

ZebraTester

Converting JMeter HTTP(S) Tests and SOAP/XML-RPC Tests to Apica ZebraTester

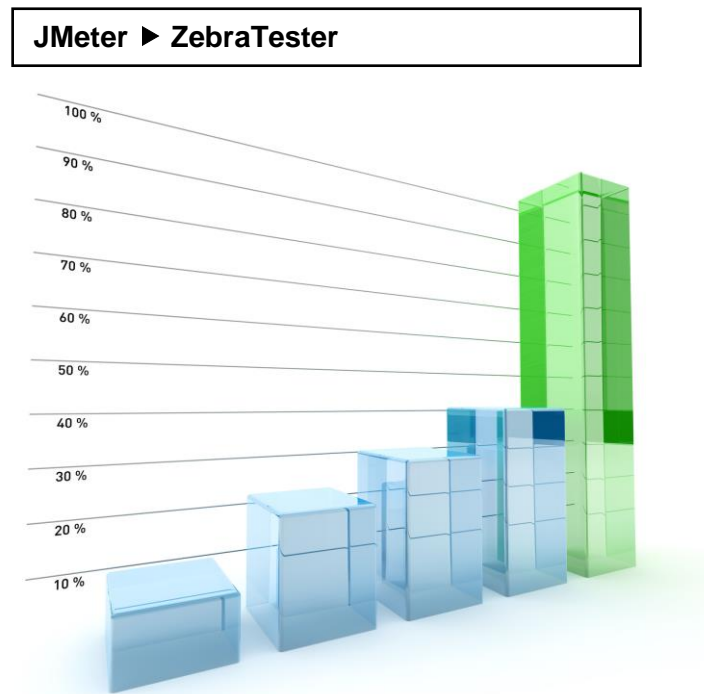


Table of Contents

1	Introduction	3
2	Basic Prerequisites	3
3	Converting a JMeter Thread Group to a ZebraTester Web Session	4
3.1	Preparing a JMeter Thread Group for Recording	4
3.1.1	Configure the Runtime Behavior of the Thread Group.....	4
3.1.2	Configure the Proxy Server	4
3.1.3	Insert Page Breaks between the HTTP Requests	5
3.2	Running the JMeter Thread Group and Recording the ZebraTester Web Session	8
3.3	Post-Processing of the Recorded Web Session	11
3.4	Generating the ZebraTester Load Test Program and Running the Converted Test	12
4	Manufacturer, Sales and Support.....	15

1 Introduction

The best way to **convert** Apache **JMeter** tests to **Apica ZebraTester** is to run the JMeter tests via the Proxy Server of Apica ZebraTester which effects that all of the **JMeter HTTP Requests** are recorded once again by ZebraTester.

The procedure of conversion is relatively simple. However, the handling of dynamically exchanged session parameters that are extracted from HTTP responses and then assigned to succeeding HTTP requests is lost by such a conversion. This means that such data or parameters must be manually postprocessed in the ZebraTester GUI.

Session Cookies are automatically supported by ZebraTester without the need of any configuration.

Note: JMeter tests containing **SOAP/XML-RPC Requests** can also converted to ZebraTester in the same manner as described in this document.

2 Basic Prerequisites

1) Install Apica ZebraTester on the same (local) machine as JMeter is already installed.

Please follow the instructions in the ZebraTester Installation Guide and don't miss to configure the Popup-Window-Blocker.
After installation, start the "ZebraTester Console".

2.) Copy the file **InsertPrxPageBreak.jar** into the JMeter installation directory **/lib/junit** .

You can get this file from the **ZebraTester installation directory** → **ConvertFromJMeter** sub-directory, or you can download this file from <http://www.proxy-sniffer.com/download/InsertPrxPageBreak.jar> .

Then restart JMeter.

3 Converting a JMeter Thread Group to a ZebraTester Web Session

3.1 Preparing a JMeter Thread Group for Recording

3.1.1 Configure the Runtime Behavior of the Thread Group

Modify the **JMeter Thread Group** in such a way that each HTTP Request is executed only once. Set

- Number of Threads (users): 1
- Loop Count: 1 (and uncheck the option "Forever")

3.1.2 Configure the Proxy Server

Configure all **JMeter HTTP Requests** in such a way that they are executed via a **Proxy Server**.

If the **JMeter Thread Group** contains a **Config Element → HTTP Request Default** then configure the following values in the "HTTP Request Default" element:

Proxy Server: Server Name or IP: **127.0.0.1**

Proxy Server: Port Number: **7997**

Thread Group

Name: Test A - One User - One Loop

Comments:

