

Oracle® Fusion Middleware

Creating Domain Templates Using the Domain Template Builder



12c (12.2.1.3)
E95527-04
November 2022

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

ORACLE®

Oracle Fusion Middleware Creating Domain Templates Using the Domain Template Builder, 12c (12.2.1.3)

E95527-04

Copyright © 2015, 2022, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

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

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

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

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

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

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

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

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	v
Documentation Accessibility	v
Diversity and Inclusion	v
Related Documents	v
Conventions	vi

1 Introduction

Overview of the Domain Template Builder	1-1
About Domain and Extension Templates	1-2
Additional Tools for Creating Templates	1-3

2 Using the Domain Template Builder to Create Domain Templates

Starting the Domain Template Builder	2-1
Creating Domain Templates Using the Domain Template Builder	2-2
Porting Security Provider Data to a New Domain	2-3

3 Creating Extension Templates by Using the Domain Template Builder

Starting the Domain Template Builder	3-1
Creating an Extension Template Using the Domain Template Builder	3-2

4 Domain Template Builder Screens

Template Type	4-1
Creating a Domain Template From an Existing Domain	4-2
Creating an Extension Template From an Existing Domain	4-2
Creating a Domain Template From an Existing Template	4-3
Creating an Extension Template From an Existing Template	4-3
Template Information	4-4
Services Selection	4-4

Topology Selection	4-4
Template Summary	4-5
Configuration Progress	4-5
End of Configuration	4-5

Preface

This preface describes the audience, document accessibility features, diversity and inclusion, related documents, and conventions used in this guide.

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

Audience

This document is intended for administrators of Oracle Fusion Middleware, who are assigned the task of installing and configuring Oracle Fusion Middleware software.

Documentation Accessibility

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

Access to Oracle Support

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

Diversity and Inclusion

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

Related Documents

You can access the documents referenced in this guide on the Oracle Technology Network at <http://www.oracle.com/technology>:

- *Creating WebLogic Domains Using the Configuration Wizard*
- *Creating Templates and Domains Using the Pack and Unpack Commands*
- *Domain Template Reference*
- *WLST Command Reference for WebLogic Server*
- *Understanding the WebLogic Scripting Tool*

Conventions

The following text conventions are used in this document:

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

1

Introduction

You can create templates in WebLogic Server by using the Domain Template Builder. The Domain Template Builder is a standalone Java application that guides you through the process.

This chapter contains the following sections:

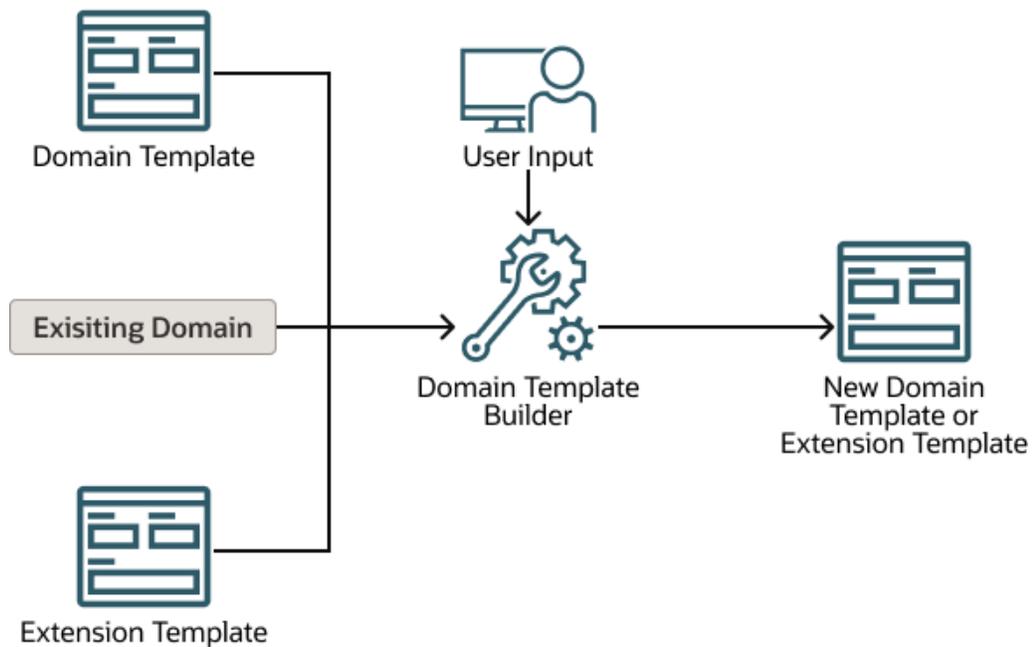
- [Overview of the Domain Template Builder](#)
The Domain Template Builder helps you create domain and extension templates. You can then use these domain or extension templates to build or extend WebLogic domains.
- [About Domain and Extension Templates](#)
In the context of the Domain Template Builder, the term template refers to a Java Archive (JAR) file. A template JAR contains the files and scripts that are used to create or extend a WebLogic domain.
- [Additional Tools for Creating Templates](#)
In addition to the Domain Template Builder, you can use other tools such as the `pack` command and WLST to create domain and extension templates.

