\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**THREAD\_DUMP**

Provides a real-time snapshot of the WebLogic Server threads that are currently running.

**Syntax**

java weblogic.Admin [[Connection and User Credentials Arguments](https://docs.oracle.com/cd/E13222_01/wls/docs61/adminguide/cli.html#1160341)]

THREAD\_DUMP

**Example:**

java weblogic.Admin -url localhost:7001 -username adminuser

-password gumby1234 THREAD\_DUMP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**UNLOCK**

Unlocks the specified WebLogic Server after a [LOCK](https://docs.oracle.com/cd/E13222_01/wls/docs61/adminguide/cli.html#1131335) operation.

**Syntax**

java weblogic.Admin [[Connection and User Credentials Arguments](https://docs.oracle.com/cd/E13222_01/wls/docs61/adminguide/cli.html#1160341)]

UNLOCK

**Example:**

java weblogic.Admin -url localhost:7001 -username adminuser

-password gumby1234 UNLOCK

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**VERSION**

Displays the version of the WebLogic Server software that is running on the machine specified by the value of URL.

**Syntax**

java weblogic.Admin [[Connection and User Credentials Arguments](https://docs.oracle.com/cd/E13222_01/wls/docs61/adminguide/cli.html#1160341)]

VERSION

**Example**

In the following example, a user requests the version of the WebLogic Server running on port 7001 on machine localhost:

java weblogic.Admin -url localhost:7001 -username guest

-password guest VERSION

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**WebLogic Server Connection Pools Administration Command**

1. [CREATE\_POOL](https://docs.oracle.com/cd/E13222_01/wls/docs61/adminguide/cli.html#1155747)
2. [DESTROY\_POOL](https://docs.oracle.com/cd/E13222_01/wls/docs61/adminguide/cli.html#1155842)
3. [DISABLE\_POOL](https://docs.oracle.com/cd/E13222_01/wls/docs61/adminguide/cli.html#1155898)
4. [ENABLE\_POOL](https://docs.oracle.com/cd/E13222_01/wls/docs61/adminguide/cli.html#1156127)
5. [EXISTS\_POOL](https://docs.oracle.com/cd/E13222_01/wls/docs61/adminguide/cli.html#1156010)
6. [RESET\_POOL](https://docs.oracle.com/cd/E13222_01/wls/docs61/adminguide/cli.html#1156066)
7. **CREATE\_POOL**

Allows creation of connection pool while WebLogic Server is running. For more information, see ["Creating a Connection Pool Dynamically](https://docs.oracle.com/cd/E13222_01/wls/docs61/jdbc/programming.html#programming004) in Programming WebLogic JDBC.

**Syntax**

java weblogic.Admin [[Connection and User Credentials Arguments](https://docs.oracle.com/cd/E13222_01/wls/docs61/adminguide/cli.html#1160341)]

CREATE\_POOL poolName aclName=aclX,

props=myProps,initialCapacity=1,maxCapacity=1,

capacityIncrement=1,allowShrinking=true,shrinkPeriodMins=15,

driver=myDriver,url=myURL

**Example:**

java weblogic.Admin -url localhost:7001 -username adminuser

-password gumby1234 CREATE\_POOL myPool

java weblogic.Admin -url t3://forest:7901 -username system

-password gumby1234 CREATE\_POOL dynapool6 "aclName=someAcl,

allowShrinking=true,shrinkPeriodMins=10,

url=jdbc:weblogic:oracle,driver=weblogic.jdbc.oci.Driver,

initialCapacity=2,maxCapacity=8,

props=user=SCOTT;password=tiger;server=bay816"

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**DESTROY\_POOL**

Connections are closed and removed from the pool and the pool dies when it has no remaining connections. Only the "system" user or users granted "admin" permission by an ACL associated with a connection pool can destroy the pool.

**Syntax**

java weblogic.Admin [[Connection and User Credentials Arguments](https://docs.oracle.com/cd/E13222_01/wls/docs61/adminguide/cli.html#1160341)]

DESTROY\_POOL poolName [true|false]

|  |  |
| --- | --- |
| **Argument** | **Definition** |
| poolName | Required. Unique name of pool. |
| false  (soft shutdown) | Soft shutdown waits for connections to be returned to the pool before closing them. |
| true  (default—hard shutdown) | Hard shutdown kills all connections immediately. Clients using connections from the pool get exceptions if they attempt to use a connection after a hard shutdown. |

 

**Example**

In the following example, a user with the name adminuser and the password gumby1234 runs the DESTROY\_POOL command temporarily freeze the active pool connections:

java weblogic.Admin -url localhost:7001 -username adminuser

-password gumby1234 DESTROY\_POOL myPool false

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**DISABLE\_POOL**

You can temporarily disable a connection pool, preventing any clients from obtaining a connection from the pool. Only the "system" user or users granted "admin" permission by an ACL associated with a connection pool can disable or enable the pool.

You have to options for disabling a pool. 1) Freezing the connections in a pool that you later plan to enable, and 2) destroy the connections.

**Syntax**

java weblogic.Admin [[Connection and User Credentials Arguments](https://docs.oracle.com/cd/E13222_01/wls/docs61/adminguide/cli.html#1160341)]

DISABLE\_POOL poolName [true|false]

|  |  |
| --- | --- |
| **Argument** | **Definition** |
| poolName | Name of the connection pool |
| false  (disables and **suspends**) | Disables the connection pool, and suspends clients that currently have a connection. Attempts to communicate with the database server throw an exception. Clients can, however, close their connections while the connection pool is disabled; the connections are then returned to the pool and cannot be reserved by another client until the pool is enabled. |
| true  (default—disables and **destroys**) | Disables the connection pool, and destroys the client's JDBC connection to the pool. Any transaction on the connection is rolled back and the connection is returned to the connection pool. |

 

**Example**

In the following example, a user with the name adminuser and the password gumby1234 runs the DISABLE\_POOL command to freeze a connection that is to be enabled later:

java weblogic.Admin -url localhost:7001 -username adminuser

-password gumby1234 DISABLE\_POOL myPool false

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**ENABLE\_POOL**

When a pool is enabled, the JDBC connection states for each in-use connection are exactly as they were when the connection pool was disabled; clients can continue JDBC operations exactly where they left off.

