies	Oracle JRF - 12.1.2.0
Template JAR file and location	<code>ORACLE_HOME/oracle_common/common/templates/wls/oracle.opss_template_12.1.2.jar</code>

3.1.5 Oracle ADF Development Mode Logging

This template is included if Oracle JDeveloper is installed. It provides development mode logging for Oracle ADF. Fine logging for `oracle.jbo` is exposed.

The following table provides key information about this template.

Table 3–5 Oracle ADF Development Mode Logging Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle ADF Development Mode Logging - 12.1.2.0.0
Template Dependencies	None
Template JAR file and location	<code>ORACLE_HOME/jdeveloper/common/templates/jdeveloper/adf.fine.logging_12.1.2.jar</code>

3.2 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 Templates screen of the Fusion Middleware Configuration Wizard.

3.2.1 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 *Administering Oracle Fusion Middleware*.

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

The following table provides key information about this template.

Table 3–6 Oracle Enterprise Manager Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle Enterprise Manager - 12.1.2.0
Template Dependencies	<ul style="list-style-type: none"> ■ Basic WebLogic Server Domain ■ + Oracle JRF - 12.1.2.0 ■ + Oracle WebCenter Composer
Template JAR file and location	<p>This template consists of multiple template JAR files. These JAR files are located in the following directory after you install Oracle Fusion Middleware:</p> <p><code>ORACLE_HOME/em/common/templates/wls</code></p> <p>The template consists of the following JAR files in the templates directory:</p> <ul style="list-style-type: none"> ■ <code>oracle.em_wls_template_12.1.2.jar</code> ■ <code>oracle.emas_wls_template_12.1.2.jar</code>

3.3 Coherence Template

This section describes the Coherence template that is provided when Coherence is installed with WebLogic Server.

3.3.1 WebLogic Coherence Cluster Extension Template

This template adds a default Coherence cluster, `defaultCoherenceCluster`, to the WebLogic domain and sets the listen port for the cluster to 8088.

Table 3–7 WebLogic Coherence Cluster Extension Template Details

Template Detail	Information
Template type	Extension Template

Table 3–7 (Cont.) WebLogic Coherence Cluster Extension Template Details

Template Detail	Information
Template name and version	WebLogic Coherence Cluster Extension - 12.1.2.0
Template Dependencies	<ul style="list-style-type: none"> ■ Basic WebLogic Server Domain
Template JAR file and location	<code>WL_HOME/common/templates/applications/wls_coherence.jar</code>

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 Java EE application.

The following table provides key information about this template.

Table 3–8 Oracle WSM Policy Manager Template Details

Template Detail	Information
Template type	Extension
Template name and version	Oracle WSM Policy Manager - 12.1.2.0
Template Dependencies	<ul style="list-style-type: none"> ■ Basic WebLogic Server Domain ■ + Oracle JRF - 12.1.2.0
Template JAR file and location	<code>ORACLE_HOME/oracle_common/common/templates/wls/oracle.wsmrpm_template_12.1.2.jar</code>

3.5 Oracle Virtual Assembly Builder Templates

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

- [Section 3.5.1, "Oracle Virtual Assembly Builder Template"](#)

3.5.1 Oracle Virtual Assembly Builder Template

With the expanded domain topology profile, this template provides a domain type that supports the Oracle Virtual Assembly Builder Deployer cluster service, and allows you to configure an Oracle WebLogic Server cluster and a Coherence cluster and target the Oracle WebLogic Server cluster to use the Coherence cluster.

You can configure the Coherence cluster with either Well Known Addresses (WKA) or multicast communication. If you do not specify a communication type, WKA is set as for cluster communication.

Oracle Virtual Assembly Builder Deployer uses Coherence cluster data partitioning to distribute work. The `deployerId` is used as a key association, where all data specific to a particular deployment id is guaranteed to exist on the same node. When you initiate an asynchronous request (such as a deploy, undeploy, or scale operation), the request is handled by the node on which the data is partitioned at that time. The same node handles the request even if data gets repartitioned to another node.

If a node with running requests goes down or crashes, the requests are cancelled, because none of the Deployer asynchronous operations are idempotent. That is, if a request is initiated, it cannot be rolled back because the work is distributed amongst multiple tiers.