Overview of the Domain Template Builder

The Domain Template Builder helps you create domain and extension templates. You can then use these domain or extension templates to build or extend WebLogic domains.

The domain and extension templates contain the key attributes and files required to build or extend a WebLogic domain. [Figure 1-1](#) illustrates the process to create templates.

Figure 1-1 Domain Template Builder



You can use the Domain Template Builder to create templates off-line, in graphical mode. The Domain Template Builder is not supported in console mode. Therefore, the console attached to the system on which you are using the Domain Template Builder must support Java-based GUIs. All consoles for Windows systems support Java-based GUIs; only some UNIX-based consoles support Java-based GUIs.

Your product installation includes a set of predefined domain and extension templates. This set of templates includes the WebLogic Server Base Domain template, and various extension templates that allow you to add product features and samples to the base domain. For more information about the templates that are available for various products, see *Fusion Middleware Product Templates* in *Domain Template Reference*.

After you create your domain and add new resources and applications to it, you can use the Domain Template Builder to create domain and extension templates. To do so, select the existing domain or template as the source for your new domain or extension templates. When your custom domain templates are complete, you can use them to create domains by using the Configuration Wizard, WLST Offline, or the `unpack` command. For more information about these tools, see [Additional Tools for Creating Templates](#).

About Domain and Extension Templates

In the context of the Domain Template Builder, the term template refers to a Java Archive (JAR) file. A template JAR contains the files and scripts that are used to create or extend a WebLogic domain.

The Configuration Wizard uses the following types of templates to create or extend domains:

- **Domain template:** defines the full set of resources within a WebLogic domain, including infrastructure components, applications, services, security options, and

general environment and operating system options. You can create this type of template from an existing domain by using the Domain Template Builder or the `pack` command. You can then create a domain based on the template.

The product distribution includes the WebLogic Server Base Domain template. This template defines the core set of resources within a domain, including an Administration Server and basic configuration information, infrastructure components, and general environment and operating system options. It does not include sample applications. You can use this template to create a basic WebLogic Server domain, which you can then extend with applications and services, or additional product component functionality.

- **Extension template:** defines applications and services that can provide additional features, such as applications, or JDBC resources, JMS components, or deployed libraries. This type of template can be used to update an existing WebLogic domain.
- **Managed Server template:** defines the subset of resources within a domain that are required to create a Managed Server domain on a remote machine. This type of template can be created by using the `pack` command.

Your product installation includes a set of predefined domain and extension templates. This set includes the WebLogic Server Base Domain template and various extension templates that allow you to add component features and samples to the base domain. For more information about these templates and how they relate to each other, see the WebLogic Server Templates in *Domain Template Reference*.

Additional Tools for Creating Templates

In addition to the Domain Template Builder, you can use other tools such as the `pack` command and WLST to create domain and extension templates.

Here is a description of the additional tools:

- `pack`

Use this command to create a template archive (JAR) file that contains a snapshot of either an entire WebLogic domain or a subset of a WebLogic domain. You can use a template that contains a subset of a domain to create a Managed Server domain directory hierarchy on a remote machine.