Action to be taken after a Sampler error

Continue

Thread Properties

Number of Threads (users): 1

Ramp-Up Period (in seconds): 1

Loop Count: Forever 1

Delay Thread creation until needed

Scheduler

Proxy Server

Server Name or IP: 127.0.0.1 Port Number: 7997 Username Password

Optional Tasks

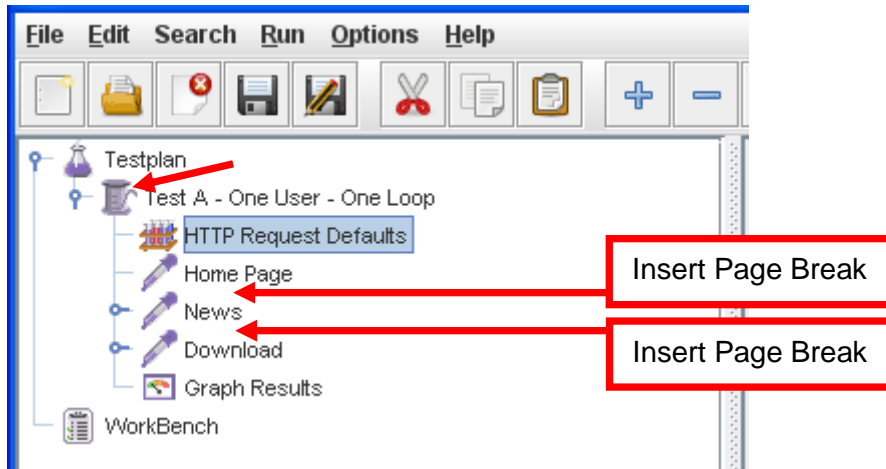
Retrieve All Embedded Resources from HTML Files Use concurrent pool. Size: 4

Alternatively, if the JMeter Thread Group doesn't contain a "HTTP Request Default" element you have to modify all **JMeter HTTP Requests** separately.

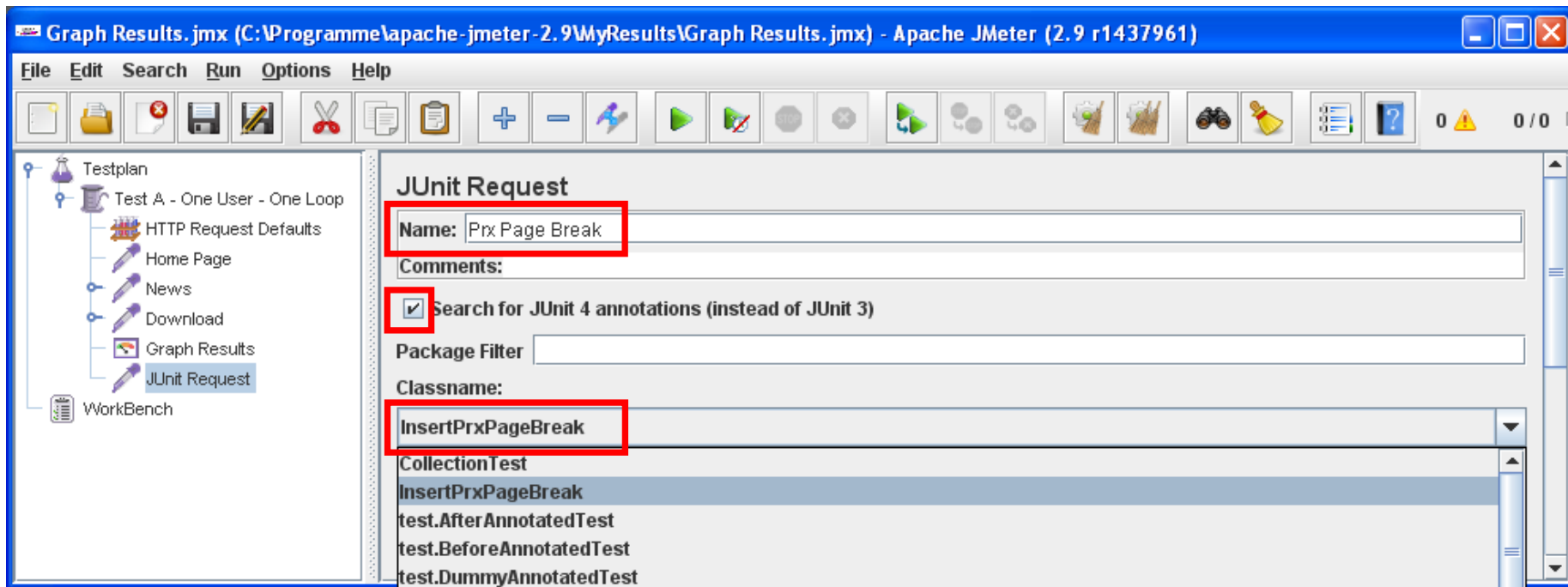
Note: If the JMeter test already uses an outbound Proxy Server configure nevertheless the new settings (127.0.0.1:7997) as shown above, but configure additionally the (other/replaced) Proxy Server in the **ZebraTester GUI** by clicking in the **Main Menu** on the **Personal Settings** icon.

3.1.3 Insert Page Breaks between the HTTP Requests

Insert Page Breaks **between** the **JMeter HTTP Requests**. The reason for this is that the proxy server of ZebraTester cannot recognize when a Web page starts, and when it finishes. The proxy server only sees single URL calls, such as requests for HTML data or image files. Adding page breaks manually is necessary in order to convert the test properly.



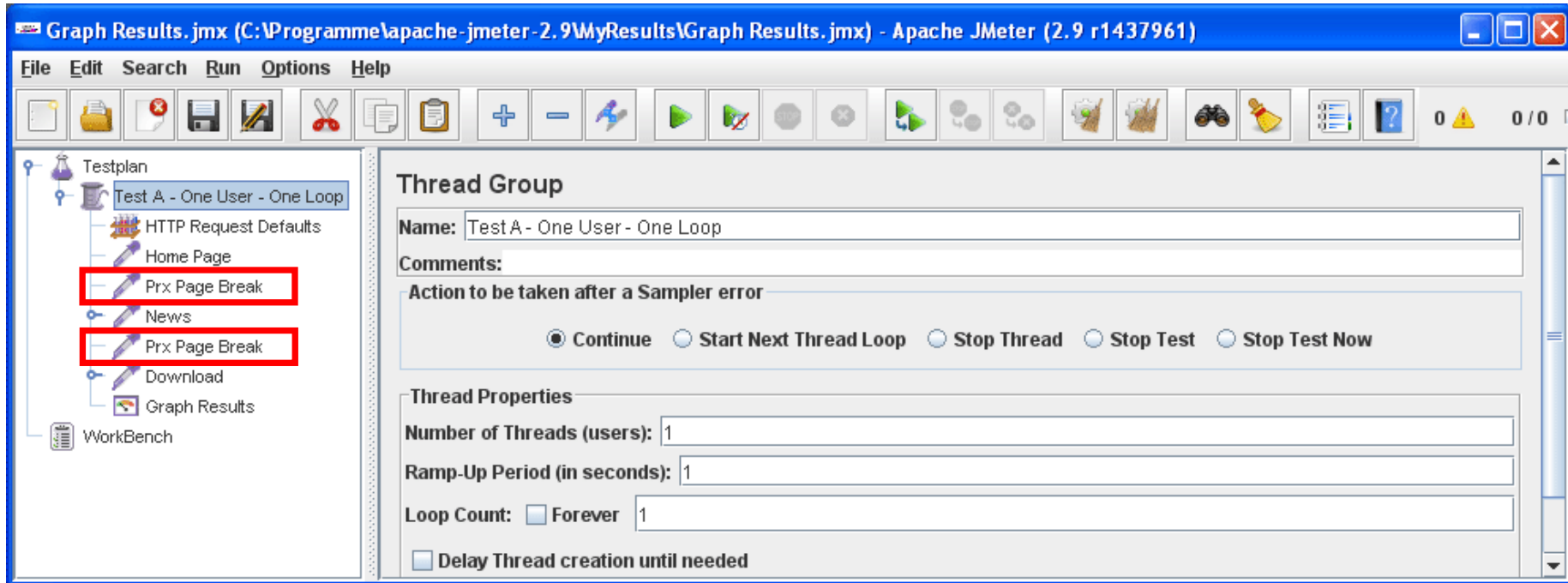
To insert a ZebraTester Page Break in JMeter click using the right mouse button on **Thread Group → Add → Sampler → JUnit Request**



Configure the JUnit Request as follows:

- Name: Prx Page Break
- Enable the Checkbox "Search for JUnit 4 annotations"
- Select the Classname "InsertPrxPageBreak"

The example of a Thread Group shown in the image above contains three JMeter HTTP Requests. This means that two Page Breaks are needed and that the first Page Break (JUnit Request) should be moved between "Home Page" and "News". After that the Page Break can be duplicated and the second Page Break should be placed between "News" and "Download".

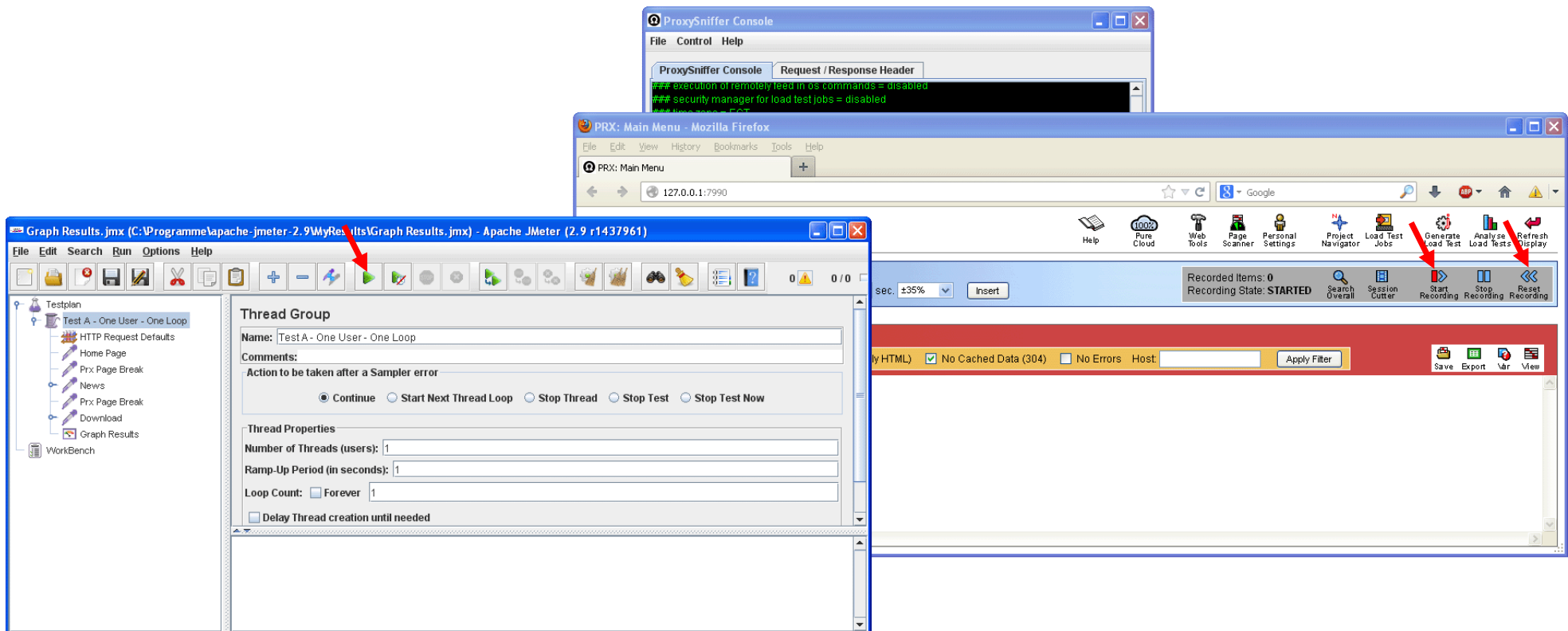


All preparations are now complete.

3.2 Running the JMeter Thread Group and Recording the ZebraTester Web Session

Proceed as follows:

1. Ensure that the "ZebraTester Console" is started.
2. Enter <http://127.0.0.1:7990> into a Web browser to access the ZebraTester GUI. If you see already some recorded requests click first on the "Reset Recording" icon in the ZebraTester GUI. Then click on "**Start Recording**" icon in the ZebraTester GUI.
3. Select the Thread Group in JMeter and click on the **Start** icon in **JMeter**.



Wait until the JMeter test is completed. Then click on the "Stop Recording" icon in the ZebraTester GUI:

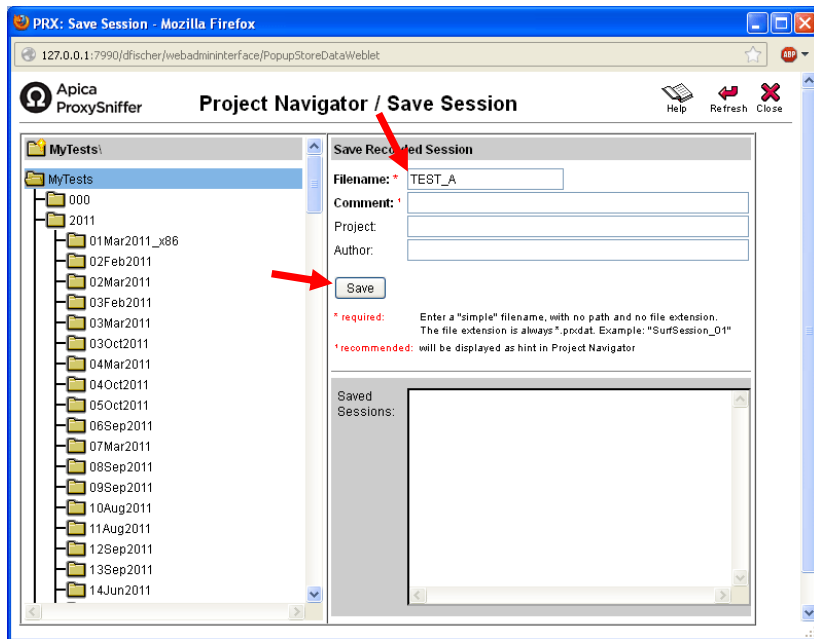
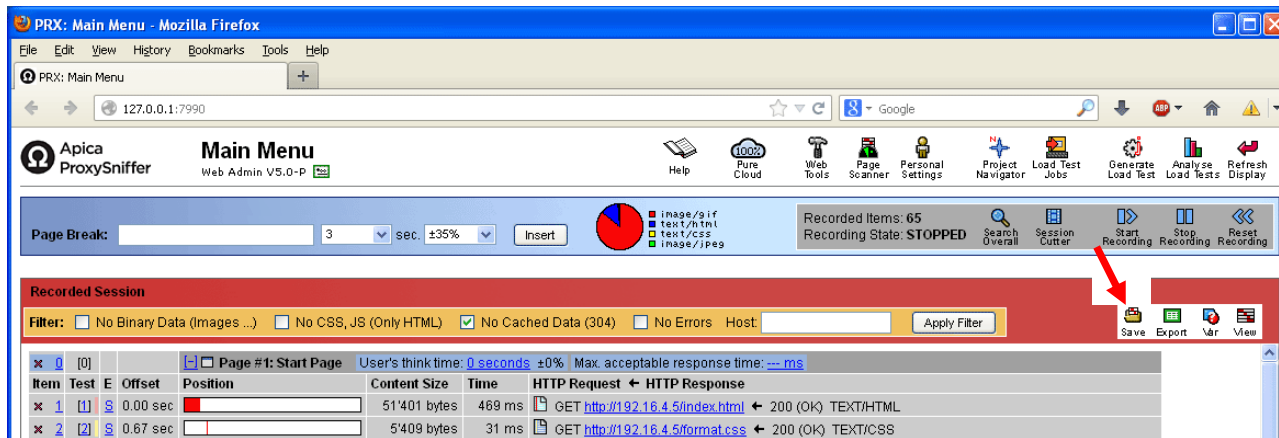
The screenshot shows the ZebraTester GUI with a recorded session of a web application. The interface includes a menu bar, a toolbar with various icons, and a main display area showing a list of recorded requests and responses. A red arrow points to the 'Stop Recording' button in the toolbar.

Recorded Session Summary:

- Recorded Items: 65
- Recording State: STOPPED
- Page Break: 3 sec. ±35%
- Filter: No Binary Data (Images ...), No CSS, JS (Only HTML), No Cached Data (304), No Errors. Host: [empty]

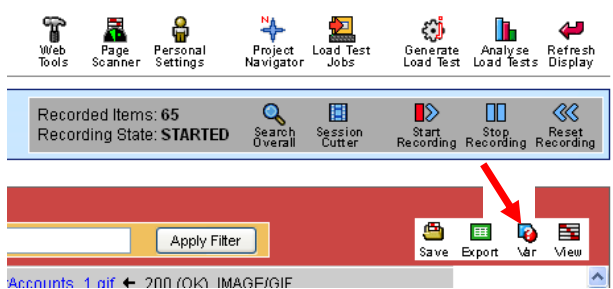
Item	Test	E	Offset	Position	Content Size	Time	HTTP Request	HTTP Response	
x 1	[1]	S	0.00 sec		51401 bytes	469 ms	GET http://192.16.4.5/index.html	← 200 (OK) TEXT/HTML	
x 2	[2]	S	0.67 sec		5409 bytes	31 ms	GET http://192.16.4.5/format.css	← 200 (OK) TEXT/CSS	
x 3	[3]	S	0.70 sec		5608 bytes	16 ms	GET http://192.16.4.5/ProxySnifferLogo.gif	← 200 (OK) IMAGE/GIF	
x 4	[4]	S	0.72 sec		43 bytes	16 ms	GET http://192.16.4.5/XXXXXXXX.gif	← 200 (OK) IMAGE/GIF	
x 5	[5]	S	0.74 sec		174 bytes	31 ms	GET http://192.16.4.5/MagGerman.gif	← 200 (OK) IMAGE/GIF	
x 6	[6]	S	0.77 sec		1'220 bytes	281 ms	GET http://192.16.4.5/MagEngland.gif	← 200 (OK) IMAGE/GIF	
x 7	[7]	S	1.06 sec		3'031 bytes	1'078 ms	GET http://192.16.4.5/logo.gif	← 200 (OK) IMAGE/GIF	
x 8	[8]	S	2.14 sec		1'750 bytes	31 ms	GET http://192.16.4.5/slogan.gif	← 200 (OK) IMAGE/GIF	
x 9	[9]	S	2.17 sec		88 bytes	953 ms	GET http://192.16.4.5/arrow_red_12x9.gif	← 200 (OK) IMAGE/GIF	
x 10	[10]	S	3.12 sec		48'980 bytes	32 ms	GET http://192.16.4.5/ProxySnifferGeneralArchitectureLg_2.gif	← 200 (OK) IMAGE/GIF	
x 11	[11]	S	3.17 sec		6'984 bytes	31 ms	GET http://192.16.4.5/noScripting.gif	← 200 (OK) IMAGE/GIF	
x 12	[12]	S	3.20 sec		2'703 bytes	32 ms	GET http://192.16.4.5/downloadRed.gif	← 200 (OK) IMAGE/GIF	
x 13	[13]	S	3.23 sec		2'720 bytes	937 ms	GET http://192.16.4.5/Button_PDF.gif	← 200 (OK) IMAGE/GIF	
x 14	[14]	S	4.17 sec		3'384 bytes	31 ms	GET http://192.16.4.5/PadSmall.gif	← 200 (OK) IMAGE/GIF	
x 15	[15]	S	4.20 sec		4'405 bytes	16 ms	GET http://192.16.4.5/ApicaWebPerformance.gif	← 200 (OK) IMAGE/GIF	
x 16	[16]	S	4.22 sec		68 bytes	16 ms	GET http://192.16.4.5/navArrowUp.gif	← 200 (OK) IMAGE/GIF	
x 17	[17]	S	4.24 sec		18'852 bytes	47 ms	GET http://192.16.4.5/Keywords.gif	← 200 (OK) IMAGE/GIF	
x 18	[18]	S	4.28 sec		44'609 bytes	31 ms	GET http://192.16.4.5/MainMenu.gif	← 200 (OK) IMAGE/GIF	
x 19	[19]	S	4.31 sec		73 bytes	31 ms	GET http://192.16.4.5/smallInfoTransp.gif	← 200 (OK) IMAGE/GIF	
x 20	[20]	S	4.34 sec		38'021 bytes	16 ms	GET http://192.16.4.5/levels_en.jpg	← 200 (OK) IMAGE/JPEG	
x 21	[21]	S	4.38 sec		77'089 bytes	938 ms	GET http://192.16.4.5/realTime2.gif	← 200 (OK) IMAGE/GIF	
x 22	[22]	S	5.31 sec		10'642 bytes	47 ms	GET http://192.16.4.5/In-DepthMeasurements.gif	← 200 (OK) IMAGE/GIF	
x 23	[23]	S	5.36 sec		5'510 bytes	31 ms	GET http://192.16.4.5/TimeProfileOfErrorTypes.gif	← 200 (OK) IMAGE/GIF	
x 24	[24]	S	5.39 sec		17'715 bytes	31 ms	GET http://192.16.4.5/ComparisonBeforeAndAfterTuning.gif	← 200 (OK) IMAGE/GIF	
x 25	[25]	S	5.42 sec		33'994 bytes	31 ms	GET http://192.16.4.5/InnerLoopControlPerUser_450x271.gif	← 200 (OK) IMAGE/GIF	
x 26	[26]	S	5.46 sec		45'807 bytes	32 ms	GET http://192.16.4.5/Cluster.gif	← 200 (OK) IMAGE/GIF	
Total:					5.49 sec	430'280 bytes	26 Requests	78.45 kbytes/sec	
x 27	[27]								
Page #2:					User's think time: 3 seconds ±35%	Max. acceptable response time: --- ms			
x 28	[28]	S	0.00 sec		49'830 bytes	407 ms	GET http://192.16.4.5/news_en.html	← 200 (OK) TEXT/HTML	

All HTTP(S) Requests are now recorded by ZebraTester. It's recommended that you save the recording by giving it almost a similar name as the JMeter test name.



3.3 Post-Processing of the Recorded Web Session

You can call the ZebraTester "Var Finder" menu to check if post-processing of dynamically exchanged session parameters is required:



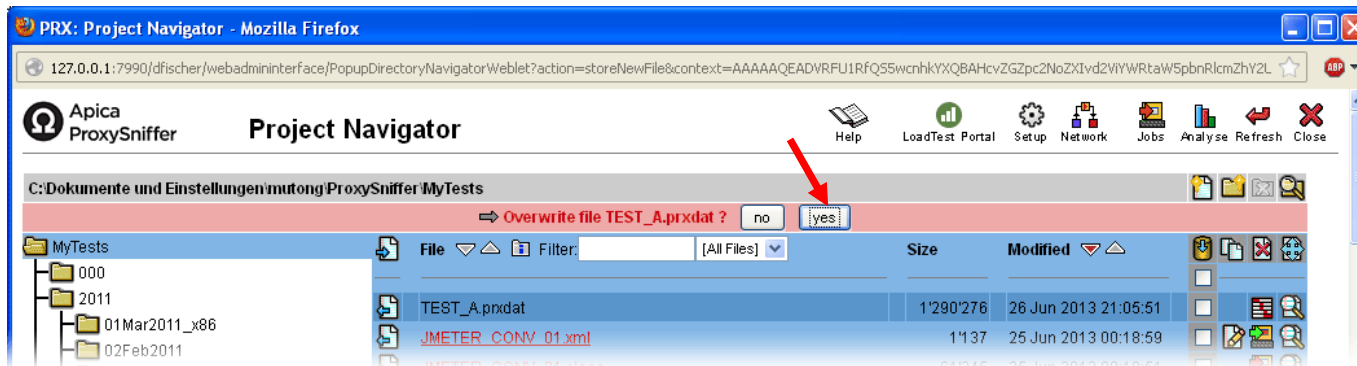
Further information is provided in the **manual** about **Handling of "Dynamically-Exchanged Session Parameters"**.

There are many other options that can be configured in the ZebraTester GUI for the recorded Web session, for example:

- Login with different user accounts into a Web application, by reading the usernames and passwords from a data file (input file).
- Define (inner-)loops around Web pages and define conditions when to continue or break a inner loop.
- Modify the automatically applied response content check of the HTTP responses.

Please take a look at the **ZebraTester User's Guide** to learn more about using the extensive product features.

If you have modified the recorded Web session you should save it again. Don't miss to confirm that the old file should be overwritten:



3.4 Generating the ZebraTester Load Test Program and Running the Converted Test

If you have reached this step the conversion from JMeter to ZebraTester is completed. The next two final steps are the generation of the ZebraTester load test program and its execution as a load test on a load generator – or on a cluster of load generators, as described in the ZebraTester User's Guide.

The screenshot displays the 'Generate HTTP(S) Load Test Program' web interface. The main content area includes the following configuration options:

- Load Test Program** - 65 Items selected: 3 Pages - 62 HTTP/S Requests/Responses
- Java™ Classname:** * **Java™ Code Model:** large
- Content Test Algorithm:** [+ apply (heuristic) methods from recorded session to check received content
- Character Encoding:** ISO-8859-1
- Generate External Files for XML and SOAP Request Data:** > 4096 Bytes
- * required:** enter a "simple" classname for the load test program, with no path and no file extension.

The **HTTP Protocol Options** section includes:

- HTTP Protocol Version:** 1.1
- Allow Keep-Alive:**
- Strip Referer Header Field:**
- Strip Accept Header Field to /*:**
- Load Test over HTTP(S) Proxy:** Apply next proxy configuration from [personal settings](#)

The **HTTP / SSL Authentication Options** section includes:

- Basic Authentication:** Apply individual Basic Authentication per user from input file (basicauth.bt)
- Digest Authentication:** Apply individual Digest Authentication per user from input file (digestauth.bt)
 use common Username: Password:
- NTLM Authentication:** use common NTLM account from Personal Settings menu
- Kerberos Authentication:** use common Kerberos account from Personal Settings menu
- HTTPS Client Certificates:** apply individual PKCS#12 certificate per user from input file (pkcs12auth.bt)

The sidebar on the left shows the **URL Execution** section with the following statistics:

Serial Executed	Parallel Executed	Threads p. User
3	59	6

Buttons for **Switch to Serial Exec.** and **Switch to Parallel Exec.** are visible. A red arrow points to the **Switch to Parallel Exec.** button. At the bottom of the main area, there is a **Program Description:** field and a **Continue** button. A note below the button states: *** recommended: will be displayed as hint in Project Navigator**.

Hint: it's recommended that you enable Parallel URL Execution when generating the load test program.

Then click on the **Continue** button at the bottom of the window.

The screenshot shows the Apica ProxySniffer web interface in a Mozilla Firefox browser window. The title bar reads "PRX: Generate HTTP(S) Load Test Program - Mozilla Firefox". The address bar shows the URL: "127.0.0.1:7990/dfischer/webadmininterface/PopupCreateLoadtestWeblet?action=createProgram&classname=TEST_A&codeStructureModel=2&selectContentTes". The page title is "Generate HTTP(S) Load Test Program".

On the left side, there is a sidebar with the Apica logo and "Excellence in Cloud Performance". Below the logo, it says "Proxy Sniffer V5.0-P". A section titled "URL Execution" contains a table:

URL Execution	
Serial Executed	3
Parallel Executed	59
Threads p. User	6

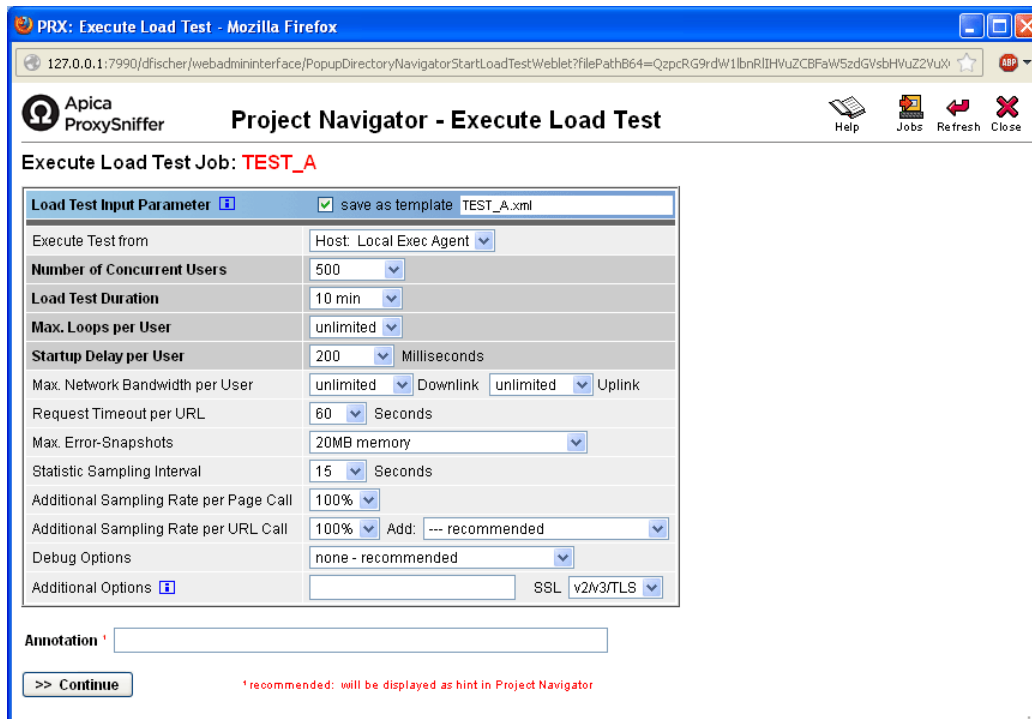
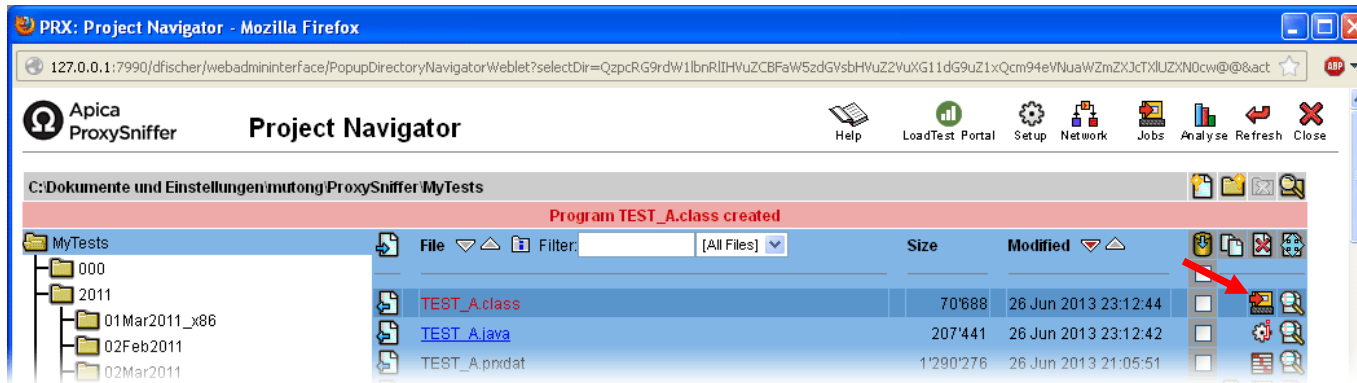
Below the table are two buttons: "Switch to Serial Exec." and "Switch to Parallel Exec.". A red arrow points from the bottom of this sidebar to the "Save Load Test Program" button.

