

# The Grinder 3 Properties File

## Table of contents

1 Table of properties.....	2
2 Specifying properties on the command line.....	5

The Grinder worker and agent processes are controlled by setting properties in the `grinder.properties` file.

All properties have default values. If you start The Grinder agent process without a `grinder.properties` file it will communicate with the console using default addresses, use one worker process, one thread, and make one run through the test script found in the file `grinder.py`. This is not much use, so read on...

## 1. Table of properties

This table lists the properties understood by The Grinder engine.

Property	Description	Default
<code>grinder.processes</code>	The number of worker processes the agent should start.	1
<code>grinder.threads</code>	The number of worker threads that each worker process spawns.	1
<code>grinder.runs</code>	The number of runs of the test script each thread performs. 0 means "run forever", and should be used when you are using the console to control your test runs.	1
<code>grinder.processIncrement</code>	If set, the agent will <i>ramp up</i> the number of worker processes, starting the number specified every <code>grinder.processesIncrement</code> milliseconds. The upper limit is set by <code>grinder.processes</code> .	Start all worker processes together.
<code>grinder.processIncrementInterval</code>	Used in conjunction with <code>grinder.processIncrement</code> , this property sets the interval in milliseconds at which the agent starts new worker processes.	60000 ms
<code>grinder.initialProcesses</code>	Used in conjunction with <code>grinder.processIncrement</code> , this property sets the initial number of worker processes to start.	The value of <code>grinder.processIncrement</code> .
<code>grinder.duration</code>	The maximum length of time in milliseconds that each worker process should run for. <code>grinder.duration</code> can be specified in conjunction with <code>grinder.runs</code> , in which case the worker processes will terminate if either the duration time or the number of runs is exceeded.	Run forever.

<code>grinder.script</code>	The file name of the Jython <a href="#">script</a> ( <code>../g3/scripts.html</code> ) to run.	<code>grinder.py</code>
<code>grinder.jvm</code>	Use an alternate JVM for worker processes. Defaults to <code>java</code> so you do not need to specify this if your <code>PATH</code> is sensible.	<code>java</code>
<code>grinder.jvm.classpath</code>	Use to adjust the classpath used for the worker process JVMs. Anything specified here will be prepended to the classpath used to start the Grinder processes.	
<code>grinder.jvm.arguments</code>	Additional arguments to worker process JVMs.	
<code>grinder.logDirectory</code>	Directory to write log files to. Created if it doesn't already exist.	The local directory.
<code>grinder.numberOfOldLogs</code>	The number of archived logs from previous runs that should be kept.	1
<code>grinder.hostID</code>	Override the "host" string used in log filenames and logs.	The host name.
<code>grinder.consoleHost</code>	The IP address or host name that the agent and worker processes use to contact the console.	All the network interfaces of the local machine.
<code>grinder.consolePort</code>	The IP port that the agent and worker processes use to contact the console.	6372
<code>grinder.useConsole</code>	Set to <code>false</code> to set the agent and worker processes not to use the console.	<code>true</code>
<code>grinder.reportToConsole.</code>	The period at which each process sends updates to the console. This also controls the frequency at which the data files are flushed.	500 ms
<code>grinder.initialSleepTime</code>	The maximum time in milliseconds that each thread waits before starting. Unlike the sleep times specified in scripts, this is varied according to a flat random distribution. The actual sleep time will be a random value between 0 and the specified value. Affected by <code>grinder.sleepTimeFactor</code> , but not <code>grinder.sleepTimeVariation</code>	0 ms

<code>grinder.sleepTimeFactor</code>	Apply a factor to all the sleep times you've specified, either through a property of in a script. Setting this to <code>0.1</code> would run the script ten times as fast.	1
<code>grinder.sleepTimeVariation</code>	The Grinder varies the sleep times specified in scripts according to a Normal distribution. This property specifies a fractional range within which nearly all (99.75%) of the times will lie. E.g., if the sleep time is specified as 1000 and the <code>sleepTimeVariation</code> is set to <code>0.1</code> , then 99.75% of the actual sleep times will be between 900 and 1100 milliseconds.	0.2
<code>grinder.logProcessStreams</code>	Set to <code>false</code> to disable the logging of output and error streams for worker processes. You might want to use this to reduce the overhead of running a client thread.	true
<code>grinder.reportTimesToConsole</code>	Set to <code>false</code> to disable reporting of <a href="#">timing information</a> ( <code>../faq.html#timing</code> ) to the console; other statistics are still reported.	true
<code>grinder.debug.singleprocess</code>	If set to <code>true</code> , the agent process spawns engines in threads rather than processes, using special class loaders to isolate the engines. This allows the engine to be easily run in a debugger. This is primarily a tool for debugging The Grinder engine, but it might also be useful to advanced users. <a href="#">GrinderStone</a> ( <a href="http://code.google.com/p/grinder">http://code.google.com/p/grinder</a> ) uses this property to allow interactive debugging.	false
<code>grinder.debug.singleprocess.classnames</code>	For advanced use only. Specifies a comma separated list of names of classes that should be shared between the worker engines when <code>grinder.debug.singleprocess</code> is <code>true</code> . Class names can end with a <code>*</code> wildcard. See <a href="#">bug 2314157</a> ( <a href="http://www.sourceforge.net/track">http://www.sourceforge.net/track</a> ) for more details.	false
<code>grinder.dcrinstrumentation</code>	For advanced use only.	false

group\_id=185

	Enables the <a href="#">new instrumentation engine</a> ( <a href="#">../g3/instrumentation.html#new-i</a> ) .	
--	--	--

## 2. Specifying properties on the command line

You can also specify these properties as Java system properties in the agent command line. For example, on UNIX systems the following command line can be used to generate log directories based on the current date.

```
java -Dgrinder.logDirectory="log/$(date +%y%m%d)" net.grinder.Grinder
```

Property values set as Java system properties override values set in the `grinder.properties` file. Only properties with names that start "grinder." are considered.