Oracle® Fusion Middleware

Configuring Oracle Fusion Middleware Using the Lifecycle Management Tools

Release 12*c* (12.2.1.1)

E70961-01

June 2016

ORACLE

Oracle Fusion Middleware Configuring Oracle Fusion Middleware Using the Lifecycle Management Tools, Release 12c (12.2.1.1)

E70961-01

Primary Authors: Phil Stubbs, Peter LaQuerre

Contributing Authors: Helen Grembowicz

Contents

Preface

This document describes how to configure and use the Oracle Fusion Middleware Lifecycle Management Tools to configure and manage a domain.

Audience

This document is intended for administrators who use the Lifecycle Management Tools to install, configure, and manage Oracle Fusion Middleware.

Related Resources

For more information, see these Oracle resources:

- *Java API Reference for the Lifecycle Management Action Framework*
- *Java API Reference for the Lifecycle Management Environment Specification*
- *Java API Reference for the Lifecycle Management Credential Management*

Conventions

The following text conventions are used in this document:

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

Part I

Understanding the Lifecycle Management Tools

The following topics describe the concepts and features of the Lifecycle Management Tools.

1

About the Lifecycle Management Tools

The Lifecycle Management Tools provide the means to install, configure, and manage an Oracle Fusion Middleware environment.

Understanding the Lifecycle Management Tools
: The following topics describe the concepts and features of the Lifecycle Management Tools.

What Are the Lifecycle Management Tools?
: The Oracle Fusion Middleware Lifecycle Management Tools provide a comprehensive set of tools that can automate the deployment and management of specific Oracle Fusion Middleware topologies.

What Can I Do With the Lifecycle Management Tools?
: With the Oracle Fusion Middleware Lifecycle Management Tools, you can define and deploy a new Fusion Middleware topology. Once you have defined the topology, by creating a topology, you can update the topology.

Understanding the Basics
: Before you begin using the Lifecycle Management Tools, you should understand some basic concepts and basic tasks you will be performing with the tools.

1.1 What Are the Lifecycle Management Tools?

The Oracle Fusion Middleware Lifecycle Management Tools provide a comprehensive set of tools that can automate the deployment and management of specific Oracle Fusion Middleware topologies.

The main components of the Lifecycle Management tools are described in the following table.

Component	Description
Fusion Middleware Environment Specification	The environment specification is a standardized, strongly-typed data structure that describes a Fusion Middleware environment.
Fusion Middleware Composer	Composer is a client application that provides a graphical user interface for describing a Fusion Middleware topology. A topology can be created from scratch by adding each element to the topology one-by-one.

Component	Description
Fusion Middleware Action Framework	The Action Framework is a standard mechanism to invoke lifecycle operations across products. Actions can be performed on targets, which are defined by the environment specification. The Lifecycle Management Tools provide a set of standard actions, such as the following: • Create an Oracle home • Create a Domain • Start and stop a server, cluster, and Node Manager. You can also create custom actions.
Fusion Middleware Provisioning APIs	The Provisioning APIs include: • The Java API for the Lifecycle Management Environment Specification • The Java API for the Lifecycle Managment Action Framework • The Java API for Lifecycle Management Credential Management For more information, see Using the Standard Actions JavaDoc

1.2 What Can I Do With the Lifecycle Management Tools?

With the Oracle Fusion Middleware Lifecycle Management Tools, you can define and deploy a new Fusion Middleware topology. Once you have defined the topology, by creating a topology, you can update the topology.

Define and Deploy a New Fusion Middleware Environment
: You can use the Oracle Fusion Middleware Lifecycle Management Tools to define and deploy a new Oracle Fusion Middleware environment. This is an alternative to manually installing and deploying the software with the Fusion Middleware installer and Configuration Wizard.

Update an Existing Fusion Middleware Environment
: Besides creating a new environment, you can also update an existing environment. For example, you can extend an existing Oracle Fusion Middleware domain to include additional components.

1.2.1 Define and Deploy a New Fusion Middleware Environment

You can use the Oracle Fusion Middleware Lifecycle Management Tools to define and deploy a new Oracle Fusion Middleware environment. This is an alternative to manually installing and deploying the software with the Fusion Middleware installer and Configuration Wizard.

To define and deploy a new Fusion Middleware environment, you need to:

1. Create a topology file that describes the topology you want to create.

   For more information, see How Do I Create a Topology File?

2. Use Java or an equivalent scripting language to invoke a set of standard actions that use the data in the topology file to deploy your new environment.

For example, when you are creating a new environment, you will likely use the `create-domain` standard action, which is part of the Lifecycle Management Tools Action Framework.

For more information, see How Do I Invoke the Standard Actions?.

1.2.2 Update an Existing Fusion Middleware Environment

Besides creating a new environment, you can also update an existing environment. For example, you can extend an existing Oracle Fusion Middleware domain to include additional components.

To extend or update an existing Fusion Middleware domain, you need to:

1. Create or locate a topology file that describes the topology you want to create.

   For more information, see How Do I Create a Topology File?.

2. Use Java or an equivalent scripting language to invoke a set of standard actions that use the data in the topology file to update or extend an existing Oracle Fusion Middleware domain.

   For more information, see How Do I Invoke the Standard Actions?.

1.3 Understanding the Basics

Before you begin using the Lifecycle Management Tools, you should understand some basic concepts and basic tasks you will be performing with the tools.

Terms and Concepts
: Before you begin using the Lifecycle Management Tools, you should learn about some important terms and concepts that are specific to the LCM Tools.

How Do I Create a Topology File?
: The first step in deploying a new Fusion Middleware environment is to create a topology file (XML or JSON) that defines the elements of the topology. You can then use the standard actions to deploy or manage the topology.

How Do I Invoke the Standard Actions?
: After you have identified a topology file, which defines the characteristics of the environment, you can then invoke one or more standard actions to create or modify the environment.

1.3.1 Terms and Concepts

Before you begin using the Lifecycle Management Tools, you should learn about some important terms and concepts that are specific to the LCM Tools.

As you learn about and use the Lifecycle Management Tools, you should be aware of the following terms and concepts.

Term or Concept	Definition
Topology	A description of the major components of a Fusion Middleware environment, including the hosts, Oracle homes, domains, Node Manager instances, data sources, and other external resources (including load balancers), as well as the binding information to map these components on to actual hosts (for example, physical or virtual machines). The topology definition is saved in a topology file, which can be a JSON or XML file, which can be created, edited, or modified with Fusion Middleware Composer.
Domain Profile	A description of the Managed Servers, clusters, and targeting information that make up a domain.
Credentials	The user name and password (or other credentials) for a particular component of the environment. Credentials are often required in order to perform life cycle tasks. For example, you need the domain administrator username and password to update a domain, and you need a database user name and password to install required schemas.
Tuning parameter	An optional setting for a component of the environment. For example, a tuning parameter might be a timeout on a data source.

1.3.2 How Do I Create a Topology File?

The first step in deploying a new Fusion Middleware environment is to create a topology file (XML or JSON) that defines the elements of the topology. You can then use the standard actions to deploy or manage the topology.

There are two methods for creating a topology file, as shown in the following table.

Method	Description	More Information
Start with a sample file	A sample topology file that represents a typical single-host Oracle SOA Suite domain is available for download from the Oracle Help Center. You can download this sample (or another sample like it) and edit specific values to match the environment you want to create. The Lifecycle Management Tools tutorial describes how to use this method of creating a new topology file.	Tutorial: Installing and Configuring Oracle SOA Suite Using the LCM Tools

Method	Description	More Information
Use Composer and its built-in tools to "draw" the topology from scratch.	If you don't want to use a sample file, you can create a new topology file by opening Composer and creating each element of the topology, one-by-one. Oracle recommends that you first familiarize yourself with the Composer user interface by going through the tutorial or by reviewing the online help for the various Composer features.	Tutorial: Installing and Configuring Oracle SOA Suite Using the LCM Tools Overview of FMW Composer in *FMW Composer Online Help*

1.3.3 How Do I Invoke the Standard Actions?

After you have identified a topology file, which defines the characteristics of the environment, you can then invoke one or more standard actions to create or modify the environment.

To invoke a standard action, you must use Java or a similar scripting language.

If you are familiar with Java programming, you can use the *Java API Reference for the Lifecycle Management Action Framework* to learn more about how to reference the standard actions in your Java code.

Note that the examples in this guide use Apache Groovy, which is a simple open source scripting language, that uses concepts similar to Java. See http://www.groovy-lang.org/index.html. To learn more about how to use Groovy to invoke standard actions, see the Tutorial: Installing and Configuring Oracle SOA Suite Using the LCM Tools.

2

Getting Started with the Lifecycle Management Tools

Use the Lifecycle Management Tools tutorial and the API JavaDocs to help you get started working with the tools.

Understanding the Lifecycle Management Tools
: The following topics describe the concepts and features of the Lifecycle Management Tools.

Locating the Lifecycle Management Tools in a Fusion Middleware Oracle Home
: The Lifecycle Management Tools are installed automatically into the Oracle home when you install any Oracle Fusion Middleware product. The files, depending on their purpose and type, are stored in various directories within the Oracle home.

Using the Lifecycle Management Tools Tutorial
: The most efficient way to get familiar with the Lifecycle Management Tools is to use the tutorial in this guide.

Getting Help While Using Composer
: When you start using Fusion Middleware Composer, you can get online help on each of the key Composer screens. You can use this information to learn about the properties and attributes that must be set for each element of the topology.

Using the Provisioning API JavaDocs
: The Lifecycle Management Tools Provisioning APIs are described in the JavaDoc for each API.

2.1 Locating the Lifecycle Management Tools in a Fusion Middleware Oracle Home

The Lifecycle Management Tools are installed automatically into the Oracle home when you install any Oracle Fusion Middleware product. The files, depending on their purpose and type, are stored in various directories within the Oracle home.

