BEAProducts

Using the Configuration Wizard in Silent Mode (Deprecated)

Copyright

Copyright © 2005 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

Trademarks or Service Marks

Contents

Using the Configuration Wizard in Silent Mode (Deprecated)

Using the Configuration Wizard in Silent Mode (Deprecated)

> **Note:** Silent-mode operation of the Configuration Wizard is deprecated in WebLogic Server 9.0. For a scripted, silent-mode method, BEA recommends that you use the BEA WebLogic Scripting Tool. For more information, see *WebLogic Scripting Tool* at the following URL:
> `http://e-docs.bea.com/wls/docs90/config_scripting/index.html`

The following sections describe how to create and extend domains using the Configuration Wizard in silent mode:

- Starting the Configuration Wizard in Silent Mode
- Creating a Script for Silent-Mode Configuration

Starting the Configuration Wizard in Silent Mode

When run in silent mode, the Configuration Wizard is executed as a noninteractive process.

Silent-mode configuration enables you to define the configuration information for a domain once and then duplicate that domain on multiple machines. In silent mode, the Configuration Wizard reads configuration settings from a script that you manually create prior to execution.

Before you can run the Configuration Wizard in silent mode, you must first create a script in which you define the settings usually entered manually by a user during graphical-mode or console-mode configuration. When running in silent mode, the Configuration Wizard queries the specified configuration script to obtain configuration settings. It does not prompt you to provide information in real time. For more information, see "Creating a Script for Silent-Mode Configuration" on page 3.

To start the Configuration Wizard in silent mode on a Windows or UNIX system, follow these steps:

1. Log in to the target system on which the product installation resides.

2. Create a configuration script, as described in "Creating a Script for Silent-Mode Configuration" on page 3.

3. Open an MS-DOS command prompt window (on Windows) or a command shell (on UNIX).

4. Go to the `\common\bin` subdirectory of the product installation directory. For example:

   ```
   cd c:\bea\weblogic90\common\bin
   ```

5. Start the silent-mode configuration process using one of the following commands:

   – Windows:
   ```
   config.cmd -mode=silent [-silent_script=scriptfile1;scriptfile2;...]
   [-log=logfile]
   ```

   – UNIX:
   ```
   sh config.sh -mode=silent
   [-silent_script=scriptfile1;scriptfile2;...] [-log=logfile]
   ```

   In both command lines:

   – *scriptfile* specifies the full or relative path of the configuration script. You can specify multiple scripts, separated by a comma (,) or a semi-colon (;). White space between filenames is not allowed. The `-silent_script` argument is optional. If you specify multiple scripts and one of them is not found, an error message indicating the name of the script not found is generated. If no script file is specified, by default, the Configuration Wizard searches the `common\lib` subdirectory of the product installation directory for the file `configscript.txt`.

   – *logfile* specifies the full or relative path of the log file. The `-log` argument is optional. If the argument is not specified, no log file is generated. If a pathname is not specified, the Configuration Wizard stores the log file within the `common\lib` subdirectory of the product installation directory.

   Note: Incorrect entries in the configuration script can cause configuration failures. To help you determine the cause of a failure, you should generate a log file when you start the configuration.

The silent script is interpreted and executed at run time. The domain is validated at various points during the execution.

Creating a Script for Silent-Mode Configuration

Before you can run the Configuration Wizard in silent mode to create or extend a domain, as described in "Starting the Configuration Wizard in Silent Mode" on page 1, you must first create a script in which you define the configuration settings for the domain.

To create a script, do one of the following:

- Copy the contents of an existing script to a text file and edit the file for your configuration. (recommended)

- Open a new file in a text editor and manually specify the domain configuration settings. (for experienced users of silent-mode scripts)

Refer to the following sections for instructions on creating scripts for creating and extending domains:

- Creating a Script to Create a New Domain in Silent Mode

- Creating a Script to Extend an Existing Domain in Silent Mode

Creating a Script to Create a New Domain in Silent Mode

When you create a script, you specify operations to be performed by the Configuration Wizard when the script is executed. The following sections describe the operations that must be defined in the script to create a new domain:

- Step 1: Select the Domain Template

- Step 2: Edit the Configuration Information

- Step 3: Create the Domain

- Step 4: Close the Domain Template

Note: The code for each operation described in this section must be terminated by a semicolon.

Step 1: Select the Domain Template

Select a domain template to use as the basis of the domain you will create and configure. To select a domain template, use the read operation, as follows:

```
read template from "template_jar_name";
```

Table 1 describes the single `read template` operation parameter for which you must specify a value.

Table 1 Parameters for read template Operation

Parameter	Description
template_jar_name	Full pathname of the domain template, enclosed in quotation marks.

In the following example, the Basic WebLogic Server Domain domain template is specified with the absolute pathname for the template:

```
read template from "C:\bea\weblogic90\common\templates\domains\wls.jar";
```

Step 2: Edit the Configuration Information

Edit the default configuration information to create new configuration objects, edit existing values for configuration object attributes, or delete configuration objects.