You must re-initiate such requests by cleaning up the previous state. For example: if a deploy operation gets cancelled, the corresponding deployment object is marked as 'Failed' with the reason as 'cancelled'. You must invoke an explicit cleanup operation by the to cleanup the deployment, then create another deployment object and start the deploy operation.

With the compact domain topology profile, this template provides for a standalone Oracle Virtual Assembly Builder Deployer.

The following table provides key information about this template.

Table 3–9 Oracle Virtual Assembly Builder Template Details

Template Detail	Information
Template type	Extension
Category	Uncategorized
Template name and version	Oracle Virtual Assembly Builder Deployer - 12.1.2.0
Template Dependencies	Oracle JRF - 12.1.2.0
Template JAR file and location	<code>OVAB_HOME/common/templates/wls/oracle.ovab.deployer.template_12.1.2.jar</code>
Domain topology profiles	Compact and Expanded

3.6 Oracle Data Integrator Templates

This section describes the following Oracle Data Integrator templates:

- [Oracle Data Integrator Standalone Agent Template](#)
- [Oracle Data Integrator Standalone Collocated Agent Template](#)
- [Oracle Data Integrator Agent Template](#)
- [Oracle Data Integrator Agent Libraries Template](#)
- [Oracle Data Integrator Console Template](#)
- [Oracle Enterprise Manager Plugin for ODI Template](#)

Note: The ODI Master Datasource template, the ODI Base template, and the ODI Work Datasource template, which are listed as dependencies for other Oracle Data Integrator templates, are internal templates. They are automatically called when you select other ODI templates in Configuration Wizard, or specify an ODI template in a WLST script. Therefore, they are not displayed as selectable components in Configuration Wizard, and you do not have to specify them in your WLST scripts.

3.6.1 Oracle Data Integrator Standalone Agent Template

Use the Oracle Data Integrator Standalone Agent when your ODI installation is not being managed by WebLogic Server.

The following table provides key information about this template.

Table 3–10 Oracle Data Integrator Standalone Agent Template Details

Template Detail	Information
Template type	Extension
Category	Oracle Data Integrator
Template name and version	Oracle Data Integrator Collocated Agent - 12.1.2.0
Template Dependencies	<ul style="list-style-type: none"> ■ Common Infrastructure Engineering Runtime - 12.1.2.0 ■ Oracle Data Integrator Standalone Agent Based - 12.1.2.0
Template JAR file and location	<code>ORACLE_HOME/odi/common/templates/wls/odi_cam_unmanaged_template_12.1.2.jar</code>

3.6.2 Oracle Data Integrator Standalone Collocated Agent Template

Use the Oracle Data Integrator Standalone Collocated Agent template when you install ODI collocated with WebLogic Server.

The following table provides key information about this template.

Table 3–11 Oracle Data Integrator Standalone Collocated Agent Template Details

Template Detail	Information
Template type	Extension
Category	Oracle Data Integrator
Template name and version	Oracle Data Integrator Standalone Collocated Agent - 12.1.2.0
Template Dependencies	<ul style="list-style-type: none"> ■ Oracle JRF 12.1.2.0
Template JAR file and location	<code>ORACLE_HOME/odi/common/templates/wls/odi_cam_managed_template_12.1.2.jar</code>

3.6.3 Oracle Data Integrator Agent Template

The Oracle Data Integrator Agent template deploys the ODI Agent application, required libraries and the ODI Master repository data source.

The following table provides key information about this template.

Table 3–12 Oracle Data Integrator Agent Template Details

Template Detail	Information
Template type	Extension
Category	Oracle Data Integrator
Template name and version	Oracle Data Integrator Agent - 12.1.2.0
Template Dependencies	<ul style="list-style-type: none"> ■ ODI Agent Libraries Template ■ ODI Master Datasource Template
Template JAR file and location	<code>ORACLE_HOME/odi/common/templates/wls/odi_agent_template_12.1.2.jar</code>

3.6.4 Oracle Data Integrator Agent Libraries Template

The Oracle Data Integrator Agent Libraries template deploys shared libraries required for the ODI Agent.

