Oracle® Fusion Middleware

Creating Domain Templates Using the Domain Template Builder

12*c* (12.2.1)

E55190-01

October 2015

This document describes how to use the Domain Template Builder to create WebLogic domain and extension templates.

ORACLE

Oracle Fusion Middleware Creating Domain Templates Using the Domain Template Builder, 12*c* (12.2.1)

E55190-01

Copyright © 2007, 2015, Oracle and/or its affiliates. All rights reserved.

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

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

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

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

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate 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 and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

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

This software or hardware and documentation may provide access to or information 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 v

Documentation Accessibility v
Related Documents v
Conventions v

1 Introduction

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

2 Using the Domain Template Builder to Create Domain Templates

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

3 Creating Extension Templates by Using the Domain Template Builder

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

4 Domain Template Builder Screens

4.1 Template Type 4-1
4.2 Template Information 4-3
4.3 Applications 4-4
4.4 Domain Content 4-6
4.5 Scripts and Files 4-8
4.5.1 Replacement Variable Definitions 4-9
4.6 Template Summary 4-10
4.7 Configuration Progress 4-11
4.8 End of Configuration 4-12

Preface

This preface describes the document accessibility features and conventions used in this guide–*Creating Domain Templates Using the Domain Template Builder*.

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.

Related Documents

You can access the documents referenced in this guide on 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*
- *Creating Domain Templates Using the Domain Template Builder*
- *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.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.

Convention	Meaning
`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

This chapter provides an overview of how to create templates in WebLogic Server by using the Domain Template Builder.

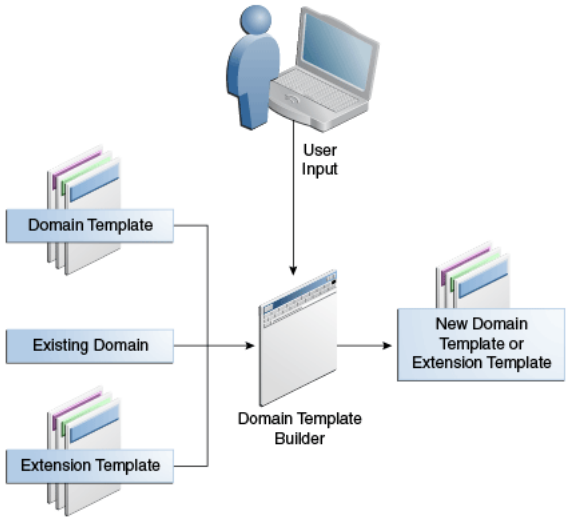
This chapter includes the following sections:

- Section 1.1, "Overview of the Domain Template Builder"
- Section 1.2, "About Domain and Extension Templates"
- Section 1.3, "Additional Tools for Creating Templates"

1.1 Overview of the Domain Template Builder

You can create WebLogic domains by using domain and extension templates that contain the key attributes and files required for building or extending a WebLogic domain. To create domain or extension templates, you use the Domain Template Builder, which is a standalone Java application that guides you through the process. This process is illustrated in Figure 1–1.

Figure 1–1 Domain Template Builder



The Domain Template Builder is designed to be used off-line in graphical mode. It is not supported in console mode. Therefore, the console attached to the machine on which you are using the Domain Template Builder must support Java-based GUIs. All

consoles for Windows systems support Java-based GUIs; 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 on the templates that are available for various products, see the *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 a custom domain template. You can also use the Domain Template Builder to customize an existing template. For example, you may want to remove applications from the template. To do so, select the existing domain or template as the source for your new custom template. 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 Section 1.3, "Additional Tools for Creating Templates."

1.2 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 needed to create or extend a WebLogic domain. The types of templates that can be used by the Configuration Wizard to create or extend domains include:

- **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 (within a domain) of resources that are required to create a Managed Server domain on a remote machine. This type of template can be created 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 *Domain Template Reference*.

1.3 Additional Tools for Creating Templates

As described earlier, you can create custom domain and extension templates by using the Domain Template Builder. You can also use the following tools to create templates:

- `pack`

  This command enables you 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, see *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, see *Understanding the WebLogic Scripting Tool* and *WLST Command Reference for WebLogic Server*.

2

Using the Domain Template Builder to Create Domain Templates

This chapter describes how to create a domain template using the Domain Template Builder.

This chapter includes the following sections:

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

2.1 Starting the Domain Template Builder

The console for the machine on which the product installation resides must support Java-based GUIs. All Windows-based consoles support Java-based GUIs; only a subset of UNIX-based consoles support 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 an MS-DOS 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:

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

   UNIX: *ORACLE_HOME*`/oracle_common/common/bin`
4. Execute the following command:

   Windows: `config_builder.cmd`

   UNIX: `sh config_builder.sh`

Note: When you run the `config_builder.cmd` or `config_builder.sh` command, the following error message might be displayed 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.

