

Tech  
Mahindra

Connected World. Connected Solutions.



# TABLE OF CONTENT

- ***Overview***
- ***Installation***
- ***Starting JMeter***
- ***Focus Areas***
- ***Test Elements : Thread Group***
- ***Test Elements : Logic Controllers***
- ***Test Elements : Samplers***
- ***Test Elements : Config Elements***
- ***Test Elements : Pre and Post Processors***
- ***Test Elements : Listeners***
- ***Test Elements : Assertions***
- ***Execution Order & Scoping Rules***
- ***Recording JMeter – Test Plan***
- ***Scaling JMeter***
- ***Best Practices***



## Apache JMeter

- ❑ A 100% pure Java application designed to load test client/server software.
- ❑ Can be used to test performance both on static and dynamic resources (*Example:- Static files, Java Servlets, ASP.NET, PHP, CGI scripts, Java objects, databases, FTP servers, and more.*)
- ❑ Can simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types.

## History

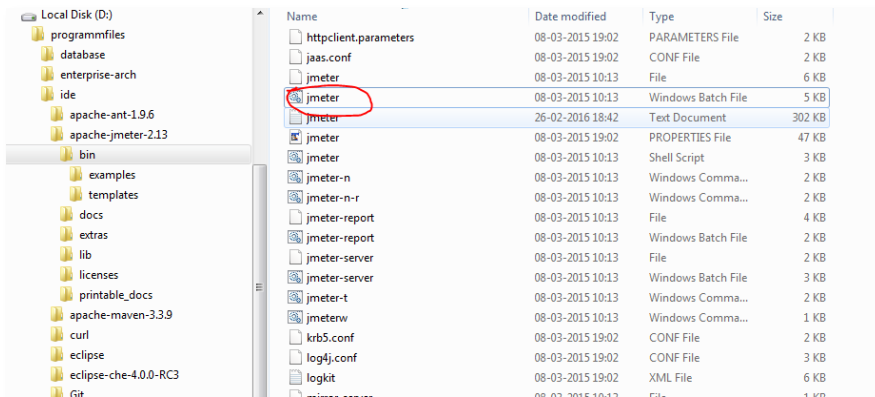
- ❑ Started by Stefano Mazzocchi of Apache Software Foundation
- ❑ Originally developed to test Apache JServ
- ❑ Functionality has been added to test a variety of server-based applications:
  - JDBC
  - LDAP
  - FTP
  - SMTP
  - Etc...

## Pre-requisites

- ❑ Requires Java 6 or Later
- ❑ Path defined to %JAVA\_HOME%\bin

## Installation

- ❑ Download latest JMeter zip (2.13) from below site
  - [http://jmeter.apache.org/download\\_jmeter.cgi](http://jmeter.apache.org/download_jmeter.cgi)
- ❑ Extract anywhere of your choice (please make sure java is set in the path)
- ❑ Several launch scripts (easiest double click jmeter.bat)



```
apache-jmeter-X.Y
apache-jmeter-X.Y/bin
apache-jmeter-X.Y/docs
apache-jmeter-X.Y/extras
apache-jmeter-X.Y/lib/
apache-jmeter-X.Y/lib/ext
apache-jmeter-X.Y/lib/junit
apache-jmeter-X.Y/licenses
apache-jmeter-X.Y/printable_docs
```

