

Oracle® Fusion Middleware

Understanding Oracle Fusion Middleware Concepts

12c (12.1.2)

E28570-02

September 2013

An overview of Oracle Fusion Middleware architecture and its key concepts. It also includes an introduction to Oracle Fusion Middleware components and tools.

Oracle Fusion Middleware Understanding Oracle Fusion Middleware Concepts, 12c (12.1.2)

E28570-02

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

Primary Authors: Helen Grembowicz, Lisa Jamen

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

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

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

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

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

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

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

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

Contents

Preface	v
Audience	v
Documentation Accessibility	v
Related Documents	v
Conventions	v
What's New in This Guide	vii
New and Changed Features for 12c (12.1.2)	vii
New and Deprecated Terminology for 12c (12.1.2)	vii
1 Overview of Oracle Fusion Middleware	
1.1 About Oracle Fusion Middleware	1-1
1.2 Understanding the Functions of Middleware	1-2
1.3 Understanding Middleware Architecture Design	1-2
1.4 Understanding the Oracle Fusion Middleware Solution	1-4
2 Understanding Key Concepts	
2.1 What is an Oracle WebLogic Server Domain?	2-1
2.1.1 What Is the Administration Server?	2-2
2.1.2 Understanding Managed Servers and Managed Server Clusters	2-3
2.1.3 What Is a Java Component?	2-3
2.1.4 What Is a System Component?	2-4
2.1.5 What Is Node Manager?	2-4
2.2 What Is a Standalone Domain?	2-4
2.3 What are the Key Oracle Fusion Middleware Directories?	2-5
2.4 What is the WebLogic Management Framework?	2-9
2.5 What Is the Metadata Repository?	2-10
3 Overview of Oracle Fusion Middleware Components	
3.1 Understanding Oracle WebLogic Server	3-1
3.2 Understanding Oracle Web Services	3-2
3.3 Understanding Oracle Coherence	3-2
3.4 Understanding Oracle HTTP Server	3-2
3.5 Understanding Oracle User Messaging Service	3-2

3.6	Understanding Oracle Platform Security Services	3-3
3.7	Understanding Oracle Data Integrator	3-3

4 Using Oracle Fusion Middleware Tools

4.1	Understanding the Installation and Configuration Tools	4-1
4.1.1	Oracle Universal Installer	4-1
4.1.2	Repository Configuration Utility	4-1
4.1.3	OPatch	4-2
4.1.4	Configuration Wizard	4-2
4.1.5	Reconfiguration Wizard	4-2
4.1.6	Upgrade Assistant	4-2
4.2	Understanding the Oracle Fusion Middleware Administration Tools	4-2
4.2.1	Oracle Enterprise Manager Fusion Middleware Control	4-3
4.2.2	Oracle WebLogic Server Administration Console	4-3
4.2.3	Oracle WebLogic Scripting Tool (WLST)	4-4
4.2.4	Oracle Enterprise Manager Cloud Control	4-4
4.3	Understanding the Development Tools	4-5
4.3.1	Oracle JDeveloper	4-5
4.3.2	Oracle Application Development Framework	4-5
4.3.3	Oracle TopLink	4-6
4.3.4	Oracle Help	4-6

Preface

This preface includes the following topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

Audience

This guide is intended for anyone interested in an overview of the key concepts and architecture of Oracle Fusion Middleware.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For more information, see the Oracle Fusion Middleware Documentation Library 12c (12.1.2), especially the following:

- *Understanding Oracle WebLogic Server*
- *Administering Oracle Fusion Middleware*

Conventions

The following text conventions are used in this guide:

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

What's New in This Guide

For existing Oracle Fusion Middleware 11g users, this preface introduces the new and changed concepts and features of Oracle Fusion Middleware 12c.

- [New and Changed Features for 12c \(12.1.2\)](#)
- [New and Deprecated Terminology for 12c \(12.1.2\)](#)

New and Changed Features for 12c (12.1.2)

Oracle Fusion Middleware 12c (12.1.2) includes the following new and changed concepts and features from previous Oracle Fusion Middleware releases:

- Middleware topology changes. See [Section 1.3](#).
- OPMN is no longer used in Oracle Fusion Middleware. Instead, system components are managed by the WebLogic Management Framework. See [Section 2.4](#).
- Redefining of the Oracle home and elimination of the Middleware home. See [New and Deprecated Terminology for 12c \(12.1.2\)](#).
- Support for a "per domain" Node Manager. See [Section 2.1.5](#).
- Oracle Web Cache is no longer part of Oracle Fusion Middleware.

New and Deprecated Terminology for 12c (12.1.2)

Terminology Used in 11g	Terminology Used in 12c
Middleware home	Oracle home
<i>MW_HOME</i>	<i>ORACLE_HOME</i>
In 11g the Middleware home is a container for the Oracle WebLogic Server home, and, optionally, one Oracle Common home and one or more Oracle homes.	The Oracle home that is created for all the Oracle Fusion Middleware products on a host computer. It includes binary and library files, the Oracle common directory and the individual product directories for each Oracle Fusion Middleware product you install.
This term is eliminated in 12c.	

Terminology Used in 11g	Terminology Used in 12c
<p>Oracle home <i>PRODUCT_ORACLE_HOME</i></p>	<p>Product directory <i>PRODUCT_DIR</i></p> <p>The product directories within the Middleware home are no longer Oracle homes. They are simply directories within the Oracle home that are created for all the Oracle Fusion Middleware products. Most Oracle Fusion Middleware components should be installed in the same Oracle home. The names of the product homes are predefined and can no longer be modified by the user during the installation.</p>
<p>Oracle instance This term is eliminated in 12c.</p>	<p>The installer will no longer create a separate instance directory for system components, such as Oracle HTTP Server. Instead, you can use the Fusion Middleware Configuration Wizard to configure your system components, just as you do for Java components. Instance information about each system component will be stored in the domain home.</p>
<p>Oracle Fusion Middleware farm This term is eliminated in 12c.</p>	<p>The term "farm" is no longer necessary for 12c. It was used in 11g to refer to a container for a WLS domain and its associated system component instances when presented in Fusion Middleware Control.</p>

Overview of Oracle Fusion Middleware

Oracle Fusion Middleware is a comprehensive family of products ranging from application development tools and integration solutions to identity management, collaboration, and business intelligence reporting. This chapter provides an introduction to Oracle Fusion Middleware. It includes the following sections:

- [Section 1.1, "About Oracle Fusion Middleware"](#)
- [Section 1.2, "Understanding the Functions of Middleware"](#)
- [Section 1.3, "Understanding Middleware Architecture Design"](#)
- [Section 1.4, "Understanding the Oracle Fusion Middleware Solution"](#)

1.1 About Oracle Fusion Middleware

Oracle Fusion Middleware is a collection of standards-based software products that spans a range of tools and services: from Java EE and developer tools, to integration services, identity management, business intelligence, and collaboration. Oracle Fusion Middleware offers complete support for development, deployment, and management.

Middleware provides the infrastructure which facilitates creation of business applications, and provides core services like concurrency, transactions, threading, and messaging. When configured appropriately, it also provides security and enables high availability functionality to your enterprise.

