

PERFORMANCE TESTING TOOLS



- (Standalone) Tool
 - Performance Testen (Load, Stress,...)
 - Apache Projekt
 - Frei verfügbar
 - Integration in andere Umgebungen (Maven,...)
 - Breites Spektrum an Funktionalität verfügbar (Web-Tests, DB-Tests, FTP-Tests,...)

Resources

- JMeter
 - <http://jmeter.apache.org/usermanual/get-started.html>
- JMeter Plugins
 - <https://jmeter-plugins.org>
- BlazeMeter
 - <https://www.blazemeter.com>

INSTALLATION

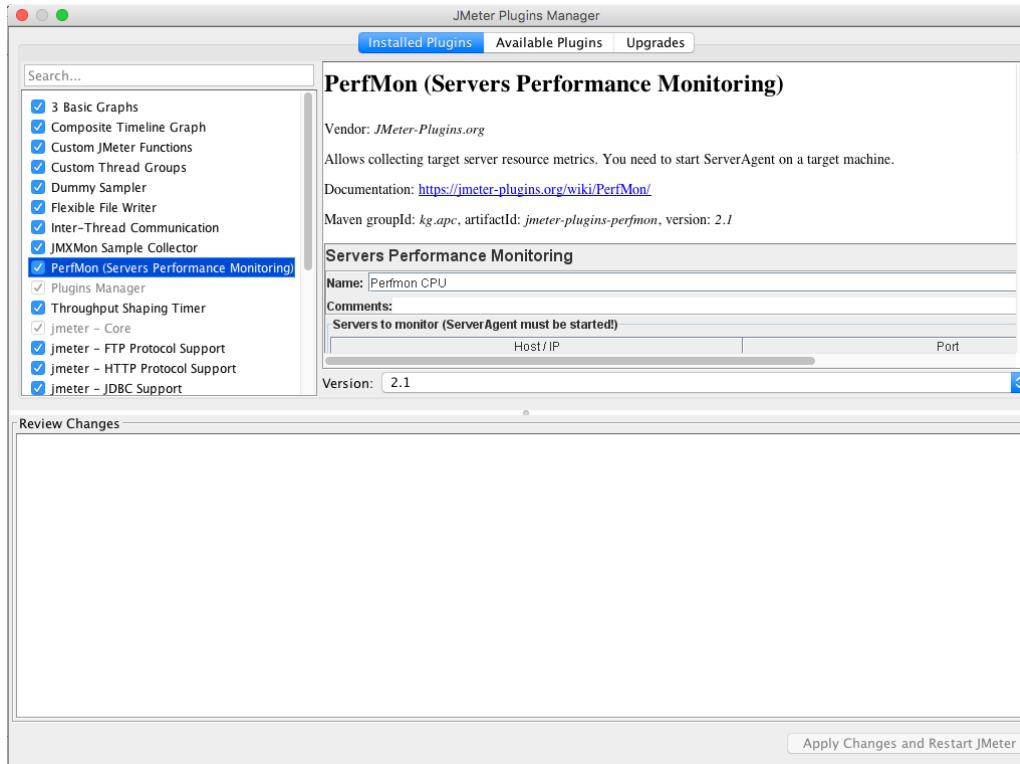
Installation

- JMeter (JDK necessary)
 - http://jmeter.apache.org/download_jmeter.cgi
- JMeter Plugins Manager
 - <https://jmeter-plugins.org/downloads/all/>
→ Copy to ext/lib → restart JMeter

Installation



Installation



Installation

- Plugins
 - „jpgc – Standard Set“
 - „3 Basic Graphs“
 - „PerfMon“

- 3 Basic Graphs
- Composite Timeline Graph
- Custom JMeter Functions
- Custom Thread Groups
- Dummy Sampler
- Flexible File Writer
- Inter-Thread Communication
- JMXTMon Sample Collector
- PerfMon (Servers Performance Monitoring)
- Plugins Manager
- Throughput Shaping Timer
- jmeter – Core
- jmeter – FTP Protocol Support
- jmeter – HTTP Protocol Support
- jmeter – JDBC Support
- jmeter – JMS Support
- jmeter – JUnit Support
- jmeter – Java Components
- jmeter – LDAP Protocol Support
- jmeter – Mail/SMTP Support
- jmeter – MongoDB Support
- jmeter – OS Process Support
- jmeter – TCP Protocol Support
- jmeter – Various Core Components
- jpgc – Standard Set

