

Apache JMeter 5.6.3 — Cheat Sheet

2026-10-14 | 5.6.3 runs on Java 8+, 17+ recommended | 6.0 unreleased (planned: Java 17+, 21 recommended)

Run properties: threads, rampup, duration, host, port | verify against jmeter.apache.org

Setup & plan | Load model | HTTP & data | Validation & scripting | Protocols & plugins | Run & results | Scale, CI & changes

1. Install, layout and heap

- Binary from jmeter.apache.org; check **SHA-512** (+ PGP). Runs on Java 8+, 17+ recommended.
- bin: launch scripts, `jmeter.properties` (do not edit), `user.properties` (your overrides), `system.properties`, `log4j2.xml`, `templates`.
- lib: JDBC drivers, JMS clients, utility jars (j. jar only), lib/ext: **plugins only** (and the Plugins Manager). lib/junit: JUnit test jars. Extra paths: `user.classpath`, `plugin_dependency_paths`, `search_paths`. Restart after changes.
- Env vars (or bin/setenv.sh): HEAP (default -Xms1g -Xmx1g -XX:MaxMetaspaceSize=256m), GC_ALGO (G1), JVM_ARGS.

```
export HEAP="-Xms2g -Xmx2g -XX:MaxMetaspaceSize=256m"
```
- Keep -Xms = -Xmx. Load order: `jmeter` -> `user` -> `system.properties`; -q/-J after user, -S/-D after system; **later wins**.

2. Elements, execution order, scoping

- Order per sampler:** 1 Config -> 2 Pre-Processors -> 3 Timers -> 4 **Sampler** -> 5 Post-Processors -> 6 Assertions -> 7 Listeners (5-7 skipped on a null result).
- Ordered:** controllers and samplers run in tree order. **Hierarchical:** the rest apply to every sampler in their scope (parent and below); under a sampler = that sampler only.
- Timers in scope add up** (1000 ms + 500 ms = 1500 ms before each sampler).
- Defaults (HTTP Request Defaults) merge field by field, closest wins. **Header Managers merge in 5.6.3** (closest wins on a duplicate name). Cookie / Authorization Manager: only one is used, no control which -> one per Thread Group.
- Families also include Test Fragments: reuse with Module Controller, or Include Controller (external .jmx).

3. Properties vs variables, functions

- Properties:** JVM-global; `user.properties`, -q file, -J (local), -G (sent to workers); read `$_P(threads,1)`, props in scripts. Run properties: `-Jthreads=50 -Jrampup=60 -Jduration=660 -Jhost=... -Jport=...`
- Variables:** per thread; Test Plan / UDV, CSV, extractors, vars. put; read `$(name)`. Case-sensitive.

Function	Use
__P __property __setProperty	read / set properties (data between groups)
__threadNum __counter __Random	thread number, counters, random data
__randomString __UUID	
__time __timeShift	CSVRead
__groovy __jexl3 __V __eval	expressions, nested refs
__groovy is cheap (cached); __BeanShell __JavaScript are not.	

4. Thread Group, setUP / tearDown

- Fields: Threads, Ramp-up (s), Loop Count (Infinite = -1), Same user on each iteration, Delay creation, **Thread lifetime:** Duration, Startup delay.
- On sampler error: Continue / Start Next Thread Loop / Stop Thread / Stop Test / Stop Test Now.
- Ramp-up:** gap = R / N; thread k starts at (k-1) R / N. 50 threads, 60 s -> one every 1.2 s, last at 58.8 s.
- Duration includes ramp-up:** 60 s ramp + 600 s hold -> Duration **660**. Threads stop only between samples: set connect/response timeouts. Empty/0 duration = "Invalid duration".
- setUP TG runs before, tearDown TG after. After **Shutdown** tearDown runs only with "Run Thread Down Thread Groups after shutdown of main threads"; never after Stop. Pass data with `__setProperty`.

