Oracle® Fusion Middleware
Creating Templates and Domains Using the Pack and Unpack Commands
Preface

This guide describes how to use the pack and unpack commands for creating WebLogic domains and templates.

- Intended Audience
- Documentation Accessibility
- Related Documents
- Conventions

Documentation Accessibility

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

Access to Oracle Support

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

Conventions

The following text conventions are used in this document:

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

Intended Audience

This guide is intended for Oracle Fusion Middleware administrators.
Related Documents

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

- Installing and Configuring Oracle WebLogic Server and Coherence
- Administering Node Manager for Oracle WebLogic Server
- Deploying Applications to Oracle WebLogic Server
- Administering Server Startup and Shutdown for Oracle WebLogic Server
- WLST Command Reference for WebLogic Server
Overview of the Pack and Unpack Commands

The pack and unpack commands provide a one-step method for creating WebLogic domains and templates from the command line. These commands are available in the ORACLE_HOME/oracle_common/common/bin directory.

Note:

You cannot use these commands to customize the contents of your WebLogic domain or template in the same way as with the other tools.

In addition to the pack and unpack commands, you can use the Configuration Wizard, Domain Template Builder, or WebLogic Scripting Tool to create WebLogic domains and templates.

You can create the following types of templates by using the pack command, and then use them in the unpack command to create either a WebLogic domain or a Managed Server domain:

- **Domain template**: This type of 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 use a domain template as the basis for new WebLogic domains.

- **Managed Server template**: This type of template defines the subset of resources within a WebLogic domain that are required to create a Managed Server domain directory on a remote system. You can create a Managed Server template by using the pack command with the -managed=true option.

When you then use the Managed Server template with the unpack command, the Managed Server domain directory that is created contains sufficient bootstrap information to start the Managed Server on the remote system. You can start the Managed Server on the remote system by using either the Node Manager or customized start scripts that were created when you unpacked the template on the remote system.

This chapter includes the following sections:

- **Introduction to the Pack Command**
  The pack command provides a method for creating a template from the command line in one step. However, it does not allow you to customize the content of your template the way you can customize by using the Domain Template Builder.

- **Introduction to the Unpack Command**
  The unpack command provides a one-step method for creating a WebLogic domain quickly from an existing template by using the default settings defined in the template.
Introduction to the Pack Command

The `pack` command provides a method for creating a template from the command line in one step. However, it does not allow you to customize the content of your template the way you can customize by using the Domain Template Builder.

The `pack` command helps you perform the following tasks quickly:

- Create a domain template that contains a snapshot of an entire working WebLogic domain.
  You can then use this template as the basis for a new WebLogic domain that you create by using either the `unpack` command, Configuration Wizard, or WLST.
- Create a Managed Server template that contains a subset of the files in a WebLogic domain that are required to create a Managed Server domain directory hierarchy on a remote system.
  You can then create the Managed Server domain directory on the remote system by using the `unpack` command.

Introduction to the Unpack Command

The `unpack` command provides a one-step method for creating a WebLogic domain quickly from an existing template by using the default settings defined in the template.

When creating a WebLogic domain, `unpack` does not provide the same customization options as the Configuration Wizard or WLST. However, if you use the `unpack` command with a domain template, you can do the following:

- Change the password for the default administrative user defined in the template.
- Add an administrative user when the default administrative user already has a password specified in the template.
- Specify the JDK and start mode for the WebLogic domain.
- Specify an applications directory, if one is supported by the template.

By using `unpack`, you can also create a Managed Server domain directory based on a Managed Server template that was created with the `pack` command. The Managed Server uses the settings defined for the Administration Server. Therefore, you cannot change the JDK or the start mode, add administrative users, or change the administrator password.

Note:

You cannot use `unpack` to extend a WebLogic domain.
Pack and Unpack Command Reference

The pack and unpack commands have unique syntaxes and parameters. The commands are located in the ORACLE_HOME/oracle_common/common/bin directory. This chapter includes the following sections:

- **The Pack Command**
  The pack command helps you create a template (a .jar archive file) that contains a snapshot of either an entire WebLogic domain or a subset of a WebLogic domain.

- **The Unpack Command**
  The unpack command helps you create a full WebLogic domain or a subset of a domain that is used for a Managed Server domain directory on a remote system.

## The Pack Command

The pack command helps you create a template (a .jar archive 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 WebLogic domain to create a Managed Server domain directory hierarchy on a remote machine.