The following table lists the locations of the key components of the Lifecycle Management Tools inside a typical Fusion Middleware Oracle home.

File or Component	Location
Fusion Middleware Composer	*ORACLE_HOME*/oracle_common/bin/fmw-composer.sh
Supporting libraries, the Action Framework, and APIs	*ORACLE_HOME*/oracle_common/modules/fmwplatform/

2.2 Using the Lifecycle Management Tools Tutorial

The most efficient way to get familiar with the Lifecycle Management Tools is to use the tutorial in this guide.

The tutorial explains, step-by-step, how to use Apache Groovy, a sample topology file, and sample code to deploy a basic Oracle Fusion Middleware domain on a single host. The domain topology is similar to the standard installation topology described in Understanding the Oracle SOA Suite Standard Installation Topology in *Installing and Configuring Oracle SOA Suite and Business Process Management*.

The tutorial demonstrates how you can:

- Open and edit a topology file in Fusion Middleware Composer.
- Use the sample code to add the Oracle SOA Suite software to an Oracle home and deploy the sample topology.
- Review sample Apache Groovy code that invokes the create-domain standard action.

When you finish the tutorial, you can then apply the concepts to your own environment.

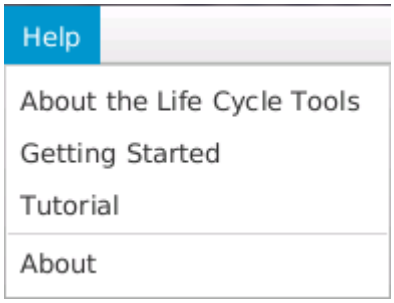
For more information, see Tutorial: Installing and Configuring Oracle SOA Suite Using the LCM Tools.

2.3 Getting Help While Using Composer

When you start using Fusion Middleware Composer, you can get online help on each of the key Composer screens. You can use this information to learn about the properties and attributes that must be set for each element of the topology.

From within Composer, you can get help in the following ways:

- From the **Help** menu, you can navigate to specific locations in the Lifecycle Management Tools documentation.



- After you have created or opened a topology file, click the Help icon in the toolbar  , and then click an object in the topology, to get help on a specific object in the topology diagram.
- After you select an object in a topology diagram, such as the host or Managed Server, you can then click the help icon in the properties panel  for information about the properties and attributes you can set for that object.

- Click the help icon in the validation pane  , at the bottom of the Composer screen, for information about how to use the validation pane to quickly address any missing information that is required for the topology.

2.4 Using the Provisioning API JavaDocs

The Lifecycle Management Tools Provisioning APIs are described in the JavaDoc for each API.

As you develop the required Java code for your environment, you can refer to the JavaDoc for specific information about each of the APIs. In addition, you can review the sample code provided as part of the Lifecycle Management Tools tutorial.

The following JavaDocs are available:

- *Java API Reference for the Lifecycle Management Action Framework*
- *Java API Reference for the Lifecycle Management Environment Specification*
- *Java API Reference for the Lifecycle Management Credential Management*

Part II

Tutorial: Installing and Configuring Oracle SOA Suite Using the LCM Tools

This tutorial shows you how to use the Lifecycle Management (LCM) Tools to install and configure Oracle SOA Suite on a single host on Oracle Linux.

In this tutorial, you will download a sample zip file that contains sample topology files, make edits to the topology files in FMW Composer, and then use Apache Groovy to invoke Lifecycle Management standard actions on the environment.

The tutorial includes the following topics:

3

About the Lifecycle Management Tools Tutorial

The Lifecycle Management Tools tutorial guides you through the steps of obtaining the sample topology and sample code from the Oracle Technology Network (OTN), editing the various elements of the topology, and configuring the topology based on the information you specify in a topology file.

Tutorial: Installing and Configuring Oracle SOA Suite Using the LCM Tools
: This tutorial shows you how to use the Lifecycle Management (LCM) Tools to install and configure Oracle SOA Suite on a single host on Oracle Linux.

Overview of the Lifecycle Management Tools Tutorial
: The Lifecycle Management Tools tutorial consists of a series of steps, where you perform some prerequisite steps and then use the Lifecycle Management Tools to automatically add Oracle SOA Suite to the Oracle home and create the Weblogic Server domain.

About the Sample Topology
: The sample topology used for this tutorial is similar to the standard installation topology used for Oracle SOA Suite. It includes a Managed Server, a cluster, and other elements of a typical domain, deployed on a single host.

About the Elements in the Sample Topology
: The sample topology includes common elements, such as a WebLogic domain, an Administration Server, and a cluster containing one Managed Server.

3.1 Overview of the Lifecycle Management Tools Tutorial

The Lifecycle Management Tools tutorial consists of a series of steps, where you perform some prerequisite steps and then use the Lifecycle Management Tools to

automatically add Oracle SOA Suite to the Oracle home and create the Weblogic Server domain.

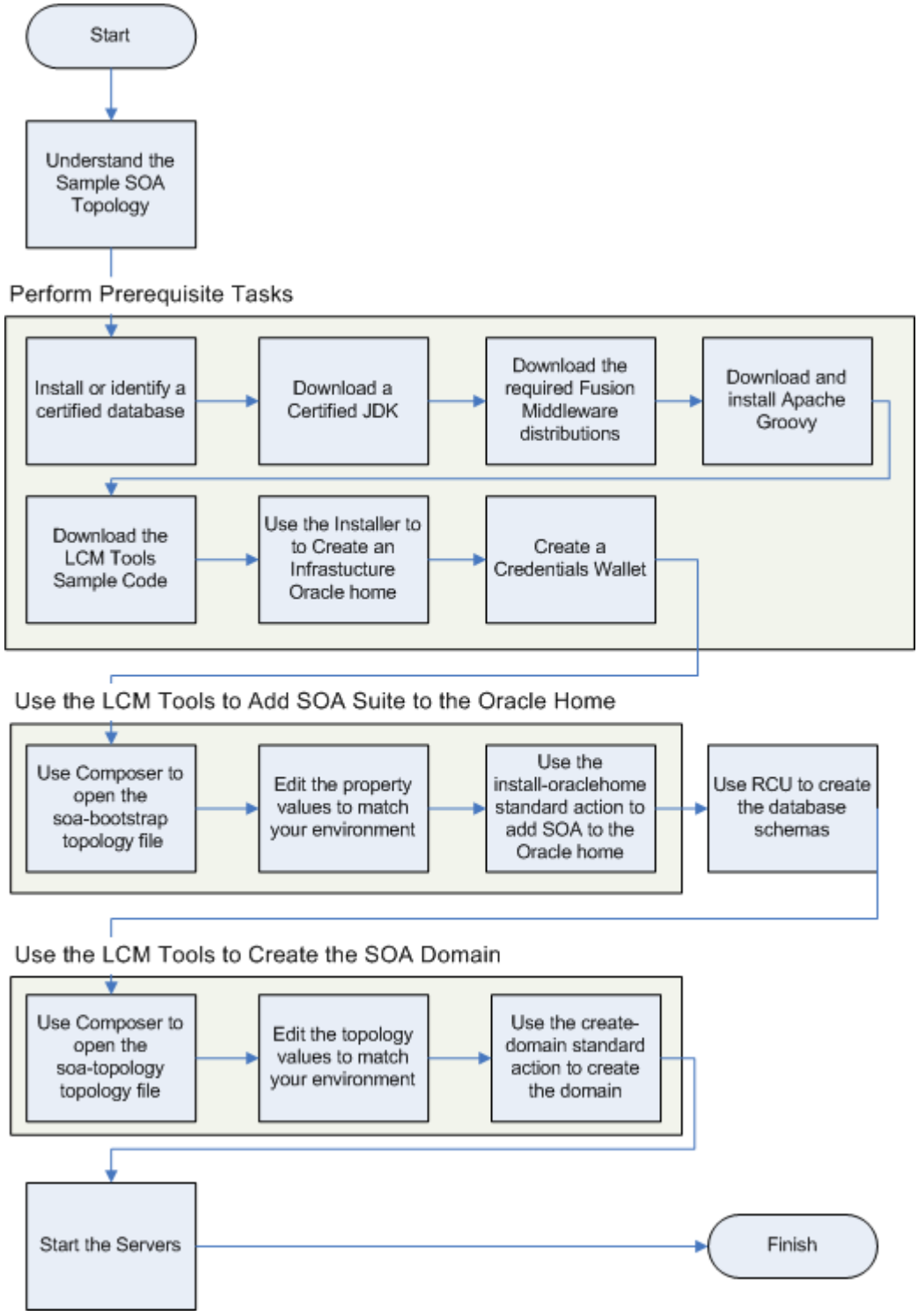
Flow Chart of the Tutorial Steps
: A flow chart is an effective way to view a summary of the tasks required to complete the tutorial.

Roadmap Table of the Tutorial Steps
: The roadmap table describes each step in the tutorial and provides links to more information.

3.1.1 Flow Chart of the Tutorial Steps

A flow chart is an effective way to view a summary of the tasks required to complete the tutorial.

The following illustration shows the structure of the Lifecycle Management Tools tutorial.


Start
Understand the Sample SOA Topology
Perform Prerequisite Tasks
Install or identify a certified database
Download a Certified JDK
Download the required Fusion Middleware distributions
Download and install Apache Groovy
Download the LCM Tools Sample Code
Use the Installer to to Create an Infrastucture Oracle home
Create a Credentials Wallet
Use the LCM Tools to Add SOA Suite to the Oracle Home
Use Composer to open the soa-bootstrap topology file
Edit the property values to match your environment
Use the install-oraclehome standard action to add SOA to the Oracle home
Use RCU to create the database schemas
Use the LCM Tools to Create the SOA Domain
Use Composer to open the soa-topology topology file
Edit the topology values to match your environment
Use the create-domain standard action to create the domain
Start the Servers
Finish

3.1.2 Roadmap Table of the Tutorial Steps

The roadmap table describes each step in the tutorial and provides links to more information.