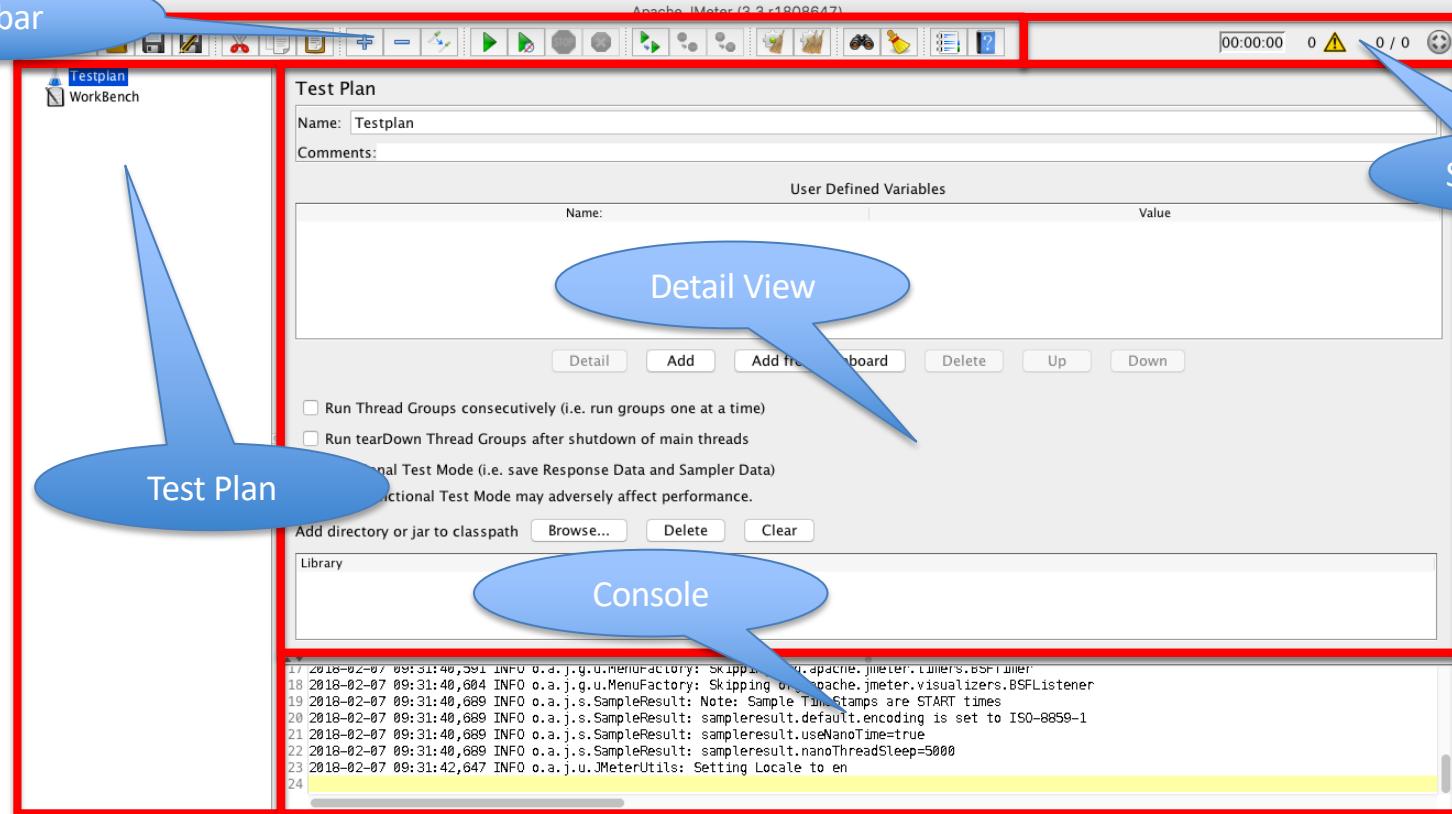
Start

- Testcreation/Testdebugging via GUI
 - jmeter.sh or jmeter.bat
- Testrun
 - Console!

```
ramirez-mon:bin chris$ ./jmeter.sh
=====
Don't use GUI mode for load testing, only for Test creation and Test debugging !
For load testing, use NON GUI Mode:
    jmeter -n -t [jmx file] -l [results file] -e -o [Path to output folder]
& adapt Java Heap to your test requirements:
    Modify HEAP="-Xms512m -Xmx512m" in the JMeter batch file
=====
```

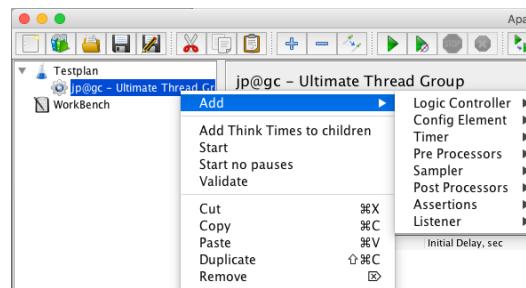
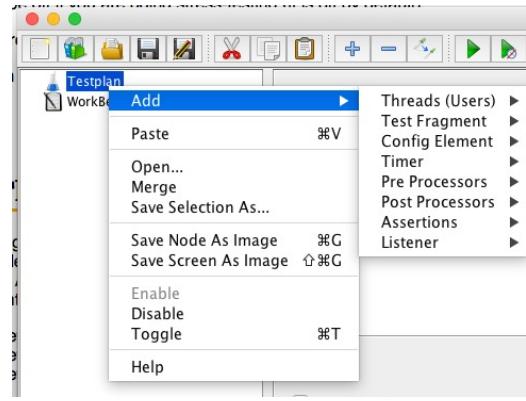
TOOL - GUI