5. Open model, throughput, Little's Law

- Open Model Thread Group** (5.5+, experimental): Schedule + Random Seed; no thread/ramp/duration fields.

```
rate(0/sec) random_arrivals(1 min) rate(10/sec)
random_arrivals(10 min) rate(10/sec)
random_arrivals(1 min) rate(0/sec)
pause(5 sec) /* comment */ // comment
```
- Units ms, sec, min, hour, day. Omitted end rate = start rate. even_arrivals not implemented. 5.6.3 times from **test** start (6.0: from TG start).
- Constant Throughput Timer:** target per minute (20/s = 1200); default mode: this thread only. Only delays.
- Precise Throughput Timer:** throughput per period, test duration; Poisson arrivals; under a 0 ms Flow Control Action.
- Little's Law:** $X = N / (R + Z)$. 20 req/s, R 0.2 s, Z 3 s -> $N = 20 \times 3.2 = 64$ threads. Bookshelf: 50 / 3.2 = 15.6 req/s.

6. Timers and think time

- | Timer | Delay |
|-----------------|--|
| Constant | Thread Delay |
| Uniform Random | offset + U x max (mean offset + max/2) |
| Gaussian Random | offset +/- Deviation (bell curve) |
| Poisson Random | offset + Poisson(lambda), tiny spread |
| JSR223 | script returns ms |
| Synchronizing | barrier: release N threads together |
- Run before **each** sampler in scope; place under one sampler to pause once. timer. factor scales the three random timers only.
 - Timers never run past the scheduled end (3.1+). Templates: "Think Time at a point", "Add Think Times to children".

7. Controllers, Transaction Controller

- Simple, Loop, **Once Only** (first iteration of each thread). If (expression mode: `While __jexl3(...)) / $(__groovy(...))`; JavaScript mode is slow), `While_ForEach` (name_1..N), Switch, Interleave, Random, Random Order, **Throughput** (a mix, not a rate), Runtime, Critical Section, Module, Include.
- Transaction Controller:** extra sample = sum of the children; OK only if all OK. Generate Parent Sample (parent mode). "Include duration of timer and pre-post processors": GUI writes false, a missing property reads **true** in 5.6.3 -> always set it.
- Example: 150+100+200+100 = 550 ms; with 4 timers of 2 s ~ 8,550 ms.

8. HTTP skeleton

- | | |
|--|--|
| Thread Group | threads, rampup, duration (\$__P) |
| HTTP Request Defaults | \$_P(host,localhost) \$_P(port,8080), timeouts |
| HTTP Header Manager | Accept, Content-Type |
| HTTP Cookie Manager | one per thread group |
| (Cache, Authorization, DNS Cache Manager, Keystore Config) | |
| GET \$(basePath)/books | method + path only |
| | assertions and extractors of this request |
- Defaults **Path is not a prefix:** use a basePath variable. Implementation HttpClient4 (Java: no PATCH, no mTLS). UTF-8 default since 5.6.1.
 - Core = HTTP/1.1 only; HTTP/2 and /3 via plugin bzm-http2 3.2.0. Same user on each iteration + `httpClient.reset_state_on_thread_group_iteration=true`.

9. REST, JSON, GraphQL, SOAP

- JSON in **Body Data** + Header Manager Content-Type: application/json: the sampler adds none. `http.post_add_content_type_if_missing = false` since 5.0 -> text/plain -> 415.
- PUT / PATCH / DELETE same pattern (PATCH needs HttpClient4). Check codes with Response Assertion, extract with JSON Extractor (\$ token).
- GraphQL HTTP Request** (5.4+): Query, Variables, Operation Name; POST (default) or GET.
- SOAP samplers removed (3.0 / 3.2): HTTP Request + envelope in Body Data + Content-Type/SOAPAction; XPath2 Extractor / Assertion.

10. Recorder and cURL import

- File > Templates > **Recording:** HTTP(S) Test Script Recorder on port **8888**; browser proxy -> JMeter.
- HTTPS: proxyserver.jks + root CA bin/ApacheJMeterTemporaryRootCA.crt; install as a trusted CA (Firefox: own store); valid 7 days (proxy.cert validity). **Remove it afterwards**. Error unknown_ca = CA not trusted.
- Include/exclude URL regexes, content-type filter, Transaction Controller grouping; then variabilise, correlate, add assertions and timers.
- Tools > Import from cURL:** paste curl commands (or a file, one per line); builds HTTP Request + Header Manager; never runs curl.