Step	Description	More Information
Understand the Sample SOA Topology	Review the sample topology you will configure using the Lifecycle Management Tools.	About the Sample Topology
Install or Identify a Certified Database	Before you can configure an Oracle Fusion Middleware domain, you must have a supported database where you can install the required product schemas.	Installing a Certified Database
Download a Certified JDK	Oracle Fusion Middleware and the Lifecycle Management Tools require a certified JDK; otherwise you cannot install or run the product software.	Installing a Certified JDK
Download and Install Apache Groovy	You can use Java or a Java-compatible scripting language to invoke the standard actions. For this tutorial, we use Apache Groovy, a simple open-source programming language that is similar to Java.	Installing Apache Groovy
Download the LCM Tools Sample Code	For this tutorial, you will use sample topology files and sample Groovy scripts to update an Oracle home and create a new Oracle SOA Suite domain.	Downloading the Lifecycle Management Tools Sample File
Use the Installer to Create an Infrastructure Oracle home	Use the Fusion Middleware Infrastructure Installer to create the initial Oracle home. The Infrastructure distribution is a prerequisite for Oracle SOA Suite; it also includes the Lifecycle Management Tools software.	Installing Oracle Fusion Middleware Infrastructure
Create a Credentials Wallet	Before you can use the Lifecycle Management Tools, you must create a wallet, which is used to store the credentials required for the installation and configuration operations. This includes the Administration Server and database connection credentials.	Creating a Wallet
Use Composer to open the soa-bootstrap topology file	Composer shows you a diagram of the Oracle home with Oracle SOA Suite software installed.	Starting FMW Composer Opening the soa-bootstrap-topology-1.0.json File

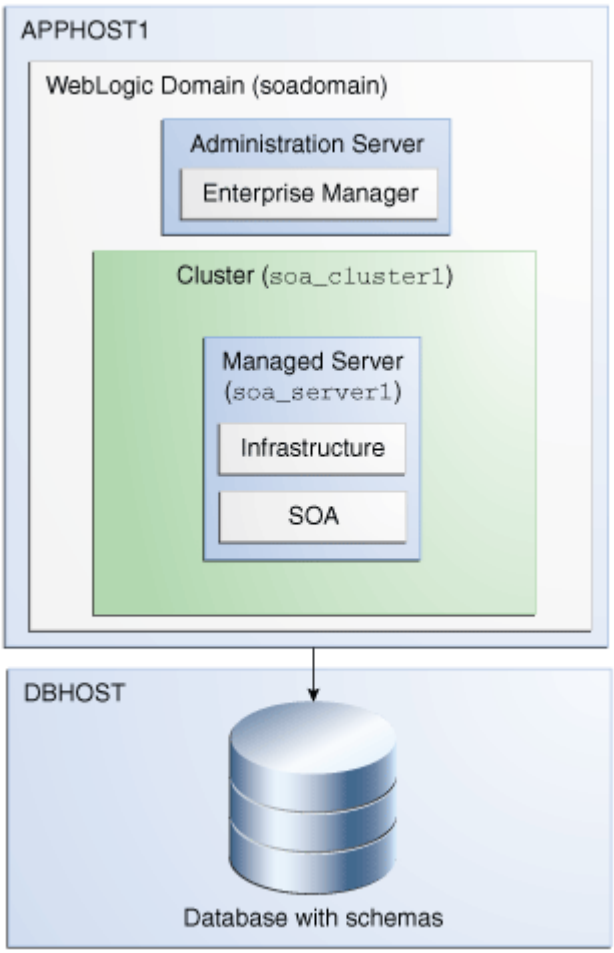
Step	Description	More Information
Edit the topology values to match your environment	In this step, you modify the properties of the Oracle home so it points to the Infrastructure Oracle home you created earlier.	Editing the soa-bootstrap-topology-1.0.json File
Use the install-oraclehome standard action to add SOA to the Oracle home	Edit the sample Groovy script so it references the path to your Oracle home, and then run the script to invoke the standard action.	Using Apache Groovy to Run the install-oraclehome Standard Action
Use RCU to create the database schemas	Use the Repository Creation Utility to create the required schemas in the database just like you would for a manual installation.	Creating the Database Schemas
Use Composer to open the soa-topology topology file	Composer displays a diagram of the SOA Suite topology.	Opening the soa-topology-1.0.json File
Edit the topology values to match your environment	You can use Composer to update the topology properties so they reference the host names, directory paths, and other characteristics of your specific environment.	Editing the soa-topology-1.0.json File
Use the create-domain standard action to create the domain	Edit the sample Groovy script so it references the path to your Oracle home, and then run the script to invoke the standard action.	Using Apache Groovy to Run the create-domain Standard Action
Start the servers	After the domain is created successfully, you can start the Administration Server and the Managed Server in the domain to verify the domain configuration.	Starting the Servers

3.2 About the Sample Topology

The sample topology used for this tutorial is similar to the standard installation topology used for Oracle SOA Suite. It includes a Managed Server, a cluster, and other elements of a typical domain, deployed on a single host.

The following diagram shows the sample Oracle SOA Suite topology and the elements in that topology.

Note that this topology is a sample topology and is not the only topology supported for the product. However, this tutorial provides specific instructions for achieving this topology.



In this topology, Oracle SOA Suite is deployed to the Managed Server.

Note that the topology also requires an external database for the required product schemas.

For information on the elements of this topology, see About the Elements in the Sample Topology.

3.3 About the Elements in the Sample Topology

The sample topology includes common elements, such as a WebLogic domain, an Administration Server, and a cluster containing one Managed Server.

The following table describes the elements of the topology illustration:

Element	Description and Links to Related Documentation
APPHOST1	A standard term used in Oracle documentation to refer to the an application server host computer.
DBHOST	A standard term used in Oracle documentation to refer to the database host computer.

Element	Description and Links to Related Documentation
WebLogic Domain	A logically related group of Java components (in this case, the Administration Server, Managed Server, and other related software components). For more information, see What is an Oracle WebLogic Server Domain? in *Understanding Oracle Fusion Middleware*.
Administration Server	Central control entity of a WebLogic domain. It maintains configuration objects for that domain and distributes configuration changes to Managed Servers. For more information, see What is the Administration Server? in *Understanding Oracle Fusion Middleware*.
Enterprise Manager	Oracle Enterprise Manager Fusion Middleware Control is a primary tool used to manage a domain. For more information, see Oracle Enterprise Manager Fusion Middleware Control in *Understanding Oracle Fusion Middleware*.
Cluster	A collection of multiple WebLogic Server instances running simultaneously and working together. For more information, see Understanding Managed Servers and Managed Server Clusters in *Understanding Oracle Fusion Middleware*.
Managed Server	A host for your applications, application components, web services, and their associated resources. For more information, see Understanding Managed Servers and Managed Server Clusters in *Understanding Oracle Fusion Middleware*.
Infrastructure	Collection of services that include the following: • Metadata repository (MDS) contains metadata for Oracle Fusion Middleware components, such as the Oracle Application Developer Framework. For more information, see What is the Metadata Repository? in *Understanding Oracle Fusion Middleware*. • Oracle Application Developer Framework (Oracle ADF) • Oracle Web Services Manager (OWSM)

4

Performing Prerequisite Tasks

There are several prerequisite tasks you need to complete to prepare your environment for running the examples in this tutorial.

Before you begin, complete the following tasks:

Tutorial: Installing and Configuring Oracle SOA Suite Using the LCM Tools
: This tutorial shows you how to use the Lifecycle Management (LCM) Tools to install and configure Oracle SOA Suite on a single host on Oracle Linux.

Installing a Certified Database
: If you do not already have a database where you can install schemas, you must install and configure a certified database.

Installing a Certified JDK
: Before you can use the Lifecycle Management Tools, you must download and install a certified Java Development Kit (JDK).

Obtaining the Oracle Fusion Middleware Software
: To install the Oracle SOA Suite sample topology, you need to obtain the necessary Oracle Fusion Middleware software.

Installing Apache Groovy
: After you have a JDK installed, download and install the latest version of Apache Groovy. Groovy is an open source tool that you will use to invoke Lifecycle Management standard actions and configure the sample topology.

Downloading the Lifecycle Management Tools Sample File
: The Lifecycle Management Tools sample file contains sample topology files, which you can edit using FMW Composer, and sample Apache Groovy code that will be used to invoke the standard actions.

Installing Oracle Fusion Middleware Infrastructure
: Using the Infrastructure distribution and JDK you downloaded, install Infrastructure to create an Oracle home. When you install Infrastructure, the Lifecycle Management Tools are automatically installed into the Oracle home.

4.1 Installing a Certified Database

If you do not already have a database where you can install schemas, you must install and configure a certified database.

For a list of certified databases that you can use, refer to the certification document for your release, located on the *Oracle Fusion Middleware Supported Systems Configuration* page.

4.2 Installing a Certified JDK

Before you can use the Lifecycle Management Tools, you must download and install a certified Java Development Kit (JDK).

You should always verify the required JDK version by reviewing the certification information on the *Oracle Fusion Middleware Supported Systems Configuration* page.

To download the required JDK, use your browser to navigate to the following URL and download the Java SE JDK:

http://www.oracle.com/technetwork/java/javase/downloads/index.html

4.3 Obtaining the Oracle Fusion Middleware Software

To install the Oracle SOA Suite sample topology, you need to obtain the necessary Oracle Fusion Middleware software.