The following sections describe the files and directories that are included in the domain and Managed Server templates that you create by using the pack command:

- **Files and Directories Included in Domain Templates**
- **Porting Security Provider Data to a New Domain**
- **Files and Directories Included in Managed Server Templates**
- **Node Manager Configuration**
- **Syntax of the Pack Command**
- **Parameters of the Pack Command**
- **Example of the Pack Command**

### Files and Directories Included in Domain Templates

All files and directories in the source WebLogic domain are included, with the following exceptions:

- Temporary files that are created when you start a server
- The servers directory
- Files in the security directory that are created automatically when you create the WebLogic domain, such as DefaultAuthenticatorInit.ldift and XACMLRoleMapperInit.ldift
- Persistent file stores, even if they are located with the domain. File stores are excluded whether they are defined for JMS or other file systems. If the file store is
located within the domain, an empty file store directory is created on the target system.

In addition to files within the domain directory, deployment plans located in external directories are included in the template that is created by pack. These resources are copied by unpack to the following standard location in the domain:

domain_home/config/deployments/deployment_name/plan

Note:

Plans already within the domain are not relocated.

External directories include locations outside of the Oracle home directory, the domain home, and any location defined in the ORACLE_HOME/oracle_common/common/lib/internalpaths.txt file.

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

Porting Security Provider Data to a New Domain

Because it is an offline command, the pack command does not export security provider data, such as embedded LDAP data, into a domain template. After using unpack 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 sign in to the Administration Console of the original domain.

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 sign 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.

Files and Directories Included in Managed Server Templates

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
- Any files with the .pem extension defined in the SSL configuration for your WebLogic domain
- All files and subdirectories in the config directory, except config.xml
- The bin directory
- The lib directory
The following files and directories are not included in a Managed Server template by default:

- Applications and certain application initialization files
- The config.xml file
- Temporary files that are created when you start a server
- The servers directory
- Files in the security directory that are created automatically when you create the WebLogic domain, such as DefaultAuthenticatorInit.ldift and XACMLRoleMapperInit.ldift

You can pack any domain, whether or not it specifies specific names or ports for the servers in the domain. You can then unpack the domain on any system, but servers that are defined in the config.xml file to use specific systems and ports will not work on other systems. Therefore, you must unpack the same template on each remote node for all servers to start successfully on their appropriate nodes and systems.

Node Manager Configuration

The pack and unpack commands do not preserve Node Manager configuration for both managed and non-managed modes. By default, the Node Manager type is always set to PerDomainNodeManager in the unpacked domain.

If the Node Manager type in the original domain was CustomLocationNodeManager when the domain was unapcked, you must include the -nodeManagerType and -nodeManagerHome parameters in the unpack command to preserve the configuration.

If the Node Manager type in the original domain was ManualNodeManagerSetup, you must manually reconfigure Node Manager after you unpack the domain on each remote system.

If a ManualNodeManagerSetup configuration was used for the original domain, the Node Manager configuration information is discarded from the template that pack Creates. When you unpack the domain, specify ManualNodeManagerSetup using the -nodeManagerType parameter. After you unpack the domain, you must manually configure Node Manager.

For information about Node Manager types, see Default Node Manager Configuration in Administering Node Manager for Oracle WebLogic Server.

Node Manager credentials are copied from the packed domain to the unpacked domain. These credentials are stored in config.xml file.

Syntax of the Pack Command

```
pack -domain=domain -template=template -template_name="template_name"
[-template_author="author"] [-template_desc="description"] [-managed=true|false]
[-log=log_file] [-log_priority=log_priority]
```

