

# Oracle® Fusion Middleware

## Domain Template Reference



14c (14.1.1.0.0)

F25729-05

February 2023

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle Fusion Middleware Domain Template Reference, 14c (14.1.1.0.0)

F25729-05

Copyright © 2014, 2023, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

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, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. 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 fail-safe, 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®, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside 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, Epyc, and the AMD 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 about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

# Contents

## Preface

---

Documentation Accessibility	v
Diversity and Inclusion	v
Conventions	v

## 1 Introduction

---

Types of Templates	1-1
Location of Installed WebLogic Server Templates	1-2
Template Tools	1-3
Template Dependencies	1-4
Files Typically Included in a Template	1-5
config-groups.xml and startup-plan.xml	1-9
Domain Topology Profiles	1-10
Application Service Groups, Server Groups, and Application Service Mappings	1-12
Add a Server To or Remove a Server From a Server Group	1-13
Startup Groups	1-13
Managing Server Startup Configuration	1-14

## 2 WebLogic Server Templates

---

Basic WebLogic Server Domain Template	2-2
Template Details	2-2
Resources and Services Configured in a Basic WebLogic Server Domain	2-3
WebLogic Server Starter Domain Template	2-5
Template Details	2-5
Resources and Services Configured in a WebLogic Server Starter Domain	2-5
WebLogic Advanced Web Services for JAX-RPC Extension Template	2-7
Template Details	2-7
Resources and Services Configured	2-7
WebLogic Advanced Web Services for JAX-WS Extension Template	2-9
Template Details	2-9
Resources and Services Configured	2-10

WebLogic JAX-WS SOAP/JMS Extension Template	2-11
Template Details	2-11
Resources and Services Configured	2-12
Avitek Medical Records Sample Domain Template	2-12
Template Details	2-13
Resources and Services Configured	2-13
Avitek Medical Records Sample Domain Template (Spring Version)	2-15
Template Details	2-16
Resources and Services Configured	2-16
WebLogic Server Default Domain Extension Template	2-18
Template Details	2-19
Resources and Services Configured	2-19
WebLogic Server Examples Extension Template	2-23
Template Details	2-23
Resources and Services Configured	2-24

# Preface

This guide provides information about the 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.

- [Documentation Accessibility](#)
- [Diversity and Inclusion](#)
- [Conventions](#)

## 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 that have purchased support 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.

## Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

## Conventions

The following text conventions are used in this document:

Convention	Meaning
<b>boldface</b>	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.

# 1

## Introduction

The Fusion Middleware domain and extension templates are Java Archive (JAR) files. These JAR files contain the files and scripts that are required to create or extend a WebLogic domain.

This chapter contains the following topics:

- [Types of Templates](#)  
Each template type performs a specific task during the installation of WebLogic Server. You can also create custom templates from existing templates.
- [Location of Installed WebLogic Server Templates](#)  
The predefined templates are located within the directory in which the WebLogic Server product is installed.
- [Template Tools](#)  
Template tools help you create templates as well as to use templates to create or extend a domain.
- [Template Dependencies](#)  
The WebLogic Server resources must be set up in your domain before you can add resources from an extension template. This dependency is known as a template dependency.
- [Files Typically Included in a Template](#)  
A domain is created or extended based on the basic files as well as the additional files that are included in the template.
- [config-groups.xml and startup-plan.xml](#)  
These files are present in many templates. The `config-groups.xml` file defines one or more of these items: domain topology profiles, application service groups, server groups, and application service mappings. The `startup-plan.xml` file defines server startup parameters at a global (domain-wide) level or a server group level.

## Types of Templates

Each template type performs a specific task during the installation of WebLogic Server. You can also create custom templates from existing templates.

The types of template include:

- **Domain template:** This 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 about the Basic WebLogic Server Domain template, see [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.

- **Extension template:** This 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 that you are installing. WebLogic Server installations include the templates described in [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:** This 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. See The Pack Command in *Creating Templates and Domains Using the Pack and Unpack Commands*.

- **Reconfiguration template:** If you upgrade a WebLogic domain from a previous WebLogic Server version, this template is automatically implemented. If a currently installed product, such as WebLogic Advanced Web Services for JAX-WS, requires updates to be compatible with the domain that 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 you run 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 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 by using the Fusion Middleware Configuration Wizard, such as SOA and Web Center. This feature enables you to update the domains for your Fusion Middleware products when you upgrade 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`.

## Location of Installed WebLogic Server Templates

The predefined templates are located within the directory in which the WebLogic Server product is installed.

For more information on location of predefined templates, see [Table 1-1](#).

The following table identifies the location of the predefined template JAR files provided with the WebLogic Server installation, where `ORACLE_HOME` is the installation directory and `WL_HOME` is the WebLogic Server directory in the product installation directory.

**Table 1-1 Location of Templates**

Type of Template	Directory Location
WebLogic Server Domain, Extension, and Reconfiguration	<code>WL_HOME\common\templates\wls</code>
Fusion Middleware Extension and Reconfiguration	<code>ORACLE_HOME\oracle_common\common\templates\wls</code> <code>ORACLE_HOME\PRODUCT_HOME\common\templates\wls</code>

## Template Tools

Template tools help you create templates as well as to use templates to create or extend a domain.

The following table provides the details of these tools:

**Table 1-2 Template Tools**

To	Use this tool
Create a domain	<ul style="list-style-type: none"> <li>• Configuration Wizard See Creating a WebLogic Domain in Graphical Mode in <i>Creating WebLogic Domains Using the Configuration Wizard</i>.</li> <li>• WLST Offline See Creating WebLogic Domains Using WLST Offline in <i>Understanding the WebLogic Scripting Tool</i>.</li> <li>• <code>unpack</code> command See The Unpack Command in <i>Creating Templates and Domains Using the Pack and Unpack Commands</i>.</li> </ul>
Extend an existing domain	<ul style="list-style-type: none"> <li>• Configuration Wizard See Updating a WebLogic Domain in Graphical Mode in <i>Creating WebLogic Domains Using the Configuration Wizard</i>.</li> <li>• WLST Offline See Creating WebLogic Domains Using WLST Offline in <i>Understanding the WebLogic Scripting Tool</i>.</li> </ul>
Create a Managed Server domain on a remote machine	<ul style="list-style-type: none"> <li>• <code>unpack</code> command See The UnPack Command in <i>Creating Templates and Domains Using the Pack and Unpack Commands</i>.</li> <li>• WLST <code>createDomain</code> command (offline) See WLST Command and Variable Reference in <i>WLST Command Reference for Oracle WebLogic Server</i>.</li> </ul>



