



The **Apache Jakarta Project**

[http:// jakarta.apache.org/](http://jakarta.apache.org/)



# for Performance and Load Testing

**Presented By:**

*Mike Taber*

*email: [mtaber@widen.com](mailto:mtaber@widen.com)*



**WIDEN<sup>®</sup>**

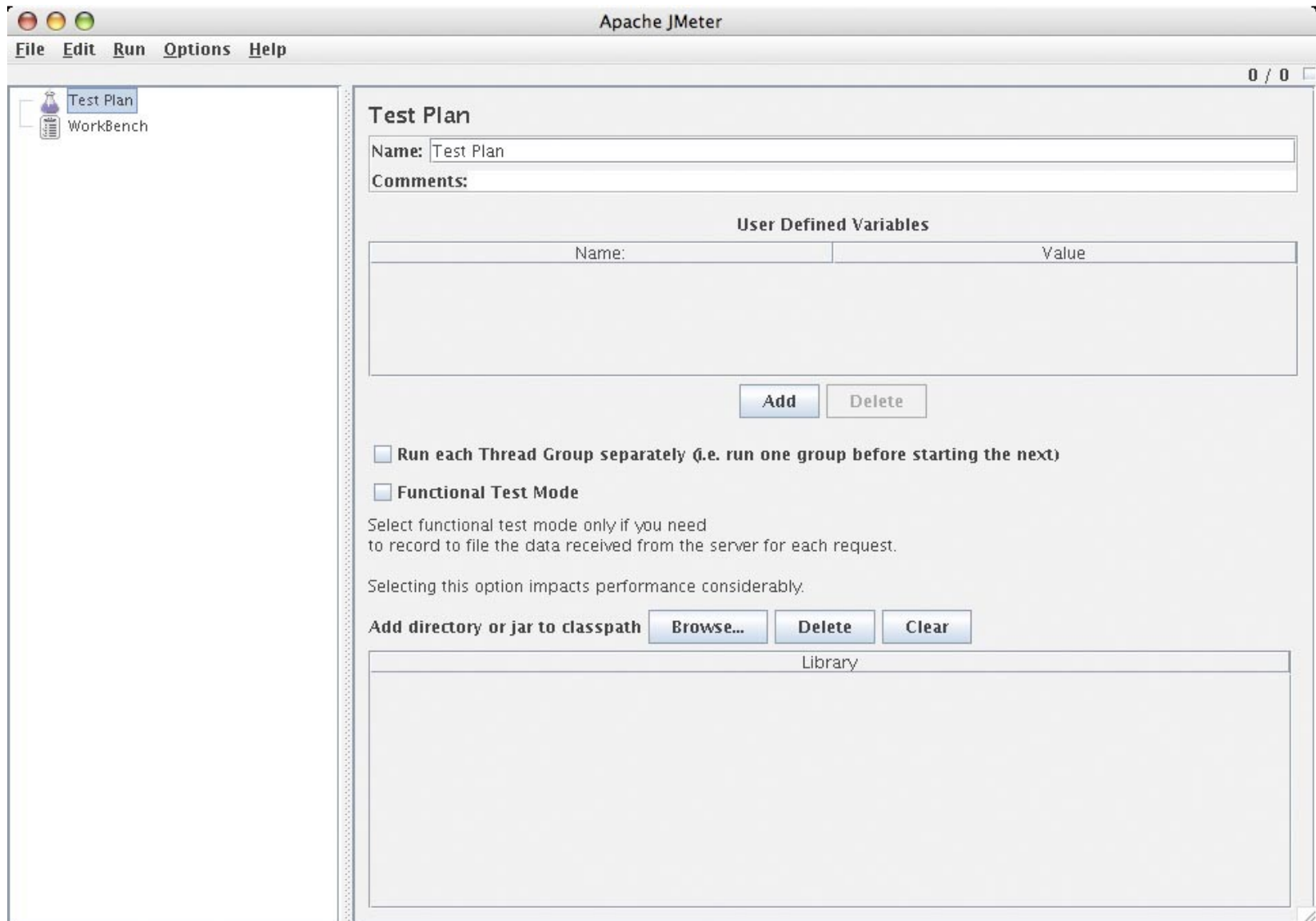


- JMeter (<http://jakarta.apache.org/jmeter>) is a free, open-source performance measurement tool written in Java.
- It can generate complex test plans, monitor and report from a range of options, and validate results via a range of assertion types. Therefore, a natural for load testing and performance analysis of Web services and applications, as well as live assertion testing of various types of data connections.

# Installing and Running JMeter

- The latest version of JMeter can be downloaded from the project website: <http://jakarta.apache.org/jmeter/index.html>
- JMeter requires a fully compliant JVM 1.4 or higher. JMeter is a 100% Java application and should run correctly on any system that has a compliant Java implementation.
- To install a release build, simply unzip the zip/tar file into the directory where you want JMeter to be installed.
- To run JMeter, run the `jmeter.bat` (for Windows) or `jmeter` (for Unix) file.