2.2 Creating a Domain Templates Using the Domain Template Builder

To create a new domain template using the Domain Template Builder in graphical mode, start the Domain Template Builder as described in Section 2.1, "Starting the Domain Template Builder."

Notes: 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. 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 2–1. For more information on each screen, refer to the related section in Chapter 4, "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. Specify the destination JAR file for the domain template. Click **Next** to continue.
Template Information	Always	Fill in each template information field. Click **Next** to continue.
Applications	Only if applications are included in the domain or template source that you selected.	Review the list of applications included in the domain. Deselect any applications that you do not want to include in the destination domain. Click **Next** to continue.
Domain Content	Always	Review the files that are included in the template. If necessary, add more files to the template. If necessary, remove files from the template. Click **Next** to continue.
Scripts and Files	Always	From the **Select File** list on the left, open a file to which you want to add a replacement variable. Select a string that you want to replace with a variable. Right-click to display the **Replacement Variable** pop-up list. In the **Select File** list on the left, select the check box for the file to indicate that it has been changed. 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 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, and then click **Finish**.
End of Configuration	Always	Click **Finish** to exit the Domain Template Builder.

2.3 Porting Security Provider Data to a New Domain

Because it is an offline utility, the Template Builder does not export security provider data, such as embedded LDAP data, into a domain template. After using the domain template to create the new domain, you must manually port the security provider data to the new domain. To do so:

1. If necessary, start the Administration Server for the original domain and log in to the original domain's 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 new domain's Administration Console.
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

This chapter describes how to create an extension template using the Domain Template Builder.

The process for creating an extension template is similar to the process for creating a domain template. The resulting 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
- Creating an Extension Template Using the Domain Template Builder

3.1 Starting the Domain Template Builder

The console for the machine on which the product installation resides must support Java-based GUIs. All Windows-based consoles support Java-based GUIs; only a subset of UNIX-based consoles support 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 an MS-DOS 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: `sh config_builder.sh`

Note: When you run the `config_builder.cmd` or `config_builder.sh` command, the following error message might be displayed 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.

3.2 Creating an Extension Template Using the Domain Template Builder

To create a new domain template using the Domain Template Builder in graphical mode, start the Domain Template Builder as described in Section 3.1, "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 Chapter 4, "Domain Template Builder Screens," or click the link in the **Screen** column.

Table 3–1 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 Extension Template** to create an extension template. Specify the source for the extension template. Specify the destination JAR file for the extension template. Click **Next** to continue.
Template Information	Always	Update each template information field as needed. Click **Next** to continue.
Applications	Only if applications are included in the domain or template source that you selected.	Select the check box for each application that you want to include in the extension template. Deselect the check box for each application that you do not want to include in the extension template. Click **Next** to continue.
Domain Content	Always	Review the files that are included in the template. If necessary, add more files to the template. If necessary, remove files from the template. Click **Next** to continue.

Table 3–1 (Cont.) Steps for Creating a Domain Template by Using the Domain Template Builder

Screen	When Does This Screen Appear?	You Perform This Action
Scripts and Files	Always	From the **Select File** list on the left, open a file to which you want to add a replacement variable. Select a string that you want to replace with a variable. Right-click to display the **Replacement Variable** pop-up list. In the **Select File** list on the left, select the check box for the file to indicate that it has been changed. 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. 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 **Finish**.
End of Configuration	Always	Click **Finish** to exit the Domain Template Builder.