Table 1-2 (Cont.) Template Tools

To	Use this tool
Create a domain template	<ul style="list-style-type: none"> <li>Domain Template Builder</li> <li>pack command See The Pack Command in <i>Creating Templates and Domains Using the Pack and Unpack Commands</i>.</li> <li>WLST Offline See Creating and Using a Domain Template (Offline) in <i>Understanding the WebLogic Scripting Tool</i>.</li> </ul>
Create an extension template	Domain Template Builder See Creating Extension Templates by Using the Domain Template Builder in <i>Creating Domain Templates Using the Domain Template Builder</i>
Create a Managed Server template	<ul style="list-style-type: none"> <li>WLST <code>writeTemplate</code> command (online)</li> <li>pack command See The Pack Command in <i>Creating Templates and Domains Using the Pack and Unpack Commands</i>.</li> </ul>
Upgrade a domain	Reconfiguration Wizard See Reconfiguring WebLogic Domains in <i>Upgrading Oracle WebLogic Server</i> .

**Note:**

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

## Template Dependencies

The WebLogic Server resources must be set up in your domain before you can add resources from an extension template. This dependency is known as a template dependency.

For example, all extension templates provided with your product depend on, at the 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 Templates screen of the Configuration Wizard, templates B, C, D, E, and F are automatically included in the domain. If any of these templates are displayed on the Templates screen, the check boxes for those templates appear automatically selected. This feature ensures that when you select a product template on the Templates screen, the Configuration Wizard automatically includes in the domain all other product templates that configure resources required by the product that you have selected.

Similarly, if you specify a template JAR in a WebLogic Scripting Tool (WLST) script, all other templates that are dependencies of that template (either directly or indirectly) are included in the domain. Based on the same example, if you specify Template A in a WLST script, templates B, C, D, E, and F are also included in the domain. You do not have to specify them explicitly in the script.

## Files Typically Included in a Template

A domain is created or extended based on the basic files as well as the additional files that are included in the template.

The basic files included in any template are `config.xml` and `template-info.xml`. 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. If the domain requires JMS resources, the *-jms.xml files must be located in the <code>config\jms</code> directory, in the template. There is one XML file for each JMS resource in the domain. These files are present only if the domain includes JMS resources.

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

Filename	Description
<code>clusters.script</code>	<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"><li>• <b>%AManagedServer%</b>—Any Managed Server</li><li>• <b>%AllManagedServers%</b>—Comma-separated list of all Managed Servers</li><li>• <b>%AdminServer%</b>—Administration Server name</li><li>• <b>%Cluster%</b>—Cluster name</li><li>• <b>%ProxyServer%</b>—Proxy server name</li><li>• <b>%HTTPProxyApp%</b>—http proxy application definition</li></ul> <p><b>Note:</b> Ensure to make the following additional considerations:</p> <ul style="list-style-type: none"><li>• You must use the name attribute of an object that is to be replaced.</li><li>• You can use an asterisk (*) as a wildcard for "All."</li></ul> <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> <p>If the template contains a <code>config-groups.xml</code> file, the <code>clusters.script</code> file, if present, is ignored.</p>
<code>config.xml</code>	<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>

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

Filename	Description
<code>config-groups.xml</code>	<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 feature ensures that all application and service dependencies are met when scaling a domain configuration.</p> <p><b>Note:</b> Do not modify this file in any way. It must be used as provided in the template.</p> <p>An Application or Service group or both 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"><li>• <b>Server group definitions</b>—Specifies a server or servers that can house functionally related sets of applications and services, thereby enabling automatic server creation.</li><li>• <b>Cluster group definitions</b>—Specifies a cluster that can house functionally related sets of applications and services, thereby enabling automatic cluster creation.</li><li>• <b>Application/Service group mapping definitions</b>—Specifies targeting of an Application or Service group or both to a specific server, group of servers, or cluster, via the name of the Application/Service group.</li></ul> <p>As of WebLogic Server 12.1.2, the domain topology section may define separate server groups, startup groups, and application or service group mapping definitions for compact and expanded domain profiles. For more information about domain profiles, see <a href="#">config-groups.xml</a> and <a href="#">startup-plan.xml</a>.</p>
<code>config-mapping.xml</code>	<p>This file is used to dynamically assign values to custom variables that are defined in a deployment plan, by using name or value pairs.</p>
<code>database.xml</code>	<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><b>Note:</b> Do not modify this file in any way. It must be used as provided in the template.</p>

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

Filename	Description
file-definition.xml	Applies only to Fusion Middleware product templates. It defines file copy and string substitution operations that are done during domain creation or extension.
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 is represented by a tilde ( ~ ) in the path name 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"> <li>• <i>Drop/Create P13N Database Objects</i> category associated with the p13nDataSource data source, which is a part of the p13n.jar domain template.</li> <li>• <i>Drop/Create Portal Database Objects</i> category associated with the p13nDataSource data source, which is a part of the wlp.jar domain template.</li> <li>• <i>Drop/Create GroupSpace Database Objects</i> category associated with the appsGroupSpaceDataSource data source, which is a part of the wlp_groupspacedb.jar domain template.</li> </ul> <p>All these template jar files are located in the <code>WL_HOME\common\templates\applications</code> directory.</p>
jvm-config.xml	This file is specific to FMW product installations in a WebSphere environment, and can be ignored in a WebLogic Server environment.
security.xml	Used to create user groups and roles that establish identity and access to domain resources. You can create the default Admin user only through the security.xml file in a domain template. However, you can create user groups and roles through the security.xml file that is included in either a domain or an extension template.
startscript.xml	Used to create the *.cmd and *.sh files that are placed into the root and bin directories of a domain.

