

Java SE Documentation

jstat - Java Virtual Machine Statistics Monitoring Tool

- [Synopsis](#)
- [Parameters](#)
- [Description](#)
- [Virtual Machine Identifier](#)
- [Options](#)
 - [General Options](#)
 - [Output Options](#)
- [Examples](#)
- [See Also](#)

SYNOPSIS

```
jstat [ generalOption | outputOptions vmid [interval[s|ms] [count]] ]
```

PARAMETERS

generalOption

A single general command-line option (-help or -options)

outputOptions

One or more output options, consisting of a single *statOption*, plus any of the -t, -h, and -J options.

vmid

Virtual machine identifier, a string indicating the target Java virtual machine (JVM). The general syntax is

```
[protocol:] [//] lvmid[@hostname[:port]/servername]
```

The syntax of the vmid string largely corresponds to the syntax of a URI. The *vmid* can vary from a simple integer representing a local JVM to a more complex construction specifying a communications protocol, port number, and other implementation-specific values. See [Virtual Machine Identifier](#) for details.

interval[s|ms]

Sampling interval in the specified units, seconds (s) or milliseconds (ms). Default units are milliseconds. Must be a positive integer. If specified, **jstat** will produce its output at each interval.

count

Number of samples to display. Default value is infinity; that is, **jstat** displays statistics until the target JVM terminates or the **jstat** command is terminated. Must be a positive integer.

DESCRIPTION

The **jstat** tool displays performance statistics for an instrumented HotSpot Java virtual machine (JVM). The target JVM is identified by its virtual machine identifier, or [vmid](#) option described below.

NOTE: This utility is unsupported and may not be available in future versions of the JDK. It is not currently available on Windows 98 and Windows ME. platforms.

VIRTUAL MACHINE IDENTIFIER

The syntax of the *vmid* string largely corresponds to the syntax of a URI:

```
[protocol:] [//] lvmid[@hostname[:port]] [/servername]
```

protocol

The communications protocol. If the *protocol* is omitted and a *hostname* is not specified, the default protocol is a platform specific optimized local protocol. If the *protocol* is omitted and a *hostname* is specified, then the default protocol is **rmi**.

lvmid

The local virtual machine identifier for the target JVM. The *lvmid* is a platform-specific value that uniquely identifies a JVM on a system. The *lvmid* is the only required component of a virtual machine identifier. The *lvmid* is typically, but not necessarily, the operating system's

process identifier for the target JVM process. You can use the **jps** command to determine the *lvmid*. Also, you can determine *lvmid* on Unix platforms with the **ps** command, and on Windows with the Windows Task Manager.

hostname

A hostname or IP address indicating the target host. If *hostname* is omitted, then the target host is the local host.

port

The default port for communicating with the remote server. If the *hostname* is omitted or the *protocol* specifies an optimized, local protocol, then *port* is ignored. Otherwise, treatment of the *port* parameter is implementation specific. For the default **rmi** protocol, the *port* indicates the port number for the rmiregistry on the remote host. If *port* is omitted, and *protocol* indicates **rmi**, then the default rmiregistry port (1099) is used.

servername

The treatment of this parameter depends on implementation. For the optimized local protocol, this field is ignored. For the **rmi** protocol, it represents the name of the RMI remote object on the remote host.

OPTIONS

The **jstat** command supports two types of options, [general options](#) and [output options](#). General options cause **jstat** to display simple usage and version information. Output options determine the content and format of the statistical output.

NOTE: All options, and their functionality are subject to change or removal in future releases.

GENERAL OPTIONS

If you specify one of the general options, you cannot specify any other option or parameter.

-help

Display help message.

-options

Display list of statistics options. See the [Output Options](#) section below.

OUTPUT OPTIONS

If you do not specify a general option, then you can specify output options. Output options determine the content and format of **jstat**'s output, and consist of a single *statOption*, plus any of the other output options (-h, -t, and -J). The *statOption* must come first.

Output is formatted as a table, with columns are separated by spaces. A header row with titles describes the columns. Use the **-h** option to set the frequency at which the header is displayed. Column header names are generally consistent between the different options. In general, if two options provide a column with the same name, then the data source for the two columns are the same.