4

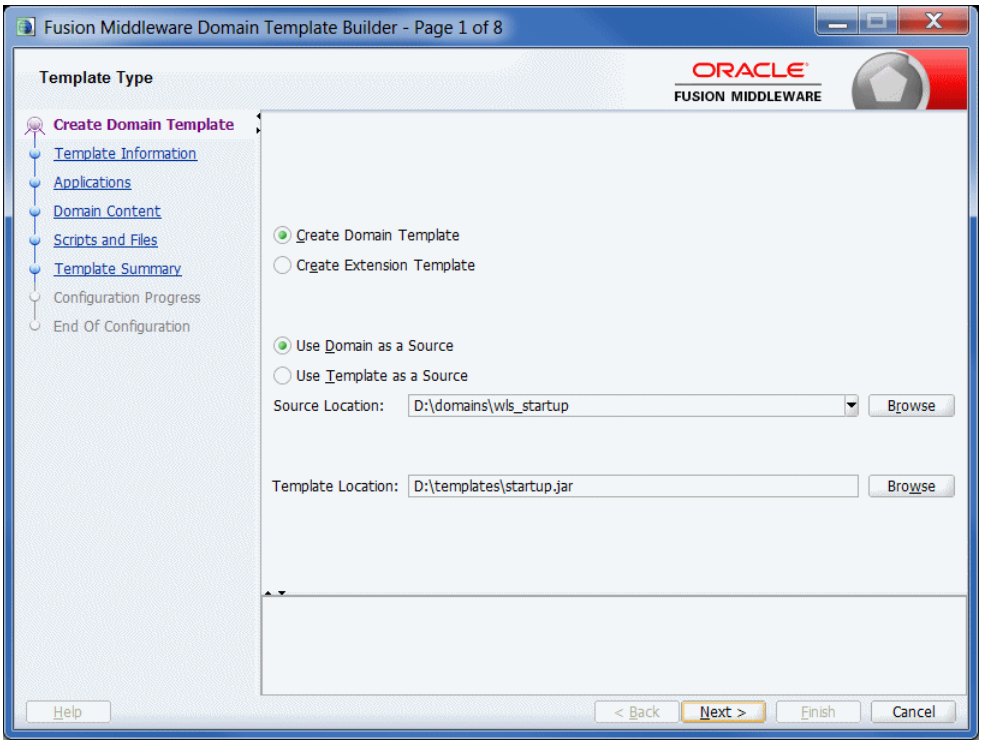
Domain Template Builder Screens

This chapter describes each of the Domain Template Builder screens.

This chapter includes the following sections:

- Template Type
- Template Information
- Applications
- Domain Content
- Scripts and Files
- Template Summary
- Configuration Progress
- End of Configuration

4.1 Template Type



Use this page to specify whether or not 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	Select this option to create a new domain template that defines the full set of resources within a WebLogic domain, including infrastructure components, applications, services, security options, and general environment and operating system parameters. You can use the template that you create as the basis for creating a WebLogic domain, by specifying the template in the Configuration Wizard, WLST, or the `unpack` command.
Create Extension Template	Select this option to create a template that defines applications and services that can be used to extend existing WebLogic domains. You can import applications and services stored in extension templates into an existing WebLogic domain by using the Configuration Wizard. For information about the Configuration Wizard, see *Creating WebLogic Domains Using the Configuration Wizard*

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** to continue.

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** to continue.

Create Domain Template From Existing Template

To create a domain template from an existing template:

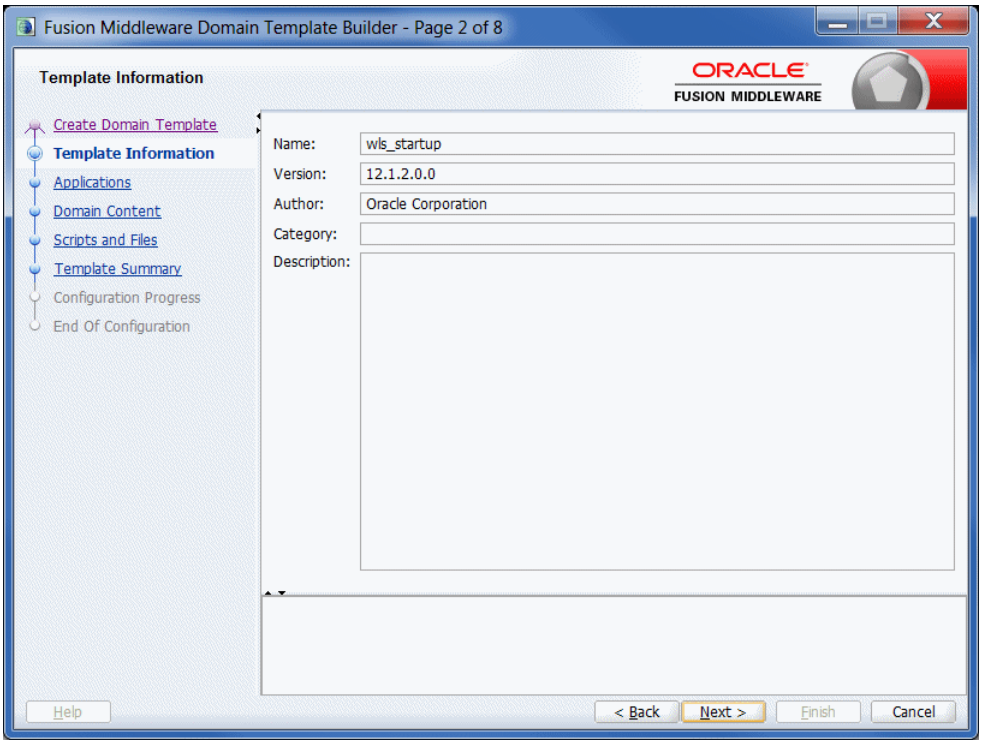
1. Select **Create Domain Template**.
2. Select **Use Template as a Source**.
3. Enter the full path of the source domain JAR file in the **Source Location** field, or use the **Browse** button to navigate to the domain directory and select the JAR file (for example, C:\domains\mydomain.jar).
4. In the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

Create Extension Template From Existing Template

To create an extension template from an existing template:

1. Select **Create Extension Template**.
2. Select **Select Template 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).
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** to continue.