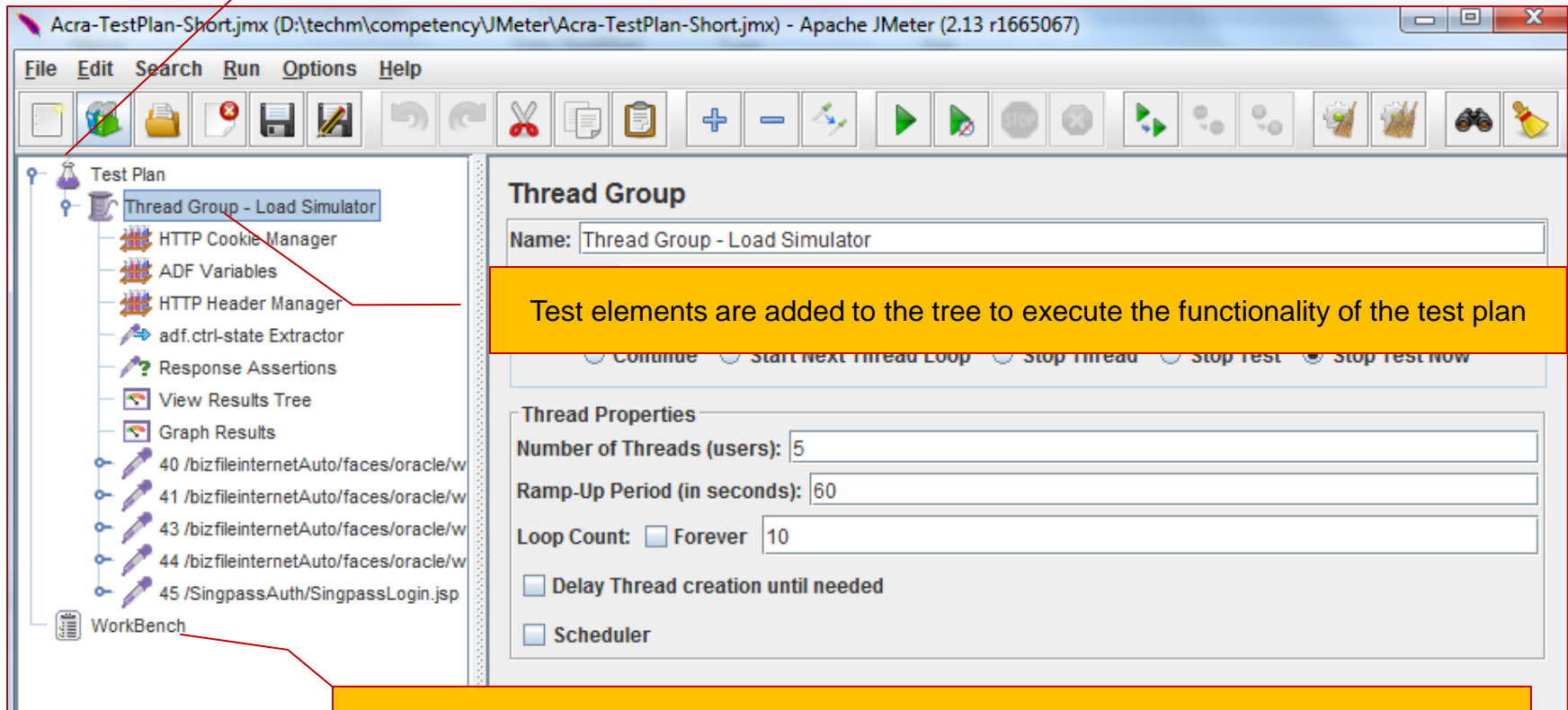
## ***Various options to Start***

- `jmeter.bat` - Standard JMeter in GUI mode
- `jmeterw.cmd` - JMeter without the windows shell console in GUI mode
- `jmeter-n.cmd` - JMeter in non-GUI mode (using a JMX file)
- `jmeter-n-r.cmd` - JMeter remotely in non-GUI mode for (using a JMX file)
- `jmeter-t.cmd` - JMeter in GUI mode (using a JMX file)
- `jmeter-server.bat` - Start JMeter in server mode
- `mirror-server.cmd` - Runs the JMeter Mirror Server in non-GUI mode
- `shutdown.cmd` - Run the Shutdown client to stop a non-GUI instance gracefully
- `stoptest.cmd` - Run the Shutdown client to stop a non-GUI instance abruptly



# Focus Areas

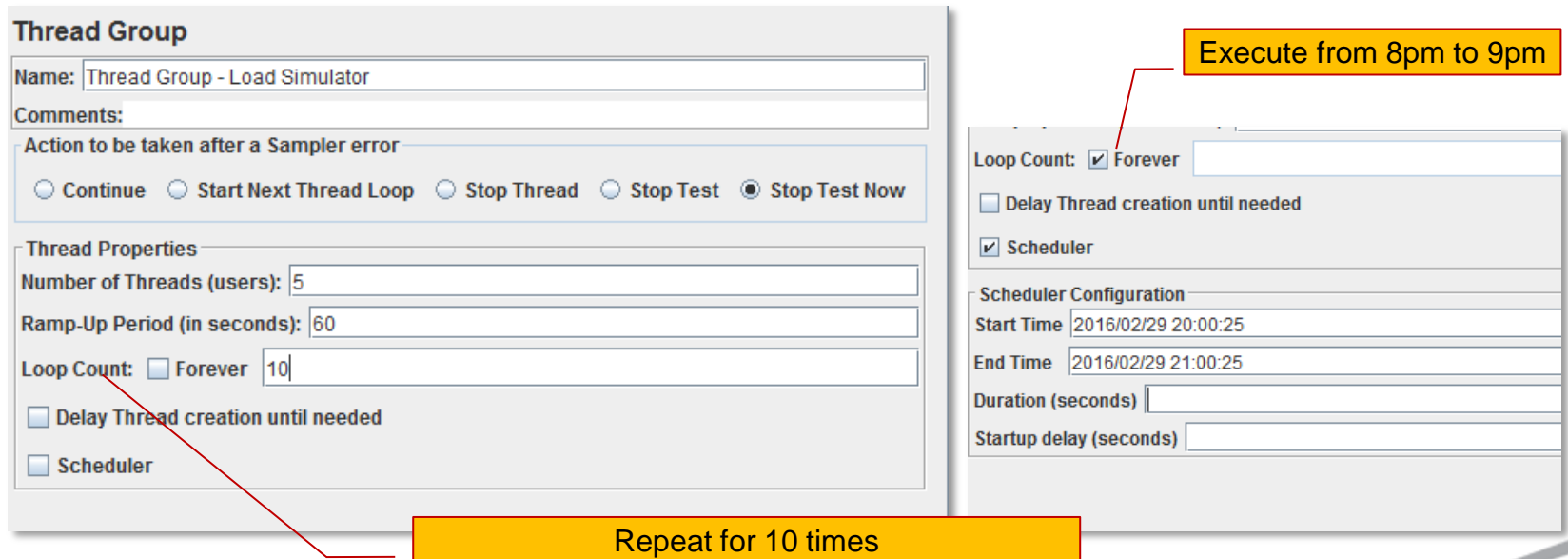
Test Plan is the parent element, describes series of steps to run



The workbench is a temporary parent element where test elements can be manipulated.