Parameters of the Pack Command

Table 2-1 lists the parameters of the pack command.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required or Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-domain=domain</td>
<td>Required</td>
<td>The full or relative path of the WebLogic domain from which the template is to be created.</td>
</tr>
</tbody>
</table>
| -template=template              | Required             | The full or relative path and file name of the template to be created. The template filename must include the .jar extension.  
**Note:** The _pack_ command does not overwrite existing files. If the file name that you specify matches the name of an existing file in the specified folder, the _pack_ command fails. |
| -template_name="template name" | Required             | The descriptive name for the template. Quotes are required only if the value contains spaces.                                                                                                               |
| -template_author="author"      | Optional             | The name of the author of the template. Quotes are required only if the value contains spaces.                                                                                                               |
| -template_desc="description"   | Optional             | The description of the template. Quotes are required only if the value contains spaces.                                                                                                                   |
| -managed=true|false                | Optional             | Specifies whether the template is to be used to create Managed Servers on remote systems. The default is _false_. When this parameter is set to _true_, a Managed Server template is created that contains a minimal set of files, including SerializedSystemIni.dat and nm_password.properties. It also includes a domain.properties file that is unique to the Managed Server template. Applications and certain application initialization files are not included. The resulting template can be used to create Managed Servers on remote systems.  
**Note:** As of WebLogic Server 12.1.2, you can use the WLST writeTemplate Command in the online mode. This feature makes it easier to create or update remote Managed Server domains. For more information about the writeTemplate command, see writeTemplate in [WLST Command Reference for WebLogic Server](https://docs.oracle.com/en/middleware/weblogic/12.2.1.5/wlst/docs/wlstcmdref.html). |
| -log=log_file                   | Optional             | The name of the log file.                                                                                                                                                                                   |
Table 2-1  (Cont.) Parameters of the Pack Command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required or Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-log_priority=log_priority</td>
<td>Optional</td>
<td>The priority setting for the log file. Use a log4j priority string. Valid log4j priority strings are debug, info, warn, error, and fatal. The priority string values correspond to the levels defined in the Level class. See <a href="http://logging.apache.org/log4j/">http://logging.apache.org/log4j/</a>.</td>
</tr>
</tbody>
</table>

Example of the Pack Command

To create a template JAR file based on an existing WebLogic domain named mydomain that is located in the C:/oracle/user_projects/domains directory, run the following command:

```
pack -domain=C:/oracle/user_projects/domains/mydomain -template=C:/oracle/user_templates/mydomain.jar -template_name="My WebLogic Domain"
```

A template file named mydomain.jar is created in the C:/oracle/user_templates directory. The name of the template is My WebLogic Domain.

The Unpack Command

The `unpack` command helps you create a full WebLogic domain or a subset of a domain that is used for a Managed Server domain directory on a remote system.

You must use `unpack` only with a template that is compatible with your current installation. The current installation binaries must exist on the system on which you are running the `unpack` command. The template can be any of the following:

- A domain template that is packaged with your current installation
- A domain template created by using the Domain Template Builder or WLST
- A domain template created by using the `pack` command
- A Managed Server template created by using the `pack` command

When you use the `unpack` command with a domain template, it creates a WebLogic domain that contains all the application and resource files defined in the template. It also creates necessary start scripts, and certain security and configuration files.

**Note:**

The source domain must have both an administrator user and an administrator password defined for the domain. Otherwise, an error is displayed when you run the `unpack` command. You can define the administrator password for the source domain in the wallet that is specified by the `unpack` command's `walletDir` option.
When you use the `unpack` command with a Managed Server template, it creates a Managed Server domain directory that includes the following:

- A customized start script for each Managed Server in the WebLogic domain
- The `config_bootstrap.xml` file (based on the `config.xml` file in the template)
- The `nm_password.properties` file
- The `SerializedSystemIni.dat` file

An entry for the Managed Server domain directory is also created in the `NM_HOME/nodemanager.domains` file, where `NM_HOME` is the node manager home directory for the product installation on the remote system. The location of this directory depends on the Node Manager type. For more information about Node Manager type and Node Manager home, see Table 2-2.

**Note:**

By default, application files are not included in a Managed Server template that is created by using the `pack` command. If an application in the WebLogic domain from which you have created a Managed Server template is deployed by using the `external_stage` mode, the Managed Server domain directory that you create with the `unpack` command does not contain any of the externally staged applications. Before you start the Managed Server, you must ensure that it has access to the externally staged application files.

For more information about the deployment staging mode, see Controlling Deployment File Copying with Staging Modes in Deploying Applications to Oracle WebLogic Server.

- **Syntax of the Unpack Command**
- **Parameters of the Unpack Command**
- **Example for the Unpack Command**

**Syntax of the Unpack Command**

```
unpack [-initialize] -template=template -domain=domain [-walletDir=directory]
[-user_name=username] [-password=password] [-app_dir=application_directory]
[-nodemanager_type=type] [-nodemanager_home=nodemanager_home_directory]
[-java_home=java_home_directory] [-server_start_mode=dev|prod]
[-overwrite_domain] [-log=log_file] [-log_priority=log_priority]
```

**Parameters of the Unpack Command**

Table 2-2 lists the parameters that you can use with the `unpack` command.
Table 2-2  Parameters of the Unpack Command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required or Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-initialize</td>
<td>Optional</td>
<td>This parameter applies only to WebLogic domains in which Fusion Middleware products are installed.</td>
</tr>
</tbody>
</table>

**Note:**

When setting this parameter to true, ensure that the source database is running. If it is not running, the unpack command fails.