4.2 Template Information

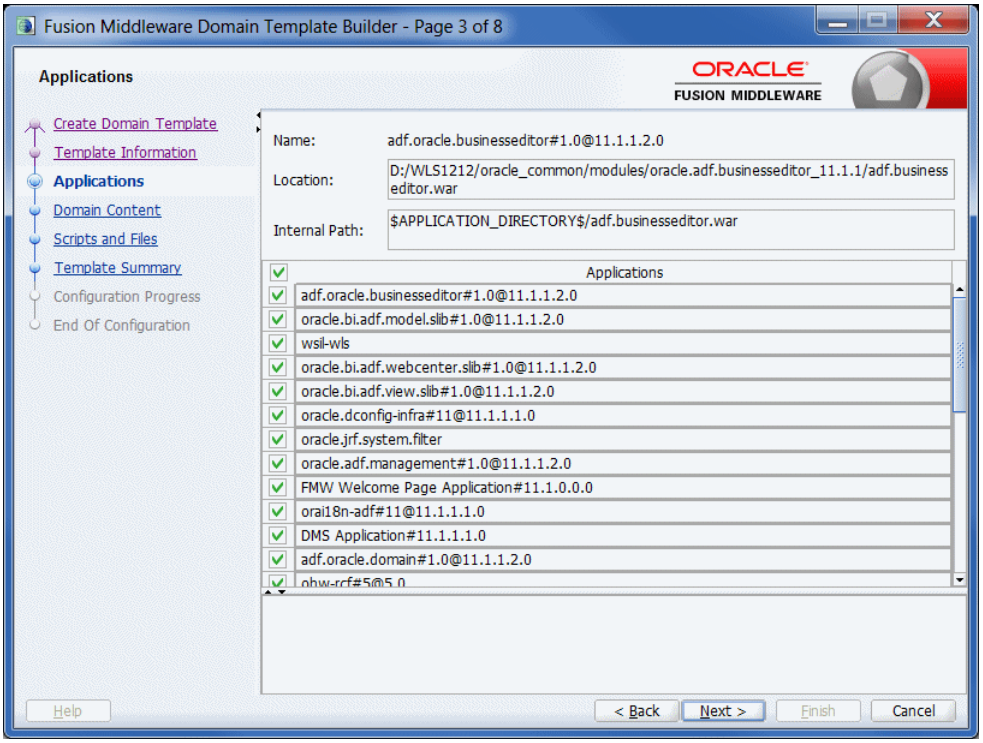


Use this screen to provide basic information about the template. The information you enter here is stored in the template-info.xml file that is generated for the domain or extension template. It is also stored in the config.xml file for the domain.

- If you selected a WebLogic domain or an application as the source for the new template, enter the information required for the domain and application, and click **Next**.
- If you selected a template as the source for the new template, information about the selected template is displayed. Review the information, and, if necessary, change it to suit the requirements of your domain or application template. When done, click **Next**.

Field	Description
Name	Required. Enter a valid name for the template. The name must be a string of characters, and can include spaces.
Version	Required. 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.
Category	Leave this field blank as it does not apply to custom templates. This field applies only to Oracle-provided templates that are selectable on the Templates screen of the Fusion Middleware Configuration Wizard.
Description	Optional. Enter a detailed description of the template. The description must be a string of characters, and can include spaces.

4.3 Applications



This screen is displayed only if applications are included in the template or domain that you selected. You can

- Review the list of applications that are currently defined in your template source.
- Review the paths of the current and imported applications. To do so, click on the name of the application in the **Applications** list.
- If necessary, omit applications from the list of files to be imported. To do so, deselect the check box.

By default, the Domain Template Builder includes applications from the domain or template you specified as the source for the template that you are creating. For example:

- If you selected a template as the source for the new template, and if the source template has a separately defined applications directory, all the application files defined in the applications directory are included by default.
- If you selected a domain as the source for the new template, applications that fulfill the following conditions are included by default:
  - Applications that are defined in the config.xml file by the app-deployment element
  - Applications that are not considered internal applications in your product installation are included by default