Toolbar



JMeter Tests

- Test consist of different elements
 - Threads
 - Configurations
 - Samplers/Logic
 - Listeners
 - Assertions
- Grouping is possible (z.B. Sampler)

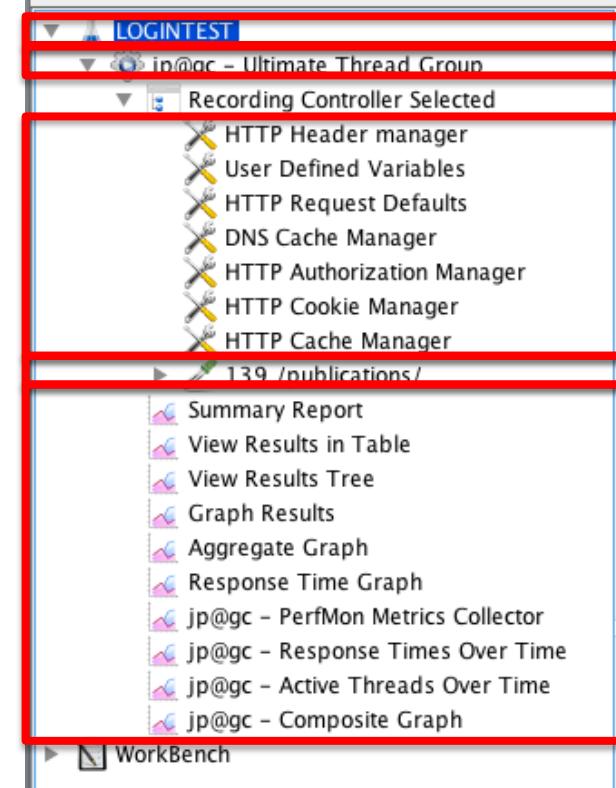


JMeter Tests

- Execution Order
 - Configuration elements
 - Pre-Processors
 - Timers
 - Sampler (in order)
 - Post-Processors (unless SampleResult is **null**)
 - Assertions (unless SampleResult is **null**)
 - Listeners (unless SampleResult is **null**)

JMeter Test Plan

- Typical Test Plan
 - Root
 - User
 - Configuration
 - Actions
 - Results/Reports



USERS/THREADS

JMeter Test Plan – Users

jp@gc – Ultimate Thread Group

Name: jp@gc – Ultimate Thread Group

Comments:

[Help on this plugin](#)

Action to be taken after a Sampler error

Continue Start Next Thread Loop Stop Thread Stop Test Stop Test Now

Threads Schedule

Start Threads Count	Initial Delay, sec	Startup Time, sec	Hold Load For, sec	Shutdown Time
50	0	300	90	10

Add Row Copy Row Delete Row

Expected parallel users count

jmeter-plugins.org

JMeter Test Plan – Users

jp@gc – Ultimate Thread Group

Name: jp@gc – Ultimate Thread Group

Comments:

[Help on this plugin](#)

Action to be taken after a Sampler error

Continue Start Next Thread Loop Stop Thread Stop Test Stop Test Now

Threads Schedule

Start Threads Count	Initial Delay, sec	Startup Time, sec	Hold Load For, sec	Shutdown Time
50	0	300	90	10
200	330	30	30	10

Add Row Copy Row Delete Row

Expected parallel users count

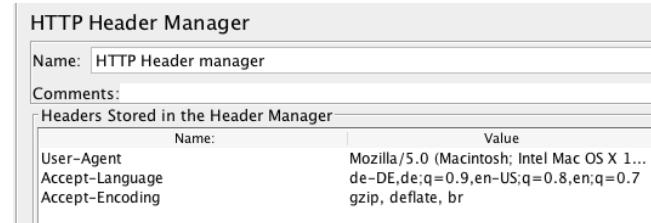
Number of active threads

Elapsed time

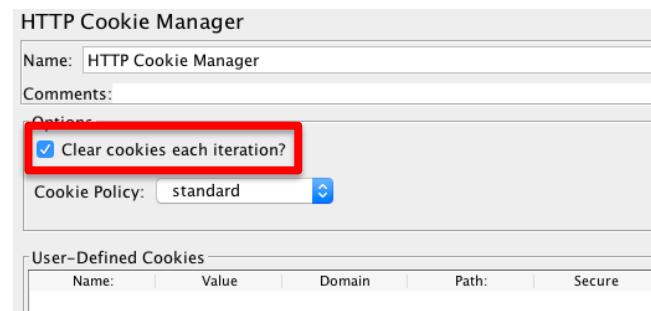
CONFIGURATION

JMeter Tests – Configuration

- HTTP Header Manager
 - User-Agent, Accept-Language, Accept-Encoding,...