Note: You can obtain information about your existing domain configuration from the `config.xml` and `security.xml` files in the template JAR file, or from the `config.xml` in the domain itself, once it is created. For information about the `config.xml` file, see *WebLogic Server Configuration Reference* at:
http://e-docs.bea.com/wls/docs81/config_xml/index.html.

You use silent mode to create and edit any valid configuration object and its child elements, except custom security objects. For more information about valid configuration objects, see *WebLogic Server Configuration Reference* at:
http://e-docs.bea.com/wls/docs81/config_xml/index.html.

Note: Because silent-mode configuration is deprecated in WebLogic Server 9.0, no 9.0-specific configuration objects are supported.

The following table summarizes the edit operations described in the following sections.

Table 2 Summary of Edit Operations for Creating a New Domain

To learn how to . . .	See . . .
Create a configuration object	Creating a Configuration Object
Edit an existing configuration object	• Assigning an Existing Configuration Object to a Variable
	• Assigning an Existing Child Element to a Variable
	• Setting Configuration Object Attributes
	• Setting a Specific Attribute for a Child Element
	• Setting an Attribute for All Instances of a Child Element
	• Setting Attributes for Unnamed Child Elements
	• Setting an HTTP Proxy Server for a Cluster
	• Setting Configuration Options When Creating a New Domain
	• Assigning Servers to Clusters
	• Assigning All Applications and Services to Servers and Clusters
	• Assigning Individual Applications to Servers and Clusters
	• Assigning Individual Services to Servers and Clusters
	• Creating Security Assignments
	• Removing Assignments Using the Unassign Operation
Delete a configuration object	Deleting a Configuration Object

Creating a Configuration Object

To create a configuration object, specify the type of object you want to create, assign it to a variable, and define the `Name` attribute. Then set the remaining attributes (or edit the `Name` attribute), as described in "Setting Configuration Object Attributes" on page 8.

To create a configuration object, use the `create` operation, as follows:

```
create object_type "name" as variable;
```

Table 3 describes the `create` operation parameters for which you must specify values.

Table 3 Parameters for create Operation

Parameter	Description
object_type	Configuration object type, as defined in *WebLogic Server Configuration Reference* at: http://e-docs.bea.com/wls/docs81/config_xml/index.html.
name	Value to assign to the Name attribute. This parameter (*name*) must be enclosed in quotation marks.
variable	User-defined variable that specifies the configuration object.

In the following example, a Server object is created and assigned to the variable s2, and the Name attribute is set to server_2:

```
create Server "server_2" as s2;
```

In the following example, a User object is created and assigned to the variable u1, and the Name attribute is set to user1:

```
create User "user1" as u1;
```

By default, the user is assigned to the group "Administrators."

Assigning an Existing Configuration Object to a Variable

Before you can edit an existing configuration object, the Configuration Wizard must find the object in the template and assign it to a variable. To assign a configuration object to a variable, use the find operation, as follows:

```
find object_type "name" as variable;
```

Table 4 describes the find operation parameters for which you must specify values.

Table 4 Parameters for find Operation

Parameter	Description
object_type	Configuration object, as defined in *WebLogic Server Configuration Reference* at: http://e-docs.bea.com/wls/docs81/config_xml/index.html.
name	Value of the Name attribute of the configuration object. This parameter (*name*) must be enclosed in quotation marks.
variable	User-defined variable that specifies the configuration object.

In the following example, a `Server` object named "`myserver`" is assigned to the variable `s1`:

```
find Server "myserver" as s1;
```

Assigning an Existing Child Element to a Variable

Some configuration object types contain child elements. Before you can edit an existing child element, the Configuration Wizard must find the element in the template and assign it to a variable. To assign a child element to a variable, use the `find` operation, as follows:

```
find object_type.child_element "name.child_element_name" as variable;
```

Table 5 describes the `find` operation parameters for which you must specify values.

Note: Silent-mode configuration supports all child elements for the configuration object types. For a complete list of the child elements for the configuration objects, see *WebLogic Server Configuration Reference* at:
http://e-docs.bea.com/wls/docs81/config_xml/index.html.

Table 5 Parameters for find Operation for Child Elements

Parameter	Description
object_type	Configuration object, as defined in *WebLogic Server Configuration Reference* at: http://e-docs.bea.com/wls/docs81/config_xml/index.html.
child_element	The child element of the configuration object. Silent-mode configuration supports all child elements for the configuration object types. For a complete list of the child elements for the configuration objects, see *WebLogic Server Configuration Reference* at: http://e-docs.bea.com/wls/docs81/config_xml/index.html.
name	Value of the Name attribute of the configuration object.
child_element _name	Value of the Name attribute of the child element.
variable	User-defined variable that specifies the child element of the configuration object.