For more information about the `pack` command, see *The Pack Command in Creating Templates and Domains Using the Pack and Unpack Commands*.

- WebLogic Scripting Tool (WLST):

WLST is a command-line scripting interface, which you can use to interact with and configure WebLogic Server instances and domains. When WLST is offline, you can create a WebLogic domain or update an existing WebLogic domain without connecting to a running WebLogic Server—supporting the same functionality as the Configuration Wizard.

For more information about WLST, see *Using the WebLogic Scripting Tool in Understanding the WebLogic Scripting Tool* and *WebLogic Server WLST Online and Offline Command Reference in WLST Command Reference for WebLogic Server*.

2

Using the Domain Template Builder to Create Domain Templates

You can create a domain template by using the Domain Template Builder in the graphical mode. The template creation process takes you through a sequence of screens for capturing specific information about the template.

This chapter includes the following sections:

- [Starting the Domain Template Builder](#)
Before you start the Domain Template Builder, ensure that the console for the machine on which the product installation resides supports Java-based GUIs. All Windows-based consoles support Java-based GUIs; only a subset of UNIX-based consoles supports Java-based GUIs.
- [Creating Domain Templates Using the Domain Template Builder](#)
You can use the Domain Template Builder in the graphical mode to create domain templates. The process takes you through a sequence of screens where you capture specific information about the template that you want to create.
- [Porting Security Provider Data to a New Domain](#)
The Domain Template Builder is an offline utility, and hence does not export security provider data, such as embedded LDAP data, into a domain template. After you use the domain template to create the new domain, you must manually port the security provider data to the new domain.

Starting the Domain Template Builder

Before you start the Domain Template Builder, ensure that the console for the machine on which the product installation resides supports Java-based GUIs. All Windows-based consoles support Java-based GUIs; only a subset of UNIX-based consoles supports Java-based GUIs.

You can start the Domain Template Builder in graphical mode from the command line.

1. Log in to the system on which the product is installed.
2. Open a command prompt window (on Windows) or a command shell (on UNIX).
3. Go to the following directory, where *ORACLE_HOME* is the Oracle home directory that you specified when installing WebLogic Server:

On Windows: *ORACLE_HOME*\oracle_common\common\bin

On UNIX: *ORACLE_HOME*/oracle_common/common/bin

4. Execute the following command:

On Windows: `config_builder.cmd`

On UNIX: `config_builder.sh`

 **Note:**

When you run the `config_builder.cmd` or `config_builder.sh` command, the following error message appears to indicate that the default cache directory is not valid:

```
*sys-package-mgr*: can't create package cache dir
```

You can change the cache directory by including the `-Dpython.cachedir=valid_directory` option in the command line.

The [Template Type](#) screen is displayed.

Creating Domain Templates Using the Domain Template Builder

You can use the Domain Template Builder in the graphical mode to create domain templates. The process takes you through a sequence of screens where you capture specific information about the template that you want to create.

Start the Domain Template Builder as described in [Starting the Domain Template Builder](#).

 **Note:**

In situations where you cannot run the Domain Template Builder in GUI mode, Oracle recommends that you use a WLST script to create the domain template as described in [Creating and Using a Domain Template \(Offline\)](#) in *Understanding the WebLogic Scripting Tool*.

The Domain Template Builder displays a sequence of screens, in the order listed in [Table 2-1](#). For more information about each screen, refer the related section in [Domain Template Builder Screens](#), or click the link in the **Screen** column.

[Table 2-1](#) summarizes the steps for creating a domain template by using the Domain Template Builder.

Table 2-1 Steps for Creating a Domain Template by Using the Domain Template Builder

Screen	When Does This Screen Appear?	You Perform This Action
Template Type	Always	Choose Create a Domain Template to create a domain template. Specify the source for the domain template and the location of the source. Specify the destination JAR file for the domain template. Click Next to continue.

Table 2-1 (Cont.) Steps for Creating a Domain Template by Using the Domain Template Builder

Screen	When Does This Screen Appear?	You Perform This Action
Template Information	Always	Fill in each template information field. Click Next to continue.
Template Summary	Always	Review the template summary. You can use the View drop-down list to restrict the view to a specific part of the template. Click Back to return to a previous screen if you want to make any changes. Click Create to continue.
Configuration Progress	Always	Wait for the template to be generated, and then click Next .
End of Configuration	Always	Click Finish to exit the Domain Template Builder.