**Syntax**

java weblogic.Admin [[Connection and User Credentials Arguments](https://docs.oracle.com/cd/E13222_01/wls/docs61/adminguide/cli.html#1160341)]

ENABLE\_POOL poolName

|  |  |
| --- | --- |
| **Argument** | **Definition** |
| poolName | Name of the connection pool. |

 

**Example**

In the following example, a user with the name adminuser and the password gumby1234 runs the ENABLE\_POOL command to reestablish connections that have been disabled (frozen):

java weblogic.Admin -url localhost:7001 -username adminuser

-password gumby1234 ENABLE\_POOL myPool

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**EXISTS\_POOL**

Tests whether a connection pool with a specified name exists in the WebLogic Server. You can use this method to determine whether a dynamic connection pool has already been created or to ensure that you select a unique name for a dynamic connection pool you want to create.

**Syntax**

java weblogic.Admin [[Connection and User Credentials Arguments](https://docs.oracle.com/cd/E13222_01/wls/docs61/adminguide/cli.html#1160341)]

EXISTS\_POOL poolName

|  |  |
| --- | --- |
| **Argument** | **Definition** |
| poolName | Name of connection pool. |

 

**Example**

In the following example, a user with the name adminuser and the password gumby1234 runs the EXISTS\_POOL command to determine wether or not a pool with a specific name exists:

java weblogic.Admin -url localhost:7001 -username adminuser

-password gumby1234 EXISTS\_POOL myPool

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**RESET\_POOL**

This command resets the connections in a registered connection pool.

This is a privileged command. You must supply the password for the WebLogic Server administrative user to use this command. You must know the name of the connection pool, which is an entry in the config.xml file.

**Syntax**

java weblogic.Admin URL RESET\_POOL poolName system password

|  |  |
| --- | --- |
| **Argument** | **Definition** |
| URL | The URL of the WebLogic Server host and port number of the TCP port at which WebLogic is listening for client requests; use "t3://host:port." |
| poolName | Name of a connection pool as it is registered in the WebLogic Server's config.xml file. |
| password | Administrative password for the user "system". You must supply the username "system" and the administrative password to use this Admin command. |

 

**Example**

This command refreshes the connection pool registered as "eng" for the WebLogic Server listening on port 7001 of the host xyz.com.

java weblogic.Admin t3://xyz.com:7001 RESET\_POOL eng system gumby

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Deploying a New J2EE Application:**

To deploy a J2EE application file (.jar, .war, or .ear) or application directory that is not deployed to WebLogic, enter the following command:

**Syntax:**

% java weblogic.deploy -port port\_number -host host\_name

deploy password application source

The values are as follows:

* application is the string you want to assign to this Application.
* source is the full pathname of the J2EE application file (.jar,.war,.ear) you want to deploy, or the full pathname of the application directory.

**Example:**

% java weblogic.deploy -port 7001 -host localhost deploy weblogicpwd Basic\_example

c:\mysamples\ejb\basic\BasicStatefulTraderBean.jar

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Viewing a Deployed J2EE Application**

To view an application that is deployed on a local WebLogic Server, enter the following command:

**Syntax:**

% java weblogic.deploy list password

The value of password is the password for the WebLogic Server system account.

To list a deployed application on a remote server, specify the port and host options, as follows:

% java weblogic.deploy -port port\_number -host host\_name list password

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Removing a Deployed J2EE Application**

To remove a deployed J2EE application, you need only reference the assigned application name, as shown in the following example:

% java weblogic.deploy -port 7001 -host localhost undeploy

weblogicpwd Basic\_example

**Note:**Removing a J2EE application does not remove the application from WebLogic Server. You cannot re-use the application name with the deploy utility. You can re-use the application name to update the deployment, as described in the following section.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Updating a Deployed J2EE Application**

To update a J2EE application, use the update argument and specify the name of the active J2EE application as follows:

**Syntax:**

% java weblogic.deploy -port 7001 -host localhost update

weblogicpwd Basic\_example

c:\updatesample\ejb\basic\BasicStatefulTraderBean.jar

To update a specific component, enter the following commands:

% java weblogic.deploy -port 7001 -host localhost -component

Basic\_example:sampleserver,exampleserver update

weblogicpwd Basic\_example

c:\updatesample\ejb\basic\BasicStatefulTraderBean.jar

update will cause the application or component to be updated on all server instances to which is targeted

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**logToZip**

The logToZip utility searches an HTTP server log file in common log format, finds the Java classes loaded into it by the server, and creates an uncompressed .zip file that contains those Java classes. It is executed from the document root directory of your HTTP server.

To use this utility, you must have access to the log files created by the HTTP server.

**Syntax**

$ java utils.logToZip *logfile* *codebase* *zipfile*

**Examples**

The following example shows how a .zip file is created for an applet that resides in the document root itself, that is, with no code base:

$ cd /HTTP/Serv/docs

$ java utils.logToZip /HTTP/Serv/logs/access "" app2.zip

The following example shows how a .zip file is created for an applet that resides in a subdirectory of the document root:

C:\>cd \HTTP\Serv

C:\HTTP\Serv>java utils.logToZip \logs\applets\classes app3.zip