Middleware includes Web servers, application servers, content management systems, and similar tools that support application development and delivery. It is especially integral to information technology based on Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), Web services, SOA, Unicode, Web 2.0 infrastructure, and Lightweight Directory Access Protocol (LDAP).

Due to continued growth and use of network-based applications by businesses, middleware technologies are increasingly important. Companies and organizations are now building enterprisewide information systems by integrating previously independent applications with new software developments. The integration process may involve legacy applications which may be used only with, or through a nonmodifiable interface. In some cases, rewriting the code for a legacy application may be cost-prohibitive.

Increasingly, information systems are composed of a collection of various specialized hardware devices interconnected by a network. Each device performs a function that involves receipt of real time data and remote interaction with other devices of the system. Some examples include computer networks, telecommunication systems, uninterrupted power supply units, and decentralized manufacturing units.

Interaction with the information system may span a wide range of performance. You can interact with Internet applications through a variety of devices, whose characteristics and performance figures span an increasingly wide range. Between a high performance personal computer, a smart telephone, and a tablet computer, the variations in bandwidth, local processing power, screen capacity, and the ability to display color pictures, are extremely large.

1.2 Understanding the Functions of Middleware

Applications use intermediate software that resides on top of the operating systems and communication protocols to perform the following functions:

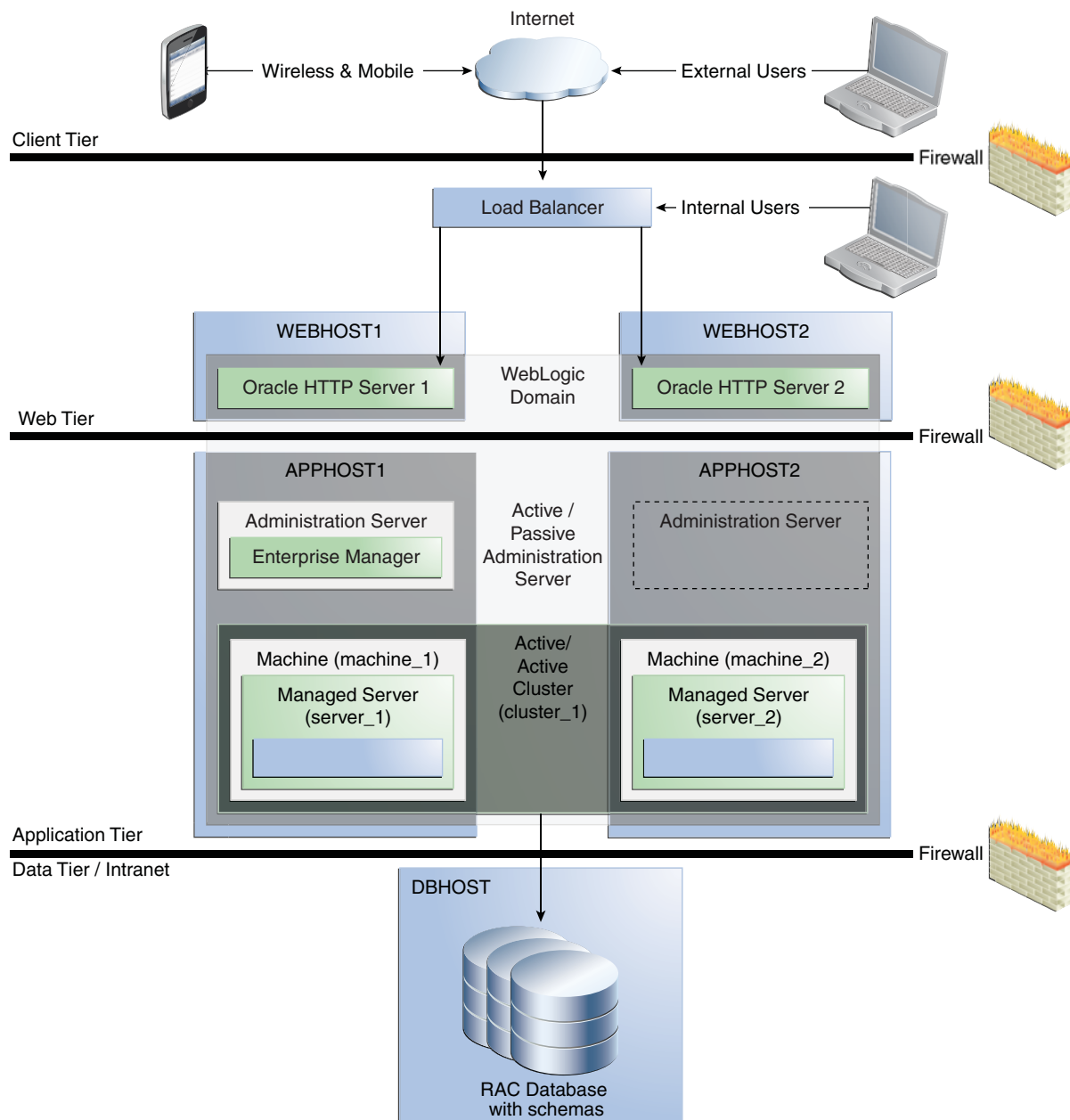
- Hide the distributed nature of the application. An application represents a collection of interconnected parts that are operational and running in distributed locations, out of view.
- Hide the heterogeneity of the enterprise. This includes the hardware components used, computer operating systems, and communication protocols.
- Provide uniform, standard, high-level interfaces to the application developers and integrators, so that applications can be easily composed, reused, ported, and made to interoperate.
- Supply a set of common services to perform various general purpose functions to avoid duplicating efforts, and to facilitate collaboration between applications.

Middleware makes application development easier, by providing common programming abstractions, by masking application heterogeneity and the distribution of the underlying hardware and operating systems, and by hiding low-level programming details.

1.3 Understanding Middleware Architecture Design

The function of middleware is to mediate interaction between the parts of an application, or between applications. Therefore, considerations for architectural structure play a central role in middleware design. The architectural design encompasses the organization, overall structure, and communication patterns, both for applications and for the middleware itself.

[Figure 1–1](#) provides an overview of the Oracle Fusion Middleware architecture, using the standard installation topology, which is a sample topology and is not the only topology supported for the product. (See "Understanding the Standard Installation Topology" in *Planning an Installation of Oracle Fusion Middleware* for more information about the standard installation topology.

Figure 1–1 Oracle Fusion Middleware Architecture Overview

In this architecture, which illustrates a highly available environment:

- From the client tier, external users, including wireless and mobile users, access Oracle Fusion Middleware through the Web tier, which is in a demilitarized zone (DMZ).
- The Web Tier contains a Load Balancer, and two instances of Oracle HTTP Server, a Web server.
- The Application Tier, contains Oracle WebLogic Server Administration Server and Managed Servers. Those Managed Servers can host applications, such as Java or Oracle ADF applications.
- The Data Tier contains databases that hold Oracle Fusion Middleware metadata and user data.

1.4 Understanding the Oracle Fusion Middleware Solution

Oracle Fusion Middleware is a collection of standards-based software products that includes a range of tools and services including developer tools, a service integration platform and data integration services. Oracle Fusion Middleware offers complete support for development, deployment, and management.