To create the Oracle SOA Suite topology, download the following software distributions:

- Oracle Fusion Middleware 12*c* (12.2.1.1.0) Infrastructure
- Oracle Fusion Middleware 12*c* (12.2.1.1.0) SOA Suite and Business Process Management

For specific information on the distributions you need to download, see the *Oracle Fusion Middleware Download, Installation, and Configuration Readme Files* on Oracle Technology Network (OTN).

4.4 Installing Apache Groovy

After you have a JDK installed, download and install the latest version of Apache Groovy. Groovy is an open source tool that you will use to invoke Lifecycle Management standard actions and configure the sample topology.

You can download the Groovy distribution from the following location:

http://groovy-lang.org/install.html

4.5 Downloading the Lifecycle Management Tools Sample File

The Lifecycle Management Tools sample file contains sample topology files, which you can edit using FMW Composer, and sample Apache Groovy code that will be used to invoke the standard actions.

To obtain the sample file for this tutorial:

1. Download the `create-domain-example.zip` file on Oracle Technology Network (OTN) to a directory on your system.
2. Open a new terminal window on your host.
3. Change directory to the directory where you downloaded the `create-domain-example.zip` file.
4. Unzip the `create-domain-example.zip` file in the current directory.
5. Verify that a folder named `create-domain-example` was created in the current directory, and note the path to this directory on your system.

4.6 Installing Oracle Fusion Middleware Infrastructure

Using the Infrastructure distribution and JDK you downloaded, install Infrastructure to create an Oracle home. When you install Infrastructure, the Lifecycle Management Tools are automatically installed into the Oracle home.

To install Oracle Fusion Middleware Infrastructure:

1. Log in to your host.
2. Go to the directory where you downloaded the installation program.
3. Launch the installation program by running the `java` executable from the JDK directory on your system, as shown in the following example:

   ```
   /home/Oracle/products/jdk1.8.0_77/bin/java -jar
   fmw_12.2.1.1.0_infrastructure_generic.jar
   ```

   Replace the JDK location in this example with the actual JDK location on your system.
4. On UNIX operating systems, the Installation Inventory Setup screen appears if this is your first Oracle installation on this host. Use this screen to specify the location of the Oracle central inventory directory.

   Make sure that the operating system group name selected has write permissions to the central inventory location. For more information, see Understanding the Oracle Central Inventory in *Installing Software with the Oracle Universal Installer*.
5. On the Welcome screen, click **Next**.
6. The Auto Updates screen allows you to search for the latest software updates, including important security updates, via your My Oracle Support account. To skip software updates, select **Skip Auto Updates**.
7. On the Installation Location screen, specify your Oracle home directory location.

   For more information about selecting an Oracle home directory, see Understanding Directories for Installation and Configuration in *Planning an Installation of Oracle Fusion Middleware*.
8. On the Installation Type screen, select **Fusion Middleware Infrastructure**.
9. Use the Prerequisite Checks screen to verify that your system meets the minimum system requirements.
10. If you have an Oracle Support account, use the Security Updates screen to indicate how you want to receive security updates.

    If you do not have an account and are sure you want to skip this step, clear the check box and verify your selection in the follow-up dialog box.
11. Verify the installation information on the Installation Summary Screen, and then click **Install** to begin installation.
12. When the installation progress bar reaches 100%, click **Next**.
13. Review the information on the Installation Complete screen, and then click **Finish** to dismiss the installer.

5

Using the Wallet Manager Tool to Set Up a Wallet

To perform lifecycle operations using the LCM tools, password credentials must be stored in a wallet. As a result, you need to set up an Oracle wallet that contains the required credentials to run the create-domain action, such as the user name and password used for connecting to the database.

To do this, you will use the Wallet Manager tool.

5.1 About the Wallet Manager Tool

You can use the Wallet Manager tool to create and manage the password credentials required to perform various lifecycle operations.

The Wallet Manager tool provides a simple command-line interface for building and maintaining an Oracle Wallet file. Depending on your security needs, you can either create an encrypted wallet or an auto-login one.

With this tool, you can:

- Create a new wallet.
- Add new credentials to a wallet.
- Update existing credentials in a wallet.
- Remove existing credentials from a wallet.

This tutorial shows you how to create a new, encrypted wallet. See Creating a Wallet.

If you are familiar with Java programming, you can also use the *Java API Reference for the Lifecycle Management Credential Management* to learn more about how to use the APIs to create a wallet.

5.2 Creating a Wallet

This example shows you how to use the Wallet Manager tool to create a wallet and add the necessary password credentials to that wallet.

For this tutorial, use the `create-wallet.sh` script in the `create-domain-example` directory to set up a wallet that will contain the credentials referenced by the sample topology.

To create a wallet:

1. Change directory to the `create-domain-example` directory.

   `cd` *`path_to_example_directory`*`/create-domain-example`

2. Open the `create-wallet.sh` file, and specify values for the `-password` option for each credential.

   Use the following table to define the `-password` option for the database user, the WebLogic administrator user, and for each schema user in the `create-wallet.sh` file:

   Make a note of the passwords you define here; you will need them later on during the domain creation process.

-alias	-user	-password
`db1/dba`	Database user name (the default user name on Oracle databases is `sys`)	database user password
`soa/wlsadmin`	WebLogic administrator user name (default is `weblogic`)	WebLogic administrator user password
`db1/soa-infra`	`${SCHEMA_PREFIX}_SOAINFRA`	SOAINFRA schema password
`db1/opss`	`${SCHEMA_PREFIX}_OPSS`	OPSS schema password
`db1/iau`	`${SCHEMA_PREFIX}_IAU`	IAU schema password
`db1/iau-append`	`${SCHEMA_PREFIX}_IAU_APPEND`	IAU_APPEND schema password
`db1/iau-viewer`	`${SCHEMA_PREFIX}_IAU_VIEWER`	IAU_VIEWER schema password
`db1/ums`	`${SCHEMA_PREFIX}_UMS`	UMS schema password
`db1/mds`	`${SCHEMA_PREFIX}_MDS`	MDS schema password
`db1/wls`	`${SCHEMA_PREFIX}_WLS`	WLS schema password
`db1/wls-runtime`	`${SCHEMA_PREFIX}_WLS_RUNTIME`	WLS_RUNTIME schema password
`db1/stb`	`${SCHEMA_PREFIX}_STB`	STB schema password

The following shows an example of the `create-wallet.sh` file:

```
"$JAVA" "$JRE_OPTIONS" -cp "$JRE_CP"
oracle.fmwplatform.credentials.wallet.WalletManager add -walletDirectory .
-alias db1/dba -user sys -password database_password -walletPassword $WALLET

"$JAVA" "$JRE_OPTIONS" -cp "$JRE_CP"
oracle.fmwplatform.credentials.wallet.WalletManager add -walletDirectory .
-alias soa/wlsadmin -user weblogic -password weblogic_password  -walletPassword
$WALLET

"$JAVA" "$JRE_OPTIONS" -cp "$JRE_CP"
oracle.fmwplatform.credentials.wallet.WalletManager add -walletDirectory .
-alias db1/soa-infra -user ${SCHEMA_PREFIX}_SOAINFRA -password soainfra_password
-walletPassword $WALLET

"$JAVA" "$JRE_OPTIONS" -cp "$JRE_CP"
oracle.fmwplatform.credentials.wallet.WalletManager add -walletDirectory .
-alias db1/opss -user ${SCHEMA_PREFIX}_OPSS -password opss_password -
walletPassword $WALLET

"$JAVA" "$JRE_OPTIONS" -cp "$JRE_CP"
oracle.fmwplatform.credentials.wallet.WalletManager add -walletDirectory .
-alias db1/iau -user ${SCHEMA_PREFIX}_IAU -password iau_password -walletPassword
$WALLET

"$JAVA" "$JRE_OPTIONS" -cp "$JRE_CP"
oracle.fmwplatform.credentials.wallet.WalletManager add -walletDirectory .
-alias db1/iau-append -user ${SCHEMA_PREFIX}_IAU_APPEND -password
iau_append_password -walletPassword $WALLET

"$JAVA" "$JRE_OPTIONS" -cp "$JRE_CP"
oracle.fmwplatform.credentials.wallet.WalletManager add -walletDirectory .
-alias db1/iau-viewer -user ${SCHEMA_PREFIX}_IAU_VIEWER -password
iau_viewer_password -walletPassword $WALLET

"$JAVA" "$JRE_OPTIONS" -cp "$JRE_CP"
oracle.fmwplatform.credentials.wallet.WalletManager add -walletDirectory .
-alias db1/ums -user ${SCHEMA_PREFIX}_UMS -password ums_password -walletPassword
$WALLET

"$JAVA" "$JRE_OPTIONS" -cp "$JRE_CP"
oracle.fmwplatform.credentials.wallet.WalletManager add -walletDirectory .
-alias db1/mds -user ${SCHEMA_PREFIX}_MDS -password mds_password -walletPassword
$WALLET

"$JAVA" "$JRE_OPTIONS" -cp "$JRE_CP"
oracle.fmwplatform.credentials.wallet.WalletManager add -walletDirectory .
-alias db1/wls -user ${SCHEMA_PREFIX}_WLS -password wls_password -walletPassword
$WALLET

"$JAVA" "$JRE_OPTIONS" -cp "$JRE_CP"
oracle.fmwplatform.credentials.wallet.WalletManager add -walletDirectory .
-alias db1/wls-runtime -user ${SCHEMA_PREFIX}_WLS_RUNTIME -password
wls_runtime_password -walletPassword $WALLET

"$JAVA" "$JRE_OPTIONS" -cp "$JRE_CP"
oracle.fmwplatform.credentials.wallet.WalletManager add -walletDirectory .
-alias db1/stb -user ${SCHEMA_PREFIX}_STB -password stb_password -walletPassword
$WALLET
```

3. Save and close the file.

4. Set the ORACLE_HOME environment variable to the full path of the Oracle home where you installed Fusion Middleware Infrastructure.