In the following example, a `Server` object named "`myserver`" with a `WebServer` child element named "`mywebserver`" is assigned to the variable `ws1`:

```
find Server.WebServer "myserver.mywebserver" as ws1;
```

Setting Configuration Object Attributes

To set a configuration attribute, specify the variable and attribute name for it, as follows:

```
set variable.attribute "value";
```

Table 6 describes the parameters for which you must specify values.

Table 6 Parameters for set Operation

Parameter	Description
variable	User-defined variable that specifies the configuration object.
	For new configuration objects, you assign a configuration object to a variable during the creation process, as described in "Creating a Configuration Object" on page 5.
	For existing configuration objects, you assign a configuration object to a variable using the find operation, as described in "Assigning an Existing Configuration Object to a Variable" on page 6.
attribute	Configuration object attribute for which you want to specify a value.
value	Value that you want to assign to the specified attribute. This parameter (*value*) must be enclosed in quotation marks.

In the following example, the ListenAddress and ListenPort attributes for the Server configuration object referenced by s2 (that is, server_2) are set to localhost and 7001, respectively:

```
set s2.ListenAdress "localhost";
set s2.ListenPort "7001";
```

Considerations for Setting the JDBC Connection Pool Properties Attribute

The JDBC Connection Pool Properties attribute specifies the list of properties passed to the JDBC Driver, and is used when creating physical database connections.

When specifying this attribute, make sure that you provide the complete list of the properties required by the JDBC Driver being used.

Because the values that you specify for the Properties attribute will override all values currently set, if you omit a value from the Properties list, that property will remain undefined, resulting in unexpected results.

Setting a Specific Attribute for a Child Element

To assign a value to an attribute for a child element, specify the variable assigned to the child element and the name of the attribute, as follows:

```
set variable.attribute "value";
```

Table 7 describes the parameters for which you must specify values.

Table 7 Parameters for set Operation for Setting Child Element Attributes

Parameter	Description
variable	User-defined variable that specifies the child element of the configuration object.
	For existing configuration objects, you assign a child element to a variable using the find operation, as described in "Assigning an Existing Child Element to a Variable" on page 7.
attribute	Child element attribute for which you want to specify a value.
value	Value that you want to assign to the specified attribute. This parameter (*value*) must be enclosed in quotation marks.

In the following example, a Server named "myserver" with a WebServer child element named "mywebserver" is assigned to the variable ws1. Then the LogFileName attribute for the webserver child element is assigned the value of c:\bea\weblogic\logs\newAccess.log:

```
find Server.WebServer "myserver.mywebserver" as ws1;
set ws1.LogFileName "c:\bea\weblogic\logs\newAcess.log";
```

Setting an Attribute for All Instances of a Child Element

To assign a value to an attribute for all instances of a particular type of child element, use the following format:

```
set variable.child_element.attribute "value";
```

Table 8 describes the parameters for which you must specify values.

Table 8 Parameters for set Operation for an Attribute of All Child Element Instances

Parameter	Description
variable	User-defined variable that specifies the configuration object.
	For new configuration objects, you assign a configuration object to a variable during the creation process, as described in "Creating a Configuration Object" on page 5.
	For existing configuration objects, you assign a configuration object to a variable using the find operation, as described in "Assigning an Existing Configuration Object to a Variable" on page 6.
child_element	The child element of the configuration object. Silent-mode configuration supports all child elements for the configuration object types. For a complete list of the child elements for the configuration objects, see *WebLogic Server Configuration Reference* at: http://e-docs.bea.com/wls/docs81/config_xml/index.html.
attribute	Child element attribute for which you want to specify a value.
value	Value that you want to assign to the specified attribute. This parameter (*value*) must be enclosed in quotation marks.

In the following example, the server named "myserver" is assigned to the variable s1. Then, all ThreadCount attributes for the ExecuteQueue child element of the server referenced by s1 are set to 25:

```
find Server "myserver" as s1;
set s1.ExecuteQueue.ThreadCount "25";
```

Setting Attributes for Unnamed Child Elements

When the Name attribute of a child element instance is not defined, you cannot reference that child element individually. In this case, the only way you can assign a value to an attribute of an unnamed child element is by setting all instances of the child element attribute, as described in "Setting an Attribute for All Instances of a Child Element" on page 9.

Setting an HTTP Proxy Server for a Cluster

You can also use the `set` operation to designate an HTTP proxy server for a cluster, as follows:

```
set variable.ProxyServer "unclustered_managed_server";
```

Note: Only an unclustered Managed Server can be used as a proxy server.

Table 9 describes the parameters for which you must specify values when setting an HTTP Proxy server.

Table 9 Parameters for set Operation for HTTP Proxy Server

Parameter	Description
variable	User-defined variable that specifies the `cluster` configuration object.
	For new configuration objects, you assign a configuration object to a variable during the creation process, as described in "Creating a Configuration Object" on page 5.
	For existing configuration objects, you assign a configuration object to a variable using the `find` operation, as described in "Assigning an Existing Configuration Object to a Variable" on page 6.
ProxyServer	Configuration object attribute that specifies the proxy server. Do not change this attribute.
unclustered_managed_server	Name of the unclustered Managed Server that you want to assign as the `ProxyServer` for the cluster. This parameter (*unclustered_managed_server*) must be enclosed in quotation marks.