Specifically, Oracle Fusion Middleware offers the following solutions through its middleware design:

- **Development Tools:** An integrated, but modular, set of development tools to build complete applications, rather than using lots of specialized tools. The design tool includes a single design environment for user interface, business logic, service composition, business process or workflow, business rules, and business intelligence. The design tool enables simplified design and debugging, and to improve productivity. Includes Oracle JDeveloper and Oracle Application Development Framework.
- **Application Server:** A standards-based Java EE application server to run the enterprise applications and provide the Web services infrastructure for interoperability.
- **Web Server:** An HTTP-based server that delivers static and dynamic content over the Internet. A Web Server either serves the content itself, or routes the request to Application Servers for further processing.
- **Enterprise Management:** Operations and administration by running on a grid architecture with grouping, backup, and other high availability technologies, and integrating with Oracle Enterprise Manager for systems management. Includes Fusion Middleware Control, Oracle WebLogic Server Administration Console, and Oracle WebLogic Scripting Tool.

For information about the management tools, see [Chapter 4](#).

Understanding Key Concepts

This chapter provides information about key Oracle Fusion Middleware concepts, such as domains, Administration Server, Managed Servers, Java components, system components, as well as the directory structure of an Oracle Fusion Middleware installation.

- [Section 2.1, "What is an Oracle WebLogic Server Domain?"](#)
- [Section 2.2, "What Is a Standalone Domain?"](#)
- [Section 2.3, "What are the Key Oracle Fusion Middleware Directories?"](#)
- [Section 2.4, "What is the WebLogic Management Framework?"](#)
- [Section 2.5, "What Is the Metadata Repository?"](#)

2.1 What is an Oracle WebLogic Server Domain?

An Oracle WebLogic Server **domain** is a logically related group of Java components. A domain includes a special WebLogic Server instance called the **Administration Server**, which is the central point from which you configure and manage all resources in the domain. Usually, you configure a domain to include additional WebLogic Server instances called **Managed Servers**. You deploy Java components, such as Web applications, EJBs, and Web services, and other resources, to the Managed Servers and use the Administration Server for configuration and management purposes only.

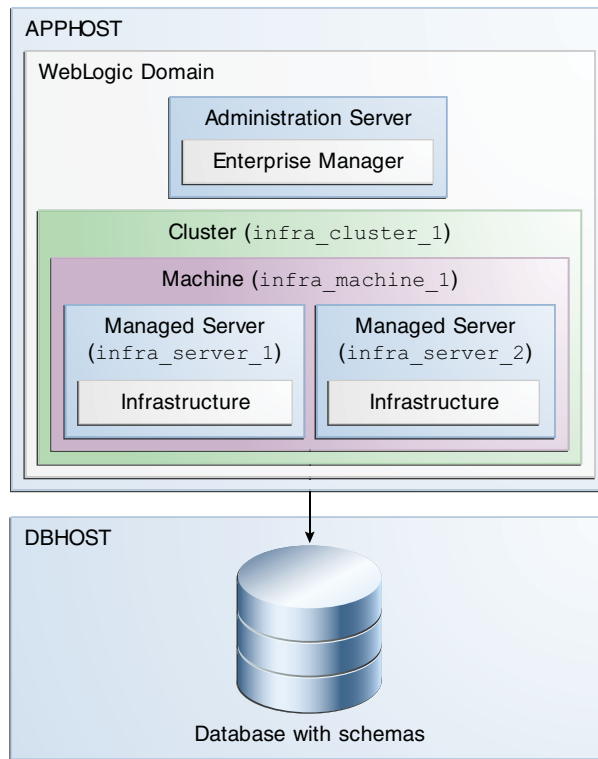
An Oracle WebLogic Server domain that was created using the Oracle Fusion Middleware Infrastructure distribution can also include system components, such as Oracle HTTP Server.

Managed Servers in a domain can be grouped together into a cluster.

The directory structure of a domain is separate from the directory structure of the WebLogic Server home. It can reside anywhere; it need not be within the Oracle home directory. The top-level directory of a domain is referred to as the **domain home**.

[Figure 2–1](#) shows an extended domain with an Administration Server, and two Managed Servers in a cluster. It also shows a database, which holds the metadata for Oracle Fusion Middleware

Figure 2–1 Oracle WebLogic Server Domain



See Also: *Understanding Domain Configuration for Oracle WebLogic Server* for more information about domain configuration

A domain can also contain system components, such as Oracle HTTP Server.

The following topics describe entities in the domain:

- [What Is the Administration Server?](#)
- [Understanding Managed Servers and Managed Server Clusters](#)
- [What Is a Java Component?](#)
- [What Is a System Component?](#)
- [What Is Node Manager?](#)

2.1.1 What Is the Administration Server?

The **Administration Server** operates as the central control entity for the configuration of the entire domain. It maintains the domain's configuration documents and distributes changes in the configuration documents to Managed Servers. The Administration Server serves as a central location from which to manage and monitor all resources in a domain.

Each domain must have one server instance that acts as the Administration Server.

To interact with the Administration Server, you can use Fusion Middleware Control, the Oracle WebLogic Server Administration Console, Oracle WebLogic Scripting Tool (WLST), or create your own JMX client.

Oracle WebLogic Server Administration Console and Fusion Middleware Control run in the Administration Server. Oracle WebLogic Server Administration Console is the

Web-based administration console used to manage the resources in an Oracle WebLogic Server domain, including the Administration Server and Managed Servers. Fusion Middleware Control is a Web-based administration console used to manage Oracle Fusion Middleware, including components such as Oracle WebLogic Server, Oracle Coherence, and Oracle HTTP Server.

See Also:

- [Section 4.2.1](#) for more information about Fusion Middleware Control
- [Section 4.2.2](#), as well as *Understanding Oracle WebLogic Server* and the Oracle WebLogic Server Administration Console Online help, for more information about Oracle WebLogic Server Administration Console

2.1.2 Understanding Managed Servers and Managed Server Clusters

Managed Servers host business applications, application components, Web services, and their associated resources. To optimize performance, Managed Servers maintain a read-only copy of the domain's configuration document. When a Managed Server starts, it connects to the domain's Administration Server to synchronize its configuration document with the document that the Administration Server maintains.

When you create a domain, you create it using a particular domain template. That template supports a particular component or group of components, such as the Oracle SOA Suite. The Managed Servers in the domain are created specifically to host those particular Oracle Fusion Middleware components.

Oracle Fusion Middleware Java components (such as Oracle Coherence), as well as customer-developed applications, are deployed to Managed Servers in the domain.

If you want to add other components to a domain that was created using a template that supports another component, you can extend the domain by creating additional Managed Servers in the domain, using a domain template for the component that you want to add. See "Extending a Domain to Support Additional Components" in *Administering Oracle Fusion Middleware* for more information.

For production environments that require increased application performance, throughput, or high availability, you can configure two or more Managed Servers to operate as a cluster. A **cluster** is a collection of multiple WebLogic Server instances running simultaneously and working together to provide increased scalability and reliability. In a cluster, most resources and services are deployed identically to each Managed Server (as opposed to a single Managed Server), enabling failover and load balancing. A single domain can contain multiple Oracle WebLogic Server clusters, as well as multiple Managed Servers that are not configured as clusters. The key difference between clustered and nonclustered Managed Servers is support for failover and load balancing. These features are available only in a cluster of Managed Servers.