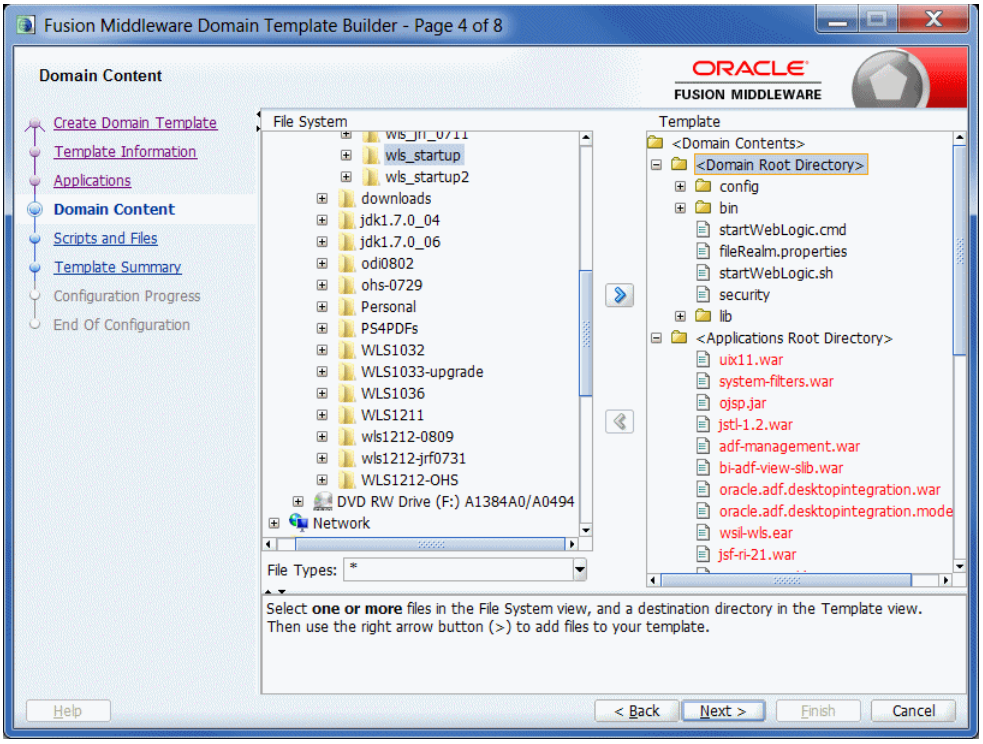
When you create or extend a clustered domain (by using the Configuration Wizard or WLST) based on a template that contains applications with application-scoped JDBC or JMS resources, you may need to perform certain additional steps after creating or extending the domain. These steps are needed to ensure that the application and its application-scoped resources are targeted and deployed properly.

For more information about targeting and deploying application-scoped modules, see *Deploying Applications to Oracle WebLogic Server*.

To include an application (or omit it from) the template, select or deselect the check box for the application as appropriate.

The listed applications reside in the directory specified in the **Location** box. The applications are copied into your template from this location. If more than one application is selected, the **Location** box contains the location of the first one in the list. To see the location of a specific application, click on the application name in the **Applications** column.

4.4 Domain Content



Use the Domain Content screen to review the files in the domain, add files to the domain, and remove files from the domain.

By default, the Domain Template Builder includes files from the WebLogic domain or template you specified as the source for the new template. You may also need to include other files from your domain or file system that are required by the applications in the domain.

- If you selected an existing template as the source for the new template, all files from the source template are automatically included. If the existing template defines a separate applications directory, the applications in the template are listed under the Applications Root Directory in the **Template** section on the right.
- If you selected a domain as the source for your new template, the following files and directories are included by default:
  - All files in the root directory with the following extensions: `.cmd`, `.sh`, `.xml`, `.properties`, and `.ini`
  - All files with the `.pem` extension defined in the SSL configuration for your domain
  - `bin` directory
  - `lib` directory
  - All files in the `security` directory that are not created automatically during domain creation

    For example, in WebLogic, `XACMLRoleMapperInit.ldift` is not created automatically, and therefore is included in the template by default when creating a template.

– All files and subdirectories in the `config` directory that are *not* included by the Domain Template Builder automatically during template creation. For example, `config.xml` and any JDBC and JMS resources defined in `config.xml` are not displayed in the **Template** section by default because the Domain Template Builder automatically includes them in the template when creating the template.

Note: By default, the `servers` directory is not included in the template. If you have configured additional security data such as users, groups, or roles, through the WebLogic Server Administration Console or other online tools, it is stored in the LDAP server and is not included in the template. You must export the data first and then import it into the target domain.

Reviewing Files in the Template

To view the current list of files in your template, expand the Domain Root Directory and Applications Root Directory under **Domain Contents** in the **Template** section on the right.

Adding Files and Directories to the Template

To add files to the template:

1. In the **File System** section on the left, navigate to the file you want to add. You can use the **Show File Types** drop-down list to restrict the list to a specific file type. For example, to display only files with the `.cmd` extension, select `*.cmd` from the drop-down list.
2. Select one or more files or directories to move into a specific subdirectory of either the Domain Root Directory or the Application Root Directory. The files you select must all be either domain or application files, not a mix of both. Select only those files that you want to move to a given subdirectory.
3. In the **Template** section on the right, select either the Domain Root Directory or Application Root Directory as appropriate for the files or directories you are adding.
4. Expand the directory and navigate to the subdirectory in which you want to store the selected files or directories.
5. Click the right arrow button to move the files or directories into the selected template directory.

   If you attempt to add a file that is created automatically by the Configuration Wizard at domain creation time, an error message is displayed.
6. Repeat steps 2 through 5 for any additional files or directories you need to add to the template.

