Name

asadmin – utility for performing administrative tasks for Oracle GlassFish Server

Synopsis

```
asadmin [--host host]
[--port port]
[--user admin-user]
[--passwordfile filename]
[--terse={true|false}]
[--secure={false|true}]
[--echo={true|false}]
[--interactive={true|false}]
[--help]
[subcommand [options] [operands]]
```

Description

Use the `asadmin` utility to perform administrative tasks for Oracle GlassFish Server. You can use this utility instead of the Administration Console interface.

Subcommands of the `asadmin` Utility

The *subcommand* identifies the operation or task that you are performing. Subcommands are case-sensitive. Each subcommand is either a local subcommand or a remote subcommand.

- A *local subcommand* can be run without a running domain administration server (DAS). However, to run the subcommand and have access to the installation directory and the domain directory, the user must be logged in to the machine that hosts the domain.
- A *remote subcommand* is always run by connecting to a DAS and running the subcommand there. A running DAS is required.

`asadmin` Utility Options and Subcommand Options

Options control the behavior of the `asadmin` utility and its subcommands. Options are also case-sensitive.

The `asadmin` utility has the following types of options:

- `asadmin` **utility options.** These options control the behavior of the `asadmin` utility, not the subcommand. The `asadmin` utility options may precede or follow the subcommand, but `asadmin` utility options after the subcommand are deprecated. All `asadmin` utility options must either precede or follow the subcommand. If `asadmin` utility options are specified both before and after the subcommand, an error occurs. For a description of the `asadmin` utility options, see the “Options” section of this help information.
- **Subcommand options.** These options control the behavior of the subcommand, not the `asadmin` utility. Subcommand options must follow the subcommand. For a description of a subcommand’s options, see the help information for the subcommand.

A subcommand option may have the same name as an `asadmin` utility option, but the effects of the two options are different.

The `asadmin` utility options and some subcommand options have a long form and a short form.

- The long form of an option has two dashes (`--`) followed by an option word.
- The short form of an option has a single dash (`-`) followed by a single character.

For example, the long form and the short form of the option for specifying terse output are as follows:

- Long form: `--terse`
- Short form: `-t`

Most options require argument values, except Boolean options, which toggle to enable or disable a feature.

Operands of `asadmin` Subcommands

Operands specify the items on which the subcommand is to act. Operands must follow the argument values of subcommand options, and are set off by a space, a tab, or double dashes (`--`). The `asadmin` utility treats anything that follows the subcommand options and their values as an operand.

Escape Characters in Options for the `asadmin` Utility

Escape characters are required in options of the `asadmin` utility for the following types of characters:

- **Meta characters in the UNIX operating system.** These characters have special meaning in a shell. Meta characters in the UNIX operating system include: `\/,.!$%^&*|{}[]"'‘~;`.

  To disable these characters, use the backslash (`\`) escape character or enclose the entire command-line argument in single quote (`'`) characters.

  The following examples illustrate the effect of escape characters on the `*` character. In these examples, the current working directory is the domains directory.

  - The following command, without the escape character, echoes all files in the current directory:

    ```
    prompt% echo *
    domain1 domain2
    ```

  - The following command, in which the backslash (`\`) escape character precedes the `*` character, echoes the `*` character:

    ```
    prompt% echo \*
    *
    ```

  - The following command, in which the `*` character is enclosed in single quote (`'`) characters, echoes the `*` character:

    ```
    prompt% echo '*'
    *
    ```

- **Option delimiters.** The `asadmin` utility uses the colon character (`:`) as a delimiter for some options. The backslash (`\`) escape character is required if the colon is part of any of the following items:

  - A property

- An option of the Virtual Machine for the Java platform (Java Virtual Machine or JVM machine)[1]

  For example, the operand of the subcommand create-jvm-options(1) specifies JVM machine options in the following format:

  (*jvm-option-name*[=*jvm-option-value*])
  [:*jvm-option-name*[=*jvm-option-value*]]*

  Multiple JVM machine options in the operand of the `create-jvm-options` subcommand are separated by the colon (`:`) delimiter. If *jvm-option-name* or *jvm-option-value* contains a colon, the backslash (`\`) escape character is required before the colon.

