

Oracle® WebLogic Server

Creating Templates and Domains Using the pack and unpack Commands
10g Release 3 (10.3)

October 2008

Oracle® WebLogic Server

Oracle® Workshop for WebLogic

Oracle® WebLogic Portal

Oracle® Service Bus

Oracle WebLogic Server Creating Templates and Domains Using the pack and unpack Commands, 10g Release 3 (10.3)

Copyright © 2007, 2008, Oracle and/or its affiliates. All rights reserved.

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

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

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

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software 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 which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on 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. 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.

Contents

1. Introduction	
2. Command Reference: pack and unpack	
pack	2-1
unpack	2-4
3. How Do I Create and Start a Managed Server on a Remote Machine?	
How Do I Create a Managed Server Template	3-1
How Do I Create a Managed Server on a Remote Machine	3-2
How Do I Start Managed Servers on a Remote Machine	3-3

Introduction

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

The `pack` and `unpack` commands provide a simple, one-step method for creating domains and templates from the command line. You cannot, however, use these commands to customize the contents of your domain or template in the same way as the other tools.

The `pack` and `unpack` commands are available in the `\common\bin` subdirectory of the product installation directory.

Note: In addition to the `pack` and `unpack` commands, you can use the Configuration Wizard, Domain Template Builder, or WebLogic Scripting Tool to create 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 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 domains.
- **Managed server template:** This type of template defines the subset of resources within a 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.

pack

The `pack` command provides an alternative method for creating a template from the command line in one simple step. Although 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 domain.
You can then use this template as the basis for a new 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 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. For more information, see [How Do I Create and Start a Managed Server on a Remote Machine?](#).

For information about the syntax and parameters of the `pack` command, see [pack](#).

unpack

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

For information about the syntax and parameters of the `unpack` command, see [unpack](#).

Related Topics

- [Creating WebLogic Domains Using the Configuration Tool](#)
- [Creating Templates Using the Domain Template Builder](#)
- [WebLogic Scripting Tool](#)

Introduction

Command Reference: pack and unpack

This section describes the purpose, syntax, and parameters of the `pack` and `unpack` commands.

pack

The `pack` command creates a template archive (`.jar`) file that contains a snapshot of either an entire domain or a subset of a domain. You can use a template that contains a subset of a domain to create a managed server domain directory hierarchy on a remote machine.

[Table 2-1](#) describes the files and directories that are included in the templates that you create by using the `pack` command.

Table 2-1 Files and Directories

Template Type	Files and Directories
Domain	<p>All files and directories in the source domain are included, with the following exceptions:</p> <ul style="list-style-type: none"> • Temporary files that are created when you start a server • The <code>servers</code> directory • Files in the <code>security</code> directory that are created automatically when you create the domain, such as <code>DefaultAuthenticatorInit.ldift</code> and <code>XACMLRoleMapperInit.ldift</code> <p>Note: If you configured additional security data, such as for users, groups, or roles, through the 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 domain.</p>

Table 2-1 Files and Directories

Template Type	Files and Directories
Managed server	<p>The following files and directories are included by default:</p> <ul style="list-style-type: none">• All files in the root directory with the following extensions: <code>.cmd</code>, <code>.sh</code>, <code>.xml</code>, <code>.properties</code>, and <code>.ini</code>.• Any files with the <code>.pem</code> extension defined in the SSL configuration for your domain• <code>bin</code> directory• <code>lib</code> directory• All files and subdirectories in the <code>config</code> directory <p>The following files and directories are not included in a managed server template by default:</p> <ul style="list-style-type: none">• Applications and certain application initialization files• Temporary files that are created when you start a server• The <code>servers</code> directory• Files in the <code>security</code> directory that are created automatically when you create the domain, such as <code>DefaultAuthenticatorInit.ldif</code> and <code>XACMLRoleMapperInit.ldif</code>. <p>Note: The <code>config.xml</code> file of the domain from which you are creating your template must contain managed server definitions that specify the IP address and port for the target remote machine. The managed server template that you create from that domain can only be used, with the <code>unpack</code> command, on the specified remote machines to create managed server domain directories for the managed servers defined in the <code>config.xml</code> file.</p>

Syntax

```
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

Table 2-2 describes the parameters that you can use with the `pack` command.

Table 2-2 Parameters: pack Command

Parameter	Required or Optional?	Description
<code>-domain=<i>domain</i></code>	Required	The full or relative path for the domain from which the template is to be created.
<code>-template=<i>template</i></code>	Required	The full or relative path and filename of the template to be created. The template filename must include the <code>.jar</code> extension.
<code>-template_name="<i>template_name</i>"</code>	Required	Descriptive name for the template. Quotes are required only if the value contains spaces.
<code>-template_author="<i>author</i>"</code>	Optional	The name of the author of the template. Quotes are required only if the value contains spaces.
<code>-template_desc="<i>description</i>"</code>	Optional	Description of the template. Quotes are required only if the value contains spaces.
<code>-managed={true false}</code>	Optional	<p>Specifies whether the template is to be used to create managed servers on remote machines. The default is <code>false</code>.</p> <p>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>, <code>config.xml</code>, and <code>nm_password.properties</code>. It also includes a <code>domain.properties</code> file that is unique to the managed server template.</p> <p>Applications and certain application initialization files are not included.</p> <p>The resulting template can be used to create managed servers on remote machines.</p>

Table 2-2 Parameters: pack Command (Continued)

Parameter	Required or Optional?	Description
<code>-log=log_file</code>	Optional	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. For more information, see http://logging.apache.org/log4j/docs/api/org/apache/log4j/Level.html .

Example

To create a template based on an existing domain named `mydomain` that is located in the `C:\bea\user_projects\domains` directory, run the following command:

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

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

Related Topic

[How Do I Create and Start a Managed Server on a Remote Machine?](#)

unpack

The `unpack` command creates a full domain or a subset of a domain 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 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 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.

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 domain
- `config_bootstrap.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` designates the node manager installation directory for the product installation on the remote machine. By default, this directory is located in `WL_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 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 will not contain any of the externally staged applications. Before you start the managed server, you will need to make sure that it has access to the externally staged application files. For more information, see [Controlling Deployment File Copying with Staging Modes in Deploying Applications to WebLogic Server](#).

For more information about using the `pack` and `unpack` commands to create a managed server domain directory, see [How Do I Create and Start a Managed Server on a Remote Machine?](#)

Syntax

