BEA

BEAProducts

Creating Templates and Domains Using the pack and unpack Commands

Contents

Introduction

This document describes how to use the `pack` and `unpack` commands to create templates and domains.

Topics in this section include:

- Overview of the pack and unpack Commands

- About Domain and Managed Server Templates

- Additional Tools for Creating Templates and Domains

Overview of the pack and unpack Commands

BEA provides several tools for creating domains and templates: the Configuration Wizard, the Domain Template Builder, WLST Offline, and the `pack` and `unpack` commands. The `pack` and `unpack` commands provide a simple, one-step method for creating domains and templates from the command line. They do not, however, allow you to customize the contents of your domain or template in the same way as the other tools. For more information about creating templates and domains using the Configuration Wizard, the Domain Template Builder, or WLST Offline, see "Additional Tools for Creating Templates and Domains" on page 1-3.

This section provides the following topics:

- Location of Command Scripts

- About the pack Command

- About the unpack Command

Location of Command Scripts

On both Windows and UNIX systems, `pack` and `unpack` are installed in the `\common\bin` subdirectory of the product installation directory, for example `C:\bea\wlserver_10.0\common\bin`.

About the pack Command

The `pack` command provides an alternative means for quickly 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 is a useful tool for quickly creating the following:

- A domain template that contains a snapshot of an entire working domain and has been customized through the Administration Console, Configuration Wizard, or WebLogic Scripting Tool (WLST). This template can then be used to create a domain using `unpack`, the Configuration Wizard, or WebLogic Scripting Tool (WLST) Offline.

- 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 use `unpack` to create the Managed Server domain directory on the remote machine. For more information, see "Creating and Starting a Managed Server on a Remote Machine: Main Steps" on page 3-1.

For a complete description of the `pack` command, see "pack" on page 2-1.

For additional information about the Domain Template Builder, the Configuration Wizard, or WLST, see "Additional Tools for Creating Templates and Domains" on page 1-3.

About the unpack Command

The `unpack` command provides a one-step method for quickly creating a domain from an existing template 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. 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 domain

- Specify an applications directory, if one is supported by the template

Using `unpack`, you can also create a Managed Server domain directory using a Managed Server template created using the `pack` command. Because a Managed Server uses the settings defined for the Administration Server, 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 a complete description of the `unpack` command, see "unpack" on page 2-5.

For additional information about the Configuration Wizard, Domain Template Builder, or WLST, see "Additional Tools for Creating Templates and Domains" on page 1-3.

About Domain and Managed Server Templates

When you are using `pack` and `unpack`, the term *template* refers to a Java Archive (JAR) file that contains the files and scripts required to create a domain.The types of templates that can be created by the `pack` command and used by the `unpack` command to create domains include:

- *Domain template*—defines the full set of resources within a domain, including infrastructure components, applications, services, security options, and general environment and operating system options. Using the `pack` command, you can create a domain template from an existing domain. A domain template is used to create a new domain.

- *Managed Server template*—defines the subset of resources within a domain that are required to create a Managed Server domain directory on a remote machine. You create a Managed Server template 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 in either of two ways: using Node Manager, or using customized start scripts created when you unpacked the template on the remote machine.

Additional Tools for Creating Templates and Domains

Your BEA software provides several tools for creating templates and domains. The `pack` and `unpack` commands provide a one-step method for creating templates and domains. You can also use the tools defined in the following table to create templates and domains.

Table 1-1 Additional Tools for Creating Templates and Domains

To perform this task . . .	Use these tools . . .
Create a template	• Domain Template Builder—a stand-alone Java application that guides you through the process of creating custom domain and extension templates using a graphical user interface. The Domain Template Builder provides more control over template customization than the `pack` command. In addition to customizing basic template information, you can also control the following: - Application and domain directory contents - Database configuration scripts - Administration Server configuration - Basic security configuration - Start menu entries - Customized scripts and variables For more information, see Creating Templates Using the Domain Template Builder. • WLST Offline—a command-line scripting interface that you use to create a new domain or update an existing domain without connecting to a running WebLogic Server. The `writeTemplate` functionality of WLST Offline provides the same capability for creating a domain template as the `pack` command. You cannot create a Managed Server template using WLST Offline. For more information, see *WebLogic Scripting Tool*.

Table 1-1 Additional Tools for Creating Templates and Domains

To perform this task . . .	Use these tools . . .
Create a new domain or extend an existing domain	• Configuration Wizard—a stand-alone Java application that guides you through the process of creating or extending a domain. When creating a new domain, you can choose to accept the default settings or to customize the domain to more closely match your particular environment by adding and configuring Managed Servers, clusters, and machine definitions, or customizing predefined JDBC data sources, and JMS file store directories. You can also use the Configuration Wizard to extend an existing domain by adding predefined applications, services, or product component functionality. For more information, see Creating WebLogic Domains Using the Configuration Tool.
	• WLST Offline—a command-line scripting interface that you use to create a new domain or update an existing domain without connecting to a running WebLogic Server—supporting the same functionality as the Configuration Wizard. For more information, in *WebLogic Scripting Tool*.

