

NAME

jmap – Prints shared object memory maps or heap memory details for a process, core file, or remote debug server. This command is experimental and unsupported.

SYNOPSIS

jmap [*options*] *pid*

jmap [*options*] *executable core*

jmap [*options*] [*pid*] *server-id@* [*remote-hostname-or-IP*

options The command-line options. See Options.

pid The process ID for which the memory map is to be printed. The process must be a Java process. To get a list of Java processes running on a machine, use the `jps(1)` command.

executable
The Java executable from which the core dump was produced.

core The core file for which the memory map is to be printed.

remote-hostname-or-IP
The remote debug server **hostname** or **IP** address. See `jsadefugd(1)`.

server-id
An optional unique ID to use when multiple debug servers are running on the same remote host.

DESCRIPTION

The **jmap** command prints shared object memory maps or heap memory details of a specified process, core file, or remote debug server. If the specified process is running on a 64-bit Java Virtual Machine (JVM), then you might need to specify the **-J-d64** option, for example: **jmap-J-d64 -heap pid**.

Note: This utility is unsupported and might not be available in future releases of the JDK. On Windows Systems where the **dbgeng.dll** file is not present, Debugging Tools For Windows must be installed to make these tools work. The **PATH** environment variable should contain the location of the **jvm.dll** file that is used by the target process or the location from which the crash dump file was produced, for example: **set PATH=%JDK_HOME%\jre\bin\client;%PATH%**.

OPTIONS

<no option>

When no option is used, the **jmap** command prints shared object mappings. For each shared object loaded in the target JVM, the start address, size of the mapping, and the full path of the shared object file are printed. This behavior is similar to the Oracle Solaris **pmap** utility.

-dump:[live,] *format=b*, *file=filename*

Dumps the Java heap in **hprof** binary format to **filename**. The **live** suboption is optional, but when specified, only the active objects in the heap are dumped. To browse the heap dump, you can use the `jhat(1)` command to read the generated file.

-finalizerinfo

Prints information about objects that are awaiting finalization.

-heap

Prints a heap summary of the garbage collection used, the heap configuration, and generation-wise heap usage. In addition, the number and size of interned Strings are printed.

-histo[:live]

Prints a histogram of the heap. For each Java class, the number of objects, memory size in bytes, and the fully qualified class names are printed. The JVM internal class names are printed with an asterisk (*) prefix. If the **live** suboption is specified, then only active objects are counted.

-clstats

Prints class loader wise statistics of Java heap. For each class loader, its name, how active it is, address, parent class loader, and the number and size of classes it has loaded are printed.

-F

Force. Use this option with the **jmap -dump** or **jmap -histo** option when the pid does not respond. The **live** suboption is not supported in this mode.

-h

Prints a help message.

-help

Prints a help message.

-Jflag

Passes **flag** to the Java Virtual Machine where the **jmap** command is running.

SEE ALSO

- [jhat\(1\)](#)
- [jps\(1\)](#)
- [jsadebugd\(1\)](#)