# Test Elements : Thread Group

- **Number of Threads** : Number of virtual users.
- **Thread Group** : This is the place where number of threads, ramp-up period, loop count are given while executing.
- **Ramp-Up Period** : It indicates the time taken by J-Meter to create all of the threads needed. If we set 10 seconds as the ramp-up period for 5 threads then the J-Meter will take 10 seconds to create those 5 threads. Also by setting its value to 0, all the threads can be created at once.
- **Loop Count** : By specifying its value J-meter gets to know that how many times a test is to be repeated.



The screenshot shows the 'Thread Group' configuration dialog in JMeter. It is divided into two panes. The left pane shows the 'Thread Group' configuration with the following settings: Name: 'Thread Group - Load Simulator', Comments: (empty), Action to be taken after a Sampler error: 'Stop Test Now' (selected), Thread Properties: Number of Threads (users): 5, Ramp-Up Period (in seconds): 60, Loop Count: 10 (with 'Forever' selected), Delay Thread creation until needed: (unchecked), and Scheduler: (unchecked). The right pane shows the 'Scheduler Configuration' with the following settings: Loop Count: 'Forever' (checked), Delay Thread creation until needed: (unchecked), Scheduler: (checked), Scheduler Configuration: Start Time: '2016/02/29 20:00:25', End Time: '2016/02/29 21:00:25', Duration (seconds): (empty), and Startup delay (seconds): (empty). Two yellow callout boxes with red arrows point to specific settings: one points to the 'Loop Count' field in the left pane with the text 'Repeat for 10 times', and another points to the 'End Time' field in the right pane with the text 'Execute from 8pm to 9pm'.

**Thread Group**

Name: Thread Group - Load Simulator

Comments:

Action to be taken after a Sampler error

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

Thread Properties

Number of Threads (users): 5

Ramp-Up Period (in seconds): 60

Loop Count: ☒ Forever 10

☐ Delay Thread creation until needed

☐ Scheduler

Loop Count: ☒ Forever

☐ Delay Thread creation until needed

☒ Scheduler

Scheduler Configuration

Start Time 2016/02/29 20:00:25

End Time 2016/02/29 21:00:25

Duration (seconds)

Startup delay (seconds)

Execute from 8pm to 9pm

Repeat for 10 times

# Test Elements : Logic Controllers

- **JMeter** has two types of Controllers:
  - Samplers and Logical Controllers.
  - These drive the processing of a test.
- **Logic Controllers** : Logic Controllers helps to customize the logic of when to send requests.

The screenshot displays the Apache JMeter interface. On the left, the 'Test Plan' tree shows a 'Thread Group' selected. A context menu is open over the 'Thread Group', with the 'Add' option expanded to show a list of controller types. The 'If Controller' is highlighted. On the right, the 'Thread Group' configuration panel is visible, showing the 'Name' field set to 'Thread Group'. Below it, the 'If Controller' configuration panel is shown, with the 'Name' field set to 'If Controller'. The 'Condition (default Javascript)' field is empty. At the bottom right, there are two checkboxes: 'Interpret Condition as Variable Expression?' and 'Evaluate for all children?'. A yellow box labeled 'Http Request Sample' is positioned to the right of the 'If Controller' panel, with a red line pointing to it.

**Thread Group**

Name: Thread Group

Comments:

**Add**

- Logic Controller
  - Critical Section Controller
  - ForEach Controller
  - If Controller**
  - Include Controller
  - Interleave Controller
  - Loop Controller
  - Module Controller
  - Once Only Controller
  - Random Controller
  - Random Order Controller
  - Recording Controller
  - Runtime Controller
  - Simple Controller
- Config Element
- Timer
- Pre Processors
- Sampler
- Post Processors
- Assertions
- Listener

**If Controller**

Name: If Controller

Comments:

Condition (default Javascript)

