

Name redeploy– redeploys the specified application

Synopsis redeploy

```

[ --terse={true|false} ] [ --echo={true|false} ]
[ --interactive={true|false} ] [ --host host ]
[ --port port ] [ --secure| -s ] [ --user admin_user ]
[ --passwordfile filename ] [ --help ]
    --name component-name [ --upload={true|false} ]
[ --retrieve local_dirpath ] [ --dbvendorname dbvendorname ]
[ --createtables={true|false} | --dropandcreatetables = {true|false} ]
[ --uniquetablenames = {true|false} ] [ --deploymentplan deployment_plan ]
[ --enabled={true|false} ]
[ --contextroot context-root ]
[ --precompilejsp={true|false} ]
[ --virtualservers virtual-servers ]
[ --libraries jar_file [ path_separator ] jar_file* ]
[ --type pkg-type ]
[ --properties (name=value) [ :name=value* ] ] [ archive-path ]

```

Description The redeploy command redeploys an application that is already deployed. The redeploy command preserves the settings and other options with which the application was originally deployed. The application must already be deployed. Otherwise, an error occurs.

This command is supported in remote mode only.

Options -t --terse

Indicates that any output data must be very concise, typically avoiding human-friendly sentences and favoring well-formatted data for consumption by a script. Default is false.

-e --echo

If set to true, the command-line statement is echoed on the standard output. Default is false.

-I --interactive

If set to true (default), only the required password options are prompted.

-H --host

The machine name where the domain administration server is running. The default value is localhost.

-p --port

The HTTP port or HTTPS port for administration. This port is the port in the URL that you specify in your web browser to manage the domain, for example, `http://localhost:4848`.

The default port number for administration is 4848.

-s --secure

If set to true, uses SSL/TLS to communicate with the domain administration server.

The default is false.

`-u --user`

The user name of the authorized administrative user of the domain administration server.

If you have authenticated to a domain by using the `asadmin login` command, you need not specify the `--user` option for subsequent operations on the domain.

`--passwordfile`

Specifies the name, including the full path, of a file that contains the password entries in a specific format.

The entry for a password must have the `AS_ADMIN_` prefix followed by the password name in uppercase letters. For example, to specify the password for the domain administration server, use an entry with the following format:

```
AS_ADMIN_PASSWORD=password
```

In this example, *password* is the actual administrator password.

The following other passwords can also be specified:

- `AS_ADMIN_MAPPEDPASSWORD`
- `AS_ADMIN_USERPASSWORD`
- `AS_ADMIN_ALIASPASSWORD`

All remote commands must specify the administration password to authenticate to the domain administration server. The password can be specified by one of the following means:

- Through the `--passwordfile` option
- Through the `asadmin login` command
- Interactively at the command prompt

The `asadmin login` command can be used only to specify the administration password. For other passwords that remote commands require, use the `--passwordfile` option or specify them at the command prompt.

After authenticating to a domain by using the `asadmin login` command, you need not specify the administration password through the `--passwordfile` option for subsequent operations on the domain. However, only the `AS_ADMIN_PASSWORD` option is not required. You still must provide the other passwords, for example, `AS_ADMIN_USERPASSWORD`, when required by individual commands, such as `update-file-user`.

For security reasons, a password that is specified as an environment variable is not read by the `asadmin` command.

The default value for `AS_ADMIN_MASTERPASSWORD` is `changeit`.

`--help`

Displays the help text for the command.

--name

The name of the application that is being redeployed. This option must specify an application that has previously been deployed. If the application does not exist, an error occurs.

--upload

When set to true (the default), uploads the deployable file to the administration server. The deployable file must be accessible from the client. If the file is accessible to both server and client, set the `--upload` option to false.

--retrieve

Retrieves the client stub JAR file from the server machine to the local directory.

--dbvendorname

Specifies the name of the database vendor for which tables are created. Supported values include `db2`, `mssql`, `oracle`, `derby`, `javadb`, `postgresql`, `pointbase`, and `sybase`, case-insensitive. If not specified, the value of the `database-vendor-name` attribute in `sun-ejb-jar.xml` is used. If no value is specified, a connection is made to the resource specified by the `jndi-name` subelement of the `cmp-resource` element in the `sun-ejb-jar.xml` file, and the database vendor name is read. If the connection cannot be established, or if the value is not recognized, SQL-92 compliance is presumed.

--createtables

Creates tables at deployment of an application with unmapped CMP beans. Default is the `create-tables-at-deploy` entry in the `cmp-resource` element of the `sun-ejb-jar.xml` file.