Use the **-t** option to display a time stamp column, labeled *Timestamp* as the first column of output. The *Timestamp* column contains the elapsed time, in seconds, since startup of the target JVM. The resolution of the time stamp is dependent on various factors and is subject to variation due to delayed thread scheduling on heavily loaded systems.

Use the *interval* and *count* parameters to determine how frequently and how many times, respectively, **jstat** displays its output.

NOTE: You are advised not to write scripts to parse **jstat**'s output since the format may change in future releases. If you choose to write scripts that parse **jstat** output, expect to modify them for future releases of this tool.

-statOption

Determines the statistics information that **jstat** displays. The following table lists the available options. Use the **-options** general option to display the list of options for a particular platform installation.

Option	Displays...
class	Statistics on the behavior of the class loader.
compiler	Statistics of the behavior of the HotSpot Just-in-Time compiler.
gc	Statistics of the behavior of the garbage collected heap.
gccapacity	Statistics of the capacities of the generations and their corresponding spaces.
gccause	Summary of garbage collection statistics (same as -gcutil), with the cause of the last and current (if applicable) garbage collection events.
gcnnew	Statistics of the behavior of the new generation.
gcnnewcapacity	Statistics of the sizes of the new generations and its corresponding spaces.
gcold	Statistics of the behavior of the old and permanent generations.
gcoldcapacity	Statistics of the sizes of the old generation.
gcpermcapacity	Statistics of the sizes of the permanent generation.
gcutil	Summary of garbage collection statistics.
printcompilation	HotSpot compilation method statistics.

-h n

Display a column header every *n* samples (output rows), where *n* is a positive integer. Default value is 0, which displays the column header above the first row of data.

-t

Display a timestamp column as the first column of output. The timestamp is the time since the start time of the target JVM.

-JjavaOption

Pass *javaOption* to the **java** application launcher. For example, **-J-Xms48m** sets the startup memory to 48 megabytes. For a complete list of options, see [java - the Java application launcher](#)

STATOPTIONS AND OUTPUT

The following tables summarize the columns that **jstat** outputs for each *statOption*.

-class Option

Class Loader Statistics

Column	Description
Loaded	Number of classes loaded.
Bytes	Number of Kbytes loaded.
Unloaded	Number of classes unloaded.
Bytes	Number of Kbytes unloaded.
Time	Time spent performing class load and unload operations.

-compiler Option

HotSpot Just-In-Time Compiler Statistics

Column	Description
Compiled	Number of compilation tasks performed.
Failed	Number of compilation tasks that failed.
Invalid	Number of compilation tasks that were invalidated.
Time	Time spent performing compilation tasks.
FailedType	Compile type of the last failed compilation.
FailedMethod	Class name and method for the last failed compilation.

-gc Option

Garbage-collected heap statistics

Column	Description
S0C	Current survivor space 0 capacity (KB).
S1C	Current survivor space 1 capacity (KB).
S0U	Survivor space 0 utilization (KB).
S1U	Survivor space 1 utilization (KB).
EC	Current eden space capacity (KB).
EU	Eden space utilization (KB).
OC	Current old space capacity (KB).
OU	Old space utilization (KB).
PC	Current permanent space capacity (KB).
PU	Permanent space utilization (KB).
YGC	Number of young generation GC Events.
YGCT	Young generation garbage collection time.
FGC	Number of full GC events.
FGCT	Full garbage collection time.
GCT	Total garbage collection time.

-gccapacity Option

Memory Pool Generation and Space Capacities

Column	Description
NGCMN	Minimum new generation capacity (KB).
NGCMX	Maximum new generation capacity (KB).
NGC	Current new generation capacity (KB).
S0C	Current survivor space 0 capacity (KB).
S1C	Current survivor space 1 capacity (KB).
EC	Current eden space capacity (KB).
OGCMN	Minimum old generation capacity (KB).
OGCMX	Maximum old generation capacity (KB).

OGC	Current old generation capacity (KB).
OC	Current old space capacity (KB).
PGCMN	Minimum permanent generation capacity (KB).
PGCMX	Maximum Permanent generation capacity (KB).
PGC	Current Permanent generation capacity (KB).
PC	Current Permanent space capacity (KB).
YGC	Number of Young generation GC Events.
FGC	Number of Full GC Events.