The main content area has a yellow banner that says "Load Test Program TEST_A.java ready to save." Below this is a file tree view under "MyTests" containing a folder "2011" with subfolders for dates from "01Mar2011_x86" to "12Sep2011". To the right of the file tree is a "Response Verification Summary" section with three entries:

- Page #1: Start Page (1) - "a clear improvement of the test quality is achieved"
- Page #2: (28) - "Further information is available in the revised"
- Page #3: (53) - "the handbook of generating load from the cloud"

At the bottom of the main area, there is a "Save Load Test Program" button and a checked checkbox for "Overwrite & Compile".

The load test program is now ready for execution:



4 Manufacturer, Sales and Support

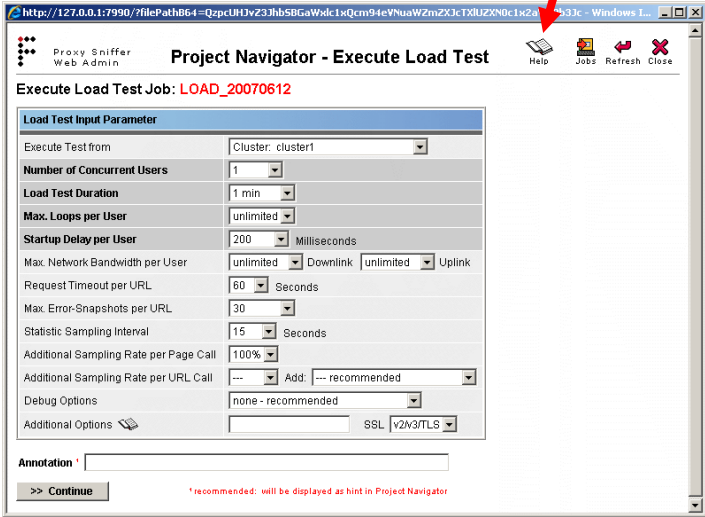
Ingenieurbüro David Fischer AG, Switzerland | A company of the [Apica Group](#)

Manufacturer's Web Site: www.zebratester.com

Support: support@apicasystem.com

Sales: sales@apicasystem.com

Note: All menus provide *context specific help text*, available using the Help Icon:



The screenshot shows the 'Project Navigator - Execute Load Test' dialog box. The title bar includes the URL 'http://127.0.0.1:7990/FilePathB64=QzpcUHVzZ3JhbSBGaWdc1xQcm94eVNuZWZmZDZjCTXlUZXR0c1x2a...' and the window title 'Project Navigator - Execute Load Test'. The dialog contains a 'Load Test Input Parameter' section with the following settings:

Parameter	Value
Execute Test from	Cluster: cluster1
Number of Concurrent Users	1
Load Test Duration	1 min
Max. Loops per User	unlimited
Startup Delay per User	200 Milliseconds
Max. Network Bandwidth per User	unlimited Downlink unlimited Uplink
Request Timeout per URL	60 Seconds
Max. Error-Snapshots per URL	30
Statistic Sampling Interval	15 Seconds
Additional Sampling Rate per Page Call	100%
Additional Sampling Rate per URL Call	--- Add: --- recommended
Debug Options	none - recommended
Additional Options	SSL v2/v3/TLS

At the bottom, there is an 'Annotation' field and a '>> Continue' button. A red arrow points to the 'Help' icon in the top right corner of the dialog.