```
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]
```

Parameter

Table 2-3 describes the parameters that you can use with the `unpack` command.

Table 2-3 Parameters: unpack Command

Parameter	Required or Optional?	Description
<code>-template=template</code>	Required	The full or relative path and filename of the template being used to create the domain.
<code>-domain=domain</code>	Required	The full or relative path of the domain to be created.
<code>-user_name=username</code> <code>-password=password</code>	Optional	<p>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 will be reset to the value specified.</p> <p>Note: If a password for the default administrator is not defined in the template, you must specify one.</p> <p>If no administrative users are defined in the template, you can create a new default administrator by specifying the <code>-user_name</code> and <code>-password</code> parameters.</p> <p>If a default administrator and password are currently defined in the template, you can add a new administrative user by specifying the <code>-user_name</code> and <code>-password</code> parameters.</p> <p>The value must not include commas, tabs, or any characters in the following list: <code>< > # & ? () { }</code>. Usernames are case sensitive.</p> <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>Note: These parameters are not applicable if you are creating a domain using a managed server template.</p>

Table 2-3 Parameters: unpack Command (Continued)

Parameter	Required or Optional?	Description
<code>-app_dir=application_directory</code>	Optional	The full path to the directory used to store the applications defined in the template. Note: This parameter is applicable only if the template supports a separate applications directory.
<code>-java_home=java_home_directory</code>	Optional	The full path to the Java home directory. When set, this parameter identifies the default JVM used by the administration server for the domain. Note: This parameter is not applicable if you are creating a domain based on a managed server template.
<code>-server_start_mode={dev prod}</code>	Optional	The start mode for the administration server: development or production. Note: This parameter is not applicable if you are creating a domain by using a managed server template.
<code>-log=log_file</code>	Optional	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. For more information, see http://logging.apache.org/log4j/docs/api/org/apache/log4j/Level.html .

Example

To create a domain named `mydomain.jar` in the `C:\bea\user_templates` directory, by using the `my_new_domain` template in the `C:\bea\user_projects\domains` directory, run the following command:

Command Reference: pack and unpack

```
unpack -template=C:\bea\user_templates\mydomain.jar  
-domain=C:\bea\user_projects\domains\my_new_domain
```

Related Topic

[How Do I Create and Start a Managed Server on a Remote Machine?](#)

How Do I Create and Start a Managed Server on a Remote Machine?

In some domains, you may want to run a managed server on a machine that is remote from the administration server for the 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 [How Do I Create 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 [How Do I Create a Managed Server on a Remote Machine](#).
3. Start the managed server on the remote machine. For more information, see [How Do I Start Managed Servers on a Remote Machine](#)

How Do I Create a Managed Server Template

You create a managed server template by executing the `pack` command on an existing 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 `BEA_HOME\wlserver_10.0\common\bin` directory.
2. Run the following command:

How Do I Create and Start a Managed Server on a Remote Machine?

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

In this command:

- *domain* is the full or relative path of the 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.
- *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 domain named `mydomain`.

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

For more information about the parameters of the `pack` command, see [pack](#).

How Do I Create a Managed Server on a Remote Machine

1. Install WebLogic Server on the machines on which you want to host managed servers for the domain.

Note: All WebLogic Server instances within a domain must run the same version of the WebLogic Server software. For more information about installing WebLogic Server, see the [Installation Guide](#).

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

Note: 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.

For information about creating managed server templates, see [How Do I Create a Managed Server Template](#).

4. On the remote machine, navigate to the `WL_HOME\common\bin` directory (`WL_HOME` is the product directory in which installed WebLogic Server is installed).

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 domain named *myManagedDomain*.

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

How Do I Start Managed Servers on a Remote Machine

When you create a 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 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 servers. Alternatively, you can use the *startManagedWebLogic* script with the required parameters.

1. Start the administration server for the domain as described in [Starting and Stopping Servers](#).
2. On the remote machine, navigate to the directory for the domain that you created in [How Do I Create 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

How Do I Create and Start a Managed Server on a Remote Machine?

`startManagedWebLogic_Readme.txt` file provides a list of all the managed servers and the `admin-url` for the domain.

Note: You can also start managed servers by using the node manager. For more information, see [Node Manager Administrator's Guide](#).