See Also: "Understanding WebLogic Server Clustering" in *Administering Clusters for Oracle WebLogic Server*

2.1.3 What Is a Java Component?

A **Java component** is an Oracle Fusion Middleware component that is deployed as one or more Java EE applications and a set of resources. Java components are deployed to an Oracle WebLogic Server domain as part of a domain template. Oracle WebLogic Server and Oracle Coherence are examples of Java components.

2.1.4 What Is a System Component?

A **system component** is a manageable process that is not deployed in a Java application container. Oracle HTTP Server is an example of a system component. System components can be deployed in a WebLogic Server domain or in a standalone domain and managed by the Weblogic Management Framework. See [Section 2.4](#) for information about the Weblogic Management Framework.

2.1.5 What Is Node Manager?

Node Manager is a Java utility that runs as a separate process from Oracle WebLogic Server and allows you to perform common operations for a Managed Server, regardless of its location with respect to its Administration Server, and for system components. You can perform those operations from a remote location. While use of Node Manager is optional in a WebLogic domain, it provides valuable benefits if your Oracle WebLogic Server environment hosts applications with high-availability requirements. It is not optional in a standalone domain.

If you run Node Manager on a computer that hosts Managed Servers, you can start and stop the Managed Servers remotely using the Administration Console, Fusion Middleware Control, or the command line. Node Manager can also automatically restart a Managed Server after an unexpected failure.

You can configure the scope of Node Manager:

- Per domain

With a per-domain Node Manager, the Node Manager is associated with a domain and is configured to control all servers for the domain on a machine. (A per-domain Node Manager is a Java-based Node Manager.)

This is the default when you configure Node Manager with the Configuration Wizard.

- Per Machine

With a per-machine Node Manager, the Node Manager process is not associated with a specific WebLogic Server domain but with a machine. You can use the same Node Manager process to control server instances in any WebLogic Server domain, as long as the server instances reside on the same machine as the Node Manager process. A per-machine Node Manager must run on each computer that hosts WebLogic Server instances—whether Administration Server or Managed Server—that you want to control with Node Manager.

See Also: *Administering Node Manager for Oracle WebLogic Server*

2.2 What Is a Standalone Domain?

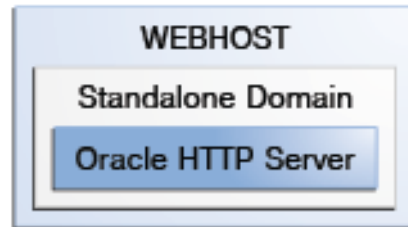
A **standalone domain** is a container for system components, such as Oracle HTTP Server. It has a directory structure similar to an Oracle WebLogic Server domain, but it does not contain an Administration Server or Managed Servers. It can contain one or more instances of system components of the same type, such as Oracle HTTP Server, or a mix of system component types.

The WebLogic Management Framework provides tools for managing standalone domains. See [Section 2.4](#) for more information about the WebLogic Management Framework.

Generally, you use a standalone domain when you do not want your Oracle HTTP Server implementation to front an Oracle WebLogic Server domain and do not need the management functionality provided by Fusion Middleware Control.

Figure 2–4 shows a standalone domain with Oracle HTTP Server installed.

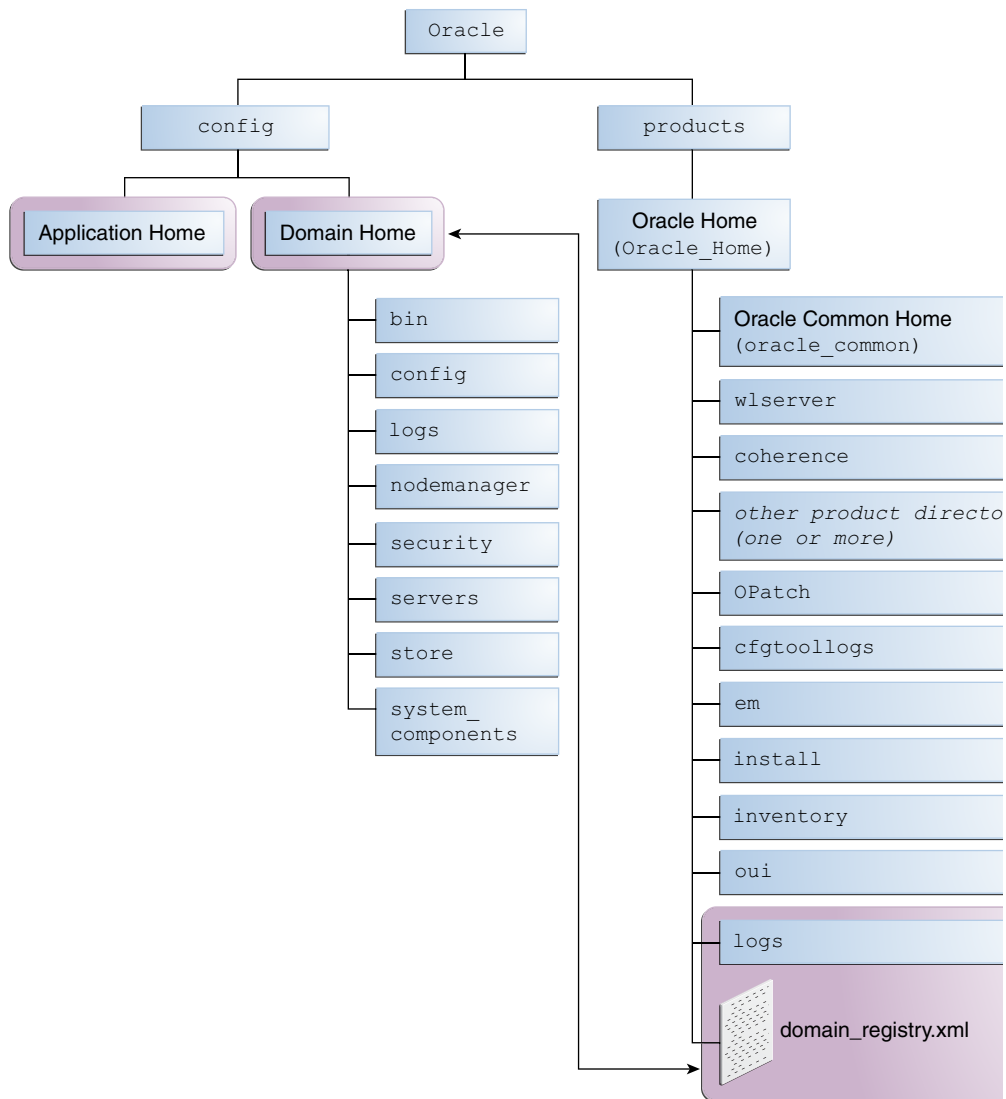
Figure 2–2 Standalone Domain



2.3 What are the Key Oracle Fusion Middleware Directories?

Figure 2–3 shows the high-level directory structure after installing and configuring Oracle Fusion Middleware with Oracle WebLogic Server and Oracle JRF.