--dropandcreatetables

If set to true, when the component is redeployed, the tables created by the previous deployment are dropped before creating the new tables. Applies to already deployed applications with unmapped CMP beans. If not set to true, the tables are dropped if the `drop-tables-at-undeploy` entry in the `cmp-resource` element of the `sun-ejb-jar.xml` file is set to true. The new tables are created if the `create-tables-at-deploy` entry in the `cmp-resource` element of the `sun-ejb-jar.xml` file is set to true.

--uniquetablenames

Guarantees unique table names for all the beans and results in a hashcode added to the table names. This is useful if you have an application with case-sensitive bean names.

--deploymentplan

Takes the deployment plan, which is a JAR containing Sun-specific descriptors, and deploys it. This should be passed along when deploying a pure EAR file. A pure EAR file is an EAR without Sun-specific descriptors.

--enabled

If set to true (default), allows users to access the application. If set to false, users will not be able to access the application. In a domain whose profile is the cluster profile or the enterprise profile, this option enables the application on the specified target instance or

cluster. If you deploy to the target domain, this option is ignored, since deploying to the domain doesn't deploy to a specific instance or cluster.

--context-root

Valid only if the archive is a web module. It is ignored for other archive types; defaults to filename without extension. The context root of the application that is being redeployed. The context root identifies the application in the server. The default is the name, without the extension, of the archive file that contains the application.

--precompilejsp

Specifies whether pages that are created with the JavaServer Pages™ technology (“JSP™ pages”) are precompiled during deployment. Possible values are as follows:

false

JSP pages are *not* precompiled during deployment, but are compiled at runtime (default).

true

JSP pages are precompiled during deployment.

--virtual-servers

One or more virtual server IDs. Multiple IDs are separated by commas.

--libraries

A comma-separated list of library JAR files. Specify the library JAR files by their relative or absolute paths. Specify relative paths relative to *instance-root/lib/applibs*. The libraries are made available to the application in the order specified.

--type

The packaging archive type of the component that is being deployed. Possible values are as follows:

osgi

The component is packaged as an OSGI Alliance bundle.

The --type option is optional. If the component is packaged as a regular archive, omit this option.

--properties --property

Optional keyword-value pairs that specify additional properties for the redeployment. The following properties are available:

keepSessions={false|true}

Specifies whether active sessions of the application that is being redeployed are preserved and then restored when the redeployment is complete. Possible values are as follows:

false

Active sessions of the application are *not* preserved and restored (default).

`true`

Active sessions of the application are preserved and restored.

If any active session of the application fails to be preserved or restored, *none* of the sessions will be available when the redeployment is complete. However, the redeployment continues and a warning is logged.

To preserve active sessions, the Enterprise Server serializes the sessions and saves them in memory. To restore the sessions, the class loader of the newly redeployed application deserializes any sessions that were previously saved.

Other available properties are determined by the implementation of the component that is being redeployed.

Note – The `--properties` option and the `--property` option are equivalent. You can use either option regardless of the number of properties that you specify.

Operands *archive-path*

The path to the archive that contains the application that is being redeployed. This path can be a relative path or an absolute path.

The archive can be in either of the following formats:

- An archive file, for example, `/export/JEE_apps/hello.war`
- A directory that contains the exploded format of the deployable archive

Whether this operand is required depends on how the application was originally deployed:

- If the application was originally deployed from a file, the *archive-path* operand is required. The operand must specify an archive file.
- If the application was originally deployed from a directory, the *archive-path* operand is optional.

If this operand is omitted, the path is retrieved from the `domain.xml` file. Otherwise, the operand can specify a directory or an archive file.

Examples EXAMPLE 1 Redeploying a Web Application From a File

This example redeploys the web application `hello` from the `hello.war` file in the current working directory. The application was originally deployed from a file. Active sessions of the application are to be preserved and then restored when the redeployment is complete.

```
asadmin> redeploy --user admin --passwordfile myfile --name hello
--properties keepSessions=true hello.war
Command redeploy executed successfully
```

EXAMPLE 2 Redeploying a Web Application From a Directory

This example redeploys the web application `helldir`. The application was originally deployed from a directory. The path is retrieved from the `domain.xml` file.

EXAMPLE 2 Redeploying a Web Application From a Directory *(Continued)*

```
asadmin> redeploy --user admin --passwordfile myfile --name hellodir
Command redeploy executed successfully
```

Exit Status 0 command executed successfully
1 error in executing the command

See Also [deploy\(1\)](#), [undeploy\(1\)](#), [list-components\(1\)](#)