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

Filename	Description
startup-plan.xml	<p>Defines the 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"> <li>• Environmental variables</li> <li>• Java system properties</li> <li>• Java protocol handlers</li> <li>• WLS PRE_CLASSPATH</li> <li>• WLS POST_CLASSPATH</li> <li>• Java library path</li> <li>• WLS JVM initial heap size</li> <li>• WLS JVM maximum heap size</li> <li>• WLS JVM perm size</li> <li>• WLS JVM maximum perm size</li> <li>• Other java arguments</li> </ul> <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 <a href="#">config-groups.xml</a> and <a href="#">startup-plan.xml</a>.</p>
stringsubs.xml	<p>Identifies string substitution values and files that receive string substitutions during domain creation or extension. The files that receive string substitutions must be prepared with replacement variables. During domain creation or extension, the Configuration Wizard framework runs macros to replace variables with the appropriate string substitution, by 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>

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

These files are present in many templates. The `config-groups.xml` file defines one or more of these items: domain topology profiles, application service groups, server groups, and application service mappings. The `startup-plan.xml` file defines server startup parameters at a global (domain-wide) level or a server group level.

When you create a domain by 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 the `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:

- [Domain Topology Profiles](#)
- [Application Service Groups, Server Groups, and Application Service Mappings](#)
- [Startup Groups](#)

## Domain Topology Profiles

A domain can have either an Expanded (clustered) or a Compact (single-instance server) domain topology. By default, domains are created as Expanded domains. When you run the Fusion Middleware Configuration Wizard from the command line, the wizard screen appears where you can select either Expanded or Compact as the topology for the domain. See [Setting the CONFIG\\_JVM\\_ARGS Environment Variable in Creating WebLogic Domains Using the Configuration Wizard](#).

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 also specify the domain topology profile if you use WLST to create the domain. See `createDomain` and `readTemplate` in *WLST Command Reference for WebLogic Server*.

Some graphical interfaces automatically configure the appropriate profile. For example, when you create a domain by using the JDeveloper domain creation utility, the domain is created with a Compact profile; when you create a domain by using the Fusion Middleware Configuration Wizard, the domain is created with an Expanded profile, by default.

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 Expanded. 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.
- The server group that the Administration Server belongs to. The Administration Server group is not user-expandable.
- Whether a server group is user-expandable, that is, whether 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 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 products 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 products 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 <code>config-groups.xml</code> , it is not used.
All application service groups defined in <code>config-groups.xml</code> are targeted. Some application service groups are targeted to the Administration Server server group, while other application service groups are targeted to the Managed Server server groups.	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 you create a domain by using the FMW Configuration Wizard, you select the profile to use on the Configuration Type screen. See Configuration Type in *Creating WebLogic Domains Using the Configuration Wizard*.



- When you create a domain by using WLST, you specify the profile to use in either the `createDomain` or `readTemplate` command. See `createDomain` or `readTemplate` in *WLST Command Reference for WebLogic Server*. The default profile type is Expanded.

## 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 groups** : Application service groups are defined in the `<app-svc-groups>` element. Each `<group>` element defines a unique application service group, which contain various applications and services that are 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** : 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, 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 run only on the Administration Server
- Application service group MAIN-APPS, which defines applications that 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-SRVS 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](#)

## 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 the use of this command.

```
# 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)
```

## 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 server startup groups 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.

You can define different startup settings for Expanded and Compact domain profiles. In addition, when you create 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 `/init-info` directory of the domain. When you start a server, this file is referenced to determine:

- The startup group of the server, if any, based on the startup group and the server group to which the server 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 by 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.

- [Managing Server Startup Configuration](#)

## 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 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. 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](#)

## Creating and Modifying a Startup Group

You can create a new startup group from an existing server group in the domain. The new startup group inherits *only* the startup parameters from the server group that 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 command displays only user-expandable server groups.
- Open the `init-info/config-groups.xml` file of the domain. Server group names are defined by the `name` attribute of each `<server group>` element in this file.

The following WLST example shows you 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:**

The `set('EnvVars',{})` command resets all customizations and reverts the environment variables for the startup group to the settings derived from the server groups that are associated with the startup group.

# 2

## WebLogic Server Templates

The WebLogic domain templates are provided with your WebLogic Server installation. You can create or extend domains by selecting these templates on the Templates screen of the Oracle Fusion Middleware Configuration Wizard or by using WLST.

**Table 2-1 Oracle WebLogic Server and Workshop for WebLogic Templates**

Template	Description
<a href="#">Basic WebLogic Server Domain Template</a>	Creates a base WebLogic Server domain.
<a href="#">WebLogic Server Starter Domain Template</a>	Creates a WebLogic Server starter domain.
<a href="#">WebLogic Advanced Web Services for JAX-RPC Extension Template</a>	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.
<a href="#">WebLogic Advanced Web Services for JAX-WS Extension Template</a>	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 by using WS-SecureConversation.
<a href="#">WebLogic JAX-WS SOAP/JMS Extension Template</a>	Extends a domain to include the resources required to use SOAP over JMS transport for JAX-WS Web services.
<a href="#">Avitek Medical Records Sample Domain Template</a>	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.
<a href="#">Avitek Medical Records Sample Domain Template (Spring Version)</a>	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.
<a href="#">WebLogic Server Default Domain Extension Template</a>	Extends the Basic WebLogic Server domain with a web application designed to guide new users through an introduction to WebLogic Server. When you run the web application, you can review informative content on various topics, including highlights of WebLogic Server functionality. From the web application, you can also run several preconfigured, precompiled examples. Resources from this extension template are required for a WebLogic Server Examples domain.
<a href="#">WebLogic Server Examples Extension Template</a>	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.

- [Basic WebLogic Server Domain Template](#)  
This template helps you configure resources such as Administration Server and Security realm at the time of creating a domain.
- [WebLogic Server Starter Domain Template](#)  
This template helps you configure resources and services such as Administration Server, Security realm, and Application Deployments at the time of creating a domain.
- [WebLogic Advanced Web Services for JAX-RPC Extension Template](#)  
When you use the Configuration Wizard or WLST, this template helps you include the resources required for advanced JAX-RPC Web services.
- [WebLogic Advanced Web Services for JAX-WS Extension Template](#)  
This template automatically configures the resources required to support the advanced Web services features.
- [WebLogic JAX-WS SOAP/JMS Extension Template](#)  
When you use the Configuration Wizard or WLST, this template helps you include the resources required to use SOAP over JMS transport for JAX-WS Web services.
- [Avitek Medical Records Sample Domain Template](#)  
When you use the Configuration Wizard or WLST, this template helps you to extend a base WebLogic Server domain to create an Avitek Medical Records Sample domain.
- [Avitek Medical Records Sample Domain Template \(Spring Version\)](#)  
When you use the Configuration Wizard or WLST, this template helps you easily extend a base WebLogic Server domain to create an Avitek Medical Records Sample domain in the Spring version.
- [WebLogic Server Default Domain Extension Template](#)  
When you use the Configuration Wizard or WLST, this template helps you extend a base WebLogic Server domain to include resources required for a default WebLogic Server domain.
- [WebLogic Server Examples Extension Template](#)  
When you use the Configuration Wizard or WLST, this template helps you easily extend a base WebLogic Server domain to create a WebLogic Server Examples domain.

## Basic WebLogic Server Domain Template

This template helps you configure resources such as Administration Server and Security realm at the time of creating a domain.

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.

- [Template Details](#)
- [Resources and Services Configured in a Basic WebLogic Server Domain](#)

### Template Details

The following table provides basic information about the Basic WebLogic Server Domain template. The template name is the name of the template as shown in the

product list in the *Select Domain Source* and *Select Extension Source* screens of the Configuration Wizard.

**Table 2-2 Basic WebLogic Server Domain Template Information**

Template Detail	Information
Template type	Domain
Category	Oracle WebLogic Server and Coherence
Template name	Basic WebLogic Server Domain
Template JAR file and location	<code>WL_HOME/common/templates/wls/wls.jar</code>
Template dependencies	None

## 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"> <li>• Listen address: All Local Addresses</li> <li>• Listen port: 7001</li> <li>• SSL is disabled</li> </ul> <p>When you use the Configuration Wizard or WLST Offline to create a domain, and if you want the Administration Server name to be different from the default name, <code>AdminServer</code>, you must configure the name manually. You cannot change the name when you apply an extension template.</p> <p>For information about customizing the Administration Server name while creating a domain with the Configuration Wizard, see <i>Creating a WebLogic Domain in Creating WebLogic Domains Using the Configuration Wizard</i>.</p> <p>For information about customizing the Administration Server name while creating a domain with WLST Offline, see <i>Creating WebLogic Domains Using WLST Offline in Understanding the WebLogic Scripting Tool</i>.</p> <p>The following sample WLST Offline code snippet shows you how to change the default Administration Server name, <code>AdminServer</code>, to <code>MedRecServer</code>.</p> <pre>#-----# -----#  Read the Basic WebLogic Server Domain template readTemplate('d:/MW_HOME/ wlsrserver/common/ templates/wls/wls.jar') #Change the Administration Server name. cd('Servers/AdminServer') set('Name', 'MedRecServer') #-----# -----#</pre>



Table 2-3 (Cont.) Resources Configured in a Basic WebLogic Server Domain

Resource Type	Name	Notes
Security realm	myrealm	This realm is the default (active) WebLogic Server security realm. The administration user account, <code>weblogic</code> , is configured in this security realm, as well as in all the default groups and roles.

## WebLogic Server Starter Domain Template

This template helps you configure resources and services such as Administration Server, Security realm, and Application Deployments at the time of creating a domain.

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.

- [Template Details](#)
- [Resources and Services Configured in a WebLogic Server Starter Domain](#)

### 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 the Configuration Wizard	This template is not available from the list of products in the Configuration Wizard. You can select it only by using the <b>Browse</b> 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

### 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"> <li>• Listen address: All Local Addresses</li> <li>• Listen port: 7001</li> <li>• SSL is disabled</li> </ul> <p>When you use the Configuration Wizard or WLST Offline to create a domain, and if you want the Administration Server name to be different from the default name, <code>AdminServer</code>, you must configure the name manually. You cannot change the name later when you apply 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 WebLogic Domains Using the Configuration Wizard</i>.</p> <p>For information about customizing the Administration Server name while creating a domain with WLST Offline, see <i>Creating WebLogic Domains Using WLST Offline in Understanding the WebLogic Scripting Tool</i>.</p> <p>The following sample WLST Offline code snippet shows how to change the default Administration Server name, <code>AdminServer</code>, to <code>MedRecServer</code>.</p> <pre>#-----# -----#  Read the Basic WebLogic Server Domain template readTemplate('d:/MW_HOME/ wlsrserver/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	wl_starter	A sample Web application deployed to the starter domain.

## WebLogic Advanced Web Services for JAX-RPC Extension Template

When you use the Configuration Wizard or WLST, this template helps you include the resources required for advanced JAX-RPC Web services.

You accomplish this task by adding the resources and services provided in the WebLogic Advanced Web Services for JAX-RPC extension template to a base WebLogic Server domain.

- [Template Details](#)
- [Resources and Services Configured](#)

### 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	Oracle WebLogic Server and Coherence
Template name and version	WebLogic Advanced Web Services for JAX-RPC Extension - 12.2.1.4.0
Template dependencies	Base WebLogic Server domain template
Template JAR file and location	<i>ORACLE_HOME</i> /oracle_common/common/templates/wls/oracle.wls-webservice-template.jar

### 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 <code>AdminServer</code>, unless changed during domain creation. The Administration Server referenced in this extension template is <code>cgServer</code>. The default configuration for the Administration Server is as follows:</p> <ul style="list-style-type: none"> <li>• Listen address: All Local Addresses</li> <li>• Listen port: 7001</li> <li>• SSL is enabled</li> <li>• SSL listen port: 7002</li> </ul> <p>For information about naming the Administration Server during domain creation, see <a href="#">Resources and Services Configured in a Basic WebLogic Server Domain</a>.</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 <code>WseeJmsServer</code> JMS server. This file store is targeted to the Administration Server.
SAF agent	ReliableWseeSAFAgent	Adds this store-and-forward agent, which uses the <code>WseeFileStore</code> , and targets it to the Administration Server. The SAF agent controls receipt and handling of reliable messages.
JMS queues	WseeMessageQueue	Adds the JMS queue to the JMS server and <code>WseeJmsServer</code> . The queues are located in <code>JMSModules/module name</code> , which are located in <code>WseeJMSModule</code> .
JMS queues	WseeCallbackQueue	Adds the JMS queue to the JMS server <code>WseeJmsServer</code> .
JMS Server	WseeJmsServer	Adds the JMS server as a system resource and targets it to the Administration Server <code>AdminServer</code> .
Work Manager	weblogic.wsee.mdb.DispatchPolicy	Add this Work Manager and targets it to the Administration Server <code>AdminServer</code> .