Figure 2–3 Directory Structure of a Oracle WebLogic Server Domain



For a diagram of the directory structure when Oracle HTTP Server is installed in a WebLogic domain, see "Understanding the Oracle HTTP Server Directory Structure in a WebLogic Server Domain" in *Installing and Configuring Oracle HTTP Server*.

Table 2–1 describes the key Oracle Fusion Middleware directories and the variables that are used when referring to those directories in an Oracle Fusion Middleware installation. When you see these variables in an example or procedure, replace the variable with the full path to the corresponding directory path in your enterprise topology.

Note: The directory paths provided in the examples below assume that Oracle Universal Installer's default directory names were used.

Table 2–1 Directories in a Oracle Fusion Middleware Installation

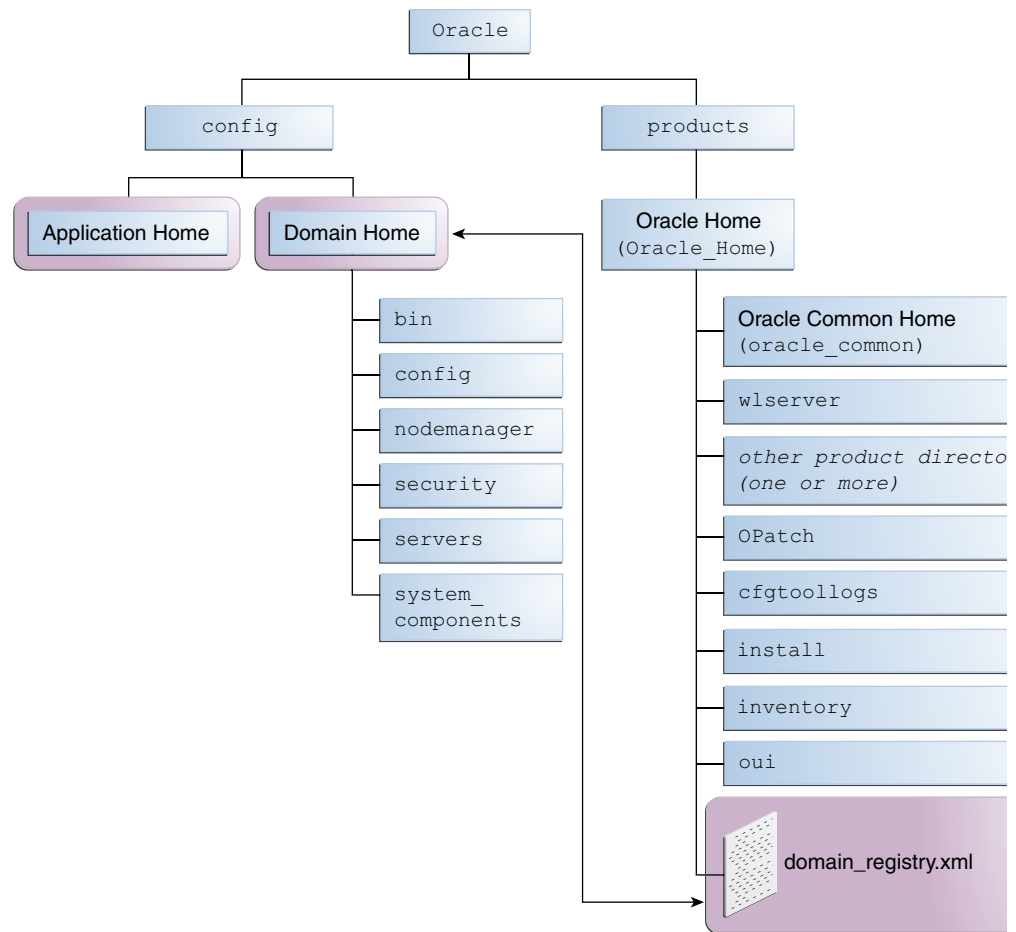
Directory Name	Variable	Description	Directory Path
Oracle home	<i>ORACLE_HOME</i>	The Oracle home that is created for all the Oracle Fusion Middleware products on a host computer. This read-only directory contains binary and library files, the Oracle common directory and the individual product directories for each Oracle Fusion Middleware product you install.	<i>/install_location/Oracle_Home</i>
Oracle Common home	<i>ORACLE_COMMON</i>	The directory that contains the binary and library files that are common to all the Oracle Fusion Middleware products and features installed in the Oracle home. In addition, the Oracle Common directory includes the files required for common tools, such as Oracle Enterprise Manager Fusion Middleware Control, WLST, the Configuration Wizard, upgrade tools, and Oracle JRF There is only one Oracle Common directory within each Oracle home.	<i>/install_location/Oracle_Home/oracle_common</i>
product directory	<i>PROD_DIR</i>	The directory within the Oracle home, which contains the binary files associated with a logical product or feature set. The name of each product directory within the Oracle home is predefined by the installer and cannot be changed.	<i>/install_location/Oracle_Home/Prod_Dir</i>
Oracle WebLogic Server home	<i>WL_HOME</i>	The specific product directory for the WebLogic Server binary files.	<i>/install_location/Oracle_Home/wlserver</i>
Fusion Middleware Control	<i>EM_DIR</i>	The Oracle Enterprise Manager Fusion Middleware Control Console directory within an Oracle Fusion Middleware Oracle home. This directory contains the binary and library files required to run Fusion Middleware Control.	<i>/install_location/Oracle_Home/em</i>

Table 2–1 (Cont.) Directories in a Oracle Fusion Middleware Installation

Directory Name	Variable	Description	Directory Path
Domain home	<i>DOMAIN_HOME</i>	<p>The location in which the domain information and configuration artifacts are stored.</p> <p>For information, about a Oracle WebLogic Server domain, see Section 2.1.</p> <p>For information, about a standalone domain, see Section 2.1.</p>	<i>/install_location/oracle/domains/ba</i> se_domain
Application home	<i>APPLICATION_HOME</i>	<p>The directory where the applications related to the domains you configure will be created. Oracle recommended that this location reside outside of the Oracle home directory; in the event that you need to upgrade or patch your software, the files inside the Oracle home would be affected.</p>	<i>install_location/oracle/</i> applications

[Figure 2–4](#) shows the directory structure when you have installed a system component, such as Oracle HTTP Server, in a standalone domain.

Figure 2–4 Directory Structure of a Standalone Domain



The directories are described in [Table 2–1](#).

2.4 What is the WebLogic Management Framework?

Oracle Fusion Middleware provides the **WebLogic Management Framework**, which provides heterogeneous management capabilities for Oracle Fusion Middleware products that require basic administrative capabilities. Its capabilities include start, stop, configuration settings and other such basic product lifecycle operations through a common command line, API and user interface. WebLogic Management Framework is comprised of specific WebLogic Server and Coherence features.