   For example:

   ```
   export ORACLE_HOME=/home/Oracle/products/fmw12211
   ```

5. Set the JAVA_HOME environment variable to the path of a certified JDK.

   For example:

   ```
   export JAVA_HOME=/home/Oracle/products/jdk1.8.0_77
   ```

6. Change directory to the `create-domain-example/mywallet` directory.

   ```
   cd path_to_example_directory/create-domain-example/mywallet
   ```

   Remove any existing wallets in this directory. This directory is empty by default.

7. Run the following command to create a wallet in the `mywallet` directory with the credentials you defined in step 2. Ensure that you have execute permissions if not already.

   ```
   sh ../create-wallet.sh FMWEX -walletPassword wallet_password
   ```

   Where:

   - `FMWEX` is the schema prefix that will be used to create the SOA schemas for the domain and will be referenced throughout this tutorial.

     If you choose a different prefix, replace `FMWEX` with the value of your actual prefix. Make a note of the prefix you enter here; you will need to make sure to update the **Prefix** fields for all the schemas when you edit the `soa-topology-1.0.json` file later.

   - *`wallet_password`* is the wallet password.

You should now have a wallet in the `create-domain-example/mywallet` directory that contains the database, WebLogic administrator, and schema user credentials.

6

Using the Lifecycle Management Tools to Install Oracle SOA Suite

Follow these steps to use Apache Groovy to run the install-oraclehome standard action, which will install Oracle SOA Suite into the Infrastructure Oracle home you created.

Tutorial: Installing and Configuring Oracle SOA Suite Using the LCM Tools
: This tutorial shows you how to use the Lifecycle Management (LCM) Tools to install and configure Oracle SOA Suite on a single host on Oracle Linux.

Starting FMW Composer
: Start FMW Composer from the *ORACLE_HOME*/oracle_common/bin directory.

Opening the soa-bootstrap-topology-1.0.json File
: After you start FMW Composer, open the sample soa-bootstrap-topology-1.0.json file. This file is located in the *path_to_example_directory*/create-domain-example/models/topologies directory.

Editing the soa-bootstrap-topology-1.0.json File
: After opening the soa-bootstrap-topology-1.0.json file in Composer, you need to specify the SOA Suite Oracle home location, the path to a certified JDK, and the location of the SOA Suite installer jar file.

Using Apache Groovy to Run the install-oraclehome Standard Action
: After editing the soa-boostrap-topology-1.0.json file, use Apache Groovy to run the create-oracle-home.groovy script and execute the install-oraclehome action.

6.1 Starting FMW Composer

Start FMW Composer from the *ORACLE_HOME*/oracle_common/bin directory.

1. Set the JAVA_HOME environment variable to the path of a certified JDK.

   For example:

   ```
   export JAVA_HOME=/home/Oracle/products/jdk1.8.0_77
   ```

2. Change directory to the *ORACLE_HOME*/oracle_common/bin directory.

   ```
   cd ORACLE_HOME/oracle_common/bin
   ```

3. Start FMW Composer (fmw-composer.sh).

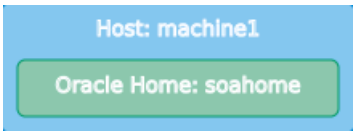
   ```
   ./fmw-composer.sh
   ```

6.2 Opening the soa-bootstrap-topology-1.0.json File

After you start FMW Composer, open the sample `soa-bootstrap-topology-1.0.json` file. This file is located in the *`path_to_example_directory`*`/create-domain-example/models/topologies` directory.

1. From the **File** menu, select **Open File...**
2. In the **Open Composer File** dialog box, navigate to the *`path_to_example_directory`*`/create-domain-example/models/topologies` directory.
3. Select **soa-bootstrap-topology-1.0.json** and click **Open**.

   When you open the file, you should see the following topology diagram on the screen:



6.3 Editing the soa-bootstrap-topology-1.0.json File

After opening the `soa-bootstrap-topology-1.0.json` file in Composer, you need to specify the SOA Suite Oracle home location, the path to a certified JDK, and the location of the SOA Suite installer jar file.

1. Select the **Oracle Home: soahome** box in the Composer topology diagram.

   This will display a panel on the right side of the screen with fields where you can specify information about the SOA Suite Oracle home.
2. In the **Path** field, enter the full path to the Oracle home where Oracle SOA Suite will be installed. This Oracle home should already contain Fusion Middleware Infrastructure.

   For example:

   ```
   /home/Oracle/products/fmw12211
   ```
3. In the **Java Home** field, enter the path to a certified JDK.

   For example:

   ```
   /home/Oracle/products/jdk1.8.0_77
   ```
4. Under **Installations**, select the **infra (Fusion Middleware Infrastructure)** entry and click  to remove this entry, as Infrastructure is already installed in this Oracle home.
5. Under **Installations**, select the **soa (SOA Suite)** entry and click  to edit the installation data for Oracle SOA Suite:
   a. For the **ID** field, keep the default value, **soa**.
   b. For the **Install Type** field, keep the default value, **SOA Suite**.

c. In the **Installer** field, specify the full path and file name of the Oracle SOA Suite installer jar file (`fmw_12.2.1.1.0_soa_generic.jar`) you downloaded (see Obtaining the Oracle Fusion Middleware Software):

```
path_to_soa_jar_file/fmw_12.2.1.1.0_soa_generic.jar
```

d. Click **OK**.

6. Select **File** and then **Save** to save the changes made to the file.

7. After saving the file, select **File** and then **Exit** to exit Composer.

6.4 Using Apache Groovy to Run the install-oraclehome Standard Action

After editing the `soa-boostrap-topology-1.0.json` file, use Apache Groovy to run the `create-oracle-home.groovy` script and execute the install-oraclehome action.

The `create-oracle-home.groovy` file is located in the `create-domain-example` directory.

To run the action:

1. Change directory to the `create-domain-example` directory.

```
cd path_to_example_directory/create-domain-example
```

2. In the `call-oh.sh` file, update the Oracle home location to point to the SOA Suite Oracle home.

   Note that there are two places in this file where you will need to update the Oracle home location, as shown in the following example:

```
#!/bin/bash

groovy -cp "ORACLE_HOME/oracle_common/modules/features/
oracle.fmwplatform.envspec_lib.jar:
ORACLE_HOME/oracle_common/modules/features/oracle.fmwplatform.common_lib.jar" $*
```

   In this example, replace *ORACLE_HOME* with the actual path to your Oracle home directory.

3. In the `create-oracle-home.groovy` file, replace `secret1@` with the actual password of the wallet you created, as shown in the following example:

```
"wallet_password".toCharArray()    // the wallet passphrase
```

4. Set the GROOVY_HOME variable to the directory where you installed the Apache Groovy distribution, as shown in the following example:

```
export GROOVY_HOME=/home/Oracle/groovy-2.4.6
export PATH=$PATH:$GROOVY_HOME/bin
```

5. Run the following command from the `create-domain-example` directory to execute the action:

```
sh call-oh.sh create-oracle-home.groovy
```

   Ensure that you have execute permissions if not already.

If the action is successful, the exit status code will be SUCCESS, as shown in the example below:

```
May 16, 2016 11:34:42 AM
oracle.fmwplatform.actionframework.v2.internal.ActionBase setInitializedState
INFO: Action install-oraclehome changed from state CREATED to state NOT_READY
May 16, 2016 11:34:42 AM
oracle.fmwplatform.actionframework.v2.internal.ActionBase setCanInvokeState
INFO: Action install-oraclehome changed from state NOT_READY to state READY
May 16, 2016 11:41:40 AM
oracle.fmwplatform.actionframework.v2.internal.AbstractActionImpl logActionResult
INFO: SUCCESS: OUI successful return code: 0
May 16, 2016 11:43:28 AM
oracle.fmwplatform.actionframework.v2.internal.AbstractActionImpl logActionResult
INFO: SUCCESS: OUI successful return code: 0
May 16, 2016 11:43:28 AM
oracle.fmwplatform.actionframework.v2.internal.ActionBase setInvokedState
INFO: Action install-oraclehome changed from state READY to state NOT_READY
May 16, 2016 11:43:28 AM
oracle.fmwplatform.actionframework.v2.internal.AbstractActionImpl logActionResult
INFO: SUCCESS: OUI successful return code: 0
May 16, 2016 11:43:28 AM
oracle.fmwplatform.actionframework.v2.internal.ActionBase setClosedState
INFO: Action install-oraclehome changed from state NOT_READY to state DESTROYED
May 16, 2016 11:43:28 AM
oracle.fmwplatform.actionframework.v2.internal.ActionBase setDoneState
INFO: Action install-oraclehome changed from state DESTROYED to state DONE
{
  "actionResultList" : [ {
    "statusCode" : "SUCCESS",
    "statusDetail" : "OUI successful return code: 0"
  } ]
}
```

At this point, you should have an Oracle home that contains both Oracle Fusion Middleware Infrastructure and Oracle SOA Suite.

7

Creating the Database Schemas

Before you configure the Oracle SOA Suite domain, use the Repository Creation Utility (RCU) to install the required schemas on a certified database.

Tutorial: Installing and Configuring Oracle SOA Suite Using the LCM Tools
: This tutorial shows you how to use the Lifecycle Management (LCM) Tools to install and configure Oracle SOA Suite on a single host on Oracle Linux.

Starting the Repository Creation Utility (RCU)
: After you have installed Oracle SOA Suite, start the Repository Creation Utility (RCU) from the *ORACLE_HOME*/oracle_common/bin directory to create the SOA Suite schemas for the domain.