Introduction

pack and unpack Command Descriptions

The following sections describe the `pack` and `unpack` commands:

- pack

- unpack

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.

The following table describes the files and directories that are included in both domain and Managed Server templates.

Table 2-1 Files and Directories

When you create this type of template . . .	The following occurs . . .
Domain	All files and directories in the source 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 domain, such as `DefaultAuthenticatorInit.ldift` and `XACMLRoleMapperInit.ldift`. **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.

Table 2-1 Files and Directories

When you create this type of template . . .	The following occurs . . .
Managed Server (using the -managed=true option)	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 domain • bin directory • lib directory • All files and subdirectories in the config directory The following files and directories are *not* included in a Managed Server template by default: • Applications and certain application initialization files • 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 domain, such as DefaultAuthenticatorInit.ldift and XACMLRoleMapperInit.ldift. **Note:** The config.xml 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 unpack command, on the specified remote machines to create Managed Server domain directories for the Managed Servers defined in the config.xml file.

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]
```

Parameter Descriptions

The following table describes the parameters that can be used with the pack command.

Table 2-2 Parameters for pack Command

Parameter	Description
The following parameters are required:	
-domain=*domain*	The full or relative path to the domain from which the template is to be created.
-template=*template*	The full or relative path and filename of the template to be created. The template filename must include the `.jar` extension.
-template_name="*template_name*"	Descriptive name for the template. Quotes are required only if the value for this parameter contains spaces.
The following parameters are optional:	
-template_author="*author*"	The name of the author of the template. Quotes are required only if the value for this parameter contains spaces.
-template_desc="*description*"	Description of the template. Quotes are required only if the value for this parameter contains spaces.
-managed={true|false}	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`, `config.xml`, 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.

Table 2-2 Parameters for pack Command (Continued)

Parameter	Description
`-log=log_file`	Name of the log file.
`-log_priority=log_priority`	The priority setting for the log file. Use a log4j priority string. Valid log4j priority strings are `debug`, `info`, `warn`, `error`, `fatal`. The priority string values correspond to the levels defined in the `Level` class. For more information, see http://logging.apache.org/log4j/docs/api/org/apache/log4j/Level.html.

Example

Executing the following command using a domain named `mydomain` in the directory `C:\bea\user_projects\domains` creates a template file named `mydomain.jar` in the directory `C:\bea\user_templates`. The name of the template is specified as `My WebLogic Domain`.

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

Related Topics

- "Creating and Starting a Managed Server on a Remote Machine: Main Steps" on page 3-1

- "How Do I: Use pack to Create a Managed Server Template" on page 3-2

unpack

Creates a full domain or a subset of a domain used for a Managed Server domain directory on a remote machine. You may use `unpack` only with a template compatible with your current installation. The template can be any of the following:

- A domain template provided by BEA and packaged with your current installation

- A domain template created using the Domain Template Builder or WLST Offline

- A domain template created using the `pack` command

- A Managed Server template created using the `pack` command. A Managed Server template, by default, contains only the files necessary for creating a Managed Server domain directory.

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

- A `config_bootstrap.xml` file (based on the `config.xml` in the template)

- A `nm_password.properties` file

- A `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`, for example, `C:\bea\wlserver_10.0\common\nodemanager\nodemanager.domains`.

Note: By default, application files are not included in a Managed Server template created using the `pack` command. If an application in the domain from which you created a Managed Server template was deployed using external_stage mode, the Managed Server domain directory that you create with the `unpack` command will not contain any of the external_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 "Creating and Starting a Managed Server on a Remote Machine: Main Steps" on page 3-1.

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 Descriptions

The following table describes the parameters that can be used with the unpack command.

Table 2-3 Parameters for unpack Command

Parameter	Description
The following parameters are required:	
-template=*template*	The full or relative path and filename of the template being used to create the domain.
-domain=*domain*	The full or relative path to the domain to be created.

Table 2-3 Parameters for unpack Command (Continued)

Parameter	Description
The following parameters are optional:	
`-user_name=`*username* `-password=`*password*	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.
	Note: If a password for the default administrator is not defined in the template, you must specify one.
	If no administrative users are defined in the template, you can create a new default administrator 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 specifying the `-user_name` and `-password` parameters.
	A valid username cannot include commas, tabs, or any characters in the following list: < > #	& ? ( ) { }. Usernames are case sensitive.
	A valid password consists of a string of *at least* 8 case-sensitive characters. The `unpack` command encrypts the password value.
	Note: These parameters are not applicable if you are creating a domain using a Managed Server template.
`-app_dir=`*application_directory*	The full path to the directory used to store the applications defined in the template.
