

# Oracle® Fusion Middleware

## Creating Templates and Domains Using the Pack and Unpack Commands



12c (12.2.1.3.0)

E95530-06

July 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 Templates and Domains Using the Pack and Unpack Commands, 12c (12.2.1.3.0)

E95530-06

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 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" or "commercial computer software documentation" 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

---

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

## 1 Overview of the Pack and Unpack Commands

---

Introduction to the Pack Command	1-2
Introduction to the Unpack Command	1-2

## 2 Pack and Unpack Command Reference

---

The Pack Command	2-1
Files and Directories Included in Domain Templates	2-1
Porting Security Provider Data to a New Domain	2-2
Files and Directories Included in Managed Server Templates	2-2
Node Manager Configuration	2-3
Syntax of the Pack Command	2-4
Parameters of the Pack Command	2-4
Example of the Pack Command	2-5
The Unpack Command	2-6
Syntax of the Unpack Command	2-7
Parameters of the Unpack Command	2-7
Example for the Unpack Command	2-13

## 3 Creating and Starting a Managed Server on a Remote System

---

Creating a Managed Server Template	3-1
Creating a Managed Server on a Remote System	3-2
Starting Managed Servers on a Remote System	3-2

# Index

---

# 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](#)
- [Diversity and Inclusion](#)

## 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:

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

## 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.

## 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*

# 1

## 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.

# 2

## 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

 **Note:**

Do not pack all the files under the `bin` directory into the template, but only the non-autogenerated files. The autogenerated files are re-created during the unpack process and you may lose the changes.

To check the complete file list, use the following command:

```
$DOMAIN_HOME/init-info/startscript.xml
```