# The Test Plan



# Thread Groups

- The Thread Group is the basic element of a JMeter Test Plan.
- Each thread represents a user.  
*Setting a thread group to 1000 simulates 1000 users.*

# Thread Group

The screenshot shows the Apache JMeter interface with the Thread Group configuration window open. The window title is "Apache JMeter". The menu bar includes "File", "Edit", "Run", "Options", and "Help". The top right corner shows "0 / 0". The left sidebar contains a tree view with "WISQA Test Plan", "Thread Grou...", and "WorkBench". The main area is titled "Thread Group" and contains the following configuration fields:

- Name:** Thread Group - WISQA
- Comments:** WISQA presentation 2008
- Action to be taken after a Sampler error:**
  - Continue
  - Stop Thread
  - Stop Test
- Thread Properties:**
  - Number of Threads (users):** 12
  - Ramp-Up Period (in seconds):** 15
  - Loop Count:**  Forever 1
  - Scheduler

# Samplers

- Samplers do the actual work in JMeter and interact with the server you are loading.
- There are a number of Samplers in JMeter by default.

*For web testing use 'HTTP Request'*

# Samplers

The screenshot shows the Apache JMeter interface with the 'HTTP Request' configuration window open. The window title is 'Apache JMeter' and the menu bar includes 'File', 'Edit', 'Run', 'Options', and 'Help'. The status bar shows '0 / 0'. On the left, a tree view shows 'WISQA Test Plan' containing 'Thread Group - WISQA' and 'Request Index' (selected). Below it is 'WorkBench'. The main configuration area is titled 'HTTP Request' and contains the following fields and options:

- Name:** Request Index
- Comments:** WISQA presentation 22 January 2008
- Web Server:**
  - Server Name or IP:** localhost
  - Port Number:** 8080
- HTTP Request:**
  - Protocol (default http):** http
  - Method:** GET
  - Content encoding:** (empty)
  - Path:** /wisqa
  - Redirect Automatically
  - Follow Redirects
  - Use KeepAlive
  - Use multipart/form-data for HTTP POST
- Send Parameters With the Request:**

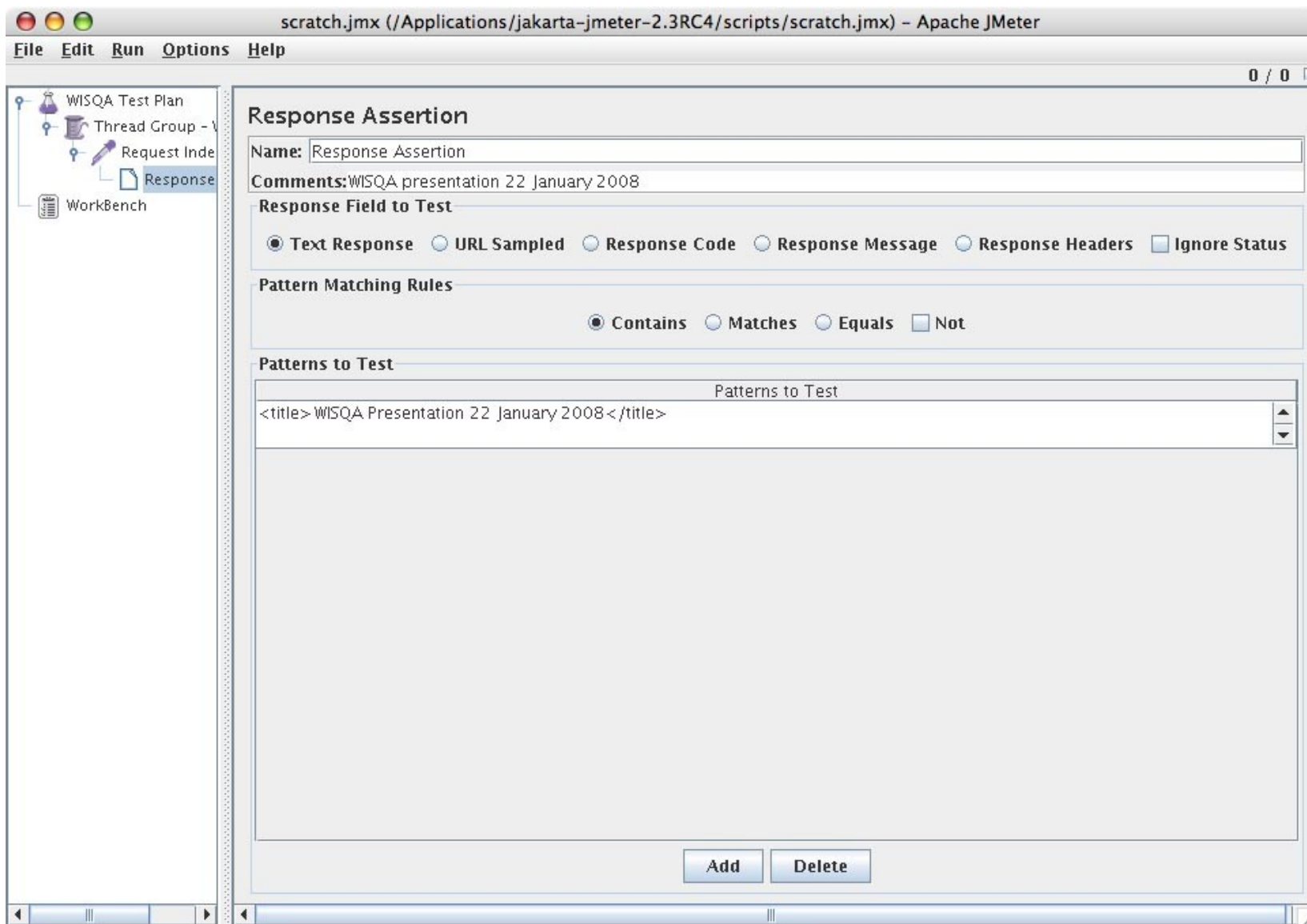
Name:	Value	Encode?	Include Eq...
-------	-------	---------	---------------