Porting Security Provider Data to a New Domain

The Domain Template Builder is an offline utility, and hence does not export security provider data, such as embedded LDAP data, into a domain template. After you use the domain template to create the new domain, you must manually port the security provider data to the new domain.

To manually port the security provider data to the new domain:

1. If necessary, start the Administration Server for the original domain and log in to its Administration Console.
2. Export the data from the original domain, as described in Export data from a security provider in the *Oracle WebLogic Server Administration Console Online Help*.
3. If necessary, start the Administration Server for the new domain and log in to the Administration Console of the new domain .
4. Import the data into the new domain, as described in Import data into a security provider in the *Oracle WebLogic Server Administration Console Online Help*.

3

Creating Extension Templates by Using the Domain Template Builder

You can create an extension template by using the Domain Template Builder. The process is similar to the process of creating a domain template.

The resulting extension template does not configure the Administration Server, administrator username and password, security realm, and Node Manager, or specify Start menu entries. These domain settings are already defined in the domain you are extending.

This chapter includes the following sections:

- [Starting the Domain Template Builder](#)
Before you start the Domain Template Builder, ensure that the console for the machine on which the product installation resides supports Java-based GUIs. All Windows-based consoles support Java-based GUIs; only a subset of UNIX-based consoles supports Java-based GUIs.
- [Creating an Extension Template Using the Domain Template Builder](#)
You can use the Domain Template Builder in the graphical mode to create extension templates. The process takes you through a sequence of screens where you capture specific information about the template that you want to create.

Starting the Domain Template Builder

Before you start the Domain Template Builder, ensure that the console for the machine on which the product installation resides supports Java-based GUIs. All Windows-based consoles support Java-based GUIs; only a subset of UNIX-based consoles supports Java-based GUIs.

You can start the Domain Template Builder in graphical mode from the command line.

1. Log in to the system on which the product is installed.
2. Open a command prompt window (on Windows) or a command shell (on UNIX).
3. Go to the following directory:

On Windows: `ORACLE_HOME\oracle_common\common\bin`

On UNIX: `ORACLE_HOME/oracle_common/common/bin`

4. Execute the following command:

On Windows: `config_builder.cmd`

On UNIX: `config_builder.sh`

 **Note:**

When you run the `config_builder.cmd` or `config_builder.sh` command, the following error message appears to indicate that the default cache directory is not valid:

```
*sys-package-mgr*: can't create package cache dir
```

You can change the cache directory by including the `-Dpython.cachedir=valid_directory` option in the command line.

The [Template Type](#) screen is displayed.

Creating an Extension Template Using the Domain Template Builder

You can use the Domain Template Builder in the graphical mode to create extension templates. The process takes you through a sequence of screens where you capture specific information about the template that you want to create.

To create a new extension template using the Domain Template Builder in graphical mode, start the Domain Template Builder as described in [Starting the Domain Template Builder](#). Then, select the **Create Extension Template** option to display the Create Extension Template screen (similar to the [Template Type](#) screen).

 **Note:**

In situations where you cannot run the Domain Template Builder in GUI mode, Oracle recommends that you use a WLST script to create the extension template. For more information, see [Creating and Using a Domain Template \(Offline\)](#) in *Understanding the WebLogic Scripting Tool*.

The Domain Template Builder displays a sequence of screens, in the order listed in [Table 3-1](#). For more information on each screen, refer to the related section in [Domain Template Builder Screens](#), or click the link in the **Screen** column.

Table 3-1 Steps for Creating an Extension Template by Using the Domain Template Builder

Screen	Screen Appears	Perform This Action
Template Type	Always	Choose Create Extension Template to create an extension template. Specify the source for the extension template and location of the source. Specify the destination JAR file for the extension template. Click Next to continue.