The WebLogic Management Framework manages both WebLogic Server domains, which can contain Java components and system components, and standalone domains, which contain system components. However, not all functionality is provided for standalone domains, as shown in the following table:

Functionality	WebLogic Server Domain	Standalone Domain	More information
Node Manager	Yes	Yes	Section 2.1.5
WebLogic Scripting Tool	Yes	Yes	Section 4.2.3

Functionality	WebLogic Server Domain	Standalone Domain	More information
Configuration Wizard	Yes	Yes	<i>Creating Domains Using the Configuration Wizard</i>
pack and unpack utilities	Yes	Yes	<i>Creating Templates and Domains Using the Pack and Unpack Commands</i>
WebLogic Server JMX and MBean Infrastructure	Yes	No	<i>Developing Custom Management Utilities Using JMX for Oracle WebLogic Server</i>
WebLogic Server RESTful Management Services	Yes	No	<i>Administering Oracle WebLogic Server with RESTful Management Services</i>
Weblogic.deployer	Yes	No	"weblogic.Deployer Command-Line Reference" in <i>Deploying Applications to Oracle WebLogic Server</i>
Coherence Caching for WebLogic Management Framework	Yes	No	<i>Managing Oracle Coherence</i>
Administration Console	Yes	No	Section 4.2.2
Fusion Middleware Control	Yes	No	Section 4.2.1

2.5 What Is the Metadata Repository?

The metadata repository contains metadata for Oracle Fusion Middleware components, such as Oracle Application Development Framework. It can also contain metadata about the configuration of Oracle Fusion Middleware and metadata for enterprise applications.

A metadata repository can be database-based or file-based. If it is database-based, repository can be installed into an existing database using the Repository Creation Utility (RCU). With RCU, you can create schemas for Oracle Fusion Middleware components.

For more information about database-based metadata repositories, see *Creating Schemas with the Repository Creation Utility*.

A particular type of repository, the MDS Repository, contains metadata for most Oracle Fusion Middleware components, such as Oracle Application Development Framework, and for certain types of applications. For more information about the MDS Repository, see "Managing the MDS Repository" in *Administering Oracle Fusion Middleware*.

Overview of Oracle Fusion Middleware Components

This chapter describes Oracle Fusion Middleware components, such as Oracle WebLogic Server, Oracle HTTP Server, and Oracle Coherence.

- [Section 3.1, "Understanding Oracle WebLogic Server"](#)
- [Section 3.2, "Understanding Oracle Web Services"](#)
- [Section 3.3, "Understanding Oracle Coherence"](#)
- [Section 3.4, "Understanding Oracle HTTP Server"](#)
- [Section 3.5, "Understanding Oracle User Messaging Service"](#)
- [Section 3.6, "Understanding Oracle Platform Security Services"](#)
- [Section 3.7, "Understanding Oracle Data Integrator"](#)

3.1 Understanding Oracle WebLogic Server

The Oracle WebLogic Server is a scalable, enterprise-ready Java Platform, Enterprise Edition (Java EE) application server. The Oracle WebLogic Server infrastructure supports the deployment of many types of distributed applications and is an ideal foundation for building applications based on SOA.

Java EE is a widely used platform for server programming in the Java programming language. The Java EE Platform differs from the standard edition of Java in that it adds libraries which provide functionality to deploy fault-tolerant, distributed, multi-tier Java software, based largely on modular components running on an application server.

In addition to the Java EE implementation, Oracle WebLogic Server enables enterprises to deploy business-critical applications in a robust, secure, highly available, manageable, and scalable environment. These features allow enterprises to configure clusters of Oracle WebLogic Server instances to distribute load and provide extra capacity in case of hardware or other failures. New diagnostic tools allow system administrators to monitor and tune the performance of deployed applications and the Oracle WebLogic Server environment itself. You can also configure Oracle WebLogic Server to monitor and tune application throughput automatically, without human intervention. Extensive security features protect access to services, keep enterprise data secure, and prevent malicious attacks.

For more information, see *Understanding Oracle WebLogic Server*.

3.2 Understanding Oracle Web Services

A Web Service is a set of functions packaged into a single application that is available to other systems on a network. The network can be a corporate intranet or the Internet. Because Web Services rely on basic, standard technologies which most systems provide, they are an excellent means for connecting distributed systems together. They can be shared by and used as a component of distributed Web-based applications. Other systems, such as customer relationship management systems, order-processing systems, and other existing back-end applications, can call a Web Service function to request data or perform an operation.

For an overview of Web services in Oracle Fusion Middleware, a description of the Web services and clients supported in each category, and a roadmap for implementing Oracle Fusion Middleware Web services, see "Introducing Oracle WebLogic Web Services" in *Understanding WebLogic Web Services for Oracle WebLogic Server*.

3.3 Understanding Oracle Coherence

Oracle Coherence provides clustered data management with a fully coherent, single system image (SSI), scalability for both read and write access, fast, transparent failover and failback, linear scalability for storage and processing, no Single-Points-of-Failure (SPOFs) and cluster-wide locking and transactions.

Built on top of this foundation are the various services that Coherence provides, including database caching, HTTP session management, grid agent invocation and distributed queries.

For information about Oracle Coherence, see "Introduction to Coherence" in *Developing Applications with Oracle Coherence*.

3.4 Understanding Oracle HTTP Server

Oracle HTTP Server is based on Apache HTTP Server infrastructure, and includes modules developed specifically by Oracle. The features of single sign-on, clustered deployment, and high availability enhance the operation of the Oracle HTTP Server. Oracle HTTP Server has the following components to handle client requests:

- HTTP listener, to handle incoming requests and route them to the appropriate processing utility.
- Modules (mods), to implement and extend the basic functionality of Oracle HTTP Server. Many of the standard Apache HTTP Server modules are included with Oracle HTTP Server. Oracle also includes several modules that are specific to Oracle Fusion Middleware to support integration between Oracle HTTP Server and other Oracle Fusion Middleware components.
- Perl interpreter, a persistent Perl runtime environment embedded in Oracle HTTP Server through `mod_perl`.

For information about Oracle HTTP Server see *Administering Oracle HTTP Server*.

3.5 Understanding Oracle User Messaging Service

Oracle User Messaging Service enables two-way communication between users and deployed applications. Key features include:

- Support for a variety of messaging channels: Messages can be sent and received through email, instant messaging (IM) (XMPP), short message service (SMS) (SMPP), and voice.
- Two-way messaging: In addition to sending messages from applications to users (referred to as outbound messaging), users can initiate messaging interactions (inbound messaging). For example, a user can send an email or text message to a specified address; the message is routed to the appropriate application that can then respond to the user or invoke another process according to its business logic.
- User messaging preferences: End users can use a web interface to define preferences for how and when they receive messaging notifications. Applications immediately become more flexible; rather than deciding whether to send to a user's email address or IM client, the application can simply send the message to the user, and let UMS route the message according to the user's preferences.
- Robust message delivery: UMS keeps track of delivery status information provided by messaging gateways, and makes this information available to applications so that they can respond to a failed delivery. Or, applications can specify one or more failover addresses for a message in case delivery to the initial address fails. Using the failover capability of UMS frees application developers from having to implement complicated retry logic.

For information about Oracle User Messaging Service, see *Administering Oracle User Messaging Service*.

3.6 Understanding Oracle Platform Security Services

Oracle Platform Security Services provides enterprise product development teams, systems integrators, and independent software vendors with a standards-based, portable, integrated, enterprise-grade security framework for Java SE and Java EE applications.