	Note: This parameter is only applicable if the template supports a separate applications directory.
`-java_home=`*java_home_directory*	The full path to the Java Home directory. When set, this identifies the default JVM used by the Administration Server for the domain.
	Note: This parameter is not applicable if you are creating a domain using a Managed Server template.

Table 2-3 Parameters for unpack Command (Continued)

Parameter	Description
`-server_start_mode={dev|prod}`	The start mode for the Administration Server: development or production.
	Note: This parameter is not applicable if you are creating a domain using a Managed Server template.
`-log=log_file`	Name of the log file.
`-log_priority=log_priority`	The priority setting for the log file. Use a log4j priority string. Valid log4j priority strings are `debug`, `info`, `warn`, `error`, `fatal`. The priority string values correspond to the levels defined in the `Level` class. For more information, see `http://logging.apache.org/log4j/docs/api/org/apache/log4j/Level.html`.

Example

Executing the following command using a template named `mydomain.jar` in the directory `C:\bea\user_templates` creates a domain named `my_new_domain` in the directory `C:\bea\user_projects\domains`.

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

Related Topics

- "Creating and Starting a Managed Server on a Remote Machine: Main Steps" on page 3-1

- "How Do I: Use unpack to Create a Managed Server on a Remote Machine" on page 3-3

pack and unpack Command Descriptions

How Do I...?

This section provides information and examples for creating templates and domains using `pack` and `unpack`:

- Creating and Starting a Managed Server on a Remote Machine: Main Steps

- How Do I: Use pack to Create a Managed Server Template

- How Do I: Use unpack to Create a Managed Server on a Remote Machine

- How Do I: Start Managed Servers on a Remote Machine

Creating and Starting a Managed Server on a Remote Machine: Main Steps

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 so by using the `pack` command to create a Managed Server template (a `.jar` file that, by default, contains only the files necessary to create a Managed Server domain directory), copying the template to the remote machine, and unpacking the template to create a domain directory customized for a Managed Server.

Note: You must create the Managed Server domain directory on the remote machine that matches the definition of the Managed Servers specified in the template.

The steps to create a Managed Server domain directory on a remote machine are as follows:

1. Create a Managed Server template using the `pack` command. A Managed Server template, by default, contains only those files necessary to create a Managed Server on a remote machine.

For instructions, see "How Do I: Use pack to Create a Managed Server Template" on page 3-2.

2. Create a Managed Server domain directory by copying and unpacking the Managed Server template on a remote machine. For a detailed procedure, see "How Do I: Use unpack to Create a Managed Server on a Remote Machine" on page 3-3.

3. Start the Managed Server on the remote machine. For a detailed procedure, see "How Do I: Start Managed Servers on a Remote Machine" on page 3-3.

How Do I: Use pack to Create a Managed Server Template

You create a Managed Server template by executing the `pack` command, with the `-managed=true` option, on an existing domain that already includes the definition of one or more Managed Servers. This domain must also contain 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. In this pathname, `BEA_HOME` is the product directory in which you installed WebLogic Server.

2. Execute the following command:

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

In this command line:

– `domain`—The full or relative path to the domain from which the template is to be created.

– `template.jar`—The full or relative path to the template, and the filename of the template to be created.

– `template_name`—Descriptive name for the template enclosed in quotes.

For example, executing 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 a complete description of the `pack` command, see "pack" on page 2-1.

How Do I: Use unpack to Create a Managed Server on a Remote Machine

1. Install WebLogic Server on each machine that you want to host a Managed Server 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 that will host the Managed Server. 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 details about how to create a Managed Server template, see "How Do I: Use pack to Create a Managed Server Template" on page 3-2.

4. On the remote machine, navigate to the `WL_HOME\common\bin` directory. In this pathname, `WL_HOME` is the product directory in which you installed WebLogic Server.

5. Execute the following command, using the Managed Server template that you copied to the machine in Step 3.

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

   In this command line:

   – `domain`—The full or relative path to the domain to be created.

   – `template.jar`—The full or relative path to the template from which the domain is to be created.

   For example, executing the following command using a template named `mydomain_managed.jar` 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 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` and `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 domain as described in "Starting and Stopping Servers" in *Managing Server Startup and Shutdown*.

2. On the remote machine, navigate to the directory for the domain that you created in "How Do I: Use unpack to Create a Managed Server on a Remote Machine" on page 3-3.

3. Start the Managed Server on the remote machine by doing one of the following:

   – On a Windows system, open an MS-DOS command prompt window and enter one of the following at the prompt:

   `start`*`my_managed_server`*

   `startManagedWebLogic `*`my_managed_server admin-url`*

   – On a UNIX system, enter one of the following at the prompt:

   `./start`*`my_managed_server`*`.sh`

   `./startManagedWebLogic.sh `*`my_managed_server admin-url`*

   In these command lines, *`my_managed_server`* is the name of the Managed Server to be started and *`admin-url`* specifies 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 domain.

Note: You can also start Managed Servers using the Node Manager. For details, see "Using Node Manager to Control Servers" in *Managing Server Startup and Shutdown*.