

Name create-jvm-options – creates options for the Java application launcher

Synopsis create-jvm-options [--help] [--target *target*] [--profiler={true|false}]
(*jvm_option_name=jvm_option_value*) [:*jvm_option_name=jvm_option_value**]

Description The create-jvm-options subcommand creates command-line options that are passed to the Java application launcher when Enterprise Server is started. The options that this subcommand creates are in addition to the options that are preset with Enterprise Server. Java application launcher options are stored in the Java configuration `java-config` element or the profiler `profiler` element of the `domain.xml` file. The options are sent to the command line in the order they appear in the `java-config` element or the profiler `profiler` element in the `domain.xml` file.

Profiler options are used to record the settings that are required to start a particular profiler. The profiler must already exist. If necessary, use the [create-profiler\(1\)](#) subcommand to create the profiler.

This subcommand can be used to create the following types of options:

- **Java system properties.** These options are set through the `-D` option of the Java application launcher. For example:
 - Djava.security.manager
 - Denvironment=Production
- **Startup parameters for the Java application launcher.** These options are preceded by the dash character (`-`). For example:
 - XX:PermSize=*size*
 - Xmx1024m
 - d64

If the subcommand specifies an option that already exists, the command does not re-create the option.

Note – Ensure that any option that you create is valid. The subcommand might allow you to create an invalid option, but such an invalid option can cause startup to fail.

An option can be verified by examining the server log after Enterprise Server starts. Options for the Java application launcher are written to the `server.log` file before any other information when Enterprise Server starts.

The addition of some options requires a server restart for changes to become effective. Other options are set immediately in the environment of the domain administration server (DAS) and do not require a restart. Whether a restart is required depends on the type of option.

- Restart is not required for Java system properties whose names do *not* start with `-Djava.` or `-Djavax.` (including the trailing period). For example, restart is *not* required for the following Java system property:

-Denvironment=Production

- Restart is required for the following options:
 - Java system properties whose names start with -Djava. or -Djavax. (including the trailing period). For example:
 - Djava.security.manager
 - Startup parameters for the Java application launcher. For example:
 - client
 - Xmx1024m
 - d64

To restart the DAS, use the `restart-domain(1)` command.

This subcommand is supported in remote mode only.

Options

- -help

- ?

Displays the help text for the subcommand.

- -target

Do not specify this option. This option is retained for compatibility with other releases. If you specify this option, a syntax error does not occur. Instead, the subcommand runs successfully and the option is silently ignored.

- -profiler

Indicates whether the Java application launcher options are for the profiler. The profiler must exist for this option to be true. Default is false.

Operands *jvm_option_name*

One or more options delimited by a colon (:). The format of an option depends on the following:

- If the option has a name and a value, the format is *option-name=value*.
- If the option has only a name, the format is *option_name*. For example, `-Xmx2048m`.

Note – If an option name or option value contains a colon, the backslash (\) must be used to escape the colon in the name or value. Other characters might also require an escape character. For more information about escape characters in subcommand options, see the [asadmin\(1M\)](#) man page.

Examples **EXAMPLE 1** Setting Java System Properties

This example sets multiple Java system properties.

```
asadmin> create-jvm-options -Dunixlocation=/root/example:
-Dvariable=$HOME:-Dwindowslocation=d:\sun\appserver:-Doption1=-value1
created 4 option(s)
```

EXAMPLE 1 Setting Java System Properties (Continued)

Command `create-jvm-options` executed successfully.

EXAMPLE 2 Setting a Startup Parameter for the Java Application Launcher

This example sets the maximum available heap size to 1024.

```
asadmin> create-jvm-options -Xmx1024m
created 1 option(s)
Command create-jvm-options executed successfully.
```

EXAMPLE 3 Setting Multiple Startup Parameters for the Java Application Launcher

This example sets the maximum available heap size to 1024 and requests details about garbage collection.

```
asadmin> create-jvm-options "-Xmx1024m:-XX:+PrintGCDetails"
created 1 option(s)
Command create-jvm-options executed successfully.
```

In this case, one of the two parameters already exists, so the subcommand reports that only one option was set.

EXAMPLE 4 Setting a JVM Startup Parameter for the Profiler

This example sets a JVM startup parameter for the profiler.

```
asadmin> create-jvm-options --profiler=true -XX:MaxPermSize=192m
created 1 option(s)
Command create-jvm-options executed successfully.
```

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

See Also `delete-jvm-options(1)`, `list-jvm-options(1)`, `create-profiler(1)`, `restart-domain(1)`
`asadmin(1M)`

For more information about the Java application launcher, see the reference page for the operating system that you are using:

- Solaris™ Operating System (Solaris OS) and Linux: *java - the Java application launcher* (<http://java.sun.com/javase/6/docs/technotes/tools/solaris/java.html>)
- Windows: *java - the Java application launcher* (<http://java.sun.com/javase/6/docs/technotes/tools/windows/java.html>)