This document describes how to use the pack command to create WebLogic domain templates and how to use the unpack command to create WebLogic domains.
## Contents

**Preface** 
Intended Audience 
Documentation Accessibility 
Related Documents 

1 **Overview of the Pack and Unpack Commands** 
   1.1 Introduction to the Pack Command 
   1.2 Introduction to the Unpack Command 

2 **pack and unpack Command Reference** 
   2.1 The pack Command 
      2.1.1 Syntax of the pack Command 
      2.1.2 Parameters of the pack Command 
      2.1.3 Example for the pack Command 
   2.2 The unpack Command 
      2.2.1 Syntax of the unpack Command 
      2.2.2 Parameters of the unpack Command 
      2.2.3 Example for the unpack Command 

3 **Creating and Starting a Managed Server on a Remote Machine** 
   3.1 Creating a Managed Server Template 
   3.2 Creating a Managed Server on a Remote Machine 
   3.3 Starting Managed Servers on a Remote Machine
Preface

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

Intended Audience

This guide is intended for Oracle Fusion Middleware administrators.

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 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
Overview of the Pack and Unpack Commands

The pack and unpack commands provide a simple, one-step method for creating WebLogic domains and templates from the command line. The pack and unpack commands are available in the \common\bin subdirectory of the product installation directory.

Notes: You cannot, however, 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 to create templates by using the unpack command:

- **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 machine. You can create a Managed Server template by using the pack command, with the `-managed=true` option.

  Subsequently, when you 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 machine. You can start the Managed Server on the remote machine by using either the node manager or customized start scripts created when you unpacked the template on the remote machine.

### 1.1 Introduction to the Pack Command

The pack command provides an alternative method for creating a template from the command line in one simple step. Though the pack command does not allow you to customize the contents of your template in the same way as the Domain Template Builder, it helps you do 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 machine.
You can then create the Managed Server domain directory on the remote machine by using the `unpack` command.

### 1.2 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. If, however, 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 created with the `pack` command. The Managed Server uses the settings defined for the Administration Server; so 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.
This chapter describes the purpose, syntax, and parameters of the `pack` and `unpack` commands.

It contains the following sections:

- Section 2.1, "The pack Command"
- Section 2.2, "The unpack Command"

### 2.1 The pack Command

The `pack` command creates 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.

Table 2–1 describes the files and directories that are included in the templates that you create by using the `pack` command.

<table>
<thead>
<tr>
<th>Template Type</th>
<th>Files and Directories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain</td>
<td>All files and directories in the source WebLogic domain are included, with the following exceptions:</td>
</tr>
<tr>
<td></td>
<td>■ Temporary files that are created when you start a server</td>
</tr>
<tr>
<td></td>
<td>■ The servers directory</td>
</tr>
<tr>
<td></td>
<td>■ Files in the security directory that are created automatically when you create the WebLogic domain, such as <code>DefaultAuthenticatorInit.ldift</code> and <code>XACMLRoleMapperInit.ldift</code></td>
</tr>
<tr>
<td></td>
<td>If you configured additional security data, such as for users, groups, or roles, through the Oracle WebLogic Server Administration Console or other online tools, 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.</td>
</tr>
</tbody>
</table>
The pack Command

2.1.1 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]

2.1.2 Parameters of the pack Command

Table 2–2 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 for the WebLogic domain from which the template is to be created.</td>
</tr>
<tr>
<td>-template=template</td>
<td>Required</td>
<td>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.</td>
</tr>
</tbody>
</table>
2.1.3 Example for 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.

2.2 The unpack Command

The unpack command creates a full WebLogic domain or a subset of a domain that is used for a Managed Server domain directory on a remote machine.

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

---

**Table 2–2 Parameters of the pack Command**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required or Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-template_name</td>
<td>Required</td>
<td>Descriptive name for the template. Quotes are required only if the value contains spaces.</td>
</tr>
<tr>
<td>-template_author</td>
<td>Optional</td>
<td>The name of the author of the template. Quotes are required only if the value contains spaces.</td>
</tr>
<tr>
<td>-template_desc</td>
<td>Optional</td>
<td>Description of the template. Quotes are required only if the value contains spaces.</td>
</tr>
<tr>
<td>-managed</td>
<td>Optional</td>
<td>Specifies whether the template is to be used to create Managed Servers on remote machines. The default is false.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>-log</td>
<td>Optional</td>
<td>Name of the log file.</td>
</tr>
<tr>
<td>-log_priority</td>
<td>Optional</td>
<td>The priority setting for the log file. Use a log4j priority string.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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/docs/api/org/apache/log4j/Level.html">http://logging.apache.org/log4j/docs/api/org/apache/log4j/Level.html</a>.</td>
</tr>
</tbody>
</table>
The unpack Command

- 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 containing all of 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 administrator password defined for the domain. Otherwise, an error is displayed when running the `unpack` command.

---

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
- `config_bootstrapper.xml` file (based on the `config.xml` in the template)
- `nm_password.properties` file
- `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 installation directory for the product installation on the remote machine. By default, this directory is located in `WLS_HOME/common/nodemanager`.