11. CSV Data Set Config

- Fields: Filename (relative to the .jmx), File Encoding, Variable Names (empty = header line), Ignore first line, Delimiter (\t = tab), Allow quoted data, Recycle on EOF, Stop thread on EOF, Sharing mode.
- | Sharing mode | Shares one read position |
|-----------------------|---------------------------|
| All threads (default) | every thread, every group |
| Current thread group | threads of one group |
| Current thread | one thread (own copy) |
| Identifier (any text) | all elements with that id |
- A new line per iteration; line order across threads not fixed. Distributed: the file must exist on **every** worker (split it for unique data).

12. Extractors and correlation

- JSON Extractor (\$ token), JSON JMESPath, Regular Expression (template \$1\$), **Boundary** (left/right text), CSS Selector, XPath2.
- Match No.:** 0 random, N the Nth, -1 **all** -> name_1..name_N + name_matchNr. After -1 read name_1, never name.
- Regex groups: name_g1. Set a visible default (TOKEN_NOT_FOUND); check with Debug Sampler / Debug PostProcessor. Result Status Action Handler acts on failures.

13. Assertions and their cost

- | Cost | Assertions |
|---------------|---|
| Very low | Response Assertion (code, substring), Size, Duration |
| Medium | JSON, JSON JMESPath, JSR223 Groovy (cached), MD5Hex |
| High | XPath2, XML Schema, Document (Tika), HTML, BeanShell, uncached JSR223 |
| Never at load | Compare Assertion |
- Each runs once per sample on the generator. Apply to main sample / sub-samples / a variable. Only assertions fill failurMessage. **Assert just enough:** one cheap check per business sampler.

14. JSR223 and Groovy

- Language **Groovy** (5.6.3 bundles 3.0.20; 6.0 plans 5.1.3). Tick **Cache compiled script if available**; script files are always compiled.
- Never \${var} in the script text** (the first value gets cached): use vars.get("var") or Parameters -> args.
- Bindings: vars, props, ctx, log, prev, sampler, SampleResult, Label, Parameters/args, OUT.
- jsr223.compiled_scripts_cache_size=100 (Caffeine since 5.6). BeanShell: migrate. BSF: deprecated since 3.1.

15. JDBC, JMS and other samplers

- JDBC:** driver jar in lib; JDBC Connection Configuration (pool variable, Max Connections = 0 = one per thread); Request types Select / Update / Callable / **Prepared Select** / **Prepared Update** / Commit / Rollback / AutoCommit.
- JMS P2P:** Request Only, Request Response, Read, Browse, Clear; Publisher / Subscriber; JNDI + provider jars in lib. 5.6.3 = javax.jms -> ActiveMQ Classic 5.19.x client.
- Mail:** SMTP, Mail Reader (POP3/IMAP). **TCP:** TCPClietImpl, Binary, LengthPrefixedBinary.
- FTP, LDAP, LDAP Extended, OS Process, Java Request, JUnit (3/4 only), Bolt (Neo4j), Access Log, Debug Sampler, Flow Control Action (no sample), MongoDB Script: deprecated, removed in 6.0.

16. Plugins Manager

- GUI: `jmeter-plugins-manager-2.0.jar` -> lib/ext, restart, **Options > Plugins Manager**. Community catalogue, not Apache.

```
cd $JMeter_HOME; cp ~/cmdrunner-2.3.jar lib/
java -cp lib/ext;jmeter-plugins-manager-2.0.jar \
org.jmeterplugins.repository.PluginManagerCMDInstaller
bin/PluginsManagerCMD.sh status | available | upgrades
bin/PluginsManagerCMD.sh install jpgc-casutg-3.1.1,jpgc-tst=2.6
bin/PluginsManagerCMD.sh install-for-jmx-plan-jmx
bin/PluginsManagerCMD.sh uninstall jpgc-perfmon
```
- Ids: jpgc-casutg 3.1.1, jpgc-tst 2.6, jpgc-autostop 0.3, bzm-http2 3.2.0, jmeter-grpc-request 2.0.2, jpgc-perfmon legacy, jpgc-json deprecated.

