

# What's new in The Grinder 3?

## Table of contents

1 Jython.....	2
2 New distribution packaging.....	2
3 Other changes.....	2

This section summarises the changes between The Grinder 2 and The Grinder 3. If you're not familiar with The Grinder 2, skim read this and move on to [Getting Started](#) ([../g3/getting-started.html](#)).

**Note:**

Please read [Should I use The Grinder 2 or The Grinder 3](#) ([../faq.html#g2vsg3](#)) ?

## 1. Jython

The most significant change that The Grinder 3 introduces is a [Jython](#) (<http://www.jython.org/>) scripting engine that is used to execute test scripts. Jython is a Java implementation of the popular Python language. Test scripts specify the tests to run. In The Grinder 2, tests were specified in the `grinder.properties` file. The `grinder.properties` file is still used to specify general control information (how to contact the console, how many worker processes to use, ..), as well as the name of the test script. Many other key features of The Grinder 2, such as the console and the process architecture, remain unchanged.

The scripting engine provides the following new capabilities:

### Test any Java code

The Grinder 3 allows any Java (or Jython) code to be encapsulated as a test. This practically removes the need to write custom plug-ins. Although plug-ins are no longer responsible for performing tests, they can still be useful to manage objects that the tests use. For example, the HTTP plug-in manages a pool of connections for each worker thread, and provides an `HttpRequest` object that makes use of these connections.

### Dynamic test scripting

The Grinder 2 worker processes execute tests in the properties file sequentially in a fixed order, and there is limited support in some of the The Grinder 2 plug-ins for checking test results. The Grinder 3 allows arbitrary branching and looping and makes test results directly available to the test script, allowing different test paths to be taken depending on the outcome of each test.

The Grinder 2 HTTP plug-in's [string bean](#) ([../g2/http-plugin.html#string-bean](#)) feature provides simple support for requests that contain dynamic data. The Grinder 3 can use the full power of Jython to create dynamic requests of arbitrary complexity.

Kind of dry, huh? If you never seen any Python or are wondering what was wrong with the old `grinder.properties` approach then skip ahead to the [Script Gallery](#) ([../g3/script-gallery.html](#)) where you can sample the power of The Grinder 3.

## 2. New distribution packaging

The Grinder is distributed in separate binary and source zip files. They can be downloaded from [SourceForge.net](#) (<http://www.sourceforge.net/projects/grinder>).

## 3. Other changes

- The TCPSniffer has been renamed *TCPProxy* for correctness. The TCPProxy can be used with other HTTP and HTTPS proxies. Many new features and fixes have been added to the TCPProxy.
- Console signals are now transmitted over a TCP socket connection. Multicast is no longer used, removing a frequent source of setup problems.
- The interface that plug-ins must implement has changed significantly. Plug-ins written for The Grinder 2 will not work with The Grinder 3.
- Many [grinder.properties](#) (../g3/properties.html) have been removed. The features formerly accessed through setting properties are now set by making calls to the plug-in from test scripts. Some of the remaining property names have been renamed.
- HTTP tests can now be directed through an HTTP proxy.
- String beans and OK strings have been removed from the HTTP plug-in. String beans and OK strings are very limited in comparison to the flexibility now provided by Jython scripts.
- The `HttpURLConnection` implementation has been removed from the HTTP plug-in.
- Many other minor updates to HTTP testing functionality.
- HTTPS and SSL contexts can now be controlled on a per thread basis.
- The JUnit and Socket plug-ins have been removed. Their functionality can be achieved directly by using the appropriate Java objects in scripts.
- From the console you can edit test scripts and distribute them to the worker processes. It is no longer necessary to copy grinder.properties files around or to use a shared disk.