Instead of using the backslash (\) escape character, you can use the double quote (") character or single quote (') character. The effects of the different types of quote characters on the backslash (\) character are as follows:

- Between double quote (") characters, the backslash (\) character is a special character.
- Between single quote (') characters, the backslash (\) character is *not* a special character.

When used without single quote (') characters, the escape character disables the delimiter in the command-line interface. The escape character is also a special character in the UNIX operating system and in the Java language. Therefore, in the UNIX operating system and in multimode, you must apply an additional escape character to every escape character in the command line. This requirement does *not* apply to the Windows operating system.

For example, the backslash (\) UNIX operating system meta character in the option argument `Test\Escape\Character` is specified on UNIX and Windows systems as follows:

- On UNIX systems, each backslash must be escaped with a second backslash:

  ```
  Test\\Escape\\Character
  ```

- On Windows systems, no escape character is required:

  ```
  Test\Escape\Character
  ```

Requirements for Using the `--secure` Option

The requirements for using the `--secure` option are as follows:

- The domain that you are administering must be configured for security.
- The `security-enabled` attribute of the “http-listener” in *Sun GlassFish Enterprise Server v3 Domain File Format Reference* element must be set to `true`.

  To set this attribute, use the set(1) subcommand. The `http-listener` element is stored in the `domain.xml` configuration file.

[1] The terms "Java Virtual Machine" and "JVM" mean a Virtual Machine for the Java platform.

Server Restart After Creation or Deletion

When you use the asadmin subcommands to create or delete a configuration item, you must restart the DAS for the change to take effect. To restart the DAS, use the restart-domain(1) subcommand.

Help Information for Subcommands and the asadmin Utility

To obtain help information for an asadmin utility subcommand, specify the subcommand of interest as the operand of the help subcommand. For example, to obtain help information for the start-domain(1) subcommand, type:

```
asadmin help start-domain
```

If you run the help subcommand without an operand, this help information for the asadmin utility is displayed.

To obtain a listing of available asadmin subcommands, use the list-commands(1) subcommand.

Options

--host
-H
The machine name where the DAS is running. The default value is localhost.

--port
-p
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, in the URLhttp://localhost:4949, the port is 4949.

The default port number for administration is 4848.

--user
-u
The user name of the authorized administrative user of the DAS.

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
-W
Specifies the name 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, an equals sign, and the password.

The entries in the file that are read by the asadmin utility are as follows:

- AS_ADMIN_PASSWORD=*administration-password*
- AS_ADMIN_MASTERPASSWORD=*master-password*

The entries in this file that are read by subcommands are as follows:

- AS_ADMIN_USERPASSWORD=*user-password* (read by the create-file-user(1) subcommand)

- `AS_ADMIN_ALIASPASSWORD=`*alias-password* (read by the `create-password-alias(1)` subcommand)
- `AS_ADMIN_MAPPEDPASSWORD=`*mapped-password* (read by the `create-connector-security-map(1)` subcommand)
- `AS_ADMIN_SSHPASSWORD=`*sshd-password* (read by the `install-node(1)` subcommand)
- `AS_ADMIN_PASSPHRASE`*sshd-passphrase* (read by the `install-node(1)` subcommand)

In domains that do not allow unauthenticated login, all remote subcommands must specify the administration password to authenticate to the DAS. The password can be specified by one of the following means:

- Through the `--passwordfile` option
- Through the `login(1)` subcommand
- Interactively at the command prompt

The `login` subcommand can be used to specify only the administration password. For other passwords that remote subcommands 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 subcommands, such as `update-file-user(1)`.

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

The master password is not propagated on the command line or an environment variable, but can be specified in the file that the `--passwordfile` option specifies.

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

`--terse`
`-t`

If true, output data is very concise and in a format that is optimized for use in scripts instead of for reading by humans. Typically, descriptive text and detailed status messages are also omitted from the output data. Default is false.

`--secure`
`-s`

If set to true, uses SSL/TLS to communicate with the DAS.

The default is false.

`--echo`
`-e`

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

`--interactive`
`-I`

If set to true, only the required options are prompted.

The default depends on how the `asadmin` utility is run:

- If the `asadmin` utility is run from a console window, the default is `true`.
- If the `asadmin` utility is run without a console window, for example, from within a script, the default is `false`.