17. Custom thread groups, shaping

- Concurrency TG:** Target Concurrency, Ramp Up Time, Ramp-Up Steps Count, Hold Target Rate Time, Time Unit (GUI default **minutes**), Thread Iterations Limit. Run time = ramp + hold.
- Ultimate TG** rows: Start Threads Count, Initial Delay, Startup Time, Hold Load For, Shutdown Time (each row from group start).

```
threads_schedule=spawn(15,1s,1s,1s,1s) spawn(40,1s,3s,1s,2s)
```
- Throughput Shaping Timer** rows: Start RPS, End RPS, Duration sec; counts every sampler in scope.

- ```
load_profile=const(10,10s) line(10,100,1m) step(5,25,5,1h)
<timer name>.load_profile=... (per timer, 2.6+)
```
- Target Concurrency: `$_tstFeedback(tst-name,1,100,10)`
- ### 18. gRPC plugin (jmeter-grpc-request 2.0.2)
- bakthava fork (zalopay 1.2.6 archived). **Add > Sampler > gRPC Request**; parses .proto at run time.
  - Fields: Server Name or IP, Port Number, SSL/TLS, Disable Cert Verification, mTLS client cert/key/CA, Proto Root Directory, Library Directory, Full Method (pkg.Service/Method), Metadata (k:v, k:v or JSON), Deadline ms (1000), request JSON.
  - Limits:** unary only, no grpc-web, no reflection, no binary metadata; codes 200 / 500 / 400; a failed call closes the channel; grpc-java 1.38.0.

## 19. CLI run and stopping

- ```
jmeter -n -t bookshelf-load.jmx \
-l results/run.jtl -j results/run.log \
-Jthreads=50 -Jrampup=60 -Jduration=660 \
-f -e -o results/report
jmeter -g results/run.jtl -o report2 # report only
```
- o must be empty or absent; -f deletes the old JTL and report. -q extra properties, -L log level, -r/-R remote.
 - bin/shutdown.sh (Shutdown: graceful, dashboard still built), bin/stopdown.sh (StopTestNow: abrupt), headump.sh, threaddump.sh; UDP port 4445 (then 4446...), loopback only.
 - Exit code is 0 even with 100 % failed samples** (and on usage errors); 1 = plan missing/unparsable or -o not empty. **Gate CI on statistics.json or the JTL**.

20. JTL results (CSV)

- ```
timestamp,elapsed,label,responseCode,responseMessage,
threadName,dataType,success,failureMessage,bytes,sentBytes,
grpThreads,allThreads,URL,Latency,IdleTime,Connect
```
- elapsed:** send -> last byte. **Latency:** send -> first byte. **Connect:** connection incl. TLS, inside latency. Connect <= Latency <= elapsed.
  - Keep CSV (not XML), save only what you need (`jmeter.save.saveservice.*`, response data off). Transaction rows: sums; IdleTime = timer time.

## 21. HTML dashboard and APDEX

- e -o dir at the end, -g jtl -o dir later, or Tools > Generate HTML Report. Override report generator.org.properties in `user.properties`:

```
jmeter.reportgenerator.apdex_satisfied_threshold=500
jmeter.reportgenerator.apdex_tolerated_threshold=1500
jmeter.reportgenerator.apdex_per_transaction=\
TX.:1000|3000|GET /api/books \\(search\\)|300|800
jmeter.reportgenerator.overall_granularity=10000
jmeter.reportgenerator.exporter.html.series_filter=\
^(TX.|.)*(success|failure)?$
```
- APDEX = (satisfied + tolerating / 2) / total; failures = frustrated. Per-transaction names are whole-label regex, ; separated. Percentiles: `aggregate_rpt_pct1..3` (90/95/99).
- Gate CI on statistics.json (errorPct, pct1..3 ResTime).