When this parameter is true and the managed parameter for pack was set to false when the domain was packed, it initializes all config-mapping.xml values in the Service Table and the Oracle Web Services Manager (OWSM) data store regardless of the reprocess attribute setting for the entry in config-mapping.xml.

When this parameter is true and the managed parameter for pack was set to true when the domain was packed, it initializes only those config-mapping.xml values in the Service Table and the Oracle Web Services Manager (OWSM) data store for which the reprocess attribute setting for the entry in config-mapping.xml is set to true.

When this parameter is false, no config-mapping.xml values are initialized.

For information about Service Tables, see Understanding Service Tables.
Table 2-2  (Cont.) Parameters of the Unpack Command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required or Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-template=template</td>
<td>Required</td>
<td>The full or relative path and name of the template from which the WebLogic domain is to be created.</td>
</tr>
<tr>
<td>-domain=domain</td>
<td>Required</td>
<td>The full or relative path of the WebLogic domain to be created.</td>
</tr>
</tbody>
</table>

Tables in *Administering Oracle Fusion Middleware*. 

Chapter 2
The Unpack Command
### Table 2-2  (Cont.) Parameters of the Unpack Command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required or Optional</th>
<th>Description</th>
</tr>
</thead>
</table>
| -walletDir  | Optional             | The directory location of the wallet file that contains the password of the administration user account. This directory is the same directory that you used when you created the wallet file by using the `configWallet.sh` command. For example:  
- `walletDir $HOME/wallet`  
If you want to create a different administrator user for the domain that you are unpacking, do the following:  
1. In the `createWallet.sh` command, include the `-create admin_name` parameter to define the administrator user name.  
2. In the `unpack` command, include the `-user_name=admin_name` parameter, using the same `admin_name` as you used in the `createWallet` command.  

---

**Note:**

The `wallet Dir` option must be used even when the `managed` parameter is set to false, whether you want to use a different administrator.
Table 2-2  (Cont.) Parameters of the Unpack Command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required or Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>user or not.</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Required or Optional</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-user_name=username</td>
<td>Optional</td>
<td>The user name for the default administrator as currently defined in the template or through the WalletDir directory. This user name is used to start the Administration Server and connect to it. If you specify a new password for the default administrator, the password is reset to the value specified. If no administrative users are defined in the template, you can create a new default administrator by using createWallet.sh and the -user_name parameter or by specifying the -user_name parameter. If a default administrator and password are currently defined in the template, you can add a new administrative user by using createWallet.sh and the -user_name parameter or by specifying the -user_name and password parameters. User names must be unique. For the WebLogic Authentication provider (DefaultAuthenticator), user names are case insensitive. User names must not include tabs or any of the following character: • Semicolons (;) • Commas (,) • Plus signs (+) • Equal signs (=) • Single backslash character ()</td>
</tr>
</tbody>
</table>

**Note:**

You can use two consecutive backslashes. For example smith\. In addition, user names must not begin with the following characters: • Pound sign (#) • Double quotations ("")
### Table 2-2  (Cont.) Parameters of the Unpack Command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required or Optional</th>
<th>Description</th>
</tr>
</thead>
</table>

**Note:**

If you create a user name with any of the preceding invalid characters, the WebLogic domain can become corrupted.

A valid password consists of a string of at least eight characters. The `unpack` command encrypts the password value.

**Note:**

This parameter is not used if you are creating a WebLogic domain by using a Managed Server template.
Table 2-2  (Cont.) Parameters of the Unpack Command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required or Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-password=password</td>
<td>Optional</td>
<td>This parameter is deprecated. For better security, use the walletDir parameter to specify the password for the administrator user. If you use this parameter:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• If you specify a new password for the default administrator, the password is reset to the value specified.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• If a password for the default administrator is not defined in the template, you must specify one.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A valid password consists of a string of at least eight case-sensitive characters. The unpack command encrypts the password value.</td>
</tr>
</tbody>
</table>

**Note:**

When you unpack a template that was created by using the managed option, you cannot change the password. The -password option is ignored.