---

**Note:** By default, application files are not included in a Managed Server template created by using the `pack` command. If an application in the WebLogic domain from which you created a Managed Server template was 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, see the "Controlling Deployment File Copying with Staging Modes" section in the *Deploying Applications to Oracle WebLogic Server* guide.

---

### 2.2.1 Syntax of the unpack Command

```
unpack -template=template -domain=domain [-user_name=username] 
[-password=password] [-app_dir=application_directory] [-java_home=java_home_directory] [-server_start_mode=dev|prod] 
[-log=log_file] [-log_priority=log_priority] [-overwrite_domain]
```

### 2.2.2 Parameters of the unpack Command

*Table 2–3* lists the parameters that you can use with the `pack` command.
### Table 2–3 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 filename 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>
<tr>
<td>-user_name=username</td>
<td>Optional</td>
<td>The username and password for the default administrator as currently defined in the template. This username-password combination is used to boot 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.</td>
</tr>
<tr>
<td>-password=password</td>
<td>Optional</td>
<td>The username-password combination is used to boot 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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong>: 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>If no administrative users are defined in the template, you can create a new default administrator by specifying the -user_name and -password parameters.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If a default administrator and password are currently defined in the template, you can add a new administrative user by specifying the -user_name and -password parameters.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The value must not include commas, tabs, or any characters in the following list: &lt; &gt; #</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>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong>: These parameters are not applicable if you are creating a WebLogic domain using a Managed Server template.</td>
</tr>
<tr>
<td>-app_dir=application_directory</td>
<td>Optional</td>
<td>The full path to the directory used to store the applications defined in the template.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This parameter is applicable only if the template supports a separate applications directory.</td>
</tr>
<tr>
<td>-java_home=java_home_directory</td>
<td>Optional</td>
<td>The full path to the Java home directory. When set, this parameter identifies the default JVM used by the Administration Server for the WebLogic domain.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This parameter is not applicable if you are creating a WebLogic domain based on a Managed Server template.</td>
</tr>
<tr>
<td>-server_start_mode=dev</td>
<td>Optional</td>
<td>The start mode for the Administration Server: development or production.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This parameter is not applicable if you are creating a WebLogic domain by using a Managed Server template.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If <strong>dev</strong> mode is specified, a boot.properties file is created for the domain.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If <strong>prod</strong> mode is specified, a boot.properties file is <strong>not</strong> created for the domain.</td>
</tr>
</tbody>
</table>
The unpack Command

Creating Templates and Domains Using the Pack and Unpack Commands

2.2.3 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
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required or Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-log=log_file</td>
<td>Optional</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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/docs/api/org/apache/log4j/Level.html">http://logging.apache.org/log4j/docs/api/org/apache/log4j/Level.html</a>.</td>
</tr>
<tr>
<td>overwrite_domain</td>
<td>Optional</td>
<td>If the specified domain directory already exists and is not empty, no warnings to that effect are displayed, and the files in the directory are automatically overwritten. No attempt is made to merge or preserve local customizations. Note: If a system error occurs after domain generation starts, the domain directory will be in an unknown state.</td>
</tr>
</tbody>
</table>
Creating and Starting a Managed Server on a Remote Machine

This chapter describes how to create and start a Managed Server on a remote machine using the pack and unpack command.

In some WebLogic domains, you may want to run a Managed Server on a machine that is remote from the Administration Server for the WebLogic domain. You can do this by performing the following steps:

1. Create a Managed Server template by using the pack command. A Managed Server template, by default, contains only those files that are necessary for creating a Managed Server on a remote machine.

   For more information, see Section 3.1, "Creating a Managed Server Template."

2. Create a Managed Server domain directory by copying and unpacking the Managed Server template on a remote machine.

   For more information, see Section 3.2, "Creating a Managed Server on a Remote Machine."

3. Start the Managed Server on the remote machine.

   For more information, see Section 3.3, "Starting Managed Servers on a Remote Machine."

3.1 Creating a Managed Server Template

You 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.

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 MW_HOME\wlserver_12.1\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, and the filename of the template to be created.
Creating a Managed Server on a Remote Machine

- `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"
```

### 3.2 Creating a Managed Server on a Remote Machine

1. Install WebLogic Server on the machines 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 the *Oracle WebLogic Server Installation Guide*.

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

   The IP address and port number of the remote machine must match the definition of the Managed Servers specified in the Managed Server template.

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

4. On the remote machine, navigate to the `WLS_HOME\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 machine 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_projects\domains\myManagedDomain\mydomain_managed.jar
   ```

### 3.3 Starting Managed Servers on a Remote Machine

When you create a WebLogic domain directory for Managed Servers by using the `unpack` command, it contains a customized start script for each Managed Server targeted to the current remote machine.

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 machine `m1`, when you create the Managed Server domain directory on machine `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...
Starting Managed Servers on a Remote Machine

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.

2. On the remote machine, navigate to the directory for the WebLogic domain that you created in Section 3.2, "Creating a Managed Server on a Remote Machine."

3. Start the Managed Server on the remote machine.
   - On a Windows system, run one of the following commands at the DOS 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 machine hosting 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 the node manager. For more information, see *Node Manager Administrator’s Guide for Oracle WebLogic Server.*