Table 3-1 (Cont.) Steps for Creating an Extension Template by Using the Domain Template Builder

Screen	Screen Appears	Perform This Action
Template Information	Always	Update each template information field as needed. Click Next to continue.
Services Selection	Only if you create an extension template	Select the custom applications and services that you want to include in the extension template. Click Next to continue.
Topology Selection	Only if you create an extension template and the selected source (domain or template) has topology information	From the topology elements listed for the configuration, select the elements you want to include in the extension template.
Template Summary	Always	Review the template summary. You can use the View drop-down list to restrict the view to a specific part of the template. If necessary, click Prev to return to a previous screen and make any changes that are needed. Click Create to continue.
Configuration Progress	Always	Wait for the template to be generated, then click Next .
End of Configuration	Always	Click Finish to exit the Domain Template Builder.

4

Domain Template Builder Screens

The Domain Template Builder helps you create domain and extension templates. During the process, it takes you through a sequence of screens where you capture specific information about the domain or extension template.

- [Template Type](#)
In the Template Type screen, first specify whether you want to create a domain template or an extension template, and then specify the source and destination for the new template.
- [Template Information](#)
Use the Template Information screen to provide basic information about the template. The information you specify here is stored in the `template-info.xml` file that is generated for the domain or extension template. It is also stored in the `config.xml` file for the domain.
- [Services Selection](#)
The Services Selection screen is available only if you are creating an extension template. You can use this screen to review the custom applications and services in the configuration, and then select the ones you want to include in the extension template.
- [Topology Selection](#)
The Topology Selection screen is available only if you are creating an extension template. You can use this screen to review the topology elements in the configuration and select the elements you want to include in the extension template.
- [Template Summary](#)
The Template Summary screen provides a summary of the settings for the new template. Review the information before you proceed to create the template.
- [Configuration Progress](#)
The Configuration Progress screen displays status messages during the template creation process.
- [End of Configuration](#)
The End of Configuration screen displays the final configuration message, depending on the success or failure of the template creation process.

Template Type

In the Template Type screen, first specify whether you want to create a domain template or an extension template, and then specify the source and destination for the new template.

Option	Description
Create Domain Template	<p>Select this option to create a new domain template that defines the full set of resources within a WebLogic domain. The resources include infrastructure components, applications, services, security options, and general environment and operating system parameters.</p> <p>You can use the template that you create as the basis for creating a WebLogic domain, by specifying the template in the Configuration Wizard, WebLogic Scripting Tool (WLST), or the <code>unpack</code> command.</p>
Create Extension Template	<p>Select this option to create a template that defines applications and services that can be used to extend existing WebLogic domains.</p> <p>You can import applications and services stored in extension templates into an existing WebLogic domain by using the Configuration Wizard.</p> <p>For information about the Configuration Wizard, see <i>Updating WebLogic Domains in Creating WebLogic Domains Using the Configuration Wizard</i>.</p>

- [Creating a Domain Template From an Existing Domain](#)
- [Creating an Extension Template From an Existing Domain](#)
- [Creating a Domain Template From an Existing Template](#)
- [Creating an Extension Template From an Existing Template](#)

Creating a Domain Template From an Existing Domain

To create a domain template from an existing domain directory:

1. Select **Create Domain Template**.
2. Select **Use Domain as a Source**.
3. Enter the full path of the source domain directory in the **Source Location** field, or use the **Browse** button to navigate to the domain directory (for example, `C:\domains\mydomain`).
4. In the **Template Location** field, enter the full path and JAR file name for the new template, or use the **Browse** button to navigate to an existing directory. Ensure that you include the JAR file name. For example, `C:\templates\domains\mynewtemplate.jar`.
5. Click **Next**.

Creating an Extension Template From an Existing Domain

To create an extension template from an existing domain directory:

1. Select **Create Extension Template**.
2. Select **Use Domain as a Source**.

3. Enter the full path of the source domain directory in the **Source Location** field, or use the **Browse** button to navigate to the domain directory (for example, C:\domains\mydomain).
4. In the **Template Location** field, enter the full path and JAR file name for the new template, or use the **Browse** button to navigate to an existing directory. Ensure that you include the JAR file name. For example, C:\templates\domains\mynewtemplate.jar.
5. Click **Next**.