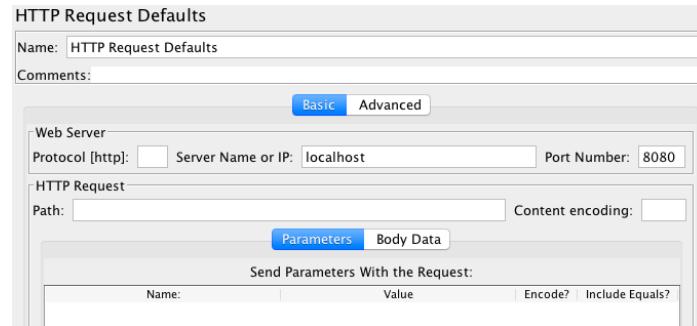


- HTTP Cookie Manager
 - Own Cookies oder Cookies of the page



JMeter Tests – Configuration

- HTTP request defaults
 - Server, IP, Port, Path,...
- User defined variables
 - Placeholder

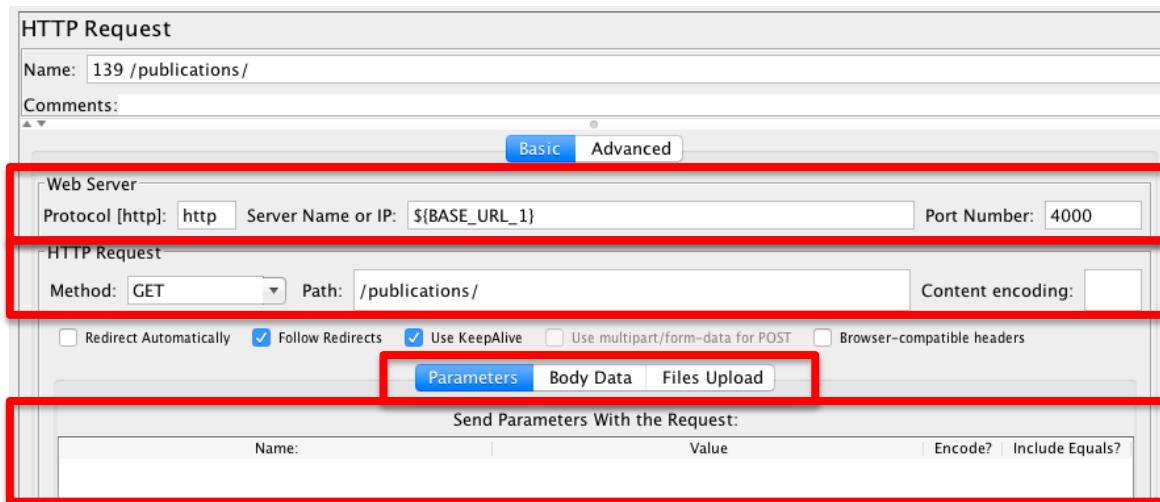


User Defined Variables		
Name:	Value	Description
BASE_URL_1	localhost	

ACTIONS (LOGIC/SAMPLER)

JMeter Tests – Actions

- HTTP Request
 - GET request to an address



JMeter Tests – Actions

- Some other actions
 - DB
 - FTP
 - JMS
 - SMTP
 - JUnit!
 - Selenium
 - ...

RESULTS AND REPORTS

JMeter Tests – Results

- Summary Report

Summary Report

Name: Summary Report

Comments:

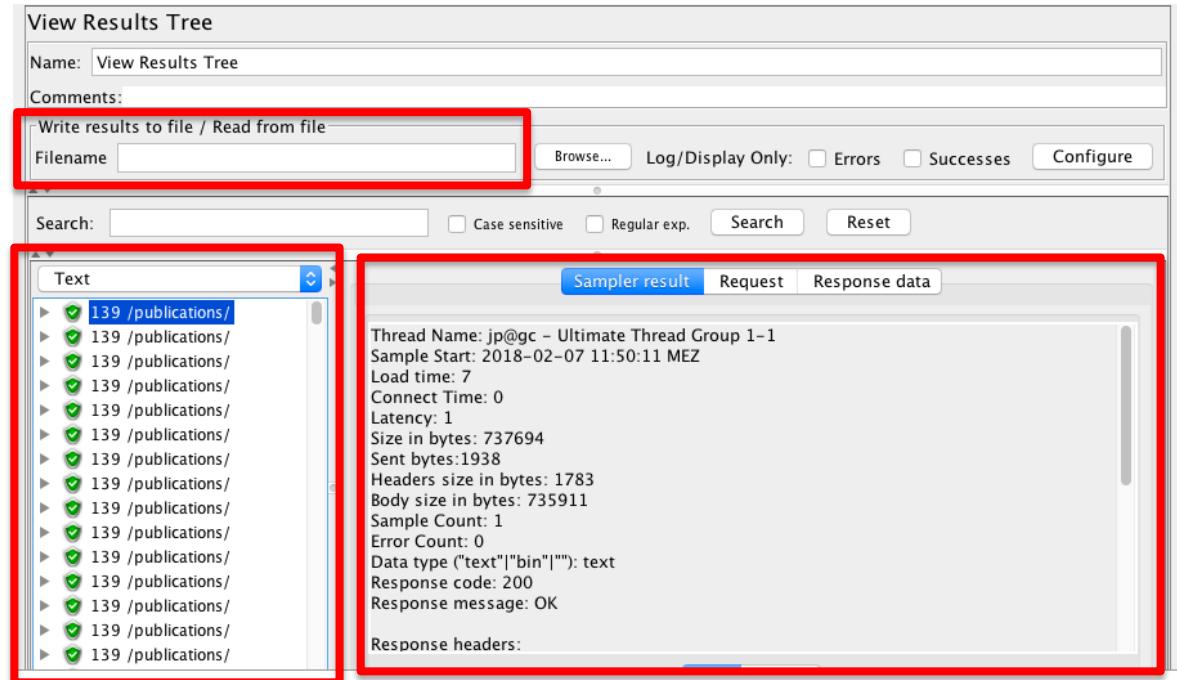
Write results to file / Read from file

Filename Browse... Log/Display Only: Errors Successes Configure

Label	# S...	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB...	Sent KB/sec	Avg. Bytes
139 /publications/	10788	18	3	351	15,44	0,01%	224,7/sec	161896,73	425,33	737660,3
TOTAL	10788	18	3	351	15,44	0,01%	224,7/sec	161896,73	425,33	737660,3

JMeter Tests – Results

- Results in Table/Tree



JMeter Tests – Reports

jp@gc – PerfMon Metrics Collector

Name: jp@gc – PerfMon Metrics Collector

Comments: [Help on this plugin](#)

Servers to Monitor (ServerAgent must be started, see help) –

Host / IP	Port	Metric to collect	Metric parameter (see help)
localhost	4444	CPU	
localhost	4444	Memory	