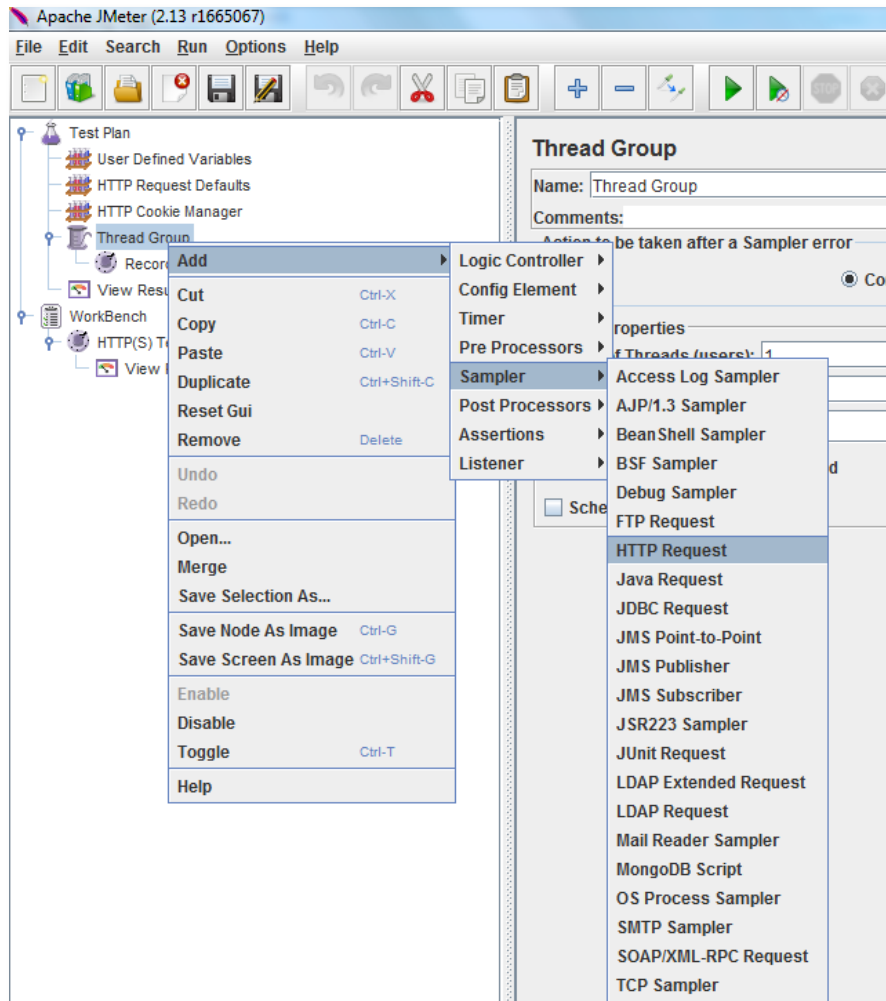
☐ Interpret Condition as Variable Expression? ☐ Evaluate for all children?

Http Request Sample

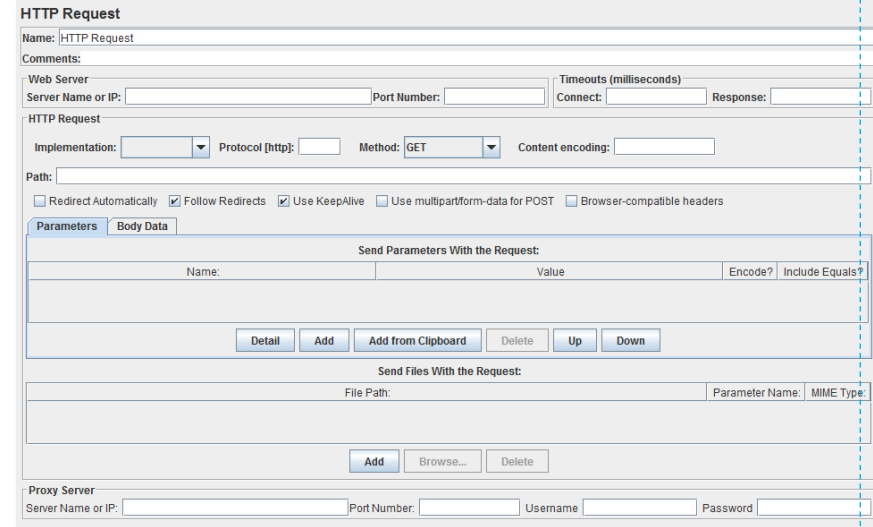


# Test Elements : Samplers

- **Samplers** : Samplers tell JMeter to send requests to a server and wait for a response.
- They are processed in the order they appear in the tree.