Navigating the RCU Screens to Create the Schemas
: Use the RCU screens to create the database schemas.

7.1 Starting the Repository Creation Utility (RCU)

After you have installed Oracle SOA Suite, start the Repository Creation Utility (RCU) from the *ORACLE_HOME*/oracle_common/bin directory to create the SOA Suite schemas for the domain.

To start RCU on Linux operating systems:

```
cd ORACLE_HOME/oracle_common/bin
./rcu
```

7.2 Navigating the RCU Screens to Create the Schemas

Use the RCU screens to create the database schemas.

7.2.1 Introducing RCU

The Welcome screen is the first screen that appears when you start RCU.

Click **Next**.

7.2.2 Selecting a Method of Schema Creation

Use the Create Repository screen to select a method to create and load component schemas into the database.

On the Create Repository screen:

- If you have the necessary permission and privileges to perform DBA activities on your database, select **System Load and Product Load**. This procedure assumes that you have SYSDBA privileges.
- If you do *not* have the necessary permission or privileges to perform DBA activities in the database, you must select **Prepare Scripts for System Load** on this screen. This option generates a SQL script that you can give to your database administrator. See Understanding System Load and Product Load in *Creating Schemas with the Repository Creation Utility*.
- If the DBA has already run the SQL script for System Load, select **Perform Product Load**.

7.2.3 Providing Database Connection Details

On the Database Connection Details screen, provide the database connection details for RCU to connect to your database.

Click **Next** to proceed, then click **OK** on the dialog window to confirm that connection to the database was successful.

7.2.4 Specifying a Custom Prefix and Selecting Schemas

On the Select Components screen, specify a prefix to use for the SOA Infrastructure schema and dependent schemas.

1. Select **Create new prefix**, and then enter `FMWEX` for your prefix. This is the same prefix you used when you created and added the schema credentials to the wallet (see Creating a Wallet).

   Note that `FMWEX` is the schema prefix used for this tutorial. If you used a different prefix, make sure to replace `FMWEX` with the value of your actual prefix.

   Note:

   Make a note of the prefix you enter here; you will need this later on during the domain creation process.

   The custom prefix logically groups these schemas together for use in this domain only. For more information about custom prefixes, see Understanding Custom Prefixes in *Creating Schemas with the Repository Creation Utility*.

   ---

2. Select the **SOA Suite** schema. This will automatically select SOA Infrastructure, along with the following schemas as dependencies:

   - User Messaging Service
   - Metadata Services
   - WebLogic Services
   - Oracle Platform Security Services
   - Audit Services
   - Audit Services Append
   - Audit Services Viewer

   A schema called Common Infrastructure Services is also automatically created; this schema is grayed out (you can't select it or deselect it). For more information, see Understanding the Service Table Schema in *Creating Schemas with the Repository Creation Utility*.

   For more information about how to organize your schemas in a multi-domain environment, see Planning Your Schema Creation in *Creating Schemas with the Repository Creation Utility*.

3. Click **Next** to proceed, then click **OK** to confirm that prerequisite checking for schema creation was successful.

7.2.5 Specifying Schema Passwords

On the Schema Passwords screen, specify how you want to set the schema passwords on your database, then enter and confirm your passwords.

On this screen, enter the same schema passwords that you created and added to the wallet (see Creating a Wallet).

Tip:

You must make a note of the passwords you set on this screen; you will need them later on during the domain creation process.

7.2.6 Specifying Custom Variables

On the Custom Variables screen, specify the custom variables for the SOA Infrastructure schema.

For the Oracle SOA Suite sample topology, accept both default values for **Database Profile** (**Small**) and **Healthcare Integration** (**No**).

For more information, see About the Custom Variables Required for the SOA Suite Schemas in *Installing and Configuring Oracle SOA Suite and Business Process Management*.

Tip:

For more information about the options on this screen, see Custom Variables in *Creating Schemas with the Repository Creation Utility*.

7.2.7 Verifying Tablespace Information

The Map Tablespaces screen shows the default and temporary tablespaces for the schemas you are about to create.

For this tutorial, review the information on this screen. If you don't want to make any changes on this screen, click **Next** to accept the default values.

Click **OK** on the Confirmation dialog box to begin tablespace creation. When the tablespaces are created, click **OK** to dismiss the progress window and continue to the Summary screen.

7.2.8 Completing Schema Creation

On the Summary screen, verify your information, then click **Create** to begin schema creation.

When you reach the Completion Summary screen, click **Close** to dismiss RCU.

8

Using the Lifecycle Management Tools to Configure the SOA Suite Domain

After installing Oracle SOA Suite and creating the database schemas, follow these steps to use Apache Groovy to run the create-domain standard action and create the domain.

Tutorial: Installing and Configuring Oracle SOA Suite Using the LCM Tools
: This tutorial shows you how to use the Lifecycle Management (LCM) Tools to install and configure Oracle SOA Suite on a single host on Oracle Linux.

Starting FMW Composer
: Start FMW Composer from the *ORACLE_HOME*/oracle_common/bin directory.

Opening the soa-topology-1.0.json File
: After you start FMW Composer, open the sample soa-topology-1.0.json file. This file is located in the *path_to_example_directory*/create-domain-example/models/topologies directory.

Editing the soa-topology-1.0.json File
: After opening the soa-topology-1.0.json file in Composer, you need to specify information about the elements in the topology.

Validating and Saving the Topology File
: If there any errors found in the topology file, these errors will be displayed in the validation panel at the bottom of the screen. Correct them before continuing.

Using Apache Groovy to Run the create-domain Standard Action
: After editing and validating the sample soa-topology-1.0.json file, use Apache Groovy to run the create-domain.groovy script and execute the create-domain action.

Starting the Servers
: To verify that your domain is properly configured, manually start the Administration Server and Managed Server.

8.1 Starting FMW Composer

Start FMW Composer from the *ORACLE_HOME*/oracle_common/bin directory.

1. Set the JAVA_HOME environment variable to the path of a certified JDK.

   For example:

   ```
   export JAVA_HOME=/home/Oracle/products/jdk1.8.0_77
   ```

2. Change directory to the *ORACLE_HOME*/oracle_common/bin directory.

   ```
   cd ORACLE_HOME/oracle_common/bin
   ```

3. Start FMW Composer (fmw-composer.sh).

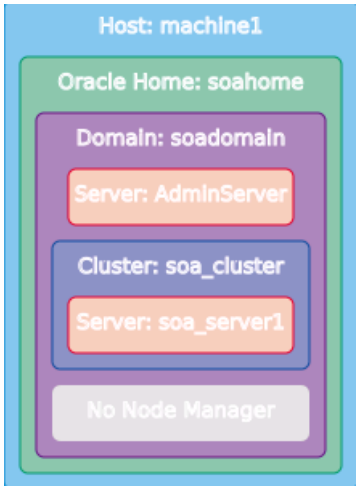
   ```
   ./fmw-composer.sh
   ```

8.2 Opening the soa-topology-1.0.json File

After you start FMW Composer, open the sample soa-topology-1.0.json file. This file is located in the *path_to_example_directory*/create-domain-example/models/topologies directory.

1. From the **File** menu, select **Settings** to open the Composer Settings page.
2. On the Composer Settings page, specify the location of the Oracle SOA Suite Oracle home.
   a. Click **select** next to **Oracle Home**.
   b. In the **Select an Oracle Home** dialog box, select the Oracle home directory and click **Open**.
3. After specifying the Oracle home location, specify the location of the wallet you created in the *path_to_example_directory*/create-domain-example/mywallet directory.
   a. Click **change** next to **Wallet**.
   b. In the **Change wallet** dialog box, select **Select an existing wallet** and click **OK**.
   a. In the **Select a Wallet Directory** dialog box, select the *path_to_example_directory*/create-domain-example/mywallet directory and click **Open**.
   b. Enter the wallet password and click **OK**.
4. Click **OK** to close the Settings page.
5. From the **File** menu, select **Open File...**
6. In the **Open Composer File** dialog box, navigate to the *path_to_example_directory*/create–domain–example/models/topologies directory.
7. Select **soa-topology-1.0.json** and click **Open**.

   When you open the file, you should see the following topology diagram on the screen:



8.3 Editing the soa-topology-1.0.json File

After opening the `soa-topology-1.0.json` file in Composer, you need to specify information about the elements in the topology.

Complete the following tasks to fill in the topology file with values specific to your environment:

Editing the Host Address
: Enter the host address for the host in the Address field.

Editing the Oracle Home Element
: Specify information about the SOA Suite Oracle home directory, such as the Oracle home and Java JDK locations.

Editing the Domain Element
: Specify information about the SOA Suite domain, such as the directory locations for where the domain and applications will be created.

Editing the Administration Server Element
: Specify the listen address and listen port to be used by the WebLogic Administration Server.

Editing the Managed Server Element
: Specify the listen address and listen port to be used by the Managed Server.

Editing the Schema Information
: Use the **Data Sources** tab in Composer to edit and verify the schema and data source information for the topology.

Editing the Database URL
: Specify the connection information for the database in which you created the schemas.

8.3.1 Editing the Host Address

Enter the host address for the host in the Address field.

Select the **Host: machine1** box in the Composer topology diagram. This will display a panel on the right side of the screen with fields where you can specify information about the host.

In the **Address** field, enter the primary IP address or DNS name of this host. The default value is `127.0.0.1`.

8.3.2 Editing the Oracle Home Element

Specify information about the SOA Suite Oracle home directory, such as the Oracle home and Java JDK locations.

1. Select the **Oracle home: soahome** box in the Composer topology diagram.

   This will display a panel on the right side of the screen with fields where you can specify information about the Oracle home.