Add Row Copy Row Delete Row

Write results to file / Read from file

Filename Browse... Log/Display Only: Errors Successes

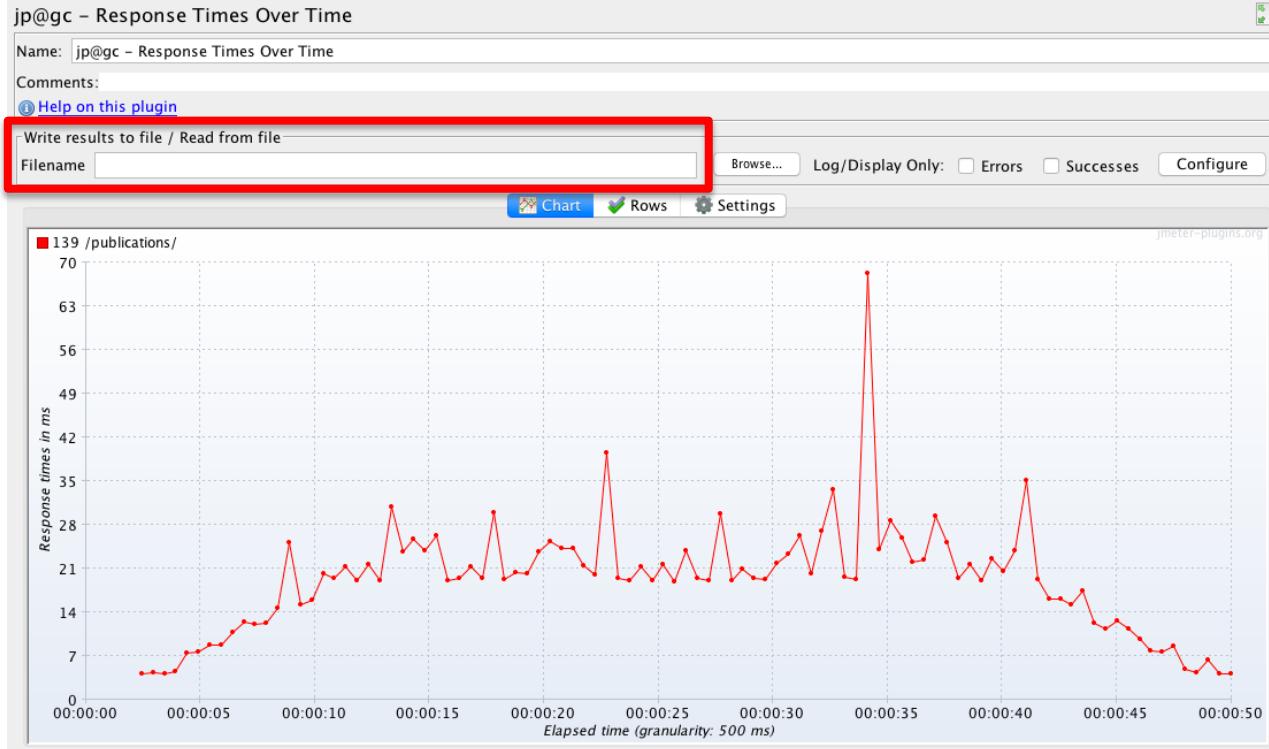
localhost CPU localhost Memory

Performance Metrics

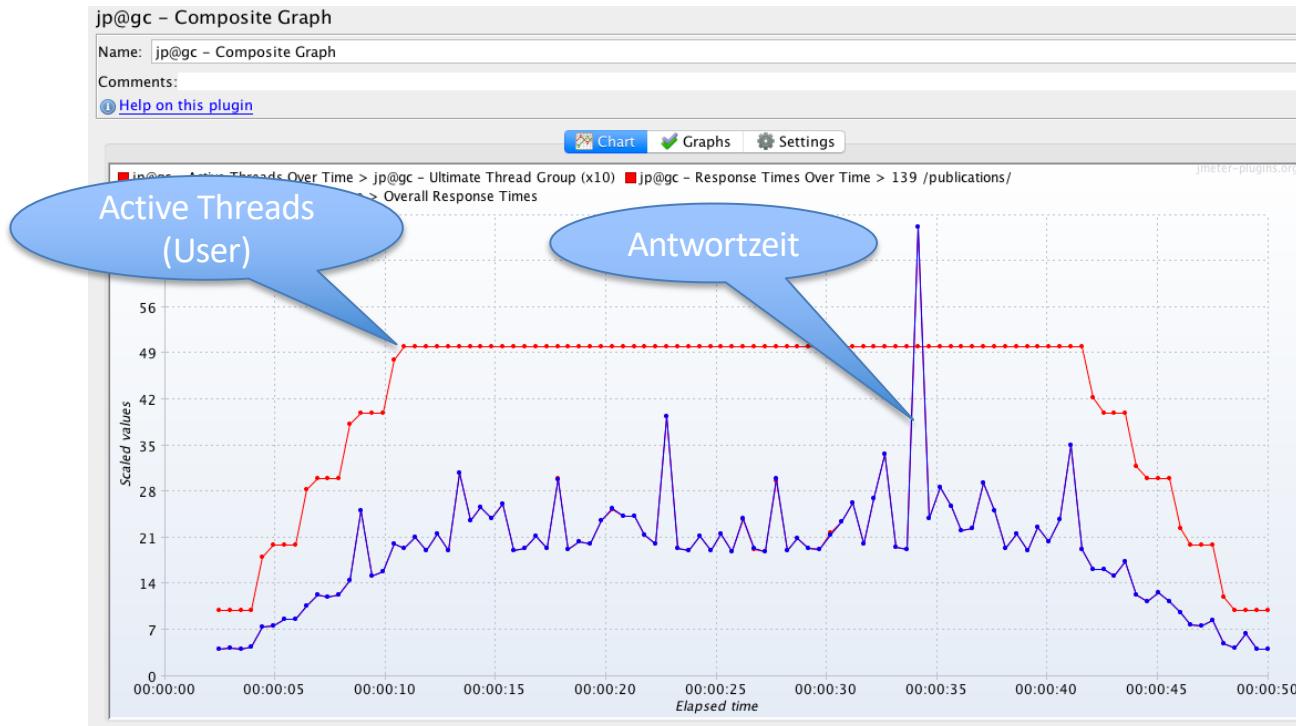
Elapsed time (granularity: 1 sec)

jmeter-plugins.org

JMeter Tests – Reports



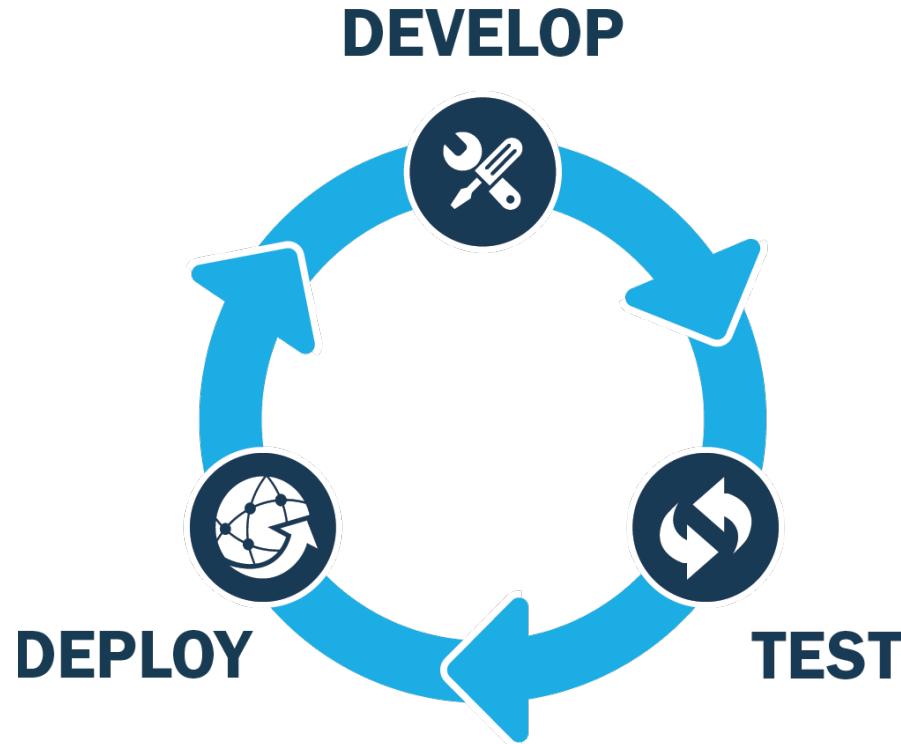
JMeter Tests – Reports



JMeter „Best Practice“

- Record tests
 - Built-in recorder (proxy)
 - BlazeMeter (browser plugin)
- Combination with other tools
 - Selenium (via JUnit tests or Selenium WebDriver plugin)

