Oracle® Fusion Middleware

Creating Templates and Domains Using the Pack and Unpack Commands

12*c* (12.2.1)

E55192-01

October 2015

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.

Oracle Fusion Middleware Creating Templates and Domains Using the Pack and Unpack Commands, 12*c* (12.2.1)

E55192-01

Contents

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

Related Documents

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

- *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

This chapter introduces 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 *ORACLE_HOME*\oracle_common\common\bin 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 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 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.

This chapter includes the following sections:

- Section 1.1, "Introduction to the Pack Command"
- Section 1.2, "Introduction to the Unpack Command"

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.

2

pack and unpack Command Reference

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

This chapter includes the following sections:

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

The commands are located in the *`ORACLE_HOME`*`/oracle_common/common/bin` directory.

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.

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.

2.1.1 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 will be created on the target system.

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

`domain_home``/config/deployments/`*`deployment_name`*`/plan`

Note: Plans already within the domain will not be 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 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.

2.1.2 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 log in to the original domain's Administration Console.
2. Export the data from the original domain as described in "Export data from a security provider" in the *Oracle WebLogic Server Administration Console Online Help*.
3. If necessary, start the Administration Server for the new domain and log in to the new domain's Administration Console.
4. Import the data into the new domain as described in "Import data into a security provider" in the *Oracle WebLogic Server Administration Console Online Help*.

2.1.3 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, with the exception of `config.xml`
- `bin` directory
- `lib` directory

The following files and directories are *not* included in a Managed Server template by default:

- Applications and certain application initialization files
- `config.xml`
- 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 machine, but servers that are defined in the `config.xml` file to use specific machines and ports will not work on other machines. You must therefore unpack the same template on each remote node for all servers to start successfully on their appropriate nodes and machines.

2.1.4 Node Manager Configuration

Node Manager configuration is not preserved by `pack` and `unpack` for both managed and non-managed modes. The Node Manager type is always set to `PerDomainNodeManager` by default in the unpacked domain.

If the Node Manager type in the original domain was `CustomLocationNodeManager`, when unpacking the domain, 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 unpacking the domain on each remote machine.

If a `ManualNodeManagerSetup` configuration was used for the original domain, the Node Manager configuration information is discarded from the template that is created by `pack`. When unpacking the domain, specify `ManualNodeManagerSetup` using the `-nodeManagerType` parameter. After unpacking 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, which are stored in config.xml, are copied from the packed domain to the unpacked domain.

2.1.5 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]
```

2.1.6 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
`-domain=domain`	Required	The full or relative path for the WebLogic domain from which the template is to be created.
`-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	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	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
`-managed=true|false`	Optional	Specifies whether the template is to be used to create Managed Servers on remote machines. 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 machines. **Note:** As of WebLogic Server 12.1.2, you can use the WLST `writeTemplate` command in online mode. This makes it easier to create or update remote Managed Server domains. For more information, see "writeTemplate" in *WLST Command Reference for WebLogic Server*.
`-log=log_file`	Optional	Name of the log file.
`-log_priority=log_priority`	Optional	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 `http://logging.apache.org/log4j/`.

2.1.7 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:

- 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_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* is the node manager home directory for the product installation on the remote machine. The location of this directory depends on the Node Manager type. For more information on Node Manager type and Node Manager home, see Table 2–2.

> **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 [-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]
```

2.2.2 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	This parameter applies only to WebLogic domains in which Fusion Middleware products are installed. **Note:** When setting this parameter to `true`, ensure that the source database is running. If it is not running, the `unpack` command will fail. 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" in *Administering Oracle Fusion Middleware*.
`-template=template`	Required	The full or relative path and filename 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.
`-walletDir`	Optional	The directory in which the wallet file that contains the password for `unpack` to use for the administration user. This is the same directory as was specified when you created the wallet file using the `configWallet.sh` command. For example: `-walletDir $HOME/wallet` If you want to create a different administrator user for the domain you are unpacking: 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.

Table 2–2 (Cont.) Parameters of the unpack Command

Parameter	Required or Optional	Description
`-user_name=username`	Optional	The username for the default administrator as currently defined in the template or via the WalletDir. This username 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. 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` and `password` parameters. If a default administrator and password are currently defined in the template, you can add a new administrative user by `screateWallet.sh` and the `-user_name` parameter or by specifying the `-user_name` and `password` parameters The value must not include commas, tabs, or any characters in the following list: < > # | & ? ( ) { }. Usernames are case sensitive. A valid password consists of a string of at least eight case-sensitive characters. The `unpack` command encrypts the password value. **Note**: This parameter is not applicable if you are creating a WebLogic domain using a Managed Server template.
`-password=password`	Optional	This parameter has been deprecated. For better security, use the `walletDir` parameter instead to specify the password for the administrator user. If you use this parameter: ■ If you specify a new password for the default administrator, the password is reset to the value specified. ■ If a password for the default administrator is not defined in the template, you must specify one. A valid password consists of a string of at least eight case-sensitive characters. The `unpack` command encrypts the password value. **Note**: This parameter is not applicable if you are creating a WebLogic domain using a Managed Server template.
`-app_dir=application_directory`	Optional	The full path to the directory used to store the applications defined in the template. This parameter is applicable only if the template supports a separate applications directory.

Table 2–2 (Cont.) Parameters of the unpack Command

Parameter	Required or Optional	Description
`-nodemanager_type=type`	Optional	Sets the Node Manager type for the domain you are creating. Specify one of the following types: ■ `PerDomainNodeManager`—If you specify this type, the Node Manager home is defined within the domain as `<domain_name>/nodemanager`. The Node Manager for each domain can have a different configuration, as determined by the files in this directory. ■ `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 `nodemanager_home` argument to specify the Node Manager home directory. The specified directory must be empty. ■ `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`.
`-nodemanager_home=home`	Conditional	If the specified Node Manager type is `CustomLocationNodeManager`, this argument is required. Enter the full path to the Node Manager location you want to use. If the specified Node Manager type is not `CustomLocationNodeManager`, this argument is not required. If included, it is ignored.
`-java_home=java_home_directory`	Optional	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. This parameter is not applicable if you are creating a WebLogic domain based on a Managed Server template.
`-server_start_mode=dev|prod`	Optional	The start mode for the Administration Server: `dev` or `prod`. The default value is `dev`. **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 will be 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

Parameter	Required or Optional	Description
`overwrite_domain`	Optional	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.
`-log=log_file`	Optional	Name of the log file.
`-log_priority=log_priority`	Optional	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 `http://logging.apache.org/log4j/`.

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
```

3 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 *`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, 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 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 *Installing and Configuring Oracle WebLogic Server and Coherence*.

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 *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 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_templates\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 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 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 Node Manager. For more information, see *Administering Node Manager for Oracle WebLogic Server*.