# WebLogic Advanced Web Services for JAX-WS Extension Template

This template automatically configures the resources required to support the advanced Web services features.

The following advanced Web services features are supported:

- 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 you apply 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*.

- [Template Details](#)
- [Resources and Services Configured](#)

## 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	Oracle WebLogic Server and Coherence
Template name and version	WebLogic Advanced Web Services for JAX-WS Extension - 12.2.1.4.0
Template dependencies	<ul style="list-style-type: none"><li>• Base WebLogic Server domain template</li><li>• Web Services State Management Memory Provider template</li></ul>
Template JAR file and location	<code>ORACLE_HOME/oracle_common/common/templates/wls/oracle.wls-webservice-jaxws-template.jar</code>

## 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 <code>AdminServer</code>, unless changed during domain creation. The Administration Server referenced in this extension template is <code>cgServer</code>.</p> <p>The default configuration for the Administration Server is as follows:</p> <ul style="list-style-type: none"> <li>• Listen address: All Local Addresses</li> <li>• Listen port: 7001</li> <li>• SSL is enabled</li> <li>• SSL listen port: 7002</li> </ul> <p>For information about naming the Administration Server during domain creation, see <a href="#">Resources and Services Configured in a Basic WebLogic Server Domain</a>.</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.
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.
File store	WseeJaxwsFileStore	Adds the file store to be used as the persistent store for the <code>WseeJaxwsJmsServer</code> JMS server. This file store is targeted to the Administration Server.

**Table 2-9 (Cont.) Resources Configured in a WebLogic Advanced Web Services for JAX-WS Domain**

Resource Type	Name	Extension Result
JMS system resource	WseeJaxwsJmsModule	Defines a distributed destination for the cluster. All associated targets are used to support JAX-WS Web services. The subdeployment name is <code>WseeJaxwsJmsServerSub</code> .  <b>Note:</b> 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).
SAF Agent	ReliableWseeJaxwsSAFAgent	Adds this store-and-forward agent, which uses the <code>WseeJaxwsFileStore</code> , and targets it to the Administration Server. The SAF agent controls receipt and handling of reliable messages.
JMS queues	<code>weblogic.wsee.BufferedRequestQueue</code> <code>weblogic.wsee.BufferedRequestErrorQueue</code> <code>weblogic.wsee.BufferedResponseQueue</code> <code>weblogic.wsee.BufferedResponseErrorQueue</code>	Adds these JMS queues to the JMS server, and targets them to <code>WseeJaxwsJmsServer</code> .  These queues are reserved for future use.

## WebLogic JAX-WS SOAP/JMS Extension Template

When you use the Configuration Wizard or WLST, this template helps you include the resources required to use SOAP over JMS transport for JAX-WS Web services.

You accomplish this task by adding the resources and services provided in the WebLogic Advanced Web Services extension template to a base WebLogic Server domain.

- [Template Details](#)
- [Resources and Services Configured](#)

### 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	Oracle WebLogic Server and Coherence
Template name and version	WebLogic JAX-WS SOAP/JMS Extension - 12.2.1.4.0
Template dependencies	Base WebLogic Server domain template
Template JAR file and location	<code>ORACLE_HOME/oracle_common/common/templates/wls/oracle.wls-webservice-soapjms-template.jar</code>

## 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.api.jms.ConnectionFactory</code>	Default JMS connection factory used to create connections for SOAP over JMS transport.
JMS queues	<code>com.oracle.webservices.api.jms.RequestQueue</code>	Default JMS request queue.
JMS queue	<code>com.oracle.webservices.api.jms.ResponseQueue</code>	Default JMS response queue.
JMS queue	<code>com.oracle.webservices.api.jms.ErrorQueue</code>	Default JMS error queue.

## Avitek Medical Records Sample Domain Template

When you use the Configuration Wizard or WLST, this template helps you to extend a base WebLogic Server domain to create an Avitek Medical Records Sample domain.



You accomplish this task 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*.

- [Template Details](#)
- [Resources and Services Configured](#)

## 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 the Configuration Wizard	This template is not available from the list of products in the Configuration Wizard. You can select it only by using the <b>Browse</b> 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.jar</code>
Template dependencies	Basic WebLogic Server Domain template

## Resources and Services Configured

The following table identifies the resources and services configured in a domain that is 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 <code>AdminServer</code>, unless changed during domain creation. The Administration Server is referenced in the template as <code>@SERVER_NAME</code>, and appears as <code>AdminServer</code> in the WebLogic Server Administration Console.</p> <p>The default configuration for the Administration Server is as follows:</p> <ul style="list-style-type: none"> <li>• Listen address: All Local Addresses</li> <li>• Listen port: 7001</li> <li>• SSL is enabled</li> <li>• SSL port: 7002</li> </ul> <p>For information about naming the Administration Server during domain creation, see <a href="#">Resources and Services Configured in a Basic WebLogic Server Domain</a>.</p>
Security realm	myrealm	Uses the default security realm provided in the Basic WebLogic Server domain.
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	chat	Adds the chat Web application, which is used for conversation between the patient and physician, and targets it to the Administration Server.
Application deployments	browser-starter	Adds the browser-starter Web application and targets it to the Administration Server.
Application deployment	SamplesSearchWebApp	Adds a sample search application and targets it 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
JDBC Data Source	MedRecGlobalDataSourceXA	Identifies the JDBC data source as a MedRecGlobalDataSourceXA system resource.
JDBC Data Source	BatchDS	Defines a JDBC data source called BatchDS. This data source is a two-phase commit XA data source with JNDI name <code>jndi/batchDS</code> .
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.
JMS Server	MedRecJMSServer	Adds the JMS server as a MedRec-jms system resource and targets it to the Administration Server
Work Manager	weblogic.wsee.mdb.DispatchPolicy	Adds this Work Manager and targets it to the Administration Server. The Work Manager defines the thread pool resources.
WLDF System Resource	MedRecWLDF	Adds the WLDF system resource, WLDF harvester for NewUserCount and WLDR instrumentation monitors for Dye Injection, and targets them to the Administration Server.
Connection Factory	MedRecConnectionFactory	Default JMS connection factory used to create connections over JMS transport.
JMS Queues	com.oracle.medrec.jms.PatientNotificationQueue	Adds the JMS queue to the JMS server, MedRecWseeJMSServer.