Advanced Usage – Continuous Integration

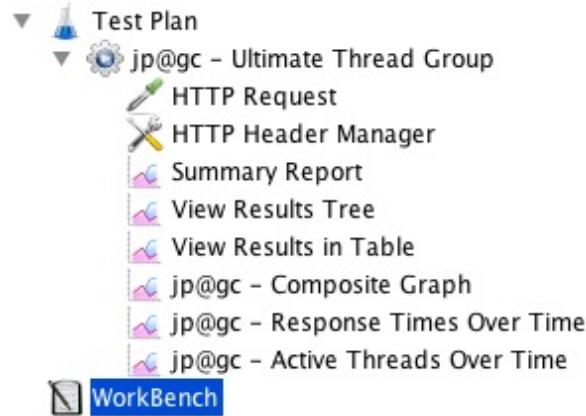


Advanced Usage – Continuous Integration

- JMeter with Apache Maven
 - Maven as build tool
 - JMeter as performance testing tool
- Uses the plugin mechanismus of Maven
 - JMeter out-of-the-box functionality is no problem
 - Extensions (plugins) not trivial (dependencies)
- Convention over Configuration
 - JMeter tests in src/test/jmeter

TESTPLAN

Advanced Usage – Continuous Integration



Advanced Usage – Continuous Integration

Test Plan

- jp@gc – Ultimate Thread Group (highlighted with a red box)
- HTTP Request
- HTTP Header Manager
- Summary Report
- View Results Tree
- View Results in Table
- jp@gc – Composite Graph
- jp@gc – Response Times Over Time
- jp@gc – Active Threads Over Time

WorkBench

Start Threads Count	Initial Delay, sec	Startup Time, sec	Hold Load For, sec	Shutdown Time
1000	0	60	180	0
500	90	30	90	10

Add Row Copy Row Delete Row

Expected parallel users count

Number of active threads

Elapsed time

Advanced Usage – Continuous Integration

The screenshot shows the JMeter Test Plan interface. A red box highlights the 'HTTP Request' item under the 'in@gc - Ultimate Thread Group'. To the right, a detailed view of the 'HTTP Request' configuration is shown. The 'Name' field is set to 'HTTP Request'. The 'Web Server' section has 'Protokoll [http:]' set to http, 'Server Name oder IP:' set to swdyn.aau.at, and 'Port Number:' set to 8080. Under the 'HTTP Request' tab, 'Methode:' is set to POST, 'Pfad:' is set to /v1/changes, and 'Content Kodierung:' is left empty. The 'Body Data' tab is selected, showing a large JSON payload:

```
1 {"src": "cGFja2FnZSBhdC5hYXU7D0oNCnB1YmxpYyBjBgfzcyBNYXZlbLByb2pLY3RNZXryawNzIHSNC1AgICBwcmL2YXRlIEExvbmcgYmxvYz0wbDNC1AgICBwcmL2YXRlIEExvbmcgbs2PMGw7DQogICAgCH3pdF0ZSBMbz5nIG51bURLGVuZGVuL1cz0wbDNC1AgICBwcmL2YXRlIEExvbmcgbnVtUGx1Z2Lucz0wbDNC1AgICBwcmL2YXRlIEExvbmcgbnVt029uZmLndJ1JZFBsdWdpbnM9Ngw7D0ogICAgchJpdmf0ZSBMbz5nIG51bU1vZHVsZxM9Mg7DQogICAgchJpdmf0ZSBMbz5nIGRpdD0wbDsQly91c2VKIGFzIG1heERpdCBpb1BvdmVYVwxsIG1ldHJpY3MNC1AgICBwcmL2YXRlIEJ1awxxSGFsc3RLyWQgaGfsC3RYw07D0ogICAgchJpdmf0ZSBMbz5nIGN1c3RvbWL6XRp25TAxpLPTB80wKL8g1CAgY2JvPyA50Z2uyYwgZGwvZw5kZW5javVzL2V4dGvbybfISGRlCgVzGVuY21cylc1ICHhcyBzdWJzdgL0dKRlIG2vc1b2hLc2lvb1BhbHQgY291Cgxpmbc/PyNcLAG1CAvLyByZwhxdL2ZSBudw1EZXAqgWSkIG51bvBsdwdpbnMgdG8gdG90TwgYmxvYBhbmQvb31gbGjsb2MNC1AgICAVL21jY2F1ZTogCG9zc2LibGld2F5cyB0byByd4gdghLIGj1awxkpYao2hlyY2sgchJvZm1sZXM1GfUzCBwBhVnaW5zP28}
```

Advanced Usage – Continuous Integration



The image shows a JMeter interface with a tree view of a test plan and a detailed configuration dialog.