`--help`
`-?`

Displays the help text for the `asadmin` utility.

Examples

EXAMPLE 1 Running an `asadmin` Utility Subcommand in Single Mode

This example runs the `list-applications(1)` subcommand in single mode. In this example, the default values for all options are used.

The example shows that the application `hello` is deployed on the local host.

```
asadmin list-applications
hello <web>

Command list-applications executed successfully.
```

EXAMPLE 2 Specifying an `asadmin` Utility Option With a Subcommand

This example specifies the `--host` `asadmin` utility option with the `list-applications` subcommand in single mode. In this example, the DAS is running on the host `srvr1.example.com`.

The example shows that the applications `basic-ezcomp`, `scrumtoys`, `ejb31-war`, and `automatic-timer-ejb` are deployed on the host `srvr1.example.com`.

```
asadmin --host srvr1.example.com list-applications
basic-ezcomp <web>
scrumtoys <web>
ejb31-war <ejb, web>
automatic-timer-ejb <ejb>

Command list-applications executed successfully.
```

EXAMPLE 3 Specifying an `asadmin` Utility Option and a Subcommand Option

This example specifies the `--host` `asadmin` utility option and the `--type` subcommand option with the `list-applications` subcommand in single mode. In this example, the DAS is running on the host `srvr1.example.com` and applications of type `web` are to be listed.

EXAMPLE 3 Specifying an asadmin Utility Option and a Subcommand Option *(Continued)*

```
asadmin --host srvr1.example.com list-applications --type web
basic-ezcomp <web>
scrumtoys <web>
ejb31-war <ejb, web>

Command list-applications executed successfully.
```

EXAMPLE 4 Escaping a Command-Line Argument With Single Quote Characters

The commands in this example specify the backslash (\) UNIX operating system meta character and the colon (:) option delimiter in the property value `c:\tools\jruby`.

For the UNIX operating system in single mode and multimode, and for all operating systems in multimode, the backslash (\) is required to escape the backslash (\) meta character and the colon (:) option delimiter:

```
asadmin deploy --property jruby.home='c\:\\tools\\jruby' bookstore
Application deployed successfully with name hello.


Command deploy executed successfully.
```

For the Windows operating system in single mode, the single quote (') characters eliminate the need for other escape characters:

```
asadmin deploy --property jruby.home='c:\tools\jruby' bookstore
Application deployed successfully with name hello.


Command deploy executed successfully.
```

EXAMPLE 5 Specifying a UNIX Operating System Meta Character in an Option

The commands in this example specify the backslash (\) UNIX operating system meta character in the option argument `Test\Escape\Character`.

For the UNIX operating system in single mode and multimode, and for all operating systems in multimode, the backslash (\) is required to escape the backslash (\) meta character:

```
asadmin --user admin --passwordfile gfpass create-jdbc-connection-pool
--datasourceclassname sampleClassName
--description Test\\Escape\\Character
sampleJDBCConnectionPool
```

For the Windows operating system in single mode, no escape character is required:

```
asadmin --user admin --passwordfile gfpass create-jdbc-connection-pool
--datasourceclassname sampleClassName
--description Test\Escape\Character
```

EXAMPLE 5 Specifying a UNIX Operating System Meta Character in an Option *(Continued)*

```
sampleJDBCConnectionPool
```

EXAMPLE 6 Specifying a Meta Character and an Option Delimiter Character in a Property

The commands in this example specify the backslash (\) UNIX operating system meta character and the colon (:) option delimiter character in the `--property` option of the `create-jdbc-connection-pool(1)` subcommand.