## Avitek Medical Records Sample Domain Template (Spring Version)

When you use the Configuration Wizard or WLST, this template helps you easily extend a base WebLogic Server domain to create an Avitek Medical Records Sample domain in the Spring version.

You accomplish this task 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*.

- [Template Details](#)
- [Resources and Services Configured](#)

## Template Details

The following table provides basic information about the WebLogic Advanced Web Services Extension template. The template name is the name of the template as 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 <b>Browse</b> 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

## 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 <code>AdminServer</code>, unless changed during domain creation. The Administration Server is referenced in the template as <code>@SERVER_NAME</code>, and appears as <code>AdminServer</code> in the WebLogic Server Administration Console.</p> <p>The default configuration for the Administration Server is as follows:</p> <ul style="list-style-type: none"> <li>• Listen address: All Local Addresses</li> <li>• Listen port: 7001</li> <li>• SSL is enabled</li> <li>• SSL port: 7002</li> </ul> <p>For information about naming the Administration Server during domain creation, see <a href="#">Resources and Services Configured in a Basic WebLogic Server Domain</a>.</p>
Security realm	myrealm	Uses the security realm provided in the base WebLogic Server domain.
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 deployments	browser-starter	Adds the browser-starter Web application and targets it to the MedRecServer.
Application deployment	SamplesSearchWebApp	Adds a sample search application and targets it to the Administration Server.
Deployed library	jsf#1.2@1.2.9.0	Adds the Java Server Faces Version 1.2 library and targets it to the MedRecServer.
Deployed library	jstl#1.2.@1.2.0.1	Adds the Java standard tagging (JSTL) Version 1.2 library and targets it to the MedRecServer.
Deployed library	weblogic-spring#12.2.1.0.0@12.2.1.0.0	Adds the WebLogic Spring Version 12.2.1 library and targets it to the MedRecServer.
Mail session	mail/MedRecMailSession	Adds this mail session.

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

Resource Type	Name	Extension Result
JDBC data sources	MedRecGlobalDataSourceXA	Identifies the JDBC data source as a MedRecGlobalDataSourceXA system resource. <ul style="list-style-type: none"> <li>Pool capacity (initial): 2</li> <li>Pool capacity (maximum): 10</li> <li>Protocol: Two Phase Commit</li> </ul>
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.
JMS Servers	MedRecJMSServer	Adds the JMS server as a MedRec-jms system resource and targets it to the MedRecServer.
Work Manager	weblogic.wsee.mdb.DispatchPolicy	Adds this Work Manager and targets it to the Administration Server. The Work Manager defines the thread pool resources.
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).
SAF Agent	WsrnAgent	Adds this store-and-forward agent, which uses the file store, MedRecWseeFileStore, and targets it to the MedRecServer.
JMS queues	com.oracle.medrec.jms.RecordToCreateQueue com.oracle.medrec.jms.PatientNotificationQueue weblogic.wsee.DefaultQueue	Adds these JMS queues to the JMS server, MedRecWseeJMSServer.
WLDF system resource	MedRecWLDF	Adds the WLDF system resource, WLDF harvester for NewUserCoun' and WLDR instrumentation monitors for Dye Injection, and targets them to the Administration Server.

## WebLogic Server Default Domain Extension Template

When you use the Configuration Wizard or WLST, this template helps you extend a base WebLogic Server domain to include resources required for a default WebLogic Server domain.

You accomplish this task by adding the resources and services provided in the WebLogic Server Default Domain extension template 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*.

- [Template Details](#)
- [Resources and Services Configured](#)

## 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 the Configuration Wizard	This template is not available from the list of products in the Configuration Wizard. You can select it only by using the <b>Browse</b> 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_default.jar</code>
Template dependencies	Basic WebLogic Server Domain template

## 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 <code>AdminServer</code>, unless changed during domain creation. The Administration Server is referenced in the template as <code>@SERVER_NAME</code>, and appears as <code>AdminServer</code> in the WebLogic Server Administration Console.</p> <p>The default configuration for the Administration Server is as follows:</p> <ul style="list-style-type: none"> <li>• Listen address: All Local Addresses</li> <li>• Listen port: 7001</li> <li>• SSL is disabled</li> </ul> <p>For information about naming the Administration Server during domain creation, see <a href="#">Resources and Services Configured in a Basic WebLogic Server Domain</a>.</p>
Security realm	myrealm	Uses the security realm provided by the base WebLogic Server domain.
Application deployment	SamplesSearchWebApp	Adds the <code>SamplesSearchWebApp</code> application and targets it to the <code>examplesServer</code> Administration Server.
Application deployment	mainWebApp	Adds the <code>mainWebApp</code> application and targets it to the <code>examplesServer</code> Administration Server.
Application deployment	examplesWebApp	Adds the <code>examplesWebApp</code> application and targets it to the <code>examplesServer</code> Administration Server.
Application deployment	entityBeanValidation	Adds the <code>entityBeanValidation</code> application and targets it to the <code>examplesServer</code> Administration Server.
Application deployment	jsfBeanValidation	Adds the <code>jsfBeanValidation</code> application and targets it to the <code>examplesServer</code> Administration Server.
Application deployment	cdi	Adds the <code>cdi</code> application and targets it to the <code>examplesServer</code> Administration Server.
Application deployment	asyncMethodOfEJB	Adds the <code>asyncMethodOfEJB</code> Enterprise application and targets it to the <code>examplesServer</code> Administration Server.