-gccause Option

This option displays the same summary of garbage collection statistics as the **-gcutil** option, but includes the causes of the last garbage collection event and (if applicable) the current garbage collection event. In addition to the columns listed for **-gcutil**, this option adds the following columns:

Garbage Collection Statistics, Including GC Events	
Column	Description
LGCC	Cause of last Garbage Collection.
GCC	Cause of current Garbage Collection.

-gcnew Option

New Generation Statistics	
Column	Description
S0C	Current survivor space 0 capacity (KB).
S1C	Current survivor space 1 capacity (KB).
S0U	Survivor space 0 utilization (KB).
S1U	Survivor space 1 utilization (KB).
TT	Tenuring threshold.
MTT	Maximum tenuring threshold.
DSS	Desired survivor size (KB).
EC	Current eden space capacity (KB).
EU	Eden space utilization (KB).
YGC	Number of young generation GC events.
YGCT	Young generation garbage collection time.

-gcnewcapacity Option

New Generation Space Size Statistics	
Column	Description
NGCMN	Minimum new generation capacity (KB).
NGCMX	Maximum new generation capacity (KB).
NGC	Current new generation capacity (KB).
S0CMX	Maximum survivor space 0 capacity (KB).
S0C	Current survivor space 0 capacity (KB).
S1CMX	Maximum survivor space 1 capacity (KB).
S1C	Current survivor space 1 capacity (KB).
ECMX	Maximum eden space capacity (KB).
EC	Current eden space capacity (KB).
YGC	Number of young generation GC events.
FGC	Number of Full GC Events.

-gcold Option

Old and Permanent Generation Statistics	
Column	Description
PC	Current permanent space capacity (KB).
PU	Permanent space utilization (KB).
OC	Current old space capacity (KB).
OU	old space utilization (KB).
YGC	Number of young generation GC events.
FGC	Number of full GC events.

FGCT	Full garbage collection time.
GCT	Total garbage collection time.

-gcoldcapacity Option

Old Generation Statistics

Column	Description
OGCMN	Minimum old generation capacity (KB).
OGCMX	Maximum old generation capacity (KB).
OGC	Current old generation capacity (KB).
OC	Current old space capacity (KB).
YGC	Number of young generation GC events.
FGC	Number of full GC events.
FGCT	Full garbage collection time.
GCT	Total garbage collection time.

-gcpermcapacity Option

Permanent Generation Statistics

Column	Description
PGCMN	Minimum permanent generation capacity (KB).
PGCMX	Maximum permanent generation capacity (KB).
PGC	Current permanent generation capacity (KB).
PC	Current permanent space capacity (KB).
YGC	Number of young generation GC events.
FGC	Number of full GC events.
FGCT	Full garbage collection time.
GCT	Total garbage collection time.

-gcutil Option

Summary of Garbage Collection Statistics

Column	Description
S0	Survivor space 0 utilization as a percentage of the space's current capacity.
S1	Survivor space 1 utilization as a percentage of the space's current capacity.
E	Eden space utilization as a percentage of the space's current capacity.
O	Old space utilization as a percentage of the space's current capacity.
P	Permanent space utilization as a percentage of the space's current capacity.
YGC	Number of young generation GC events.
YGCT	Young generation garbage collection time.
FGC	Number of full GC events.
FGCT	Full garbage collection time.
GCT	Total garbage collection time.

-printcompilation Option

HotSpot Compiler Method Statistics

Column	Description
Compiled	Number of compilation tasks performed by the most recently compiled method.
Size	Number of bytes of bytecode of the most recently compiled method.
Type	Compilation type of the most recently compiled method.
Method	Class name and method name identifying the most recently compiled method. Class name uses "/" instead of "." as namespace separator. Method name is the method within the given class. The format for these two fields is consistent with the HotSpot - XX:+PrintCompilation option.

EXAMPLES

This section presents some examples of monitoring a local JVM with a *lvmid* of 21891.

