

# JMeterAutomatedRemoteTesting

This document shows how you can use JMeter to perform more automated (non-interactive) testing.

## Requirements

This article assumes that you:

- know the basics of JMeter
- have an existing test plan
- have a bit of knowledge on using variables in test plans
- have touched on remote testing (from GUI is fine)

## Set up

- In your existing test plan, make sure that any variations in testing make use of functions or variables. For example, if running a HTTP sampler, use HTTP Request Defaults to specify a host as `${+P(targetHost,localhost)}`. Other useful places for parameterisation might include number of threads, ramp-up period or scheduler duration in a thread group, using a format of `${+P(threadgroup.threads,500)}` (The `+P` function is shorthand for `+property`. See the [http://jmeter.apache.org/usermanual/functions.html#\_\_P user manual] for more info on using this function).
- Save your test plan and properties file to a directory.
- Create a properties file containing all your variables. E.g. `mytest.properties` could contain `threadgroup.threads=100, targetHost=my-target-host.com`
- The test plan does not need Listeners, as this will be configured via parameters. This will improve performance on the testing.
- Run the test mode in stand-alone mode (i.e. no remote servers): `jmeter -n -t load_test.jmx -l load_test_report.jtl -q mytest.properties -j mytest.log`
- The `-q` parameter defines an additional property file, which is the equivalent of setting properties for the contents of the file using the `-J` parameter. These parameters are not sent to remote servers; for that one can use the `-G` parameter. The `-G` parameter can reference either a property in the format of `-Gprop=val` or a properties file, such as `-G./myglobal.properties`
- Properties can be referenced anywhere in the test plan using the `+P()` function - e.g. `${+P(propname)}`
- There is no way currently to predefine variables, however the User Defined Variables table on the Test Plan can be used to create variables from properties. For example:

Name :	Value:
HOST	<code>\${__P(host,localhost)}</code>

- On all client machines, start up JMeter server, ensuring that firewall is not blocking connections (clients must all be on the same subnet).
- On JMeter controller (the host initializing the test), run the test with the `-R` parameter (can be run using `-r` and specifying hosts in `jmeter properties file`):  
`jmeter -n -t load_test.jmx -l load_test_report.jtl -Gmytest.properties -j mytest.log -R remotehost1, remotehost2 ${GLOBAL_VARS}`

Note that load injection to targets will be a increased as a ratio of number of threads per thread group multiplied by number of client machines.

A bash script similar to the following could be used to start up jmeter (called `jmeter_test_wrapper.sh` for example):

```

#!/bin/sh

loadtest="myloadtest"
GLOBAL_VARS=""
REPORT_DIR=/tmp
JMETER_PROPERTIES=jmeter.properties
JMETER_CUSTOM_PROPERTIES=mytest.properties
# If set as an environment var, then use it, otherwise leave it unset.
HOST_LIST=${HOST_LIST:+"-R ${HOST_LIST}"}

# Had an issue with -Gglobal.properties file, so parsed this into individual properties.
for var in `grep -vE "^#|^$" ${JMETER_CUSTOM_PROPERTIES}`; do
    GLOBAL_VARS="-G${var} ${GLOBAL_VARS}"
done

# But after testing, it does work as expected.
GLOBAL_VARS="-G${JMETER_CUSTOM_PROPERTIES}"

if [ -f ${loadtest}.jmx ]; then
    echo "Running ${loadtest} with:"
    echo -e "
    $JMETER -n
        -t ${loadtest}.jmx
        -l ${REPORT_DIR}/${loadtest}.jtl
        -p ${JMETER_PROPERTIES}
        ${GLOBAL_VARS}
        ${HOST_LIST}
        -j ${REPORT_DIR}/_${loadtest}.log
    -----"
    $JMETER -n \
        -t ${loadtest}.jmx \
        -l ${REPORT_DIR}/${loadtest}.jtl \
        -p ${JMETER_PROPERTIES} \
        ${GLOBAL_VARS} \
        ${HOST_LIST} \
        -j ${REPORT_DIR}/_${loadtest}.log
else
    echo "Could not find test plan for ${loadtest}"
fi

```

For a local test, you could run: `./jmeter_test_wrapper.sh`

For a remote test, it can be modified as `HOST_LIST="192.168.1.10,192.168.1.12" ./jmeter_test_wrapper.sh`

Now all you need to do is make sure the servers are running and add the above command into scheduler.