Creating a Domain Template From an Existing Template

To create a domain template from an existing template:

1. Select **Create Domain Template**.
2. Select **Use Template as a Source**.
3. Enter the full path of the source domain JAR file in the **Source Location** field, or use the **Browse** button to navigate to the domain directory and select the JAR file (for example, C:\domains\mydomain.jar).

 **Note:**

Ensure that you do not select a product template when you use a template as the source for creating domain templates. If you select a product template, an error message appears when you click **Next**.

4. In the **Template Location** field, enter the full path and JAR file name for the new template, or use the **Browse** button to navigate to an existing directory. Ensure that you include the JAR file name, for example, C:\templates\domains\mynewtemplate.jar.
5. Click **Next**.

Creating an Extension Template From an Existing Template

To create an extension template from an existing template:

1. Select **Create Extension Template**.
2. Select **Use Template as a Source**.
3. Enter the full path of the source application JAR file in the **Select Source Location** field, or use the **Browse** button to navigate to the application directory and select the jar file (for example, C:\applications\myapplication.jar).

 **Note:**

Ensure that you do not select a product template when you use a template as the source for creating extension templates. If you select a product template, an error message appears when you click **Next**.

4. In the **Select Template Location** field, enter the full path and JAR file name for the new template, or use the **Browse** button to navigate to an existing directory. Ensure that you

include the JAR file name. For example,
C:\templates\applications\mynewapplication.jar.

5. Click **Next**.

Template Information

Use the Template Information screen to provide basic information about the template. The information you specify here is stored in the `template-info.xml` file that is generated for the domain or extension template. It is also stored in the `config.xml` file for the domain.

When you select a domain or a template as the source for the new domain or extension template, information about the selected template is displayed. Review the information and modify, if necessary, and then click **Next**.

Field	Description
Name	Enter a valid name for the template. The name must be a string of characters, and can include spaces.
Version	Enter a version number for the template. The version number can be anything you want, as long as it contains four or five digits, for example, 12.1.2.0 or 12.1.2.0.0.
Author	(Optional) Enter the name of the author of the template.
Description	(Optional) Enter a detailed description of the template. The description must be a string of characters, and can include spaces.

Services Selection

The Services Selection screen is available only if you are creating an extension template. You can use this screen to review the custom applications and services in the configuration, and then select the ones you want to include in the extension template.

By default, the Domain Template Builder includes only the custom applications and services from the WebLogic domain or the template that you specified as the source for the extension template.

Expand each application or service to view its contents. Select an item from the list and review the associated details on the **Description** pane on the right. After reviewing, select the elements you want to include in the extension template, and then click **Next**.

Topology Selection

The Topology Selection screen is available only if you are creating an extension template. You can use this screen to review the topology elements in the configuration and select the elements you want to include in the extension template.

By default, the Domain Template Builder includes all the topology elements from the WebLogic domain or template you specified as the source for the extension template.

Expand each topology element to view its contents. Select an element from the list and review the associated details on the **Description** pane on the right. After reviewing, select the files you want to include in the extension template, and then click **Next**.

Template Summary

The Template Summary screen provides a summary of the settings for the new template. Review the information before you proceed to create the template.

To review the settings for the new template:

1. Select an item in the **Template Summary** pane on the left and review the associated details in the **Details** pane on the right. If you want to change any settings that you defined in previous windows, click **Back** to return to the required configuration window.

 **Note:**

You can limit the type of information displayed in the Template Summary screen by using the **View** drop-down list.

2. Click **Create**.
The [Configuration Progress](#) screen is displayed.

Configuration Progress

The Configuration Progress screen displays status messages during the template creation process.

The Domain Template Builder generates a JAR file that contains all the components required to generate a new WebLogic domain, including the template settings that you have provided.

When the process completes, click **Finish**.

The files that are included in a template vary, depending on the WebLogic domain or template on which the template is based.

End of Configuration

The End of Configuration screen displays the final configuration message, depending on the success or failure of the template creation process.

- If the template was created successfully, a link to the template JAR file is provided under **Template Location**.
- If the template was not created successfully, an error message and an error code is displayed. In this case, try to determine the cause of the failure from the message and correct the issue.