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

Resource Type	Name	Extension Result
Application deployment	calendarStyledTimer	Adds the <code>calendarStyledTimer</code> application and targets it to the <code>examplesServer Administration Server</code> .
Application deployment	noInterfaceViewInWAR	Adds the <code>noInterfaceViewInWAR</code> application and targets it to the <code>examplesServer Administration Server</code> .
Application deployment	portableGlobalJNDIName	Adds the <code>portableGlobalJNDIName</code> application and targets it to the <code>examplesServer Administration Server</code> .
Application deployment	singletonBean	Adds the <code>singletonBean</code> application and targets it to the <code>examplesServer Administration Server</code> .
Application deployment	jaxrs	Adds the <code>jaxrs</code> application and targets it to the <code>examplesServer Administration Server</code> .
Application deployment	stockAdapter	Adds the <code>stockAdapter</code> application and targets it to the <code>examplesServer Administration Server</code> .
Application deployment	stockFrontEnd	Adds the <code>stockFrontEnd</code> application and targets it to the <code>examplesServer Administration Server</code> .
Application deployment	stockBackEnd	Adds the <code>stockBackEnd</code> application and targets it to the <code>examplesServer Administration Server</code> .
Application deployment	jdbcDataSource.war	Adds the <code>jdbcDataSource.war</code> application and targets it to the <code>examplesServer Administration Server</code> .
Application deployment	criteriaQuery	Adds the <code>criteriaQuery</code> application and targets it to the <code>examplesServer Administration Server</code> .
Application deployment	elementCollection	Adds the <code>elementCollection</code> application and targets it to the <code>examplesServer Administration Server</code> .
Application deployment	ajaxJSF	Adds the <code>ajaxJSF</code> application and targets it to the <code>examplesServer Administration Server</code> .

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

Resource Type	Name	Extension Result
Application deployment	bookmarkingJSF	Adds the <code>bookmarkingJSF</code> application and targets it to the <code>examplesServer Administration Server</code> .
Application deployment	faceletsJSF	Adds the <code>faceletsJSF</code> application and targets it to the <code>examplesServer Administration Server</code> .
Application deployment	annotation	Adds the <code>annotation</code> application and targets it to the <code>examplesServer Administration Server</code> .
Application deployment	asyncServlet30	Adds the <code>asyncServlet30</code> application and targets it to the <code>examplesServer Administration Server</code> .
Application deployment	programmaticSecurity	Adds the <code>programmaticSecurity</code> application and targets it to the <code>examplesServer Administration Server</code> .
Application deployment	multipartFileHandling	Adds the <code>multipartFileHandling</code> application and targets it to the <code>examplesServer Administration Server</code> .
Application deployment	webFragment	Adds the <code>webFragment</code> application and targets it to the <code>examplesServer Administration Server</code> .
Application deployment	sample-echo	Adds the <code>sample-echo</code> application and targets it to the <code>examplesServer Administration Server</code> .
Application deployment	sample-cdi	Adds the <code>sample-cdi</code> application and targets it to the <code>examplesServer Administration Server</code> .
Application deployment	sample-auction	Adds the <code>sample-auction</code> application and targets it to the <code>examplesServer Administration Server</code> .
Application deployment	message-board	Adds the <code>message-board</code> application and targets it to the <code>examplesServer Administration Server</code> .
Application deployment	jaxrs-async	Adds the <code>jaxrs-async</code> application and targets it to the <code>examplesServer Administration Server</code> .
Application deployment	jaxrs-provider	Adds the <code>jaxrs-provider</code> application and targets it to the <code>examplesServer Administration Server</code> .

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

Resource Type	Name	Extension Result
Application deployment	jaxrs-sse	Adds the <code>jaxrs-sse</code> application and targets it to the <code>examplesServer</code> Administration Server.
JDBC system resource	examples-demo	Identifies this JDBC data source, which has the following configuration: <ul style="list-style-type: none"> <li>JNDI name: <code>examples-datasource-demoPool</code></li> <li>Global transaction protocol: Two-Phase Commit</li> </ul> The connection pool settings are: <ul style="list-style-type: none"> <li>Initial capacity: 1</li> <li>Maximum capacity: 10</li> </ul>
JDBC system resource	examples-demoXA	Identifies this JDBC data source, which has the following configuration: <ul style="list-style-type: none"> <li>JNDI Name: <code>examples-datasource-demoXAPool</code></li> <li>Global transaction protocol: Two-Phase Commit</li> </ul> The connection pool settings are: <ul style="list-style-type: none"> <li>Initial capacity: 2</li> <li>Maximum capacity: 10</li> </ul>
Deployed library	jax-rs	Adds the <code>jax-rs</code> library dependency to this domain.

## WebLogic Server Examples Extension Template

When you use the Configuration Wizard or WLST, this template helps you easily extend a base WebLogic Server domain to create a WebLogic Server Examples domain.

You accomplish this task by adding the resources and services provided in WebLogic Server Examples extension template 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*.

- [Template Details](#)
- [Resources and Services Configured](#)

### 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 the Configuration Wizard	This template is not available from the list of products in the Configuration Wizard. You can select it only by using the <b>Browse</b> 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	Basic WebLogic Server Domain template

## 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 WebLogic Server Administration Console.</p> <p>The default configuration for the Administration Server is as follows:</p> <ul style="list-style-type: none"> <li>Listen address: All Local Addresses</li> <li>Listen port: 7001</li> <li>SSL is disabled</li> </ul> <p>For information about naming the Administration Server during domain creation, see <a href="#">Resources and Services Configured in a Basic WebLogic Server Domain</a>.</p>
Security realm	myrealm	Uses the security realm provided by the base WebLogic Server domain.
Application deployment	SamplesSearchWebApp	Adds this application and targets it to the Administration Server.
Application deployment	mainWebApp	Adds this application and targets it to the Administration Server.