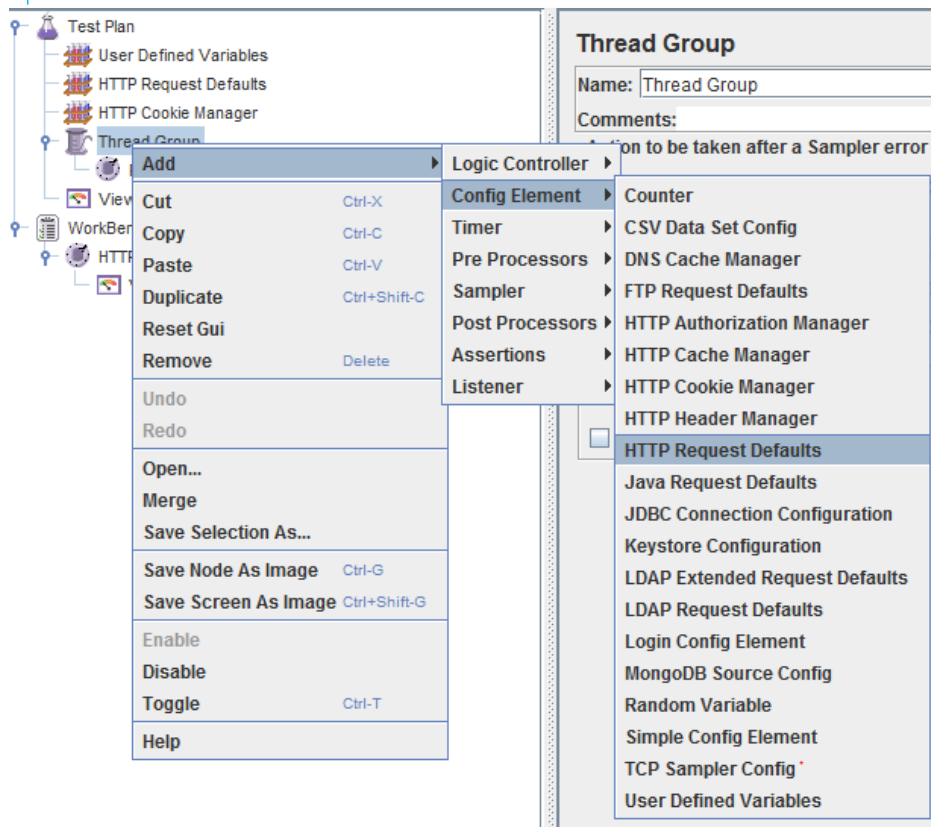
Http Request Sample



The screenshot shows the 'HTTP Request' configuration dialog box. The 'Name' field is 'HTTP Request'. The 'Comments' field is empty. The 'Web Server' section has 'Server Name or IP' and 'Port Number' fields. The 'Timeouts (milliseconds)' section has 'Connect' and 'Response' fields. The 'HTTP Request' section has 'Implementation' (dropdown), 'Protocol (http)' (dropdown), 'Method' (dropdown), and 'Content encoding' (dropdown). The 'Path' field is empty. The 'Redirect Automatically' checkbox is unchecked, 'Follow Redirects' is checked, 'Use KeepAlive' is checked, 'Use multipartform-data for POST' is unchecked, and 'Browser-compatible headers' is unchecked. The 'Parameters' tab is selected, showing a table with 'Name' and 'Value' columns. The 'Send Parameters With the Request' section has 'Detail', 'Add', 'Add from Clipboard', 'Delete', 'Up', and 'Down' buttons. The 'Send Files With the Request' section has 'File Path', 'Parameter Name', and 'MIME Type' fields. The 'Proxy Server' section has 'Server Name or IP', 'Port Number', 'Username', and 'Password' fields.

# Test Elements : Config Elements

- **Config Element:** A configuration element works closely with a Sampler.
- It does not send requests (except for HTTP(S) Test Script Recorder), it can add to or modify requests.
- A configuration element is accessible from only inside the tree branch where it is placed the element



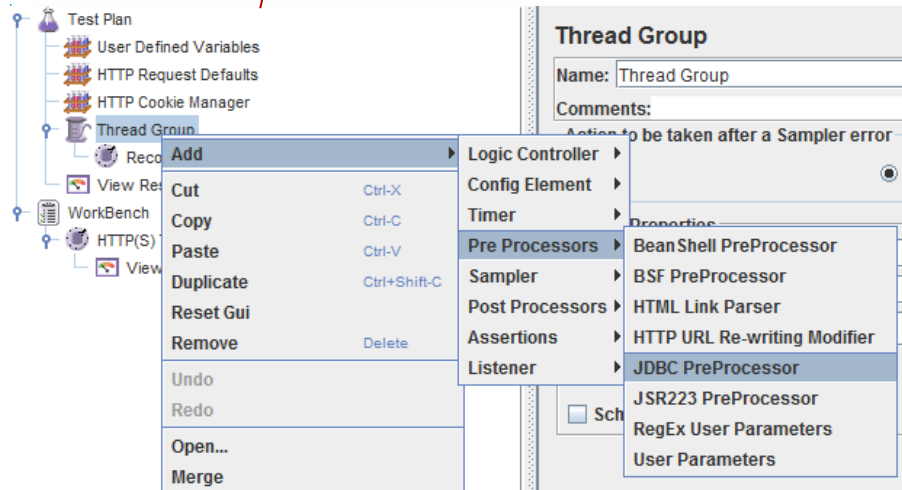
Http Request Defaults

The screenshot shows the 'HTTP Request Defaults' configuration dialog. The 'Name' field is set to 'HTTP Request Defaults'. The 'Comments' field is empty. The 'Web Server' section includes fields for 'Server Name or IP', 'Port Number', 'Connect' (with a dropdown), and 'Response' (with a dropdown). The 'HTTP Request' section includes a dropdown for 'Implementation', a dropdown for 'Protocol [http]', and a text field for 'Content encoding'. The 'Path' field is empty. The 'Parameters' section is expanded, showing a table with columns 'Name', 'Value', and 'Encode?'. The table is currently empty.

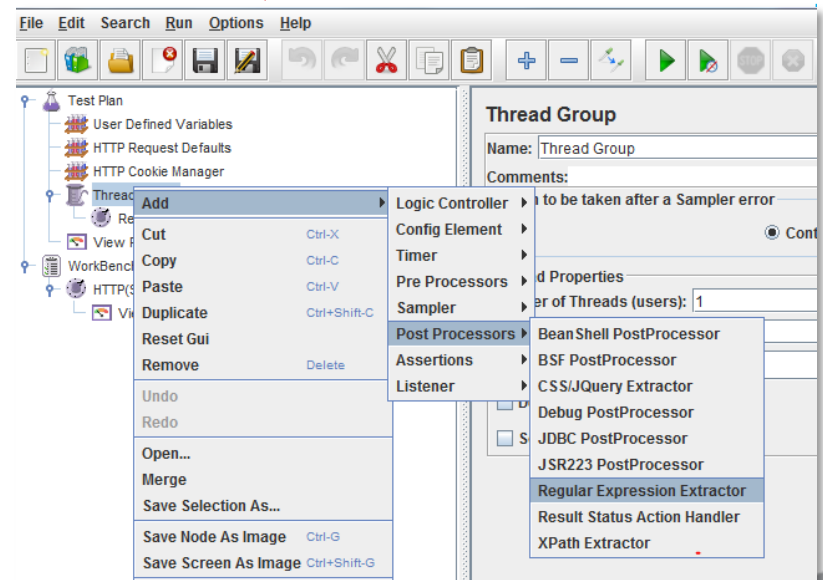
# Test Elements : Pre and Post Processors

- **Pre-processor :**
  - Pre-Processor executes some action prior to a Sampler Request being made.
  - Pre-Processor is most often used to modify the settings of a Sample Request just before it runs, or to update variables that aren't extracted from response text..
- **Post-processor :**
  - Post-Processor executes some action after a Sampler Request has been made.
  - Post-Processor is most often used to process the response data, often to extract values from it.

Pre-Processor

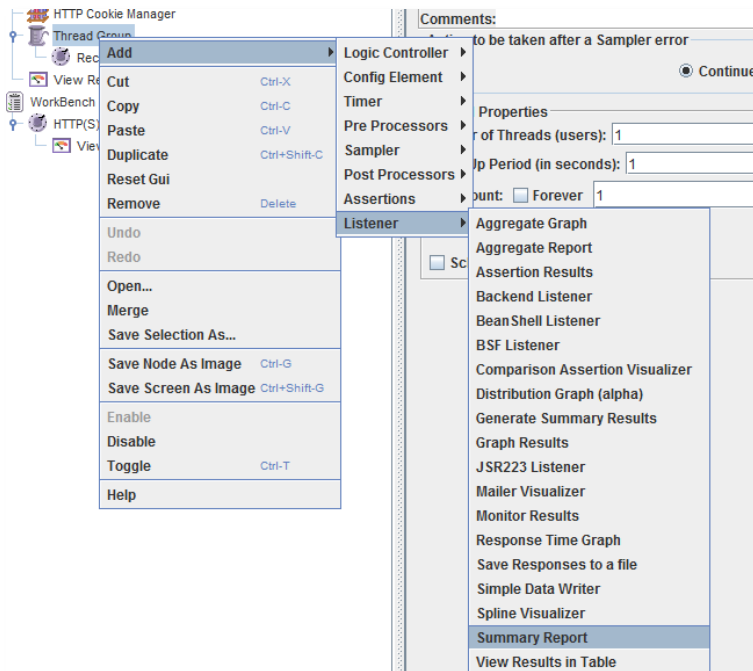


Post-Processor



# Test Elements : Listeners

- **Listeners** : Listeners provide access to the information JMeter gathers about the test cases while JMeter runs.
  - The Graph Results listener plots the response times on a graph.
  - The "View Results Tree" Listener shows details of sampler requests and responses, and can display basic HTML and XML representations of the response.
  - Other listeners provide summary or aggregation information.
- Listeners can direct the data to a file for later use, and configuration button helps to choose which fields to save, and whether to use CSV or XML format



Sample Summary Report

**Summary Report**

Name: Summary Report

Comments:

Write results to file / Read from file

Filename:   Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
40 /bizfileintem...	1	296	296	296	0.00	0.00%	3.4/sec	18.90	5730.0
41 /bizfileintem...	1	2222	2222	2222	0.00	0.00%	27.0/min	39.25	89312.0
43 /bizfileintem...	1	78	78	78	0.00	0.00%	12.8/sec	70.26	5612.0
44 /bizfileintem...	1	35327	35327	35327	0.00	100.00%	1.7/min	0.07	2567.0
TOTAL	4	9480	78	35327	14945.65	25.00%	6.3/min	2.64	25805.2

# Test Elements : Assertions

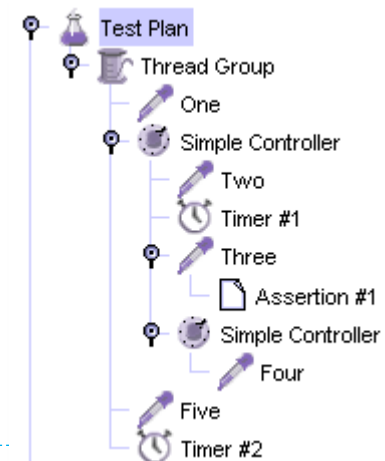
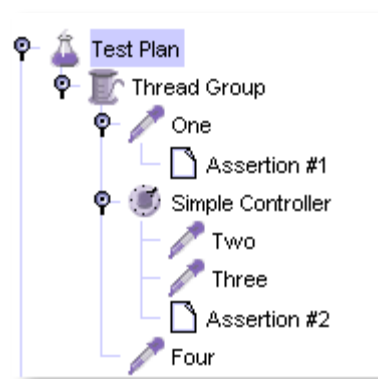
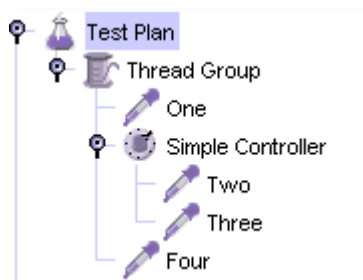
- **Assertions:**
  - Allow assert facts about responses received from the server being tested.
  - We can "test" that application is returning the results you expect it to.
  - *Example : We can assert that the response to a query will contain some particular text.*

The screenshot displays the Apache JMeter GUI. The 'Test Plan' tree on the left shows a 'Thread Group' containing a 'Load Simulator'. The 'Add' menu is open, and the 'Assertions' sub-menu is selected, listing various assertion types: BeanShell Assertion, BSF Assertion, Compare Assertion, Duration Assertion, HTML Assertion, JSR223 Assertion, MD5Hex Assertion, **Response Assertion** (highlighted), Size Assertion, SMIME Assertion, XML Assertion, XML Schema Assertion, and XPath Assertion. To the right, the 'Sample Summary Report' window is visible, showing a table of test results. The table has columns for 'Name', 'Comments', 'Apply to:', 'Response Field to Test', 'Pattern Matching Rules', and 'Patterns to Test'. The first row shows an exception: 'The following exception occurred: Server Exception during PPR'. The second row shows 'ADFC-12000: java.net.SocketException'. The third row shows 'java.io.IOException'. The fourth row contains the text: 'Because of inactivity, your session has timed out and is no longer active'.

Name	Comments	Apply to:	Response Field to Test	Pattern Matching Rules	Patterns to Test
The following exception occurred					
Server Exception during PPR					
ADFC-12000					
java.net.SocketException					
java.io.IOException					
Because of inactivity, your session has timed out and is no longer active					

# Execution Order & Scoping Rules

- **Execution Order:**
  1. Configuration elements
  2. Pre-Processors
  3. Timers
  4. Sampler
  5. Post-Processors
  6. Assertions
  7. Listeners
- **Scoping Rules:**
  - The JMeter test tree contains elements that are both hierarchical and ordered.
    - Some elements in the test trees are strictly hierarchical (Listeners, Config Elements, Post-Processors, Pre-Processors, Assertions, Timers), and
    - Some are primarily ordered (controllers, samplers).





# Recording JMeter – Test Plan

- Test Plans can be created from existing template.
- Use the menu **File > Templates...** or Templates icon

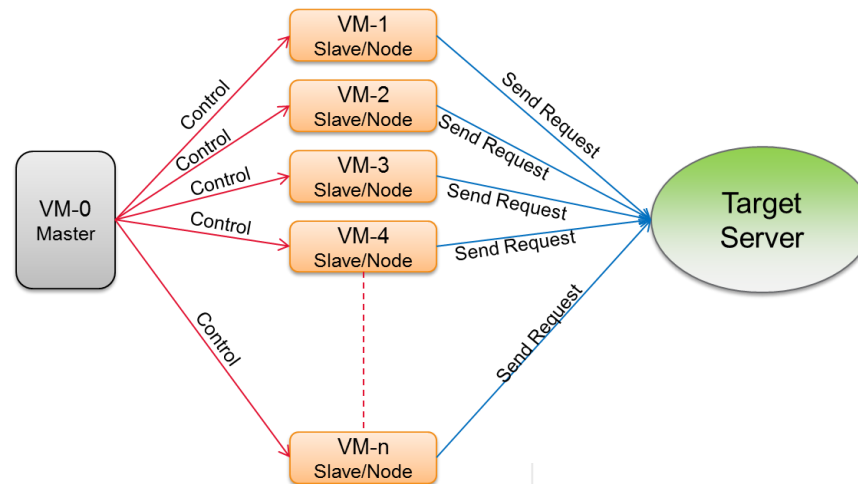
The screenshot displays the Apache JMeter (2.13.1) interface. The main window shows the 'Test Plan' configuration, including the 'Name' field set to 'Test Plan' and the 'Comments' field. The 'Test Plan' tree on the left includes 'User Defined Variables', 'HTTP Request Defaults', 'HTTP Cookie Manager', 'Thread Group', 'Recording Controller', 'View Results Tree', 'WorkBench', and 'HTTP(S) Test Script Recorder'. The 'HTTP(S) Test Script Recorder' is selected, showing its configuration: 'Name: HTTP(S) Test Script Recorder', 'Comments:', 'Global Settings', 'Port: 8888', 'Test plan controller', 'Target Controller', 'Grouping: Put', 'HTTP Sampler', 'Type:', 'Content-type file', 'Include:', 'URL Patterns to', and 'Notify Child Lis'. A red arrow points from the 'File > Templates...' menu path to the 'File' menu. Another red arrow points from the 'Review Workbench' label to the 'WorkBench' item in the tree. A third red arrow points from the 'Update proxy in browser' label to the 'Connection Settings' dialog box. The 'Connection Settings' dialog box is open, showing the 'Configure Proxies to Access the Internet' section. It has four radio buttons: 'No proxy', 'Auto-detect proxy settings for this network', 'Use system proxy settings', and 'Manual proxy configuration:'. The 'Manual proxy configuration' option is selected. Below it, there are fields for 'HTTP Proxy: localhost', 'Port: 8080', 'SSL Proxy:', 'Port: 0', 'FTP Proxy:', 'Port: 0', and 'SOCKS Host:', 'Port: 0'. There are also checkboxes for 'Use this proxy server for all protocols' and 'SOCKS v4' / 'SOCKS v5'. At the bottom, there is a 'No Proxy for:' field with 'localhost, 127.0.0.1' and an 'Automatic proxy configuration URL:' field. The 'OK', 'Cancel', and 'Help' buttons are at the bottom right.

Select File > Templates

Review Workbench

Update proxy in browser

- **Distributed Testing :** To distribute the load of JMeter across multiple slaves/nodes, run JMeter in server mode on the remote node(s), and then control the server(s) from the JMeter Master GUI.
- Code base for master and slave JMeter are same, except how they are started and the configurations defined



```
# Remote Hosts - comma delimited  
remote_hosts=192.168.0.10,192.168.0.11,192.168.0.12  
#remote_hosts=localhost:1099,localhost:2010
```

*IP addresses of each slave* →

- **Steps for distributed testing :**
  1. On **Slave** : Go to JMeter/bin and execute `jmeter-server.sh`
  2. On **master** systems: Go to /bin directory and edit file `jmeter.properties`, add IP of slave machine
  3. On the master machine, run JMeter GUI and open the test plan
  4. Click Run on the menu bar; select Remote start -> select the IP address of slave machine

# ***Best Practices***

- Always use latest version of JMeter
- Use the correct Number of Threads
- Where to Put the Cookie Manager
- Where to Put the Authorization Manager
- Using the HTTP(S) Test Script Recorder
- User variables
- Reducing resource requirements
- BeanShell server

*Demo*

# Thank you

Visit us at [www.techmahindra.com](http://www.techmahindra.com)

## Disclaimer

Tech Mahindra Limited, herein referred to as TechM provide a wide array of presentations and reports, with the contributions of various professionals. These presentations and reports are for informational purposes and private circulation only and do not constitute an offer to buy or sell any securities mentioned therein. They do not purport to be a complete description of the markets conditions or developments referred to in the material. While utmost care has been taken in preparing the above, we claim no responsibility for their accuracy. We shall not be liable for any direct or indirect losses arising from the use thereof and the viewers are requested to use the information contained herein at their own risk. These presentations and reports should not be reproduced, re-circulated, published in any media, website or otherwise, in any form or manner, in part or as a whole, without the express consent in writing of TechM or its subsidiaries. Any unauthorized use, disclosure or public dissemination of information contained herein is prohibited. Unless specifically noted, TechM is not responsible for the content of these presentations and/or the opinions of the presenters. Individual situations and local practices and standards may vary, so viewers and others utilizing information contained within a presentation are free to adopt differing standards and approaches as they see fit. You may not repackage or sell the presentation. Products and names mentioned in materials or presentations are the property of their respective owners and the mention of them does not constitute an endorsement by TechM. Information contained in a presentation hosted or promoted by TechM is provided “as is” without warranty of any kind, either expressed or implied, including any warranty of merchantability or fitness for a particular purpose. TechM assumes no liability or responsibility for the contents of a presentation or the opinions expressed by the presenters. All expressions of opinion are subject to change without notice.