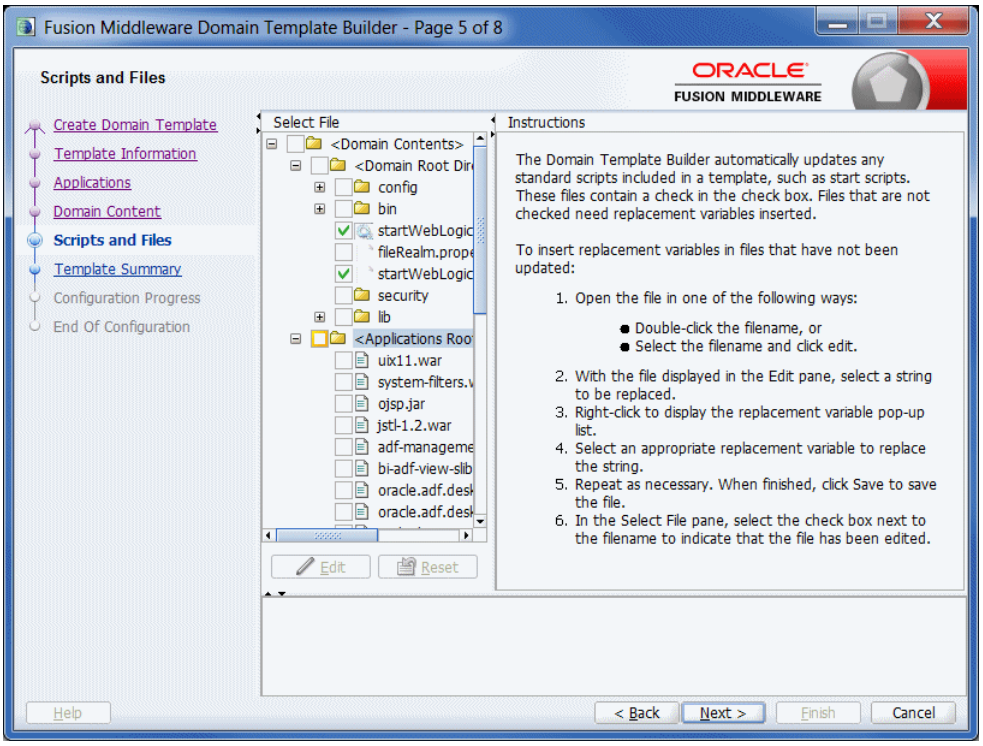
Removing Files and Directories From the Template

To remove files and directories:

1. In the **Template** section, navigate to and select all files and directories that you want to remove from the template.
2. Click the left arrow to remove the selected files or directories. The files are removed from the template. They are not copied or moved into any directory that may be currently selected in the **File System** section

You cannot remove files shown in red or their parent directories as they consist of the applications defined in the `config.xml` file in the template source. These files are included by default in the template being created. If you do not want these files included in your template, you must remove the application with which they are associated. To do so, click **Previous** to return to the Applications screen and deselect the appropriate check box for the application.

4.5 Scripts and Files



When you are creating a template, you want the scripts and files that you are packaging with your template to be free of local domain environment settings and therefore be ready for use by the Configuration Wizard.

The Domain Template Builder automatically updates any standard scripts included in a template (such as start scripts) by replacing hard-coded values for various domain environment settings with replacement variables. The Configuration Wizard can replace these variables with new hard-coded values during the configuration of a new WebLogic domain.

Use the Scripts and Files screen to replace hard-coded strings with replacement variables in files that have not been updated by the Domain Template Builder. The check box for any file that was automatically updated by the Domain Template Builder is already selected.

To insert replacement variables into your files:

1. Open the file in one of the following ways:
   - Double-click the appropriate filename in the **Select File** list.
   - Select the appropriate filename in the list and click **Edit**.

   The **Instructions** pane is replaced by an **Edit File** pane, in which the file is displayed. An editing toolbar is provided at the top of the **Edit File** pane.

2. To insert a replacement variable:

   a. Select the string to be replaced.

   b. Click the right mouse button. A list of replacement variables is displayed.

   c. Select the required variable.

   The replacement variable replaces the selected string. The Configuration Wizard later substitutes literal strings for the replacement variables to set up a specific WebLogic domain.

   For definitions of the available replacement variables, see Section 4.5.1, "Replacement Variable Definitions."

3. Repeat steps 1 and 2 for each string for which you want to substitute a variable.

4. Review the changes and click **Save** to preserve your changes, or **Revert** to replace the current version of the file with the last saved version. If you edited the file since the last time you saved it, you are prompted to confirm the revert operation.

   Note: You can also click **Reset** in the **Select File** pane to revert to the saved version of the file.

5. If you saved the edited file, select the check box next to the filename in the **Select File** pane.

6. Repeat steps 1 through 5 for each file in which you want to insert replacement variables.

7. Click **Next**.

   The Template Summary screen is displayed.

4.5.1 Replacement Variable Definitions

The following table defines the replacement variables most commonly used in files.