The name and value pairs for the `--property` option are as follows:

```
user=dbuser
passwordfile=dbpasswordfile
DatabaseName=jdbc:derby
server=http://localhost:9092
```

For the UNIX operating system in single mode and multimode, and for all operating systems in multimode, a backslash (\) is required to escape the colon (`:`) and the backslash (\):

```
asadmin --user admin --passwordfile gfpass create-jdbc-connection-pool
--datasourceclassname com.derby.jdbc.jdbcDataSource
--property user=dbuser:passwordfile=dbpasswordfile:
DatabaseName=jdbc\\:derby:server=http\\://localhost\\:9092 javadb-pool
```

Alternatively, the entire argument to the `--property` option can be enclosed single quote (') characters:

```
asadmin --user admin --passwordfile gfpass create-jdbc-connection-pool
--datasourceclassname com.derby.jdbc.jdbcDataSource
--property 'user=dbnuser:passwordfile=dbpasswordfile:
DatabaseName="jdbc:derby":server="http://locahost:9092"'
```

For the Windows operating system in single mode, a backslash (\) is required to escape only the colon (`:`), but *not* the backslash (\):

```
asadmin --user admin --passwordfile gfpass create-jdbc-connection-pool
--datasourceclassname com.derby.jdbc.jdbcDataSource
--property user-dbuser:passwordfile-dbpasswordfile:
DatabaseName=jdbc\:derby:server=http\://localhost\:9092 javadb-pool
```

For all operating systems, the need to escape the colon (`:`) in a value can be avoided by enclosing the value in double quote characters or single quote characters:

```
asadmin --user admin --passwordfile gfpass create-jdbc-connection-pool
--datasourceclassname com.derby.jdbc.jdbcDataSource
--property user=dbuser:passwordfile=dbpasswordfile:
DatabaseName=\"jdbc:derby\":server=\"http://localhost:9092\" javadb-pool
```

EXAMPLE 7 Specifying an Option Delimiter and an Escape Character in a JVM Machine Option

The commands in this example specify the following characters in the `-Dlocation=c:\sun\appserver` JVM machine option:

- The colon (`:`) option delimiter
- The backslash (\) escape character

For the UNIX operating system in single mode and multimode, and for all operating systems in multimode, these characters must be specified as follows:

- To pass a literal backslash to a subcommand, two backslashes are required. Therefore, the colon (`:`) must be escaped by two backslashes (\\).
- To prevent the subcommand from treating the backslash as a special character, the backslash must be escaped. As a result, two literal backslashes (\\) must be passed to the subcommand. To prevent the shell from interpreting these backslashes as special characters, each backslash must be escaped. Therefore, the backslash must be specified by a total of four backslashes (\\\\).

The resulting command is as follows:

```
asadmin create-jvm-options --target test-server
-e -Dlocation=c\\:\\\\sun\\\\appserver
```

For the Windows operating system in single mode, a backslash (\) is required to escape the colon (`:`) and the backslash (\):

```
asadmin create-jvm-options --target test-server
-e -Dlocation=c\:\\sun\\appserver
```

EXAMPLE 8 Specifying an Option That Contains an Escape Character

The commands in this example specify the backslash (\) character and the double quote (") characters in the `"Hello\App"\authentication` option argument.

For the UNIX operating system in single mode and multimode, and for all operating systems in multimode, a backslash (\) is required to escape the double quote character (") and the backslash (\):

```
asadmin set-web-env-entry --name="Hello User" --type=java.lang.String
--value=techscribe --description=\"Hello\\App\"\\authentication hello
```

For the Windows operating system in single mode, a backslash (\) is required to escape only the double quote ("), but *not* the backslash (\):

```
asadmin set-web-env-entry --name="Hello User" --type=java.lang.String
--value=techscribe --description=\"Hello\App\"\authentication hello
```

Environment Variables

Environment variables modify the default values of asadmin utility options as shown in the following table.

Environment Variable	asadmin Utility Option
AS_ADMIN_TERSE	--terse
AS_ADMIN_ECHO	--echo
AS_ADMIN_INTERACTIVE	--interactive
AS_ADMIN_HOST	--host
AS_ADMIN_PORT	--port
AS_ADMIN_SECURE	--secure
AS_ADMIN_USER	--user
AS_ADMIN_PASSWORDFILE	--passwordfile
AS_ADMIN_HELP	--help

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Unstable

See Also

create-connector-security-map(1), create-file-user(1), create-jdbc-connection-pool(1), create-jvm-options(1), create-password-alias(1), deploy(1), list-applications(1), list-commands(1), login(1), restart-domain(1), set(1), set-web-env-entry(1), start-domain(1), update-file-user(1)

attributes(5)

“http-listener” in *Sun GlassFish Enterprise Server v3 Domain File Format Reference*