Oracle Platform Security Services is the underlying security platform that provides security to Oracle Fusion Middleware including WebLogic Server, Server Oriented Architecture (SOA) applications, Oracle WebCenter, Oracle Application Development Framework (ADF) applications, and Oracle Entitlement Server.

Oracle Platform Security Services provides an abstraction layer in the form of application programming interfaces (APIs) that insulate developers from security and identity management implementation details. With OPSS, developers do not need to know the details of, for example, cryptographic key management, repository interfaces, or other identity management infrastructures. Using OPSS, in-house developed applications, third-party applications, and integrated applications benefit from the same, uniform security, identity management, and audit services across the enterprise.

3.7 Understanding Oracle Data Integrator

Oracle Data Integrator (ODI) provides a fully unified solution for building, deploying, and managing complex data warehouses or as part of data-centric architectures in a SOA or business intelligence environment. In addition, it combines all the elements of data integration — data movement, data synchronization, data quality, data management, and data services—to ensure that information is timely, accurate, and consistent across complex systems.

Oracle Data Integrator features an active integration platform that includes all styles of data integration: data-based, event-based and service-based. ODI unifies silos of

integration by transforming large volumes of data efficiently, processing events in real time through its advanced Changed Data Capture (CDC) capability, and providing data services to the Oracle SOA Suite. It also provides robust data integrity control features, assuring the consistency and correctness of data. With powerful core differentiators — heterogeneous E-LT, Declarative Design and Knowledge Modules — Oracle Data Integrator meets the performance, flexibility, productivity, modularity and hot-pluggability requirements of an integration platform.

For more information, see "Introduction to Oracle Data Integrator" in *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*.

Using Oracle Fusion Middleware Tools

This chapter discusses the Oracle Fusion Middleware tools used to perform administration tasks such as installing, configuring, and managing your middleware components. It also describes tools for developing applications.

- [Section 4.1, "Understanding the Installation and Configuration Tools"](#)
- [Section 4.2, "Understanding the Oracle Fusion Middleware Administration Tools"](#)
- [Section 4.3, "Understanding the Development Tools"](#)

4.1 Understanding the Installation and Configuration Tools

Oracle Fusion Middleware offers the following primary tools for installing, configuring and upgrading Oracle Fusion Middleware:

- [Oracle Universal Installer](#)
- [Repository Configuration Utility](#)
- [OPatch](#)
- [Configuration Wizard](#)
- [Reconfiguration Wizard](#)
- [Upgrade Assistant](#)

4.1.1 Oracle Universal Installer

Oracle Universal Installer (OUI) is a Java-based installer that enables you to install, and deinstall Oracle components. It performs component-based installations as well as complex installations, such as integrated bundle and suite installations, and installations over the Web.

One of the key features of OUI is that it provides a prerequisite checking tool to diagnose the readiness of an environment for installation. The prerequisite checks are run as part of the installation process, but can also be run as a separate application.

For more information, see *Installing Software with the Oracle Universal Installer*.

4.1.2 Repository Configuration Utility

Many of the Oracle Fusion Middleware components require the existence of schemas in a database prior to installation. These schemas are created and loaded in your database using the Repository Creation Utility (RCU). See *Creating Schemas with the Repository Creation Utility* for more information.

4.1.3 OPatch

OPatch is a utility that assists you with the process of applying interim patches to Oracle's software. OPatch 11.1 is a Java-based utility that can run on either OUI-based Oracle homes or standalone homes. It works on all operating systems for which Oracle releases software. see *Patching with OPatch*.

4.1.4 Configuration Wizard

The Configuration Wizard creates the appropriate directory structure for a WebLogic Server domain or a standalone domain, a domain configuration file, and scripts you can use to start the servers in the domain.

The Configuration Wizard guides you through the process of creating or updating a domain for your target environment by selecting the product components to include in your domain, or by selecting template JAR files. If necessary, you can also customize the domain to suit your environment by adding and configuring Managed Servers, clusters, and machine definitions, or customizing predefined JDBC data sources and JMS file store directories.

For more information, see "Overview of the Configuration Wizard" in *Creating Domains Using the Configuration Wizard*.

4.1.5 Reconfiguration Wizard

The Reconfiguration Wizard reconfigures a WebLogic Server domain. The following items are automatically updated, depending on the applications in the domain:

- WLS core infrastructure
- Domain version

If your domain also includes Oracle Fusion Middleware products, other items may also be updated, depending on the particular product. This lets you take advantage of new features that are included with the newest version of Oracle Fusion Middleware. For more information, refer to the installation and upgrade guide for the Oracle Fusion Middleware component.

For more information, see "Reconfiguring a WebLogic Domain in Graphical Mode" in *Upgrading Oracle WebLogic Server*.

4.1.6 Upgrade Assistant

The Upgrade Assistant automates the upgrade of many aspects of Oracle Fusion Middleware environment. The Oracle Fusion Middleware upgrade Assistant guides you through the process of upgrading from previous versions. For more information see *Planning an Upgrade of Oracle Fusion Middleware*.

4.2 Understanding the Oracle Fusion Middleware Administration Tools

After you install and configure Oracle Fusion Middleware, you can use the graphical user interfaces or command-line tools to manage your environment.

Oracle offers the following primary tools for managing your Oracle Fusion Middleware installations:

- [Oracle Enterprise Manager Fusion Middleware Control](#)
- [Oracle WebLogic Server Administration Console](#)

- [Oracle WebLogic Scripting Tool \(WLST\)](#)
- [Oracle Enterprise Manager Cloud Control](#)

Note that you should use these tools, rather than directly editing configuration files, to perform all administrative tasks unless a specific procedure requires you to edit a file. Editing a file may cause the settings to be inconsistent and generate problems.

4.2.1 Oracle Enterprise Manager Fusion Middleware Control

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

It can manage the Administration Server, Managed Servers, clusters, and the Oracle Fusion Middleware components that are installed, configured, and running in the domain.

Fusion Middleware Control organizes a wide variety of performance data and administrative functions into distinct, Web-based home pages for the domain, clusters, servers, components, and applications. The Fusion Middleware Control home pages make it easy to locate the most important monitoring data and the most commonly used administrative functions all from your Web browser.

Fusion Middleware Control provides direct access to Oracle WebLogic Server Administration Console. The Web pages in the Fusion Middleware Control interface contain links that take enable you to access the Administration Console. For example, on the Domain home page, if you go to the Summary area there is a link that takes you to the Administration Console.

For more information, see "Getting Started Using Oracle Enterprise Manager Fusion Middleware Control" in *Administering Oracle Fusion Middleware*.

4.2.2 Oracle WebLogic Server Administration Console

Oracle WebLogic Server Administration Console is a Web browser-based, graphical user interface that you use to manage an Oracle WebLogic Server domain. It is accessible from any supported Web browser with network access to the Administration Server.