Using the gcutil option

This example attaches to *lvmid* 21891 and takes 7 samples at 250 millisecond intervals and displays the output as specified by the **-gcutil** option.

```
jstat -gcutil 21891 250 7
```

S0	S1	E	O	P	YGC	YGCT	FGC	FGCT	GCT
12.44	0.00	27.20	9.49	96.70	78	0.176	5	0.495	0.672
12.44	0.00	62.16	9.49	96.70	78	0.176	5	0.495	0.672
12.44	0.00	83.97	9.49	96.70	78	0.176	5	0.495	0.672
0.00	7.74	0.00	9.51	96.70	79	0.177	5	0.495	0.673
0.00	7.74	23.37	9.51	96.70	79	0.177	5	0.495	0.673
0.00	7.74	43.82	9.51	96.70	79	0.177	5	0.495	0.673
0.00	7.74	58.11	9.51	96.71	79	0.177	5	0.495	0.673

The output of this example shows that a young generation collection occurred between the 3rd and 4th sample. The collection took 0.001 seconds and promoted objects from the eden space (E) to the old space (O), resulting in an increase of old space utilization from 9.49% to 9.51%. Before the collection, the survivor space was 12.44% utilized, but after this collection it is only 7.74% utilized.

Repeating the column header string

This example attaches to *lvmid* 21891 and takes samples at 250 millisecond intervals and displays the output as specified by **-gcutil** option. In addition, it uses the **-h3** option to output the column header after every 3 lines of data.

```
jstat -gcnew -h3 21891 250
```

S0C	S1C	S0U	S1U	TT	MTT	DSS	EC	EU	YGC	YGCT
64.0	64.0	0.0	31.7	31	31	32.0	512.0	178.6	249	0.203
64.0	64.0	0.0	31.7	31	31	32.0	512.0	355.5	249	0.203
64.0	64.0	35.4	0.0	2	31	32.0	512.0	21.9	250	0.204
S0C	S1C	S0U	S1U	TT	MTT	DSS	EC	EU	YGC	YGCT
64.0	64.0	35.4	0.0	2	31	32.0	512.0	245.9	250	0.204
64.0	64.0	35.4	0.0	2	31	32.0	512.0	421.1	250	0.204
64.0	64.0	0.0	19.0	31	31	32.0	512.0	84.4	251	0.204
S0C	S1C	S0U	S1U	TT	MTT	DSS	EC	EU	YGC	YGCT
64.0	64.0	0.0	19.0	31	31	32.0	512.0	306.7	251	0.204

In addition to showing the repeating header string, this example shows that between the 2nd and 3rd samples, a young GC occurred. Its duration was 0.001 seconds. The collection found enough live data that the survivor space 0 utilization (S0U) would have exceeded the desired survivor Size (DSS). As a result, objects were promoted to the old generation (not visible in this output), and the tenuring threshold (TT) was lowered from 31 to 2.

Another collection occurs between the 5th and 6th samples. This collection found very few survivors and returned the tenuring threshold to 31.

Including a time stamp for each sample

This example attaches to *lvmid* 21891 and takes 3 samples at 250 millisecond intervals. The **-t** option is used to generate a time stamp for each sample in the first column.

```
jstat -gcoldcapacity -t 21891 250 3
```

Timestamp	OGCMN	OGCMX	OGC	OC	YGC	FGC	FGCT	GCT
150.1	1408.0	60544.0	11696.0	11696.0	194	80	2.874	3.799
150.4	1408.0	60544.0	13820.0	13820.0	194	81	2.938	3.863
150.7	1408.0	60544.0	13820.0	13820.0	194	81	2.938	3.863

The **Timestamp** column reports the elapsed time in seconds since the start of the target JVM. In addition, the **-gcoldcapacity** output shows the old generation capacity (OGC) and the old space capacity (OC) increasing as the heap expands to meet allocation and/or promotion demands.

The old generation capacity (OGC) has grown to from 11696 KB to 13820 KB after the 81st Full GC (FGC). The maximum capacity of the generation (and space) is 60544 KB (OGCMX), so it still has room to expand.

Monitor instrumentation for a remote JVM

This example attaches to *lvmid* 40496 on the system named *remote.domain* using the **-gcutil** option, with samples taken every second indefinitely.

```
jstat -gcutil 40496@remote.domain 1000
```

... output omitted

The *lvmid* is combined with the name of the remote host to construct a *vmid* of *40496@remote.domain*. This *vmid* results in the use of the **rmi** protocol to communicate to the default **jstatd** server on the remote host. The **jstatd** server is located using the **rmiregistry** on *remote.domain* that is bound to the default **rmiregistry** port (port 1099).

SEE ALSO

- [java](#) - the Java Application Launcher
- [jps](#) - the Java Process Status Application
- [jstatd](#) - the jvmstat daemon
- [rmiregistry](#) - the Java Remote Object Registry