Test Plan Structure:

- Test Plan
 - jp@gc - Ultimate Thread Group
 - HTTP Request
 - HTTP Header Manager** (highlighted with a red box)
 - Summary Report
 - View Results Tree
 - View Results in Table
 - jp@gc - Composite Graph
 - jp@gc - Response Times Over Time
 - jp@gc - Active Threads Over Time
- WorkBench

HTTP Header Manager Dialog:

Name: HTTP Header Manager

Kommentare:

Name:	Wert
Origin	http://swdyn.aau.at:81
Accept-Encoding	gzip, deflate
Accept-Language	de-DE,de;q=0.9,en-US;q=0.8,en;q=0.7
User-Agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_5...)
Content-Type	application/json;charset=UTF-8
Accept	application/json, text/plain, */*
Referer	http://swdyn.aau.at:81/
Connection	keep-alive

Advanced Usage – Continuous Integration

```
<plugin>
  <groupId>com.lazerycode.jmeter</groupId>
  <artifactId>jmeter-maven-plugin</artifactId>
  <version>2.6.0</version>
</plugin>

<executions>
  <execution>
    <id>jmeter-tests</id>
    <goals>
      <goal>jmeter</goal>
    </goals>
  </execution>
</executions>

<configuration>
  ...
</configuration>
</plugin>
```

The diagram illustrates the structure of a Maven plugin configuration. It consists of three main sections, each highlighted with a red border:

- Plugin ID:** Contains the `<groupId>`, `<artifactId>`, and `<version>` elements.
- Execution command:** Contains the `<executions>` section, which includes an `<execution>` block. This block contains the `<id>` (set to `jmeter-tests`) and `<goals>` (containing a single `<goal>jmeter</goal>`) elements.
- Configuration:** Contains the `<configuration>` section, which includes an ellipsis (`...`).

Brackets on the right side of the diagram group these sections into three categories: **Plugin ID**, **Execution command**, and **Configuration**.

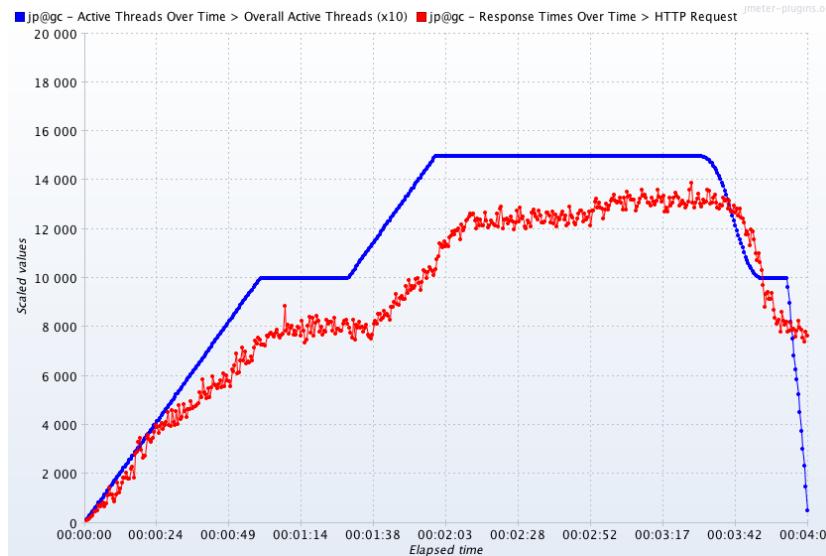
Advanced Usage – Continuous Integration

```
<plugin>
...
<configuration>
<jmeterVersion>3.3</jmeterVersion>
<jmeterExtensions>
<artifact>kg.apc:jmeter-plugins-casutg:2.5</artifact>
<artifact>kg.apc:jmeter-plugins-dummy:0.2</artifact>
<artifact>kg.apc:jmeter-plugins-ffw:2.0</artifact>
<artifact>kg.apc:jmeter-plugins-fifo:0.2</artifact>
<artifact>kg.apc:jmeter-plugins-functions:2.0</artifact>
<artifact>kg.apc:jmeter-plugins-graphs-basic:2.0</artifact>
<artifact>kg.apc:jmeter-plugins-graphs-composite:2.0</artifact>
<artifact>kg.apc:jmeter-plugins-cmn-jmeter:0.5</artifact>
<artifact>kg.apc:jmeter-plugins-jmxmon:0.2</artifact>
<artifact>kg.apc:jmeter-plugins-manager:0.19</artifact>
<!--<artifact>kg.apc:jmeter-plugins-perfmon:2.1</artifact>-->
<artifact>kg.apc:jmeter-plugins-tst:2.1</artifact>
</jmeterExtensions>
<downloadExtensionDependencies>false</downloadExtensionDependencies>
</configuration>
</plugin>
```

DEMO (REST SERVICE)

REST Service

Label	Anz. d...	Durc...	Min	Max	Std. Dev.	% Feh...	Durchsatz	KB/sek	Sent KB/...
HTTP Request	29218	9307	29	20189	3695,23	0,00%	118,5/sec	87,70	1484,43
Gesamt	29218	9307	29	20189	3695,23	0,00%	118,5/sec	87,70	1484,43



REST Service



Outlook?

- Other Samplers/Actions
 - Java Tests
 - Selenium
 - ...
- Distributed performance testing
 - Many „physical“ Clients
- Assertions