Use the Administration Console to:

- Configure, start, and stop Oracle WebLogic Server instances
- Configure Oracle WebLogic Server clusters
- Configure Oracle WebLogic Server services, such as database connectivity (JDBC) and JMS messaging
- Configure security parameters, including creating and managing users, groups, and roles
- Configure and deploy Java EE applications
- Monitor server and application performance
- View server and domain log files
- View application deployment descriptors
- Edit selected run time application deployment descriptor elements

For more information, see "Getting Started Using Oracle WebLogic Server Administration Console" in *Administering Oracle Fusion Middleware*.

4.2.3 Oracle WebLogic Scripting Tool (WLST)

The Oracle WebLogic Scripting Tool (WLST) is a command-line scripting environment that you can use to create, manage, and monitor Oracle WebLogic Server domains and standalone domains. It is based on the Java scripting interpreter, Jython. In addition to supporting standard Jython features such as local variables, conditional variables, and flow control statements, WLST provides a set of scripting functions (commands) that are specific to Oracle WebLogic Server and Oracle Fusion Middleware components. You can extend the WebLogic scripting language to suit your needs by following the Jython language syntax.

You can use any of the following techniques to invoke WLST commands:

- Interactively, on the command line
- In script mode, supplied in a file
- Embedded in Java code

For more information, see "Getting Started Using Oracle WebLogic Server Scripting Tool (WLST)" in *Administering Oracle Fusion Middleware* and the *WLST Command Reference for WebLogic Server*.

4.2.4 Oracle Enterprise Manager Cloud Control

Oracle Enterprise Manager Cloud Control provides a comprehensive management solution for Oracle WebLogic Server, Oracle Fusion Middleware, non-Oracle middleware technology, and lifecycle management including configuration management, compliance management, provisioning, and patching. Oracle's offering encompasses out-of-the-box availability and performance monitoring, robust diagnostics, configuration management, and lifecycle management across middleware software such as Oracle WebLogic Server, Oracle Coherence, and Oracle HTTP Server.

Oracle Enterprise Manager Cloud Control is a Web browser-based, graphical user interface that you can use to monitor multiple Oracle Fusion Middleware environments and Oracle WebLogic Server domains. In fact, Cloud Control provides deep management solutions for Oracle technologies including Oracle packaged applications, Oracle Database and Oracle VM. Cloud Control also offers extensive support for non-Oracle technologies through more than two dozen heterogeneous management plug-ins and connectors including Microsoft MOM, IBM WebSphere, JBoss, EMC storage, F5 BIG IP, Check Point Firewall, and Remedy.

Beyond managing your entire data center from a single interface, Cloud Control offers critical features that help you manage Oracle Fusion Middleware and Oracle WebLogic Server more effectively and efficiently. Such additional management capabilities include:

- Analyze and report on trends based upon collected availability and performance data.
- Receive alert notifications (via email, page, SNMP) for metrics which have crossed thresholds previously defined by you.
- Automate common administrative operations (e.g. start/stop, WLST scripts).
- Resolve problems faster through visibility into all Java activity - including in-flight transactions - and tracing transactions from Java to Database and vice-versa.
- Detect, validate, and report authorized and unauthorized configuration changes in real time.

- Ensure configuration consistency across development and production environments.

For more information about Oracle Enterprise Manager Cloud Control refer to the *Oracle Enterprise Manager Cloud Control Introduction* available on OTN.

4.3 Understanding the Development Tools

This section describes Oracle Fusion Middleware support for building Fusion Web applications with Oracle ADF and the integrated development environment, Oracle JDeveloper. A Fusion Web application is any database-centric application that is targeted to run in the Oracle Fusion Middleware environment.

This section includes the following topics:

- [Oracle JDeveloper](#)
- [Oracle Application Development Framework](#)
- [Oracle TopLink](#)
- [Oracle Help](#)

4.3.1 Oracle JDeveloper

Oracle JDeveloper is an integrated development environment (IDE) for building SOA applications using the latest standards for Java, XML, Web services, and SQL. It supports the complete development lifecycle with integrated features for modeling, coding, debugging, testing, profiling, tuning, and deploying applications. Oracle JDeveloper is the main development platform for the Oracle Fusion Middleware suite of products. It is a cross-platform IDE that runs on Windows, Linux, Mac OS X, and other UNIX-based systems.

Oracle JDeveloper provides a visual and declarative development approach and works together with the Oracle ADF to simplify development.

Key features of Oracle JDeveloper include:

- A consistent development environment that can be used for various technology stacks including Java, SOA, Oracle WebCenter Portal, SQL and PL/SQL, HTML, and JavaScript.
- XML-based application development.
- A full development and modeling environment for building database objects and stored procedures.
- A wide range of application deployment options, including Integrated Oracle WebLogic Server, an integrated run time service for running and testing applications before deploying to a production environment.
- Extension capabilities that enable customization of the IDE based on development needs and add additional functionality.

For more information, see *Developing Applications with Oracle JDeveloper*.

4.3.2 Oracle Application Development Framework

Oracle ADF is an end-to-end application framework that builds on Java EE standards and open-source technologies to simplify and accelerate implementing service-oriented applications. Oracle ADF is suitable for enterprise developers who want to create applications that search, display, create, modify, and validate data using

Web, wireless, desktop, or Web services interfaces. Used in tandem, Oracle JDeveloper 11g and Oracle ADF provide an environment that covers the full development lifecycle from design to deployment, with drag-and-drop data binding, visual UI design, and team development features built in.

For more information, see *Understanding Oracle Application Development Framework*.

4.3.3 Oracle TopLink

Oracle TopLink is an advanced, object-persistence and object-transformation framework that provides development tools and run time capabilities that reduce development and maintenance efforts, and increase enterprise application functionality.

Oracle TopLink builds high-performance applications that store persistent object-oriented data in a relational database. It successfully transforms object-oriented data into either relational data, or Extensible Markup Language (XML) elements.

For more information about Oracle TopLink, see "Overview of Oracle TopLink" in *Understanding Oracle TopLink*.

4.3.4 Oracle Help

Oracle Help provides the means for developing and displaying HTML-based help systems for Java applications and Web applications. Authors can create a single help system that can be displayed - without modification - both in a Java environment, using Oracle Help for Java, and in a Web environment, using Oracle Help for the Web. Oracle Help for the Web is available in two formats: Oracle Help for the Web - UIX and Oracle Help for the Web - Rich Client.

Oracle Help for Java (OHJ) is a set of Java components, a Java API, and a file formats specification for developing and displaying HTML-based help content in a Java environment. OHJ is designed primarily for displaying help for Java applications, although it can also be implemented as a standalone document viewer for use in a Java environment.

Oracle Help for the Web - UIX (OHW-UIX) is a Java servlet and a file formats specification for developing and delivering HTML-based help content in a Web environment. OHW-UIX can be used to provide context-sensitive help for Web applications or as a means for processing and displaying structured views of independent HTML content on the Web.

Oracle Help for the Web - Rich Client (OHW-RC), similar to OHW-UIX, delivers HTML-based Help content in a Web environment. It uses Oracle Application Development Framework (Oracle ADF), which is based on the JavaServer Faces (JSF) technology, to build a user interface that follows Oracle's Browser Look And Feel Plus (BLAF+) guidelines.

For more information about Oracle Help, see "Introduction to Oracle Help" in *Developing Help Systems with Oracle Help*.