## 22. Backend Listener -> InfluxDB / Grafana

- Implementation ...influxdb.InfluxdbBackendListenerClient (or Graphite, InfluxDBRaw); Async Queue size 5000; aggregates sent every 5 s.

```
influxdbMetricsSender=...influxdb.HttpMetricsSender
influxdbUrl=http://host:8086/write?db=jmeter (1.x)
or http://host:8086/api/v2/write?org=ORG&bucket=jmeter (2.x)
influxdbToken= (2.x, 5.2+; add the row)
application=bookshelf measurement=jmeter summaryOnly=false
samplerRegex=.* (find) percentiles=99;95;90
```
- Grafana reads the measurement and events annotations; tag runs with TAG\_runId=....

## 23. Analysis checklist

- Per transaction, never Total; percentiles (p90/p95/p99), not averages; mean >>> median = slow tail; p99 >>> p90 = separate slow population.
- Steady state only (cut warm-up/ramp-down with start\_date/end\_date); errors by type and code; Little's Law check  $X \sim N / (R + Z)$ .
- Saturation knee:** throughput flattens, response time climbs.  $N^* \sim X\_max (R0 + Z)$ ; 60 x 3.2 ~ 192 users.
- Coordinated omission:** closed model under-samples stalls (throughput dips while RT spikes); use an open model.
- Generator healthy? CPU < -80 %, GC, network. Then server: pools, GC, CPU, DB queries, locks.

## 24. Distributed testing (RMI)

- The whole plan runs on every worker:** total load = threads x workers. Listeners/results on the controller.

```
bin/create-rmi-keystore.sh # copy to all nodes
jmeter-server -Djava.rmi.server.hostname=10.0.1.11 \
-Jserver.rmi.localport=4000 # registry 1099
jmeter -n -t plan.jmx -R gen1,gen2 -Gthreads=25 \
-l res.jtl -e -o report -X
```
- G reaches workers, -J does not. CSV files, plugins, drivers on **every** node. Default sender mode StrippedBatch.
- Ports: 1099 + server.rmi.localport on workers. With SSL keep client.rmi.localport=0 (a fixed port failed in 5.6.3). Exit code 0 even if no worker started.

## 25. jmeter-maven-plugin 3.8.0

- ```
<plugin>groupId=com.lazerycode.jmeter</groupId>
<artifactId>jmeter-maven-plugin</artifactId>
<version>3.8.0</version>
<executions>configure | jmeter | results </executions>
<configuration>
  <jmeterVersion>5.6.3</jmeterVersion>
  <jmeterExtensions>{plugin artifacts}</jmeterExtensions>
  <propertiesUser>threads>50</threads></propertiesUser>
  <errorRateThresholdInPercent>1</errorRateThresholdInPercent>
</configuration>
</plugin>
```
- Default jmeterVersion is **5.6.2**; pin 5.6.3, mvn clean verify after changing it. Plans in src/test/jmeter. results fails on 1 failed sample unless a threshold is set.

26. Lean load-generator rules

- CLI mode only (GUI to build and debug); few listeners, no View Results Tree at load; CSV JTL with minimal fields.
- Few cheap assertions; Groovy with caching; no functional mode; test data prepared in CSV files.
- Heap -Xms = -Xmx; timeouts on every request; watch generator CPU, GC, file descriptors, ephemeral ports; never generate load from the server under test.

27. From 2.x / 3.0 books to 5.6 and 6.0

- 5.5 Open Model TG. 5.6: ServiceLoader plugins, Caffeine caches, Groovy 3.0.17. 5.6.1 UTF-8 default. 5.6.3 (Jan 2024): Groovy 3.0.20. No Groovy 4 in any release.
- 6.0 (planned, unreleased):** Java 17+ (21 rec.); MongoDB elements removed; Cookie Manager drops expired cookies; 307/308 keep method; Open Model / Precise Throughput Timer timed from TG start; MySQL com.mysql.cj.jdbc.Driver; Groovy 5.1.3, Tika 4.x; JMS to jakarta.jms (re-check).