**Table 2-19 (Cont.) Additional Resources Configured by the WebLogic Server Examples Domain**

Resource Type	Name	Notes
Application deployment	examplesWebApp	Adds this application and targets it to the Administration Server.
Application deployment	sample-echo	Adds this application and targets it to the Administration Server.
Application deployment	sample-cdi	Adds this application and targets it to the Administration Server.
Application deployment	sample-auction	Adds this application and targets it to the Administration Server.
Application deployment	message-board	Adds this application and targets it to the Administration Server.
Application deployment	jaxrs-async	Adds this application and targets it to the Administration Server.
Application Deployment	jaxrs-provider	Adds this application and targets it to the Administration Server.
Application deployment	jaxrs-sse	Adds this application and targets it to the Administration Server.
Application deployment	batch10-joboperator	Adds this application and targets it to the Administration Server.
Application deployment	batch10-partition	Adds this application and targets it to the Administration Server.
Application deployment	cdi11-event	Adds this application and targets it to the Administration Server.
Application deployment	cdi11-transactional	Adds this application and targets it to the Administration Server.
Application deployment	cdi11-transaction-scoped	Adds this application and targets it to the Administration Server.
Application deployment	concurrency10-executor	Adds this application and targets it to the Administration Server.
Application deployment	concurrency10-schedule	Adds this application and targets it to the Administration Server.
Application deployment	concurrency10-threads	Adds this application and targets it to the Administration Server.
Application deployment	concurrency10-dynamicproxy	Adds this application and targets it to the Administration Server.
Application deployment	ejb32-lifecycle	Adds this application and targets it to the Administration Server.
Application deployment	ejb32-connector	Adds this application and targets it to the Administration Server.
Application deployment	ejb32-mdb	Adds this application and targets it to the Administration Server.
Application deployment	mailconnector	Adds this application and targets it to the Administration Server.
Application deployment	jsf22-contracts	Adds this application and targets it to the Administration Server.

**Table 2-19 (Cont.) Additional Resources Configured by the WebLogic Server Examples Domain**

Resource Type	Name	Notes
Application deployment	jsf22-fileupload	Adds this application and targets it to the Administration Server.
Application deployment	jsf22-flows	Adds this application and targets it to the Administration Server.
Application deployment	jsf22-html5	Adds this application and targets it to the Administration Server.
Application deployment	jsonp-jaxrs	Adds this application and targets it to the Administration Server.
Application deployment	servlet31-httpupgrade	Adds this application and targets it to the Administration Server.
Application deployment	servlet31-nonblockread	Adds this application and targets it to the Administration Server.
Application deployment	servlet31-nonblockwrite	Adds this application and targets it to the Administration Server.
Application deployment	servlet31-sessionidchange	Adds this application and targets it to the Administration Server.
Application deployment	servlet31-uncoveredhttpmethod	Adds this application and targets it to the Administration Server.
Application deployment	beanvalidation11-cdi	Adds this application and targets it to the Administration Server.
Application deployment	json-ejb	Adds this application and targets it to the Administration Server.
Application deployment	json-servlet	Adds this application and targets it to the Administration Server.
JDBC system resource	examples-demo	Identifies this JDBC data source, which has the following configuration: <ul style="list-style-type: none"> <li>JNDI name: examples-datasource-demoPool</li> <li>Global transaction protocol: Two-Phase Commit</li> </ul> The database driver is configured as <code>org.apache.derby.jdbc.ClientDriver</code> .
JDBC system resource	examples-demoXA	Identifies this JDBC data source, which has the following configuration: <ul style="list-style-type: none"> <li>JNDI name: examples-datasource-demoXAPool</li> <li>Global transaction protocol: Two-Phase Commit</li> </ul> The database driver is configured as <code>org.apache.derby.jdbc.ClientXADataSource</code> .

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

Resource Type	Name	Notes
JMS Server	examplesJMSServer	<p>Adds this JMS server as an <code>examples-jms</code> system resource and targets it to the Administration Server, as listed below:</p> <ul style="list-style-type: none"> <li>Persistent Store: <code>exampleJDBCStore</code></li> <li>JMS Message log file: <code>examplesJMSServer</code></li> </ul>
JDBC store	exampleJDBCStore	<p>Adds the JDBC store to be used as the persistent store for the <code>examples-demo</code> JDBC data source and the <code>examplesJMSServer</code> JMS server, and targets the store to the <code>examplesServer</code> Administration Server.</p>
JMS system resources	examples-jms	<p>Identifies the JMS servers, connection factories, queues, and topics to be used for JMS system resources.</p>
JDBC system resource	examples-demoXA-2	<p>Identifies this JDBC data source, which is targeted to the Administration Server and has the following configuration:</p> <ul style="list-style-type: none"> <li>Connection pool maximum: 100</li> <li>Global transaction protocol: Two-Phase Commit</li> </ul> <p>This data source is mapped to the <code>examples-multiDataSource-demoXAPool</code> multi-data source.</p>
JDBC system resource	examples-multiDataSource-demoXAPool	<p>Identifies this JDBC multi-data source, which is targeted to the Administration Server. It is configured for failover, and maps to the <code>examples-oracleXA</code> and <code>examples-demo-XA-2</code> data sources.</p>
File store	WseeFileStore	<p>Adds the file store to be used as the persistent store for the <code>WSEEJMSServer</code> JMS server and the <code>ReliableWseeSAFAgent</code> SAF Agent. This file store is targeted to the <code>examplesServer</code> Administration Server.</p>
JMS Server	WseeJMSServer	<p>Adds this JMS server as an <code>examples-jms</code> system resource and targets it to the Administration Server.</p> <p>This server is configured to use the <code>WSeeFileStore</code> persistent store.</p>

**Table 2-19 (Cont.) Additional Resources Configured by the WebLogic Server Examples Domain**

Resource Type	Name	Notes
SAF agent	ReliableWseeSAFAgent	Adds this store-and-forward agent, which uses the <code>WseeFileStore</code> file store, and targets it to the Administration Server.
Work Manager	<code>weblogic.wsee.mdb.DispatchPolicy</code>	Adds this Work Manager, but does not target it to any servers.
JMS Connection Factories	<code>exampleTopic</code> <code>exampleTrader</code> <code>weblogic.examples.jms.QueueConnectionFactory</code> <code>weblogic.examples.ejb30.QueueConnectionFactory</code>	Adds these connection factories as <code>examples-jms</code> system resources and targets them to the <code>examplesServer</code> server.
JMS queues	<code>exampleQueue</code> <code>jms/MULTIDATASOURCE_MDB_QUEUE</code> <code>weblogic.examples.ejb30.ExampleQueue</code>	Adds these JMS queues to the <code>examplesJMSServer</code> JMS server.
JMS queue	<code>weblogic.wsee.wseeExamplesDestinationQueue</code> <code>WseeBufferedRequestQueue</code> <code>WseeBufferedResponseQueue</code>	Adds these JMS queues to the <code>WseeJMSServer</code> JMS server.
JMS topics	<code>exampleTopic</code> <code>quotes</code> <code>stockTopic</code>	Adds these JMS topics and targets them to the <code>examplesJMSServer</code> JMS server.