In the following example, the cluster named `myCluster` is assigned to the variable `mycluster`. Then the server `AnUnClusteredServer` is assigned to the attribute `ProxyServer` for the cluster referenced by `mycluster`:

```
create Cluster "myCluster" as mycluster;
set mycluster.ProxyServer "AnUnClusteredServer";
```

Setting Configuration Options When Creating a New Domain

You can also use the set operation to specify configuration options for the domain, as follows:

```
set configuration_option value;
```

Table 10 describes the parameters for which you must specify values.

Table 10 Parameters for set Operation for Configuration Options

Parameter	Description
configuration_option	Name of the configuration option to be defined. The configuration options that you can specify when creating a new domain are listed in Table 11.
value	Value that you want to assign to the configuration option. This parameter (value) must be enclosed in quotation marks.

In the following example, a Windows Start Menu shortcut for the domain is not created:

```
set CreateStartMenu "false";
```

Table 11 describes the configuration options that you can specify using the set operation.

Table 11 Configuration Options Specified Using the Set Operation When Creating a New Domain

For this configuration option . . .	Specify the following . . .
AllowCasualUpdate	Whether you want to allow a domain to be updated without adding an extension template. The valid value for this option is always true, which allows you to update a domain with or without adding an extension template. **Note:** This option is provided for backwards compatibility.
AppDir	The full pathname for the application directory when a separate directory is required for applications, as indicated in the specified template. If a separate application directory is not required, this setting has no effect. **Note:** This option can be used when you are creating a new domain and extending an existing domain.

Table 11 Configuration Options Specified Using the Set Operation When Creating a New Domain

For this configuration option . . .	Specify the following . . .
CreateStartMenu	Whether you want to create a shortcut for the domain on the Windows Start menu. Valid values for this option are: • true—create shortcut (default) • false—do not create shortcut
JavaHome	The full pathname for the directory for the Java JDK to be used when starting the server.
OverwriteDomain	Whether you want to overwrite an existing domain. Make sure you create a backup copy of any existing domain that you want to persist. The default is false.
ServerStartMode	The startup mode for your domain. Valid values for this option are: • dev—Start the server in development mode. (default) • prod—Start the server in production mode. For more information about choosing the start mode for your domain, see "Specify the Server Start Mode and Java JDK" in *Creating WebLogic Domains Using the Configuration Wizard* at the following URL: http://e-docs.bea.com/common/docs90/confgwiz/newdom. html#startmode

Assigning Servers to Clusters

You can assign servers to clusters by using the assign Server operation as follows:

```
assign Server "server" to Cluster "cluster";
```

Table 12 describes the assign Server operation parameters for which you must specify a value.

Table 12 Parameters for assign Server Operation

Parameter	Description
server	Name of the server that you want to assign to the cluster. You can assign multiple servers to a cluster by specifying multiple server names, separated by commas. The server names must be enclosed in quotation marks.

Table 12 Parameters for assign Server Operation (Continued)

Parameter	Description
cluster	Name of the cluster to which you want the server(s) assigned. A server can be assigned to only one cluster. The cluster name must be enclosed in quotation marks.

In the following example, the servers named `myServer1` and `myServer2` are assigned to the cluster `myCluster`:

```
assign Server "myServer1,myServer2" to Cluster "myCluster";
```

Assigning All Applications and Services to Servers and Clusters

You can assign all the applications and services in a domain to servers and clusters by using the `assign all` operation as follows:

- To assign all applications to a server or cluster, use the `assign all Applications` operation as follows:

  ```
  assign all Applications to Target "server_or_clusters"
  ```

  In the following example, all applications in the domain are assigned to the targets `AdminServer` and `Cluster1`:

  ```
  assign all Applications to Target "adminServer,cluster1";
  ```

- To assign all services to a server or cluster, use the `assign all Services` operation, as follows:

  ```
  assign all Services to Target "server_or_clusters";
  ```

  In the following example, all services in the domain are assigned to the targets `AdminServer` and `Cluster1`:

  ```
  assign all Services to Target "adminServer,cluster1";
  ```

Table 13 describes the single `assign all` operation parameter for which you must specify a value.

Table 13 Parameter for assign all Operation

Parameter	Description
server_or_clusters	Name of the server or cluster to which you want to target applications or services. You can specify multiple server or cluster names, separated by commas. Target server and cluster names must be enclosed in quotation marks.

If you use the `assign all Services` operation, any of the following services that are defined in the `config.xml` for the domain, or that you added in this script, are assigned to the target server or cluster:

- MigratableRMIService

- Shutdownclass

- Startupclass

- FileT3

- RMCFactory

- MailSession

- MessagingBridge

- JMSConnectionFactory

- JDBCConnectionPool

- JDBCMultipool

- JDBCTxDataSource

- JDBCDataSource

- JDBCPoolComp

- JoltConnectionPool

- WLECConnectionPool

- WTCServer

Assigning Individual Applications to Servers and Clusters

You can assign individual applications and application components (such as Web applications and EJB modules) to a server or cluster by using the `assign Application` operation, as follows:

```
assign Application "application_names" to Target "server_or_clusters";
```

Table 14 describes the `assign Application` operation parameters for which you must specify a value.

Table 14 Parameters for assign Application Operation

Parameter	Description
application_names	Name of the application or component that you want to assign to the server or cluster. You can specify multiple application names, separated by commas. To specify an application component, use the following format:
	application_name.component_name
	Note: Because the . (period) is used as a separator in the specification of an application component, you cannot use periods in your application or component names.
server_or_clusters	Name of the server or cluster to which you want to target applications or services. You can specify multiple server or cluster names, separated by commas. Target server and cluster names must be enclosed in quotation marks.

In the following example, the applications named `MedRecEAR` and `PhysicianEAR` are assigned to the server `myServer1`:

```
assign Application "MedRecEAR, PhysicianEAR" to Target "myServer1";
```

In the following example, the application component `SessionEJB` for the application `MedRecEAR` is assigned to the server `MedRecServer`:

```
assign Application "MedRecEAR.SessionEJB" to Target "MedRecServer";
```

Assigning Individual Services to Servers and Clusters

You can assign an individual service to a server or cluster by name or category, as follows:

```
assign service_type "service_names" to Target "server_or_clusters";
```

Table 15 describes the `assign` operation parameters for which you must specify a value when you are assigning individual services to servers or clusters.

Table 15 Parameters for assign Operation for Assigning Individual Services

Parameter	Description
service_type	Type of service to be assigned. In addition to the services listed in "Assigning All Applications and Services to Servers and Clusters" on page 14, you can also individually assign the following service categories: • JMSServer • JMSDistributedQueue • JMSDistributedTopic
service_names	Name of the services that you want to assign to the server or cluster. You can specify multiple service names, separated by commas. If you want to assign all services of the type specified using the `service_type` parameter, specify an asterisk for this parameter. The `service_names` parameter must be enclosed in quotation marks.
server_or_clusters	Name of the server or cluster to which you want to target service(s). You can specify multiple server or cluster names, separated by commas. Target server and cluster names must be enclosed in quotation marks.

In the following example, the service `MedRecJMSServer` is assigned to the server `myServer1`:

```
assign JMSServer "MedRecJMSServer" to Target "myServer1";
```

In the following example, all JMS servers are assigned to `myServer1`:

```
assign JMSServer "*" to Target "myServer1";
```

Creating Security Assignments

When you are creating a new domain, you can perform the following assignments:

- Assign users to groups—You can designate individual users as members of a particular group. Groups allow you to manage multiple users simultaneously. This is generally more efficient than managing each user individually.

  Note: Any user that you create is assigned, automatically, to the Administrators group.

- Assign groups to groups—You can designate one group a subgroup of another, to refine security management.

To assign users to groups, use the `assign User` operation as follows:

```
assign User "username" to Group "group";
```

To assign groups to groups, use the assign Group operation as follows:

```
assign Group "group" to Group "group";
```

Table 16 describes the `assign` operation parameters for which you must specify a value when you are assigning users and groups to groups.

Table 16 Parameters for assign Users and assign Groups Operations

Parameter	Description
username	Name of the user that you want to assign to a particular group. The username must be enclosed in quotation marks.
group	Name of the group to which you want to assign a user or group, or that you want to assign to another group. The group name must be enclosed in quotation marks.

In the following example, the user `weblogic` is assigned to the group `Monitors` and the group `Deployers` is designated a subgroup of the group `Monitors`:

```
assign User "weblogic" to Group "Monitors";
assign Group "Deployers" to Group "Administrators";
```

Removing Assignments Using the Unassign Operation

You can use the `unassign` operation to remove assignments using the same operations as the `assign` command. To do so, change `assign` to `unassign`, and the `to` operator to `from`.

The following examples illustrate sample usage for the `unassign` operation:

```
unassign Application "MedRecEAR, PhysicianEAR" from Target "myServer1";
unassign all Services from Target "adminServer,cluster1";
unassign User "weblogic" from Group "Monitors";
```

Deleting a Configuration Object

To delete a configuration object, use the `delete` operation, as follows:

```
delete variable;
```

Table 17 describes the single `delete` operation parameter for which you must specify a value.

Table 17 Parameters for delete Operation

Parameter	Description
variable	User-defined variable that specifies the configuration object.
	For new configuration objects, you assign a configuration object to a variable during the creation process, as described in "Creating a Configuration Object" on page 5.
	For existing configuration objects, you assign a configuration object to a variable using the `find` operation, as described in "Assigning an Existing Configuration Object to a Variable" on page 6.
	Note: You cannot delete an administration server; you can only modify its name or configuration.

In the following example, the `Server` configuration object referenced by `s2` (that is, `server_2`) is deleted:

```
delete s2;
```

Step 3: Create the Domain

To create the domain using the configuration settings that you have defined in step 2, use the `write` operation, as follows:

```
write domain to "domain_name";
```

Table 18 describes the single `write` operation parameter for which you must specify a value.

Table 18 Parameters for write Operation

Parameter	Description
domain_name	Full pathname for the directory in which you want to create the domain, including the name of the domain. The pathname must be enclosed in quotation marks.

In the following example, the domain is created in the `C:\bea\user_projects\domains` directory and is named `wlsDomain`:

```
write domain to "C:\bea\user_projects\domains\wlsDomain";
```

Step 4: Close the Domain Template

To close the domain template, use the `close` operation, as follows:

```
close template;
```

When you have finished defining the configuration, save the script. You can name the script as desired, and pass the relative or full pathname for it to the `config` command, as described in "Starting the Configuration Wizard in Silent Mode" on page 1.

Creating a Script to Extend an Existing Domain in Silent Mode

The following sections describe the steps that you must perform in the script to create a new domain:

- Step 1: Select the Domain Directory

- Step 2: Add the Extension Template to the Domain

- Step 3: Edit the Existing Configuration Information (Optional)

- Step 4: Update the Domain

- Step 5: Close the Domain

Note: Each operation described in this section must be terminated by a semicolon.

Step 1: Select the Domain Directory

To select a domain directory that you want to update with additional applications and services, use the `read` operation, as follows:

```
read domain from "domain_directory_name";
```

Table 19 describes the single `read domain` operation parameter for which you must specify a value.

Table 19 Parameters for read domain Operation

Parameter	Description
`domain_directory_name`	Full pathname for an existing domain directory that you want to update. The pathname must be enclosed in quotation marks.

In the following example, the absolute pathname to a domain named `mydomain` is specified:

```
read domain from "C:\bea\user_projects\domains\mydomain";
```

Step 2: Add the Extension Template to the Domain

An extension template defines applications and services, such as JDBC or JMS components, and startup/shutdown classes, that can be used to extend an existing domain. Use the `add` operation to add an extension template to the domain you specified using the `read domain` operation in step 1:

```
add template extension_template_jar_name;
```

Table 20 describes the single `add template` operation parameter for which you must specify a value.

Table 20 Parameter for add template Operation

Parameter	Description
`extension_template_jar _name`	Full pathname of the extension template, enclosed in quotation marks.

In the following example, an extension template that adds a basic Web application (including an HTML file, image files, and required XML descriptors) is specified, using the absolute pathname for the template:

```
add template
"C:\bea\weblogic90\common\templates\applications\wls_default.jar";
```

Step 3: Edit the Existing Configuration Information (Optional)

You can edit the default configuration information to create new configuration objects, edit existing values for configuration object attributes, delete configuration objects, and assign applications and services to servers or clusters. The operations available for these tasks are the same ones available for extending a domain and creating a new domain. For instructions about using these operations, see the following sections:

- "Creating a Configuration Object" on page 5

- "Assigning an Existing Configuration Object to a Variable" on page 6

- "Setting Configuration Object Attributes" on page 8

- "Deleting a Configuration Object" on page 19

- "Assigning All Applications and Services to Servers and Clusters" on page 14

- "Assigning Individual Applications to Servers and Clusters" on page 16

- "Assigning Individual Services to Servers and Clusters" on page 17

You can also set configuration options when extending a domain, as described in "Setting Configuration Options When Extending a Domain" on page 23.

You can use silent mode to create and edit any valid configuration object and its child elements, except security objects. For more information about valid configuration objects, see *WebLogic Server Configuration Reference* at:

http://e-docs.bea.com/wls/docs81/config_xml/index.html.

Setting Configuration Options When Extending a Domain

When you are extending a domain, you can use the `set` operation to specify configuration options for the domain as follows:

set *configuration_option value;*

Table 21 describes the parameters for which you must specify values.

Table 21 Parameters for set Operation for Configuration Options

Parameter	Description
configuration_option	Name of the configuration option to be defined. The configuration options that you can specify when you are extending an existing domain are listed in Table 22.
value	Value that you want to assign to the configuration option. This parameter (*value*) must be enclosed in quotation marks.

Table 22 lists the configuration options that you can specify using the `set` operation when you are extending an existing domain.

Table 22 Configuration Options Specified Using the Set Operation When Extending a Domain

For this configuration option . . .	Specify the following . . .
AppDir	The full pathname for the application directory when a separate directory is required for applications, as indicated in the specified template. If a separate application directory is not required, this setting has no effect.
	Note: This option can be used when you are creating a new domain or extending an existing domain.
ReplaceDuplicates	Whether you want to keep the original configuration in the domain or overwrite it with updated configuration information from the extension template when there is a conflict between the two.

Step 4: Update the Domain

To update the domain opened in the `read domain` operation, use the `update` operation, as follows:

```
update domain;
```

Note: You must specify the `add template` operation before you can update the domain.

Step 5: Close the Domain

To close the domain, use the `close` operation, as follows:

```
close domain;
```

When you have finished defining the configuration, save the script. You can name the script as desired, and pass a relative or full pathname for it to the `config` command, as described in .

Sample Configuration Scripts for Silent-Mode Configuration

This section provides two sample scripts that demonstrate the basic concepts required to create and update domains using silent-mode configuration:

- Sample Script for Creating a New Domain from a Domain Template
- Sample Script for Creating a New Domain and Updating It Using an Extension Template

Sample Script for Creating a New Domain from a Domain Template

The following sample script shows how to create a new domain, based on the Basic WebLogic Server Domain template.

The numbers shown along the left-hand margin mark the beginning of each section of code that performs a major task within the script. For details about a particular task, see the description corresponding to the numeric callout in Table 23.

Listing 1   Sample Script for Creating a New Domain Based on the Basic WebLogic Server Domain

```
1 read template from "C:\bea\weblogic90\common\templates\domains\wls.jar";
  /////////////////////////////////////////////////////////////////////////
  //Find and configure the Admin Server.
  /////////////////////////////////////////////////////////////////////////
2 find Server "myserver" as s1;
  set s1.ListenAddress "";
  set s1.ListenPort "7001";
  set s1.SSL.Enabled "true";
  set s1.SSL.ListenPort "7002";
  /////////////////////////////////////////////////////////////////////////
  //Create a JMSQueue.
  /////////////////////////////////////////////////////////////////////////
  //A JMSServer has to be created first.
3 create JMSServer "myJMSServer" as jmsserver;
  create JMSQueue "myJMSQueue" as myq;
  //required attribute
  set myq.JNDIName "jms/myjmsqueue";
  //required attribute
```

```
 set myq.JMSServer "myJMSServer";
 //optional attribute
 //set myq.StoreEnabled "false";
 //target "myJMSServer" to server "myserver"
 assign JMSServer "myJMSServer" to target "myserver";
 //////////////////////////////////////////////////////////////////////
 //Create a JDBCConnectionPool.
 //////////////////////////////////////////////////////////////////////
 create JDBCConnectionPool "demoPool" as mypool;
 //required attribute
 set mypool.DriverName "com.pointbase.jdbc.jdbcUniversalDriver";
 //required attribute
 set mypool.URL "jdbc:pointbase:server://localhost:9092/demo";
 //required attribute
 set mypool.Password "PBPUBLIC";
 //optional attribute (but it's recommended you set the db user...)
 set mypool.Properties "user=PBPUBLIC";
 //target all JDBC connection pools to server "myserver"
 assign JDBCConnectionPool "*" to target "myserver";
 //////////////////////////////////////////////////////////////////////
 //target existing applications.
 //////////////////////////////////////////////////////////////////////
 //target applications only when they exist in current domain template
 //assign application "*" to target "myserver";
 //////////////////////////////////////////////////////////////////////
 //Find the admin user and set the password, since it has not been set in the
 // template.
 //////////////////////////////////////////////////////////////////////
 find User "weblogic" as u1;
 set u1.password "weblogic";
 //////////////////////////////////////////////////////////////////////
 //Write out the domain.
 //////////////////////////////////////////////////////////////////////
 set OverwriteDomain "true";
 write domain to "C:\bea\user_projects\domains\wls";
 //////////////////////////////////////////////////////////////////////
 //Close domain template to indicate completion of work.
```

```
/////////////////////////////////////////////////////////////////////
9 close template;
```

The following table defines the main tasks performed by the sample script shown in Listing 1.

Table 23 Major Tasks Shown in Sample Script Based on WebLogic Server Domain Template

In this section of code . . .	The script . . .
1	Selects the Basic WebLogic Server Domain template. For more information, see "Step 1: Select the Domain Template" on page 3.
2	Configures the Administration Server. For more information, see the following relevant sections in "Step 2: Edit the Configuration Information": • "Assigning an Existing Configuration Object to a Variable" on page 6 • "Setting Configuration Object Attributes" on page 8
3	Creates a JMS queue. For more information, see the following relevant sections in "Step 2: Edit the Configuration Information": • "Creating a Configuration Object" on page 5 • "Setting Configuration Object Attributes" on page 8 • "Assigning Individual Services to Servers and Clusters" on page 17

Table 23 Major Tasks Shown in Sample Script Based on WebLogic Server Domain Template (Continued)

In this section of code . . .	The script . . .
4	Creates a JDBC connection pool.
	When specifying the JDBC connection pool `Properties` attribute, make sure you provide the complete list of properties required by the JDBC driver being used. For guidelines for setting the JDBC connection pool `Properties` attribute, see "Considerations for Setting the JDBC Connection Pool Properties Attribute" on page 8. In Listing 1, the following `Properties` specification is acceptable because the JDBC PointBase driver only requires definition of the `user`:
	`set mypool.Properties "user=PBPUBLIC";`
	Alternatively, you can assign the properties separately. For example, to define the user name, you can also specify the following command:
	`set mypool.UserName "myname";`
	For more information, see the following relevant sections in "Step 2: Edit the Configuration Information":
	• "Creating a Configuration Object" on page 5
	• "Setting Configuration Object Attributes" on page 8
	• "Assigning Individual Services to Servers and Clusters" on page 17
5	Optionally assigns applications that exist in the domain to `myserver`. By default, the Basic WebLogic Server Domain contains no applications. For more information, see "Assigning Individual Applications to Servers and Clusters" on page 16.
6	Updates the password for the user `weblogic`. For more information, see the following relevant sections in "Step 2: Edit the Configuration Information":
	• "Assigning an Existing Configuration Object to a Variable" on page 6
	• "Creating Security Assignments" on page 18
	Note: To review the predefined security information for a domain, view the `security.xml` file included in the domain template JAR file. If a user password is set to `"????????"` (indicating that it is not set), it must be defined in the silent script.
7	Sets the `OverwriteDomain` configuration option to `true` and stipulates that if the specified domain exists, it can be overwritten. For more information, see "Setting Configuration Options When Creating a New Domain" on page 12.

Table 23 Major Tasks Shown in Sample Script Based on WebLogic Server Domain Template (Continued)

In this section of code . . .	The script . . .
8	Creates the domain in `C:\bea\user_projects\domains\wls`. For more information, see "Step 3: Create the Domain" on page 20.
9	Closes the domain template. For more information, see "Step 4: Close the Domain Template" on page 20.

Sample Script for Creating a New Domain and Updating It Using an Extension Template

The following sample script shows how to create a new domain, based on the Basic WebLogic Server Domain template, and then update it using the WebLogic Server Default Domain extension template.

The numbers shown along the left-hand margin mark the beginning of each section of code that performs a major task within the script. For details about a particular task, see the description corresponding to the numeric callout in Table 24.

Listing 2 Sample Script for Creating a New Domain and Updating It Using an Extension Template

```
/*Create a domain from a domain template and update it */

read template from "C:\bea\weblogic90\common\templates\domains\wls.jar";

find User "weblogic" as admin;
set admin.Password "newPassword";
create User "user1" as u1;
set u1.Password "password";
assign User "user1" to Group "Deployers";

set ServerStartMode "dev";
set JavaHome "C:\bea\jrockit90_150_03";
set CreateStartMenu "false";

write domain to "C:\bea\user_projects\domains\wls";
```

```
5 close template;

  /* update an existing domain */
6 read domain from "C:\bea\user_projects\domains\wls";

  //keep originals if there are duplicates from added extension template
  //We do not really need this here since wls template does not have
    applications.
7 set ReplaceDuplicates "false";

  //add an extension template
8 add template
      "C:\bea\weblogic90\common\templates\applications\wls_default.jar";

9 update domain;
10 close domain;
```

The following table defined the main tasks performed by the sample script performs the tasks shown in Listing 2.

Table 24 Major Tasks Shown in Sample Script For Creating a New Domain and Updating it Using an Extension Template

In this section of code . . .	The script . . .
1	Selects the Basic WebLogic Server Domain template. For more information, see "Step 1: Select the Domain Template" on page 3.
2	Defines users for the domain. For more information, see the following relevant sections in "Step 2: Edit the Configuration Information": • "Creating a Configuration Object" on page 5 • "Assigning an Existing Configuration Object to a Variable" on page 6 • "Creating Security Assignments" on page 18 **Note:** To review the predefined security information for a domain, view the security.xml file included in the domain template JAR file. If a user password is set to "????????" (indicating that it is not set), it must be defined in the silent script.

Table 24 Major Tasks Shown in Sample Script For Creating a New Domain and Updating it Using an Extension Template (Continued)

In this section of code . . .	The script . . .
3	Sets several configuration options. For more information, see "Setting Configuration Options When Creating a New Domain" on page 12.
4	Creates the domain in `C:\bea\user_projects\domains\wls`. For more information, see "Step 3: Create the Domain" on page 20.
5	Closes the domain template. For more information, see "Step 4: Close the Domain Template" on page 20.
6	Selects the domain directory to be extended. For more information, see "Step 1: Select the Domain Directory" on page 21.
7	Sets the `ReplaceDuplicate` configuration option to `false` specifying that duplicate configuration options defined in the extension template should not be overwritten. For more information, see "Setting Configuration Options When Extending a Domain" on page 23.
8	Extends the domain to support a basic Web application by importing resources from the WebLogic Server Default Domain extension template. For more information, see "Step 2: Add the Extension Template to the Domain" on page 22.
9	Updates the domain. For more information, see "Step 4: Update the Domain" on page 24.
10	Closes the domain. For more information, see "Step 5: Close the Domain" on page 24.