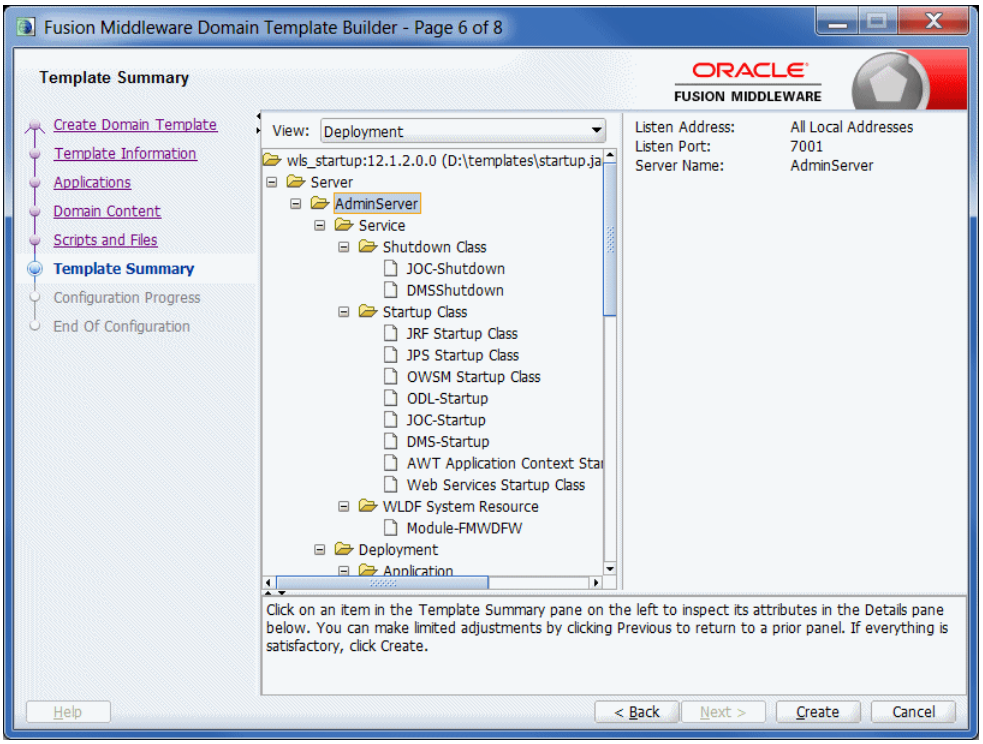
Table 4–1 List of Replacement Variables

Variable	Definition
ADMIN_SERVER_URL	URL specifying the listen address (host name or IP address) and port number of the Administration Server for the WebLogic domain.
APPLICATIONS_ DIRECTORY	Directory containing the applications required by the domain. By default, the path name for this directory is *ORACLE_HOME*\user_ projects\applications*domain_name*.
BEAHOME	The Oracle Home directory (*ORACLE_HOME*) in which the products associated with the domain are installed. On a Windows system, the default path for this directory is C:\Oracle\Middleware.
DOCSWEBROOT	The URL specifying the location of the product documentation on the Internet.
DOMAIN_HOME	Directory in which the domain resides.
DOMAIN_NAME	Name of the domain.
DOMAIN_USER	Administration username required to start the server.
JAVA_HOME	Location of the JDK used to start WebLogic Server.
JAVA_VENDOR	Vendor of the JVM, for example, Oracle, HP, IBM, and Sun.

Table 4–1 (Cont.) List of Replacement Variables

Variable	Definition
JAVA_VM	Mode in which you want the JVM to run, such as `-server` or `-hotspot`.
LISTEN_PORT	Port to be used to listen for regular, non-secure requests (through protocols such as HTTP and T3). The default value is 7001.
PRODUCTION_MODE	Mode in which the server is started. If this variable is set to true, the server starts in production mode; if set to false, the server starts in development mode.
SERVER_HOST	Name or address of the machine hosting the server.
SERVER_NAME	Name of the server to be started.
SERVER_PORT	Port to be used to listen for regular, non-secure requests (through protocols such as HTTP and T3). The default value is 7001.
SSL_PORT	Port to be used to listen for secure requests (through protocols such as HTTPS and T3S). The default value is 7002.
STARTMODE	Mode in which the server is started: development or production.
USERDOMAIN_HOME	Directory containing the user domains created with the Configuration Wizard. By default, the path for this directory is *ORACLE_HOME*\user_projects\domains.
WL_HOME	Root directory of your WebLogic Server installation. By default, this directory is defined as *ORACLE_HOME*\wlserver.

4.6 Template Summary



Use the Template Summary screen to review the settings for the new template before it is created.