Buttons: Add, Delete
- Send a File With the Request:**
  - Filename:** (empty)
  - Value for "name" attribute:** (empty)
  - MIME Type:** (empty)
- Optional Tasks:**
  - Retrieve All Embedded Resources from HTML Files
  - Use as Monitor
  - Embedded URLs must match:** (empty)

# Assertions

- A test isn't a test unless it checks something.  
*Assertions are what do this checking*
- For web applications, these are the useful ones
  - Response
  - Duration
  - HTML Assertion

# Assertions



# Listeners

- The information produced by Samplers is consumed by Listeners.
- Commonly used Listeners are:
  - Graph Results
  - View Results Tree
  - Simple Data Writer

# Listeners

The screenshot shows the Apache JMeter interface with the 'Assertion Results' listener configuration window open. The window title is 'scratch.jmx (/Applications/jakarta-jmeter-2.3RC4/scripts/scratch.jmx) - Apache JMeter'. The menu bar includes 'File', 'Edit', 'Run', 'Options', and 'Help'. The left sidebar shows a tree view of the test plan: 'WISQA Test Plan' containing 'Thread Group - WISQA' with 'Request Index', 'Response Assertion', and 'Assertion Results' (selected). 'WorkBench' is also visible. The main area is titled 'Assertion Results' and contains the following fields and controls:

- Name:** Assertion Results
- Comments:** WISQA presentatin 22 Januray 2008
- Write All Data to a File:** A section with a 'Filename' input field, a 'Browse...' button, a 'Log Errors Only' checkbox, and a 'Configure' button.
- Assertions:** A large empty text area for listing assertions.

The status bar at the top right shows '0 / 0'.

# Logic Controllers

- Logic Controllers determine the order that Samplers are processed.
- More complex scripts will make use of
  - Loops
  - Interleave
  - Random
  - If

# Configuration Elements

- Configuration Elements set default values for other parts of the Test Plan as well as configure variables.
  - CSV Data Set Config allows you to do Data Driven Testing within JMeter
  - HTTP Cookie Manager will automatically intercept and send cookies with requests
  - HTTP Request Defaults saves you time when doing lots of HTTP Samplers

# Variables

- In JMeter, you can reference variables as such:

`${VARIABLE}`

# Variables

The screenshot shows the Apache JMeter GUI for a test plan named 'scratch.jmx'. The left sidebar contains a tree view of the test plan elements, including 'WISQA Test Plan', 'Initialize \${DataDir}', 'Site Config', 'HTTP Request Defaults', 'Initialize File Location', 'Thread Group - WISQA', 'Get New Login from \${Lo...', 'Request Index', 'Submit Zodiac Form', 'Assertion Results', 'View Results Tree', and 'WorkBench'. The main area displays the configuration for the 'Get New Login from \${Lo...' element, specifically the 'User Parameters' section.