2. In the **Path** field, enter the full path to the Oracle home where you installed Oracle SOA Suite.

   For example:

   ```
   /home/Oracle/products/fmw12211
   ```

3. In the **Java Home** field, enter the path to a certified JDK.

   For example:

   ```
   /home/Oracle/products/jdk1.8.0_77
   ```

8.3.3 Editing the Domain Element

Specify information about the SOA Suite domain, such as the directory locations for where the domain and applications will be created.

1. Select the **Domain: soadomain** box in the Composer topology diagram.

   This will display a panel on the right side of the screen with fields where you can specify information about the domain you will create.

2. In the **Path** field, enter the full path to the domain home directory. This is the directory where the domain will be created and will be referred to as *DOMAIN_HOME* in this guide.

   For example:

   ```
   /home/Oracle/config/domains/soadomain
   ```

   Oracle recommends that you place your domain home outside of the Oracle home directory. For more information on selecting a domain home, see About the Domain Home Directory in *Planning an Installation of Oracle Fusion Middleware*

   Make a note of the domain location; you need it later to access the scripts that start the servers.

3. In the **Application Path** field, enter the full path to the application home directory. This is the directory where applications associated with the domain will be created and will be referred to as *APPLICATION_HOME* in this guide.

   For example:

   ```
   /home/Oracle/config/applications/soadomain
   ```

   Oracle recommends that you place your application home outside of the Oracle home directory. For more information on selecting an application home, see About

the Application Home Directory in *Planning an Installation of Oracle Fusion Middleware*.

4. In the **Admin Server Url** field, enter the URL that will be used to connect to the domain's Administration Server. Use the following format:

   `http://adminserver_host:adminserver_port`

   Make a note of the Administration Server URL; you need it later to access the Administration Server.

8.3.4 Editing the Administration Server Element

Specify the listen address and listen port to be used by the WebLogic Administration Server.

1. Select the **Server: AdminServer** box in the Composer topology diagram.
2. On the right side of the screen, verify that the **Is Admin Server** option is checked (checked by default).
3. In the **Name** field, enter a name for the Administration Server. The default name is `AdminServerBinding`.
4. In the **Listen Address** field, specify the IP address or DNS name of the host where the Administration Server will reside. The default value is `127.0.0.1`.
5. In the **Listen Port** field, enter the port number to be used by the Administration Server. The default port is `7001`.

8.3.5 Editing the Managed Server Element

Specify the listen address and listen port to be used by the Managed Server.

1. Select the **Server: soa_server1** box in the Composer topology diagram.
2. On the right side of the screen, verify that the **Is Admin Server** option is unchecked (unchecked by default).
3. In the **Name** field, enter a name for the Managed Server. The default name is `soa_server1Binding`.
4. In the **Listen Address** field, specify the IP address or DNS name of the host where the Managed Server will reside. The default value is `127.0.0.1`.
5. In the **Listen Port** field, enter the port number to be used by the Managed Server. The default port is `7002`.

8.3.6 Editing the Schema Information

Use the **Data Sources** tab in Composer to edit and verify the schema and data source information for the topology.

1. Navigate to the **Data Sources** tab, and use the following table to verify the **Prefix** fields for each of the schema components.

   Note that `FMWEX` is the schema prefix used for this tutorial. If you used a different prefix, make sure to replace `FMWEX` with the value of your actual prefix in each of the **Prefix** fields.

Component	Credential	Prefix
IAU	db1/iau	FMWEX
IAU_APPEND	db1/iau-append	FMWEX
IAU_VIEWER	db1/iau-viewer	FMWEX
MDS	db1/mds	FMWEX
OPSS	db1/opss	FMWEX
SOAINFRA	db1/soa-infra	FMWEX
STB	db1/stb	FMWEX
UCSUMS	db1/ums	FMWEX
WLS	db1/wls	FMWEX

2. The dummy schema component groups data sources that are not associated with a particular schema in their templates. Similar to the other components, you will need to assign a database to this component as well as credentials to run the action.

   a. Select **Component: dummy**, and on the right side of the screen, click **select** next to the **Credential** field to assign credentials to this component.

      A dialog box appears that lists the credentials in the wallet. You can assign an existing credential from the list, or click **New** to add a new one to the wallet for this component.

   b. To assign a database, right-click **Component: dummy** on the screen and select **Assign Existing Database to Schema Component 'dummy'** to select a database for this component.

3. Select the **DS: SOALocalTxDataSource** data source associated with the SOAINFRA component, and deselect the **XA** check box on the right side of the screen.

   Repeat this step to deselect this option for **DS: EDNLocalTxDataSource** as well.

4. The WLSSchemaDataSource data source associated with the WLS component is not defined in any template for the domain. Right-click **DS: WLSSchemaDataSource** on the screen to remove it.

8.3.7 Editing the Database URL

Specify the connection information for the database in which you created the schemas.

1. Navigate to the **Data Sources** tab, and select the **Node** element on the screen.

2. In the **Connect String** field, enter the JDBC URL to the database repository in the following format:

   ```
   jdbc:oracle:thin:@db_host:db_port/service_name
   ```

   Where *db_host* is the host name of the machine on which the database resides, *db_port* is the listener port of the database, and *service_name* is the service name identified for the database.

8.4 Validating and Saving the Topology File

If there any errors found in the topology file, these errors will be displayed in the validation panel at the bottom of the screen. Correct them before continuing.

To locate an error in the file, select the error message from the list, and then click  . You can also double-click the error message to go straight to the panel that contains the error.

After correcting any errors, save the file and exit Composer.

8.5 Using Apache Groovy to Run the create-domain Standard Action

After editing and validating the sample `soa-topology-1.0.json` file, use Apache Groovy to run the `create-domain.groovy` script and execute the create-domain action.

The `create-domain.groovy` script is located in the `create-domain-example` directory.

To run the action:

1. Change directory to the `create-domain-example` directory.

   ```
   cd path_to_example_directory/create-domain-example
   ```

2. In the `call-domain.sh` file, update the Oracle home location to point to the SOA Suite Oracle home.

   Note that there are two places in this file where you will need to update the Oracle home location, as shown in the following example.

   ```
   #!/bin/bash

   groovy -cp "ORACLE_HOME/oracle_common/modules/features/
   oracle.fmwplatform.envspec_lib.jar:
   ORACLE_HOME/oracle_common/modules/features/oracle.fmwplatform.common_lib.jar" $*
   ```

   In this example, replace *ORACLE_HOME* with the actual path to your Oracle home directory.

3. In the `create-domain.groovy` file in the `create-domain-example` directory, update the Oracle home location to point to the SOA Suite Oracle home.

   For example:

   ```
   // get ready to run the action

   def af = DefaultActionFactoryLocator.locateActionFactory()
   def action = af.getAction("create-domain", "ORACLE_HOME")
   def target = ModelTargetFactory.createDomainTarget("soadomain")
   def targets = new ArrayList<ModelTarget>()
   targets.add(target)
   ```

   In this example, replace *ORACLE_HOME* with the actual path to your Oracle home directory.

4. In the `create-domain.groovy` file, replace `secret1@` with the actual password of the wallet you created, as shown in the following example:

```
"wallet_password".toCharArray()    // the wallet passphrase
```

5. Verify that the ORACLE_HOME variable is not set in your environment.

   ```
   echo $ORACLE_HOME
   ```

   If the ORACLE_HOME variable is set in your environment, unset it before running `create-domain.groovy`:

   ```
   unset ORACLE_HOME
   ```

6. If not already set, set the JAVA_HOME variable to the path of a certified JDK, as shown in the following example:

   ```
   export JAVA_HOME=/home/Oracle/products/jdk1.8.0_77
   ```

7. If not already set, set the GROOVY_HOME variable to the directory where you installed the Apache Groovy distribution, as shown in the following example:

   ```
   export GROOVY_HOME=/home/Oracle/groovy-2.4.6
   export PATH=$PATH:$GROOVY_HOME/bin
   ```

8. Run the following command from the `create-domain-example` directory to execute the action:

   ```
   sh call-domain.sh create-domain.groovy
   ```

   If the action is successful, the exit status code will be SUCCESS, as shown in the example below:

   ```
   May 18, 2016 9:39:28 AM
   oracle.fmwplatform.actionframework.v2.internal.AbstractActionImpl logActionResult
   INFO: SUCCESS: JAVA_WLST_ACTION: Init:
   May 18, 2016 9:39:28 AM oracle.fmwplatform.actionframework.v2.internal.ActionBase
   setInitializedState
   INFO: Action create-domain changed from state CREATED to state NOT_READY
   May 18, 2016 9:39:28 AM oracle.fmwplatform.actionframework.v2.internal.ActionBase
   setCanInvokeState
   INFO: Action create-domain changed from state NOT_READY to state READY
   May 18, 2016 9:39:28 AM
   oracle.fmwplatform.actionframework.v2.logging.internal.ActionsLoggerImpl
   logComponentCall
   INFO: Action create-domain: Calling the component wlst.selectTemplate(Basic
   WebLogic Server Domain) {"domainName":"soadomain"}
   May 18, 2016 9:39:39 AM
   oracle.fmwplatform.actionframework.v2.logging.internal.ActionsLoggerImpl
   logComponentReturn
   INFO: Action create-domain: Resuming execution upon the return from the component
   wlst.selectTemplate(Basic WebLogic Server Domain) {"domainName":"soadomain"}
   May 18, 2016 9:39:39 AM
   oracle.fmwplatform.actionframework.v2.logging.internal.ActionsLoggerImpl
   logComponentCall
   INFO: Action create-domain: Calling the component wlst.selectTemplate(Oracle SOA
   Suite) {"domainName":"soadomain"}
   May 18, 2016 9:39:39 AM
   oracle.fmwplatform.actionframework.v2.logging.internal.ActionsLoggerImpl
   logComponentReturn
   INFO: Action create-domain: Resuming execution upon the return from the component
   wlst.selectTemplate(Oracle SOA Suite)
   {"domainName":"soadomain"}
   May 18, 2016 9:39:39 AM
   oracle.fmwplatform.actionframework.v2.logging.internal.ActionsLoggerImpl
   ```

```
logComponentCall
INFO: Action create-domain: Calling the component wlst.loadTemplates()
{"domainName":"soadomain"}
May 18, 2016 9:39:52 AM
oracle.fmwplatform.actionframework.v2.logging.internal.ActionsLoggerImpl
logComponentReturn
INFO: Action create-domain: Resuming execution upon the return from the component
wlst.loadTemplates() {"domainName":"soadomain"}
drw-   AppDeployment
drw-   CoherenceClusterSystemResource
drw-   Credential
drw-   FileStore
drw-   JDBCSystemResource
drw-   JMSServer
drw-   JMSSystemResource
drw-   Keystore
drw-   Library
drw-   NMProperties
drw-   Security
drw-   SecurityConfiguration
drw-   SelfTuning
drw-   Server
drw-   ServerTemplate
drw-   ShutdownClass
drw-   StartupClass
drw-   StartupGroupConfig
drw-   WLDFSystemResource
drw-   WebAppContainer
drw-   AdminServer
drw-   soa_server1
drw-   AppDeployment
drw-   CoherenceClusterSystemResource
drw-   Credential
drw-   FileStore
drw-   JDBCSystemResource
drw-   JMSServer
drw-   JMSSystemResource
drw-   Keystore
drw-   Library
drw-   NMProperties
drw-   Security
drw-   SecurityConfiguration
drw-   SelfTuning
drw-   Server
drw-   ServerTemplate
drw-   ShutdownClass
drw-   StartupClass
drw-   StartupGroupConfig
drw-   WLDFSystemResource
drw-   WebAppContainer
drw-   AdminServer
May 18, 2016 9:39:53 AM
oracle.fmwplatform.actionframework.v2.logging.internal.ActionsLoggerImpl logInfo
INFO: Action create-domain: Server AdminServerBinding is not mapped to a domain
node manager {"domainName":"soadomain"}
May 18, 2016 9:39:54 AM
oracle.fmwplatform.actionframework.v2.logging.internal.ActionsLoggerImpl logInfo
INFO: Action create-domain: Server soa_server1Binding is not mapped to a domain
node manager {"domainName":"soadomain"}
drw-   AppDeployment
drw-   CoherenceClusterSystemResource
```

```
drw-   Credential
drw-   FileStore
drw-   JDBCSystemResource
drw-   JMSServer
drw-   JMSSystemResource
drw-   Keystore
drw-   Library
drw-   NMProperties
drw-   Security
drw-   SecurityConfiguration
drw-   SelfTuning
drw-   Server
drw-   ServerTemplate
drw-   ShutdownClass
drw-   StartupClass
drw-   StartupGroupConfig
drw-   WLDFSystemResource
drw-   WebAppContainer
drw-   AppDeployment
drw-   CoherenceClusterSystemResource
drw-   Credential
drw-   FileStore
drw-   JDBCSystemResource
drw-   JMSServer
drw-   JMSSystemResource
drw-   Keystore
drw-   Library
drw-   NMProperties
drw-   Security
drw-   SecurityConfiguration
drw-   SelfTuning
drw-   Server
drw-   ServerTemplate
drw-   ShutdownClass
drw-   StartupClass
drw-   StartupGroupConfig
drw-   WLDFSystemResource
drw-   WebAppContainer
drw-   AppDeployment
drw-   CoherenceClusterSystemResource
drw-   Credential
drw-   FileStore
drw-   JDBCSystemResource
drw-   JMSServer
drw-   JMSSystemResource
drw-   Keystore
drw-   Library
drw-   NMProperties
drw-   Security
drw-   SecurityConfiguration
drw-   SelfTuning
drw-   Server
drw-   ServerTemplate
drw-   ShutdownClass
drw-   StartupClass
drw-   StartupGroupConfig
drw-   WLDFSystemResource
drw-   WebAppContainer
drw-   Property
drw-   Property
drw-   Property
```

```
drw-   Property
drw-   Property
drw-   Property
drw-   Property
drw-   Property
drw-   Property
drw-   Property
drw-   Property
drw-   Property
drw-   Property
drw-   Property
drw-   Property
May 18, 2016 9:39:57 AM
oracle.fmwplatform.actionframework.v2.logging.internal.ActionsLoggerImpl
logComponentCall
INFO: Action create-domain: Calling the component wlst.writeDomain()
{"domainName":"soadomain"}
May 18, 2016 9:42:21 AM
oracle.fmwplatform.actionframework.v2.logging.internal.ActionsLoggerImpl
logComponentReturn
INFO: Action create-domain: Resuming execution upon the return from the component
wlst.writeDomain() {"domainName":"soadomain"}
May 18, 2016 9:42:21 AM
oracle.fmwplatform.actionframework.v2.internal.AbstractActionImpl logActionResult
INFO: SUCCESS: Action create-domain succeeded for Target soadomain
May 18, 2016 9:42:21 AM oracle.fmwplatform.actionframework.v2.internal.ActionBase
setInvokedState
INFO: Action create-domain changed from state READY to state NOT_READY
May 18, 2016 9:42:21 AM
oracle.fmwplatform.actionframework.v2.internal.AbstractActionImpl logActionResult
INFO: SUCCESS: Action create-domain succeeded for Target soadomain
May 18, 2016 9:42:21 AM oracle.fmwplatform.actionframework.v2.internal.ActionBase
setClosedState
INFO: Action create-domain changed from state NOT_READY to state DESTROYED
May 18, 2016 9:42:21 AM oracle.fmwplatform.actionframework.v2.internal.ActionBase
setDoneState
INFO: Action create-domain changed from state DESTROYED to state DONE
{
  "actionResultList" : [ {
    "statusCode" : "SUCCESS",
    "statusDetail" : "JAVA_WLST_ACTION: Init: "
  }, {
    "statusCode" : "SUCCESS",
    "statusDetail" : "Action create-domain succeeded for Target soadomain",
    "target" : {
      "domainName" : "soadomain"
    }
```

8.6 Starting the Servers

To verify that your domain is properly configured, manually start the Administration Server and Managed Server.

1. To start the WebLogic Administration Server, change directory to the *DOMAIN_HOME*/bin directory and run the following command:

   ```
   ./startWebLogic.sh
   ```

2. To verify that the Administration Server is running, enter the following URL in a browser to access the WebLogic Administration Console:

   ```
   http://adminserver_host:adminserver_port/console
   ```

Log in using the WebLogic administrator credentials that you created and added to the wallet for this tutorial. See Creating a Wallet.

3. After starting the Administration Server, start the Managed Server from the *DOMAIN_HOME*/bin directory:

```
./startManagedWebLogic.sh managed_server_name http://
adminserver_host:adminserver_port
```

You are prompted for the administrator user login credentials. Enter the WebLogic administrator credentials that you created and added to the wallet.

Part III

Appendices Additional Information About the Lifecycle Management Tools

A

Additional Information About Standard Actions

The Lifecycle Management Tools provide standard actions that you can use to configure and manage your Oracle Fusion Middleware environment.

This appendix describes important information about each standard action.

A.1 Using the Standard Actions JavaDoc

The Lifecycle Management Tools provide a set of Java APIs.

These APIs are documented in the following JavaDoc:

- *Java API Reference for the Lifecycle Management Action Framework*
- *Java API Reference for the Lifecycle Management Environment Specification*
- *Java API Reference for the Lifecycle Management Credential Management*

A.2 More Information About Commonly Used Standard Actions

A.2.1 About the Create Domain Standard Action

With the create-domain standard action, you can create an Oracle Fusion Middleware domain.

The create-domain standard action creates the specified domain by loading any templates used in the definition of that the domain in the model. It creates any data sources and other resources defined in the templates. It also creates the servers, clusters, partitions, and other structures as specified in the domain definition. This standard action does not perform any additional configuration steps, such as component-specific configuration. You can use custom actions to perform the component-specific configuration.

This standard action is equivalent to running the Domain Configuration Wizard.

A.2.2 About the Install Oracle Home Standard Action

With the install-oracle-home standard action, you can create an Oracle home for Oracle Fusion Middleware.

This standard action creates an Oracle Home by installing products as specified by the Installer and Install Types.

A.2.3 About the Create Schema Standard Action

With the create-schema standard action, you can create a schema in the target database. The schema is used by Oracle Fusion Middleware.

This action creates the schema, but does not load data into the newly created schema.

A.2.4 About the Load Schema Standard Action

The load-schema standard action loads data into a schema.

This is equivalent to running the second phase of the Repository Creation Utility (RCU), the `-dataLoad` operation. This standard action does not create the schema if it does not exist. Use the create-schema standard action to create the schema.

A.2.5 About the Start Standard Action

With the start standard action, you can start all servers in a domain, a specific server, a cluster, all servers on a host, or you can start Node Manager.

You can start the following using the start standard action:

- Domain: All servers within the domain, across all hosts
- Server: A specified server within the domain.
- Cluster: All servers in a specified cluster within the domain across all hosts.
- Host: All servers on a specified host across all domains on the host.
- Node Manager: A specified Node Manager on the local host.

A.2.6 About the Stop Standard Action

With the stop standard action, you can stop all servers in a domain, a specific server, a cluster, all servers on a host, or you can stop Node Manager.

You can stop the following using the stop standard action:

- Domain: All servers within the domain, across all hosts
- Server: A specified server within the domain.
- Cluster: All servers in a specified cluster within the domain across all hosts.
- Host: All servers on a specified host across all domains on the host.
- Node Manager: A specified Node Manager on the local host.