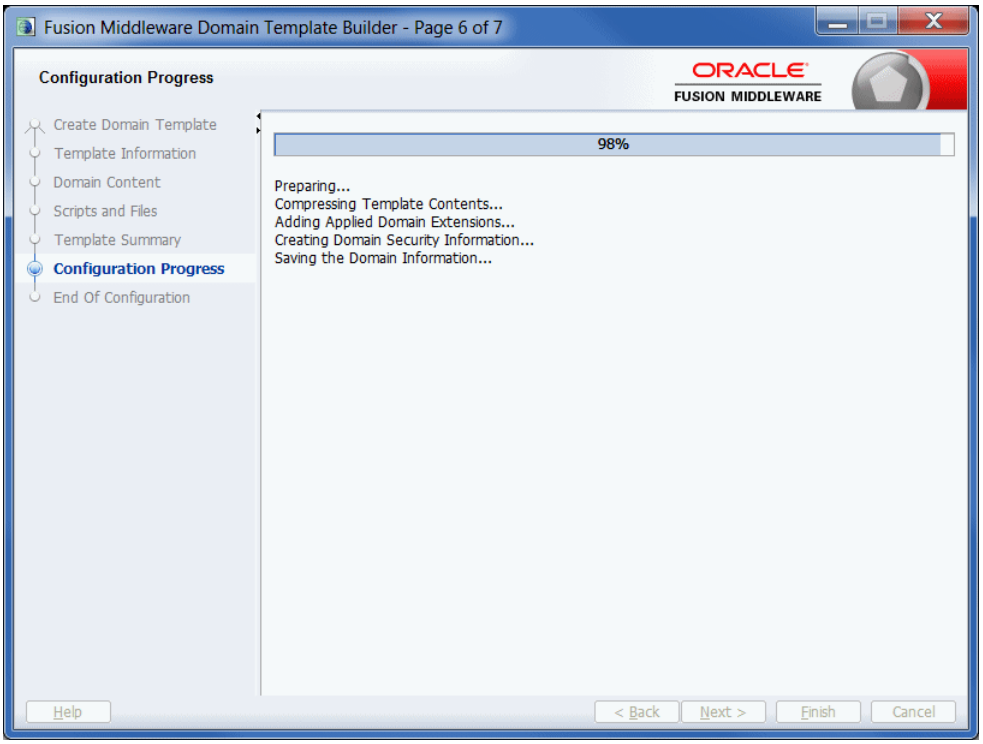
1. Select an item in the **Template Summary** pane on the left and review the associated details in the **Details** pane on the right. If you need to change any settings that you defined in previous windows, select **Previous** to return to the required configuration window.

   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.

4.7 Configuration Progress



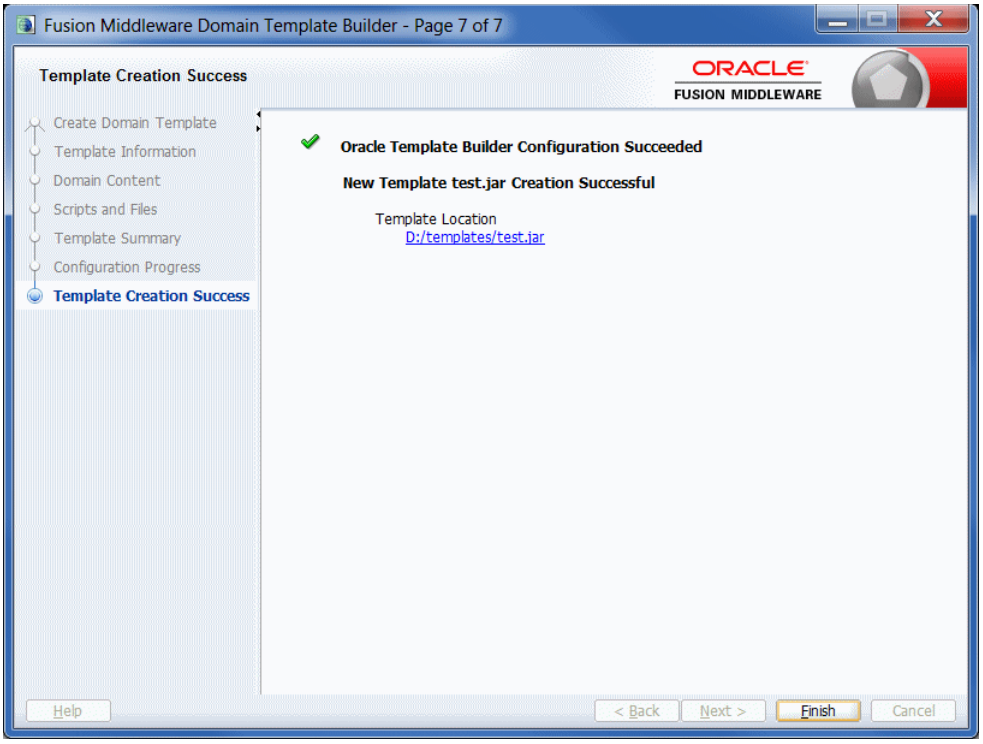
The Configuration Progress screen displays status messages during the template creation process. The Domain Template Builder generates a JAR file that contains all the components required to generate a new WebLogic domain, including the template settings that you provided.

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

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

For more information, see "Files Typically Included in a Template" in the *Domain Template Reference*.

4.8 End of Configuration



This screen displays the final configuration status.

- 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 are displayed. In this case, try to determine the cause of the failure from the message and correct the issue.