---

Chapter 2

The Unpack Command

2-13
### Table 2-2 (Cont.) Parameters of the Unpack Command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required or Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-app_dir=application_directory</td>
<td>Optional</td>
<td>The full path of the directory that is used to store the applications that are defined in the template.</td>
</tr>
</tbody>
</table>

**Note:**

If the template includes applications that are located outside of the product installation directory and in a separate applications directory, then this option specifies where to copy the applications from the template. If the template does not contain any of these applications, then specify
Table 2-2 (Cont.) Parameters of the Unpack Command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required or Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-nodemanager_type=type</td>
<td>Optional</td>
<td>Sets the Node Manager type for the domain that you are creating. Specify one of the following types:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• PerDomainNodeManager: If you specify this type, the Node Manager home is defined within the domain as <code>&lt;domain_name&gt;/nodemanager</code>. The Node Manager for each domain can have a different configuration, as determined by the files in this directory.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• CustomLocationNodeManager: Specify this type if you want the Node Manager configuration files to be created in a specific location. If you specify this type, you must include the <code>nodemanager_home</code> argument to specify the Node Manager home directory. The specified directory must be empty.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ManualNodeManagerSetup: Specify this type if you want to use an existing Node Manager configuration. If specified, the Node Manager configuration for the domain is ignored, and you must manually configure the Node Manager configuration. The default value is PerDomainNodeManager.</td>
</tr>
<tr>
<td>-nodemanager_home=home</td>
<td>Conditional</td>
<td>If the specified Node Manager type is CustomLocationNodeManager, this argument is required. Enter the full path of the Node Manager location you want to use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the specified Node Manager type is not CustomLocationNodeManager, this argument is not required. If included, it is ignored.</td>
</tr>
</tbody>
</table>
### Table 2-2  (Cont.) Parameters of the Unpack Command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required or Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-java_home=java_home_directory</code></td>
<td>Optional</td>
<td>The full path of the Java home directory. When set, this parameter identifies the default Java Virtual Machine (JVM) used by the Administration Server for the WebLogic domain. This parameter is not applicable if you are creating a WebLogic domain based on a Managed Server template.</td>
</tr>
</tbody>
</table>
Table 2-2 (Cont.) Parameters of the Unpack Command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required or Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-server_start_mode=dev</td>
<td>prod</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The start mode for the Administration Server: dev or prod. The default value is dev.</td>
</tr>
</tbody>
</table>

**Note:**

If you want the unpacked domain to run in production mode, you must include this parameter in the unpack command and specify prod. Otherwise, the unpacked command is created in development mode.

This parameter is not applicable if you are creating a WebLogic domain by using a Managed Server template.

If dev mode is specified or if this parameter is omitted, a boot.properties file is created for the domain. If prod mode is specified, a boot.properties file is not created for the domain.
Table 2-2  (Cont.) Parameters of the Unpack Command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required or Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-overwrite_domain</td>
<td>Optional</td>
<td>If the specified domain directory exists and is not empty, the files in the directory are automatically overwritten without warning. No attempt is made to merge or preserve local customizations.</td>
</tr>
<tr>
<td>-log=log_file</td>
<td>Optional</td>
<td>The name of the log file.</td>
</tr>
<tr>
<td>-log_priority=log_priority</td>
<td>Optional</td>
<td>The priority setting for the log file. Use a log4j priority string. Valid log4j priority strings are debug, info, warn, error, and fatal. The priority string values correspond to the levels defined in the Level class. For more information, see <a href="http://logging.apache.org/log4j/">http://logging.apache.org/log4j/</a>.</td>
</tr>
</tbody>
</table>

Note:
If a system error occurs after domain generation starts, the domain directory will be in an unknown state.

Example for the Unpack Command

To create a WebLogic domain named mynewdomain in the C:/oracle/user_projects/domains directory by using the mydomain.jar template in the C:/oracle/user_templates directory, run the following command:

```
unpack -template=C:/oracle/user_templates/mydomain.jar
-domain=C:/oracle/user_projects/domains/mynewdomain
```
Creating and Starting a Managed Server on a Remote System

You can create and start a Managed Server on a remote system by using the `pack` and `unpack` commands. In some WebLogic domains, if you want to run a Managed Server on a system that is remote from the Administration Server for the WebLogic domain, perform the steps described in this chapter.

This chapter includes the following sections:

- **Creating a Managed Server Template**
  You can create a Managed Server template by executing the `pack` command on an existing WebLogic domain that includes the definition of one or more Managed Servers and contains Managed Server definitions in the `config.xml` file.

- **Creating a Managed Server on a Remote System**
  Before you create a Managed Server on a remote system, ensure that the IP address and port number of the remote system matches the definition of the Managed Servers that are specified in the Managed Server template.

- **Starting Managed Servers on a Remote System**
  When you use the `unpack` command to create a WebLogic domain directory for Managed Servers, it contains a customized start script for each Managed Server that is targeted on the current remote system. You can use these customized scripts to start the associated Managed Server.

### Creating a Managed Server Template

You can create a Managed Server template by executing the `pack` command on an existing WebLogic domain that includes the definition of one or more Managed Servers and contains Managed Server definitions in the `config.xml` file.

To create a Managed Server template:

1. From the command line on the local machine (that is, the machine that contains the Administration Server and the definition of Managed Servers), navigate to the `ORACLE_HOME/oracle_common/common/bin` directory.

2. Run the following command:

   ```bash
   pack -managed=true -domain=domain -template=template.jar -template_name="template_name"
   ```

   In this command:
   - `domain` is the full or relative path of the WebLogic domain from which the template is to be created.
   - `template.jar` is the full or relative path of the template including the file name of the template to be created.
   - `template_name` is a descriptive name for the template, enclosed in quotes.
For example, the following command creates a Managed Server template named mydomainManaged.jar from a WebLogic domain named mydomain.

```
pack -managed=true -domain=C:/oracle/user_projects/domains/mydomain -template=C:/oracle/user_templates/mydomain_managed.jar -template_name="My Managed Server Domain"
```

Creating a Managed Server on a Remote System

Before you create a Managed Server on a remote system, ensure that the IP address and port number of the remote system matches the definition of the Managed Servers that are specified in the Managed Server template.

1. Install WebLogic Server on the systems on which you want to host Managed Servers for the WebLogic domain.

   Note:
   All WebLogic Server instances within a WebLogic domain must run the same version of the WebLogic Server software. For more information about installing WebLogic Server, see Installing the Oracle WebLogic Server and Coherence Software in Installing and Configuring Oracle WebLogic Server and Coherence.

2. Establish a session with the remote system. You can use any valid method, such as telnet, to do so.

3. Copy the Managed Server template to the remote system.

4. On the remote system, navigate to the ORACLE_HOME\oracle_common\common\bin directory.

5. Run the following command:

   ```
   unpack -domain=domain -template=template.jar
   ```

   In this command:
   - `domain` is the full or relative path of the domain to be created.
   - `template.jar` is the full or relative path of the Managed Server template that you copied to the system in step 3.

   For example, the following command creates a WebLogic domain named myManagedDomain.

   ```
   unpack -domain=C:\oracle\user_projects\domains\myManagedDomain -template=C:\oracle\user_templates\mydomain_managed.jar
   ```

Starting Managed Servers on a Remote System

When you use the `unpack` command to create a WebLogic domain directory for Managed Servers, it contains a customized start script for each Managed Server that
is targeted on the current remote system. You can use these customized scripts to start the associated Managed Server.

For example, if you create a WebLogic domain that contains two Managed Servers, my_managed_server1 and my_managed_server2, and you target the servers to system m1, when you create the Managed Server domain directory on system m1, four custom start scripts are created: startmy_managed_server1.cmd, startmy_managed_server1.sh, startmy_managed_server2.cmd, and startmy_managed_server2.sh. You can use these scripts to start the corresponding Managed Servers. Alternatively, you can use the startManagedWebLogic script with the required parameters.

1. Start the Administration Server for the WebLogic domain as described in Starting and Stopping Servers in Administering Server Startup and Shutdown for Oracle WebLogic Server.

2. On the remote system, navigate to the directory for the WebLogic domain that you created in Creating a Managed Server on a Remote System.

3. Start the Managed Server on the remote system.
   - On a Windows system, run one of the following commands at the command prompt:
     
     ```
     startmy_managed_server
     startManagedWebLogic my_managed_server admin-url
     ```
   - On a UNIX system, run one of the following commands:
     
     ```
     ./startmy_managed_server.sh
     ./startManagedWebLogic.sh my_managed_server admin-url
     ```

   In these commands, my_managed_server is the name of the Managed Server to be started and admin-url is the listen address (host name or IP address) and port number of the system that hosts the Administration Server. For your convenience, the startManagedWebLogic_Readme.txt file provides a list of all the Managed Servers and the admin-url for the WebLogic domain.

   **Note:**

   You can also start Managed Servers by using Node Manager as described in Use Node Manager to Start a Managed Server in Administering Node Manager for Oracle WebLogic Server.