- 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 unpacked, 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=log_file] [-log_priority=log_priority]
```

## Parameters of the Pack Command

Table 2-1 lists the parameters of the `pack` command.

**Table 2-1 Parameters of the Pack Command**

Parameter	Required or Optional	Description
<code>-domain=domain</code>	Required	The full or relative path of the WebLogic domain from which the template is to be created.
<code>-template=template</code>	Required	The full or relative path and file name of the template to be created. The template filename must include the <code>.jar</code> extension.  <b>Note:</b> The <code>pack</code> 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 <code>pack</code> command fails.
<code>-template_name="template name"</code>	Required	The descriptive name for the template. Quotes are required only if the value contains spaces.
<code>-template_author="author"</code>	Optional	The name of the author of the template. Quotes are required only if the value contains spaces.
<code>-template_desc="description"</code>	Optional	The description of the template. Quotes are required only if the value contains spaces.

Table 2-1 (Cont.) Parameters of the Pack Command

Parameter	Required or Optional	Description
<code>-managed=true false</code>	Optional	Specifies whether the template is to be used to create Managed Servers on remote systems. The default is <code>false</code> . When this parameter is set to <code>true</code> , a Managed Server template is created that contains a minimal set of files, including <code>SerializedSystemIni.dat</code> and <code>nm_password.properties</code> . It also includes a <code>domain.properties</code> 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.  <b>Note:</b> As of WebLogic Server 12.1.2, you can use the WLST <code>writeTemplate</code> command in the online mode. This feature makes it easier to create or update remote Managed Server domains. For more information about the <code>writeTemplate</code> command, see <code>writeTemplate</code> in <i>WLST Command Reference for WebLogic Server</i> .
<code>-log=log_file</code>	Optional	The name of the log file.
<code>-log_priority=log_priority</code>	Optional	The priority setting for the log file. Use a log4j priority string. Valid log4j priority strings are <code>debug</code> , <code>info</code> , <code>warn</code> , <code>error</code> , and <code>fatal</code> . The priority string values correspond to the levels defined in the <code>Level</code> class. See <a href="http://logging.apache.org/log4j/">http://logging.apache.org/log4j/</a> .

## 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

Parameter	Required or Optional	Description
-initialize	Optional	<p>This parameter applies only to WebLogic domains in which Fusion Middleware products are installed.</p> <p><b>Note:</b> When setting this parameter to <code>true</code>, ensure that the source database is running. If it is not running, the <code>unpack</code> command fails.</p> <p>When this parameter is <code>true</code> and the managed parameter for <code>pack</code> was set to <code>false</code> 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 <code>reprocess</code> attribute setting for the entry in config-mapping.xml.</p> <p>When this parameter is <code>true</code> and the managed parameter for <code>pack</code> was set to <code>true</code> 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 <code>reprocess</code> attribute setting for the entry in config-mapping.xml is set to <code>true</code>.</p> <p>When this parameter is <code>false</code>, no config-mapping.xml values are initialized.</p> <p>For information about Service Tables, see Understanding Service Tables in <i>Administering Oracle Fusion Middleware</i>.</p>
-template=template	Required	The full or relative path and name of the template from which the WebLogic domain is to be created.
-domain=domain	Required	The full or relative path of the WebLogic domain to be created.

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

Parameter	Required or Optional	Description
<code>-walletDir</code>	Optional	<p>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 <code>configWallet.sh</code> command. For example:</p> <pre><code>-walletDir \$HOME/wallet</code></pre> <p>If you want to create a different administrator user for the domain that you are unpacking, do the following:</p> <ol style="list-style-type: none"><li>1. In the <code>configWallet.sh</code> command, include the <code>-create admin_name</code> parameter to define the administrator user name.</li><li>2. In the <code>unpack</code> command, include the <code>-user_name=admin_name</code> parameter, using the same <code>admin_name</code> as you used in the <code>configWallet.sh</code> command.</li></ol> <p><b>Note:</b> The <code>walletDir</code> option must be used even when the <code>managed</code> parameter is set to <code>false</code>, whether you want to use a different administrator user or not.</p>

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

Parameter	Required or Optional	Description
<code>-user_name=username</code>	Optional	<p>The user name for the default administrator as currently defined in the template or through the <code>WalletDir</code> 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.</p> <p>If no administrative users are defined in the template, you can create a new default administrator by using <code>configWallet.sh</code> and the <code>-user_name</code> parameter or by specifying the <code>-user_name</code> parameter.</p> <p>If a default administrator and password are currently defined in the template, you can add a new administrative user by using <code>configWallet.sh</code> and the <code>-user_name</code> parameter or by specifying the <code>-user_name</code> and <code>password</code> parameters.</p> <p>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:</p> <ul style="list-style-type: none"> <li>• Semicolons (;)</li> <li>• Commas (,)</li> <li>• Plus signs (+)</li> <li>• Equal signs (=)</li> <li>• Single backslash character (\)</li> </ul> <p><b>Note:</b> You can use two consecutive backslashes. For example <code>smith\\</code>.</p> <p>In addition, user names must not begin with the following characters:</p> <ul style="list-style-type: none"> <li>• Pound sign (#)</li> <li>• Double quotations (")</li> </ul> <p><b>Note:</b> If you create a user name with any of the preceding invalid characters, the WebLogic domain can become corrupted.</p> <p>A valid password consists of a string of at least eight characters. The <code>unpack</code> command encrypts the password value.</p> <p><b>Note:</b> This parameter is not used if you are creating a WebLogic domain</p>

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

Parameter	Required or Optional	Description
		by using a Managed Server template.
<code>-password=password</code>	Optional	<p>This parameter is deprecated. For better security, use the <code>walletDir</code> parameter to specify the password for the administrator user. If you use this parameter:</p> <ul style="list-style-type: none"> <li>• If you specify a new password for the default administrator, the password is reset to the value specified.</li> <li>• If a password for the default administrator is not defined in the template, you must specify one.</li> </ul> <p>A valid password consists of a string of at least eight case-sensitive characters. The <code>unpack</code> command encrypts the password value.</p> <p><b>Note:</b> When you unpack a template that was created by using the managed option, you cannot change the password. The <code>-password</code> option is ignored.</p>
<code>-app_dir=application_directory</code>	Optional	<p>The full path of the directory that is used to store the applications that are defined in the template.</p> <p><b>Note:</b> 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 specifying the <code>app_dir</code> option has no effect and is ignored.</p>

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

Parameter	Required or Optional	Description
<code>-nodemanager_type=type</code>	Optional	<p>Sets the Node Manager type for the domain that you are creating. Specify one of the following types:</p> <ul style="list-style-type: none"> <li><code>PerDomainNodeManager</code>: 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.</li> <li><code>CustomLocationNodeManager</code>: 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.</li> <li><code>ManualNodeManagerSetup</code>: 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.</li> </ul> <p>The default value is <code>PerDomainNodeManager</code>.</p>
<code>-nodemanager_home=home</code>	Conditional	<p>If the specified Node Manager type is <code>CustomLocationNodeManager</code>, this argument is required. Enter the full path of the Node Manager location you want to use.</p> <p>If the specified Node Manager type is not <code>CustomLocationNodeManager</code>, this argument is not required. If included, it is ignored.</p>
<code>-java_home=java_home_directory</code>	Optional	<p>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.</p> <p>This parameter is not applicable if you are creating a WebLogic domain based on a Managed Server template.</p>

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

Parameter	Required or Optional	Description
<code>-server_start_mode=dev prod</code>	Optional	<p>The start mode for the Administration Server: <code>dev</code> or <code>prod</code>. The default value is <code>dev</code>.</p> <p><b>Note:</b> If you want the unpacked domain to run in production mode, you must include this parameter in the <code>unpack</code> command and specify <code>prod</code>. Otherwise, the unpacked command is created in development mode.</p> <p>This parameter is not applicable if you are creating a WebLogic domain by using a Managed Server template.</p> <p>If <code>dev</code> mode is specified or if this parameter is omitted, a <code>boot.properties</code> file is created for the domain. If <code>prod</code> mode is specified, a <code>boot.properties</code> file is <i>not</i> created for the domain.</p>
<code>-overwrite_domain</code>	Optional	<p>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.</p> <p><b>Note:</b> If a system error occurs after domain generation starts, the domain directory will be in an unknown state.</p>
<code>-log=log_file</code>	Optional	The name of the log file.
<code>-log_priority=log_priority</code>	Optional	<p>The priority setting for the log file. Use a log4j priority string.</p> <p>Valid log4j priority strings are <code>debug</code>, <code>info</code>, <code>warn</code>, <code>error</code>, and <code>fatal</code>. The priority string values correspond to the levels defined in the <code>Level</code> class. For more information, see <a href="http://logging.apache.org/log4j/">http://logging.apache.org/log4j/</a>.</p>

## 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
```

# 3

## 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.

### 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:

```
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 `mydomain_managed.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.

The custom start scripts are generated for the servers that either have no machine name assigned or the machine name of the server matches the host name or IP

address of the machine where `unpack` is run. 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`, with none of the servers targeted to a machine, and then when you create the Managed Server domain, four custom start scripts are created:

- `startmy_managed_server1.cmd`
- `startmy_managed_server1.sh`
- `startmy_managed_server2.cmd`
- `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*.

# Index