The following table provides key information about this template.

Table 3–13 Oracle Data Integrator Agent Libraries Template Details

Template Detail	Information
Template type	Extension
Category	Oracle Data Integrator
Template name and version	Oracle Data Integrator Agent Libraries - 12.1.2.0
Template Dependencies	<ul style="list-style-type: none"> ■ ODI Base Template ■ Oracle Data Integrator SDK Shared Libraries Template
Template JAR file and location	<code>ORACLE_HOME/odi/common/templates/wls/odi_agent_libraries_template_12.1.2.jar</code>

3.6.5 Oracle Data Integrator Console Template

The Oracle Data Integrator Console template deploys the ODI Console application.

The following table provides key information about this template.

Table 3–14 Oracle Data Integrator Console Template Details

Template Detail	Information
Template type	Extension
Category	Oracle Data Integrator
Template name and version	Oracle Data Integrator Console - 12.1.2.0
Template Dependencies	<ul style="list-style-type: none"> ■ ODI Work Datasource ■ Oracle Data Integrator SDK Shared Libraries Template
Template JAR file and location	<code>ORACLE_HOME/odi/common/templates/wls/odi_repository_explorer_template_12.1.2.jar</code>

3.6.6 Oracle Enterprise Manager Plugin for ODI Template

The Oracle Enterprise Manager Plugin for ODI template deploys the ODI Plugin for Oracle Enterprise Manager.

The following table provides key information about this template.

Table 3–15 Oracle Enterprise Manager Plugin for ODI Template Details

Template Detail	Information
Template type	Extension
Category	Oracle
Template name and version	Oracle Enterprise Manager Plugin for ODI - 12.1.2.0
Template Dependencies	<ul style="list-style-type: none"> ▪ Oracle Enterprise Manager 12.1.2.0
Template JAR file and location	<code>ORACLE_HOME/em/common/templates/wls/oracle.em_odi_template_12.1.2.jar</code>

3.7 Oracle HTTP Server Templates

The Oracle HTTP Server (OHS) domain templates are extension templates used to add OHS resources and services to a Basic WebLogic Server domain. OHS is the Web server component for Oracle Fusion Middleware. It provides a listener for Oracle WebLogic Server and the framework for hosting static pages, dynamic pages, and applications over the Web. The specific OHS template you will use depends on whether you are implementing OHS in a collocated or standalone environment.

This chapter contains these sections:

- [Section 3.7.1, "Oracle HTTP Server \(Collocated\)"](#)
- [Section 3.7.2, "Oracle HTTP Server \(Standalone\)"](#)

3.7.1 Oracle HTTP Server (Collocated)

Use the Oracle HTTP Server (Collocated) domain template when you want your OHS implementation to serve one or more FMW domains (for example, when you want OHS to front requests for a SOA domain) or when you want to take advantage of the advanced management capabilities provided by FMW Control.

Table 3–16 Oracle HTTP Server (Collocated) Domain Template Details

Template Detail	Information
Template type	Extension
Category	Uncategorized
Template name and version	Oracle HTTP Server (Collocated)—12.1.2.0
Template Dependencies	Oracle Enterprise Manager for WEBTIER—12.1.2.0
Template JAR file and location	<code>MW_HOME/ohs/common/templates/wls/ohs_managed_template_12.1.2.jar</code>
Domain topology profiles	Compact and Expanded.
Category	Undefined

3.7.2 Oracle HTTP Server (Standalone)

Use the Oracle HTTP Server (Standalone) domain template when you do not want your OHS implementation to front an FMW domain and do not need the management functionality provided by FMW Control or when you want to keep OHS in a DMZ and you do not want to open management ports used by the Node Manager.

Table 3–17 Oracle HTTP Server (Standalone) Domain Template Details

Template Detail	Information
Template type	Extension
Category	Oracle HTTP Server
Template name and version	Oracle HTTP Server (Standalone)—12.1.2.0
Template Dependencies	None
Template JAR file and location	MW_HOME/ohs/common/templates/wls/ohs_standalone_template_12.1.2.jar
Domain topology profiles	None defined.
Server groups	Undefined