**User Parameters**

Name:

Comments:

Update Once Per Iteration

**Parameters**

Name:	User_1
firstName	<code>\${_CSVRead(\${InputFile}, 0)}</code>
lastName	<code>\${_CSVRead(\${InputFile}, 1)}</code>
gender	<code>\${_CSVRead(\${InputFile}, 2)}</code>
zodiacSign	<code>\${_CSVRead(\${InputFile}, 3)}</code>
DummyVar	<code>\${_CSVRead(\${InputFile}, next)}</code>

Buttons: Add Variable, Delete Variable, Add User, Delete User

# Variables

scratch.jmx (/Applications/jakarta-jmeter-2.3RC4/scripts/scratch.jmx) - Apache JMeter

File Edit Run Options Help

0 / 12

WISQA Test Plan

- Initialize \${DataDir}
- Site Config
- HTTP Request Defaults
- Initialize File Location
- Thread Group - WISQA
  - Get Data from \${Input}
  - Request Index
  - Submit Zodiac Form
  - Response Assertion
  - Assertion Results
  - View Results Tree
- WorkBench

### HTTP Request

Name: Submit Zodiac Form

Comments: Submit the more complex form

Web Server

Server Name or IP: Port Number:

HTTP Request

Protocol (default http): Method: POST Content encoding:

Path: /wisqa/form\_results.cgi

Redirect Automatically  Follow Redirects  Use KeepAlive  Use multipart/form-data for HTTP POST

Send Parameters With the Request:

Name:	Value	Encode?	Include Eq...
NameFirst	\${firstName}	<input type="checkbox"/>	<input checked="" type="checkbox"/>
NameLast	\${lastName}	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Gender	\${gender}	<input type="checkbox"/>	<input checked="" type="checkbox"/>
ZodiacSign	\${zodiacSign}	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Add Delete

Send a File With the Request:

Filename: Browse...

Value for "name" attribute:

MIME Type:

Optional Tasks

Retrieve All Embedded Resources from HTML Files  Use as Monitor

Embedded URLs must match:

# Batch

- JMeter can run in a non-gui mode to save resources on the load generating machines.
- JMeter may also be integrated into Ant:  
<http://www.programmerplanet.org/pages/projects/jmeter-ant-task.php>

# Recording

- JMeter allows you record tests (as in LoadRunner) but many find it easier to build scripts one element at a time...

# Recording

The screenshot displays the Apache JMeter GUI for configuring an HTTP Proxy Server. The window title is "scratch.jmx (/Applications/jakarta-jmeter-2.3RC4/scripts/scratch.jmx) - Apache JMeter". The menu bar includes "File", "Edit", "Run", "Options", and "Help". The left sidebar shows a tree view of the test plan: "WISQA Test Plan" (expanded) contains "Initialize \${DataDir}", "Site Config", "HTTP Request Defaults", "Initialize File Location", "Thread Group - WISQA" (expanded), "WorkBench" (expanded), "HTTP Proxy Server" (selected), and "Recording Controller".

The main configuration area is titled "HTTP Proxy Server". It includes the following fields and options:

- Name:** HTTP Proxy Server
- Comments:** Need this to record our actions
- Port:** 4444  Attempt https Spoofing
- Test plan content:**
  - Target Controller:** WorkBench > Recording Controller
  - Grouping:** Do not group samplers
  - Capture HTTP Headers  Add Assertions  Regex matching
- HTTP Sampler settings:**
  - Type:** HTTP Request  Redirect Automatically  Follow Redirects  Use KeepAlive  Retrieve All Embedded Res
- Content-type filter:**
  - Include:**
  - Exclude:**
- URL Patterns to Include:**
  - URL Patterns to Include
  - 
  -
- URL Patterns to Exclude:**
  - URL Patterns to Exclude
  - /\* .jpg
  -
- Buttons:** Start, Stop, Restart

# Recording

scratch.jmx (/Applications/jakarta-jmeter-2.3RC4/scripts/scratch.jmx) - Apache JMeter

File Edit Run Options Help

0 / 12

WISQA Test Plan

- Initialize \${DataDir}
- Site Config
- HTTP Request Defaults
- Initialize File Location
- Thread Group - WISQA
  - WorkBench
    - HTTP Proxy Server
    - Recording Controller
      - /wisqa
      - /wisqa/
      - /wisqa/form\_results.**
      - /wisqa

## HTTP Request

Name: /wisqa/form\_results.cgi

Comments:

Web Server

Server Name or IP: localhost Port Number: 8080

HTTP Request

Protocol (default http): http Method: POST Content encoding:

Path: /wisqa/form\_results.cgi

Redirect Automatically  Follow Redirects  Use KeepAlive  Use multipart/form-data for HTTP POST

### Send Parameters With the Request:

Name:	Value	Encode?	Include Eq...
NameFirst	Bob	<input type="checkbox"/>	<input checked="" type="checkbox"/>
NameLast	Smith	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Gender	Male	<input type="checkbox"/>	<input checked="" type="checkbox"/>
ZodiacSign	Aries	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Add Delete

Send a File With the Request:

Filename: Browse...

Value for "name" attribute:

MIME Type:

Optional Tasks

Retrieve All Embedded Resources from HTML Files  Use as Monitor

Embedded URLs must match: