



IBM Rational Software Conference 2009
As Real as It Gets!



IBM Rational Software Conference 2009

Know more about IBM Rational Performance Tester 8.1

Snehamoy Karmakar
Associate IT Architect

Rational software

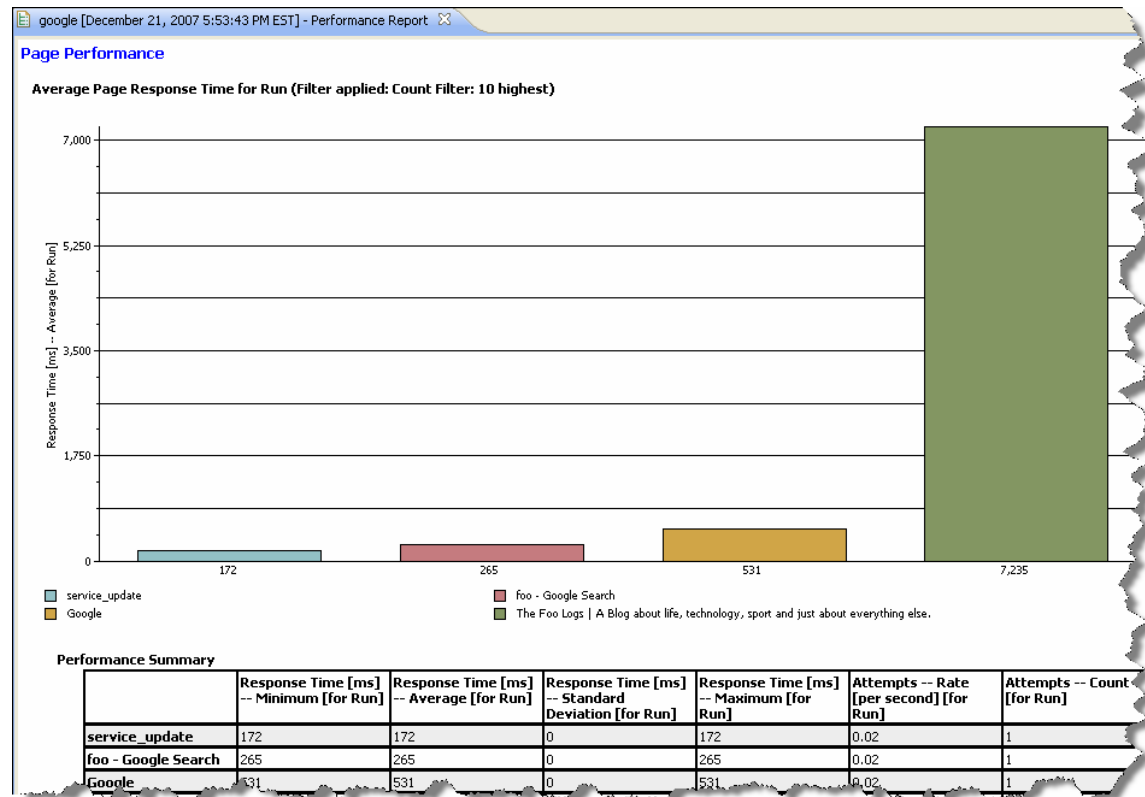
Agenda

- Let me know ...
 - What is IBM Rational Performance Tester first
- Hhmm...Do you have a list of features !!!
- Great... is there more ?
- How about some tips and techniques.
- Any Guidance to our team ?
- I have few questions.



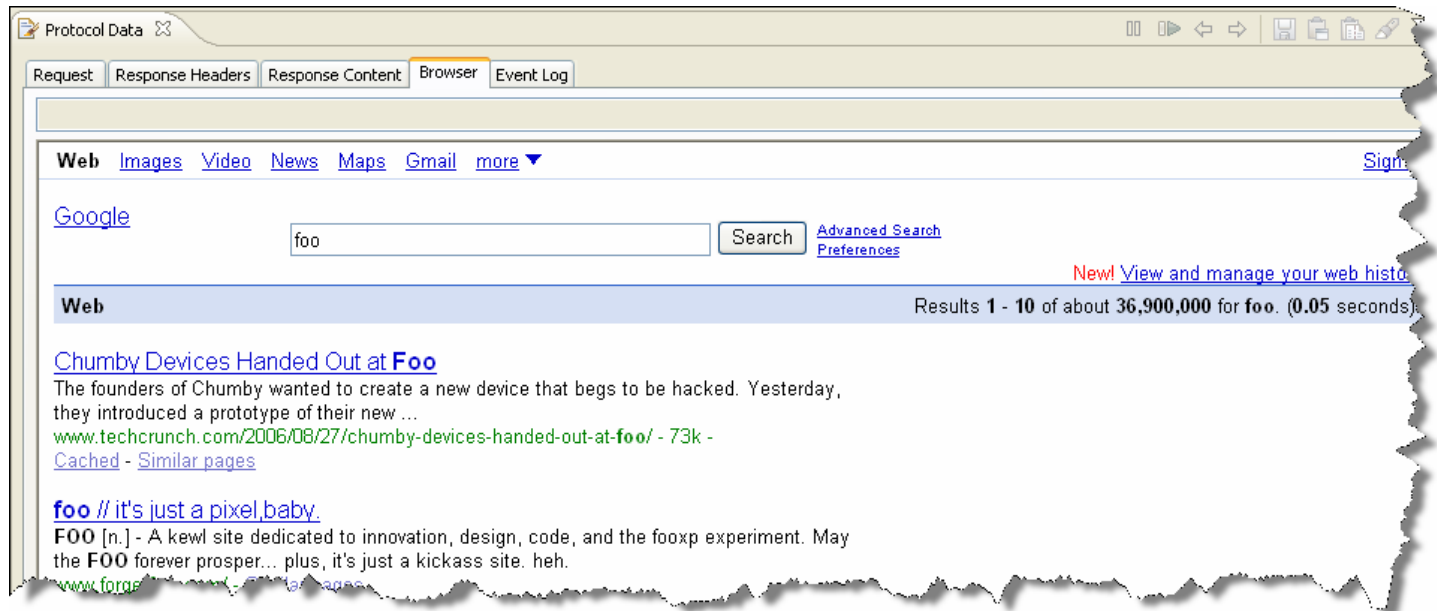
Overview: Rational Performance Tester (RPT)

- RPT is a performance test creation, execution and analysis tool that helps teams validate the scalability and reliability of their web-based applications



Overview: Rational Performance Tester (RPT)

- RPT works with most web applications based on the HTTP protocol, and also supports numerous other application frameworks via protocol extensions:
 - SOA/WebServices
 - SAP ®
 - SIP
 - Siebel ®
 - Citrix ®
 - Socket
- RPT also provides a protocol SDK to allow partners and customers to build support for other protocols.



Overview: Rational Performance Tester (RPT)

- Can be used to accurately and scalably simulate tens of thousands of “virtual users” to verify application performance.
- Can also be used to perform functional verification of web / protocol based applications.
- Combines optimal access to underlying protocol data and remote system statistics with the ability to insert custom Java code, enabling experts to create highly advanced test scenarios.

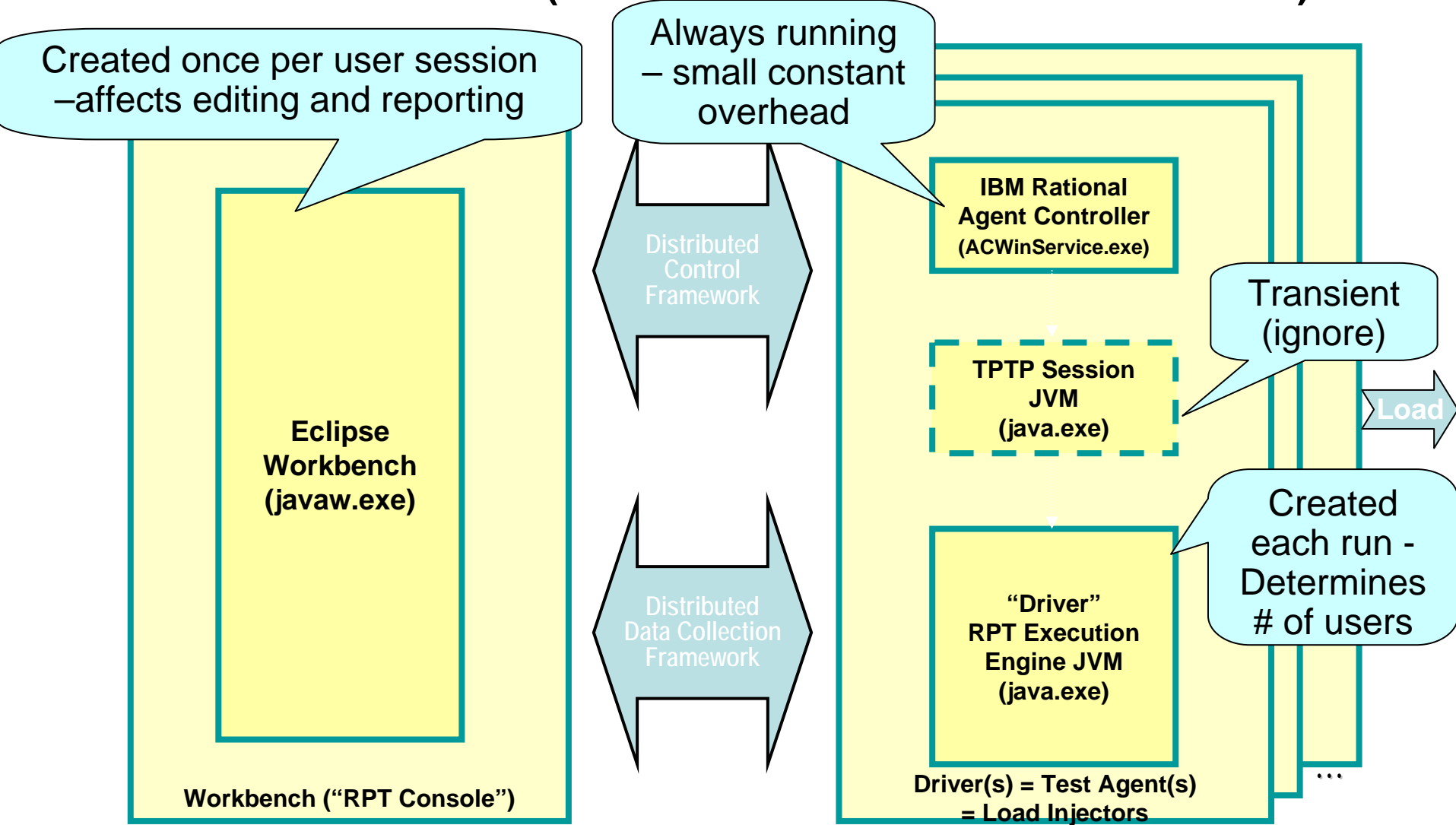
The screenshot displays the Rational Performance Tester (RPT) interface for a performance test named "plants". The interface is divided into several sections:

- Test Contents:** A tree view showing the test structure. The selected element is "9.37.207.17:9080/PlantsByWebSphere/servlet/ShoppingServlet". Below the tree are navigation buttons: Add, Insert, Remove, Up, Down, Prev, and Next.
- Test Element Details:** Shows the selected test element: "9.37.207.17:9080/PlantsByWebSphere/servlet/ShoppingServlet".
- Request Attributes:** Fields for Version (1.1), Method (POST), Host (9.37.207.17:9080), and URL (/PlantsByWebSphere/servlet/ShoppingServlet).
- Data:** A text area containing the request body: "qty=1&itemID=A0011&itemQuantity=1&action=addtocart&submit.x=74&submit.y=23".
- Request Headers:** A table listing headers and their values.

Header Name	Value
Accept	image/gif, image/x- bitmap, image/jpeg,
Referer	http://9.37.207.17:9080/PlantsByWebSp
Accept-Language	en-us



RPT Process Architecture (Machine and Process/JVM boundaries)



Agenda

- Let me know What is IBM Rational Performance Tester first
- Hmm...Do you have a list of features !!!
- Great... is there more ?
- How about some tips and techniques.
- Any Guidance to our team ?
- I have few questions.



Agenda

- Let me know What is IBM Rational Performance Tester first
- Hmm...Do you have a list of features !!!
- Great... is there more ?
- How about some tips and techniques.
- Any Guidance to our team ?
- I have few questions.



What's new

- **Enhanced RQM integration**
- **Automated Performance Requirements Analysis**
- **Conditional color palettes of Reports.**

- **Resource monitoring enhancements**
 - ▶ JBoss Application Server monitoring
 - ▶ IBM WebSphere PMI monitoring
 - ▶ Aggregated resource monitoring data

- **Enhanced Percentile response analysis**
 - ▶ Real-time percentile response analysis
 - ▶ Not log dependant
 - ▶ Available for all protocols

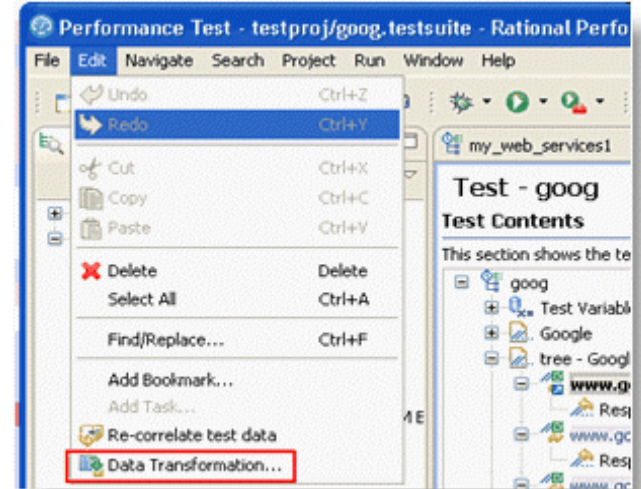
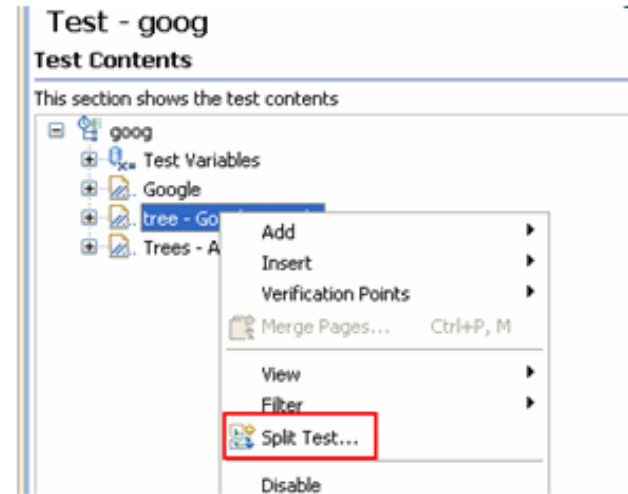
- **HTTP Browser rendering at schedule playback**
- **“Split Test”**
- **Data transforms**
- **Recording annotations**



Rational Performance Tester

Simplifying the test development experience

- Test Modularity with test variable support
- Binary Editor view supporting UTF8, EBCDIC, GB 18030
- Replay of http live-browser view
- Custom/Conditional color pallets for reports



Rational Performance Tester

Encoded Data support for Web 2.0 applications

The typical scenario ...



- Encoding prevents the user from finding or modifying test data points
- Users must create their own code and methods for dealing with obscured data (such as binary data)

With IBM Rational Performance Tester ...



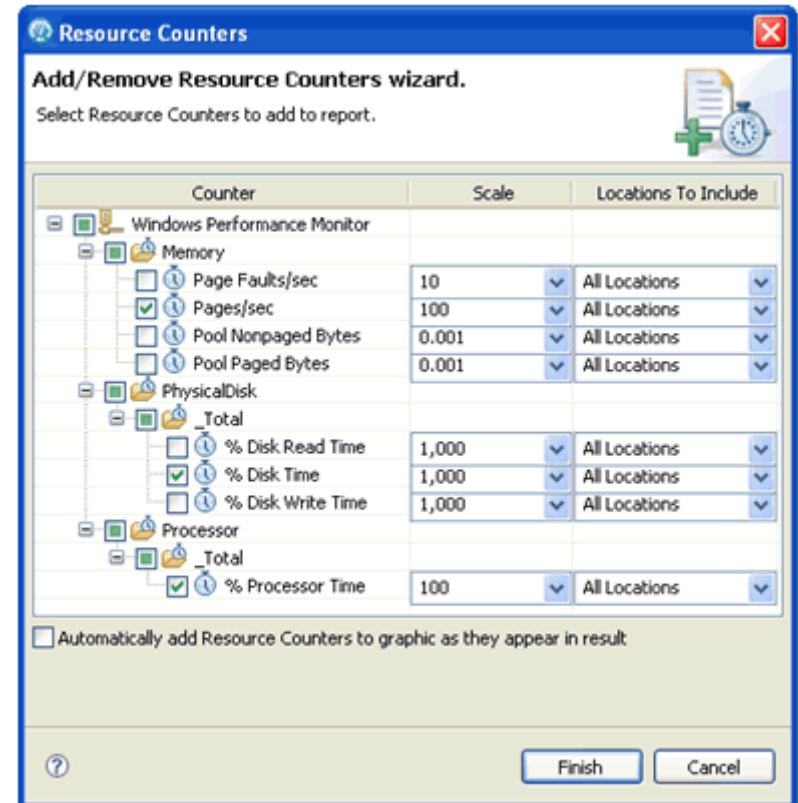
- Data transformations allows user to “see” into encoded data for editing, data variation, and data correlation
- Built-in transformations for Java Serialized Objects and Binary data
- Extensible to accommodate custom formats



Rational Performance Tester

Accelerate problem identification with agent-less resource monitoring

- **Agent-less monitoring for**
 - ***NEW*** WebSphere and JBoss application servers
 - Windows Perfmon
 - Unix rstatd
 - IBM Tivoli Monitoring
- **Aggregated (per-run) counters for resource monitoring**
- **Overlay counters on performance reports**



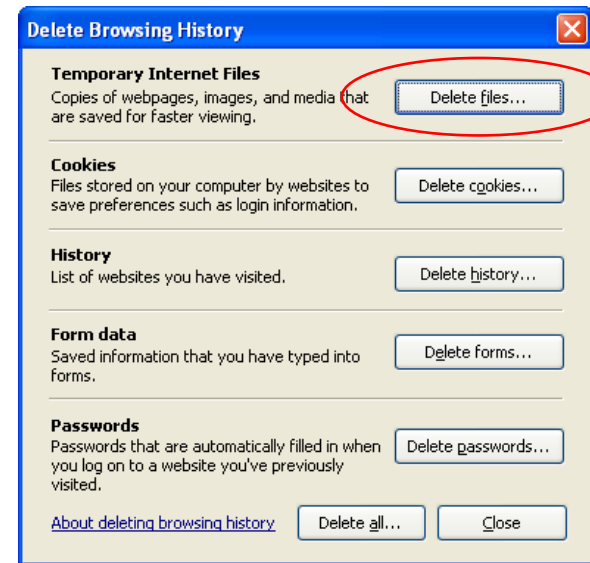
Agenda

- Let me know What is IBM Rational Performance Tester first
- Hmm...Do you have a list of features !!!
- Great... is there more ?
- How about some tips and techniques.
- Any Guidance to our team ?
- I have few questions.



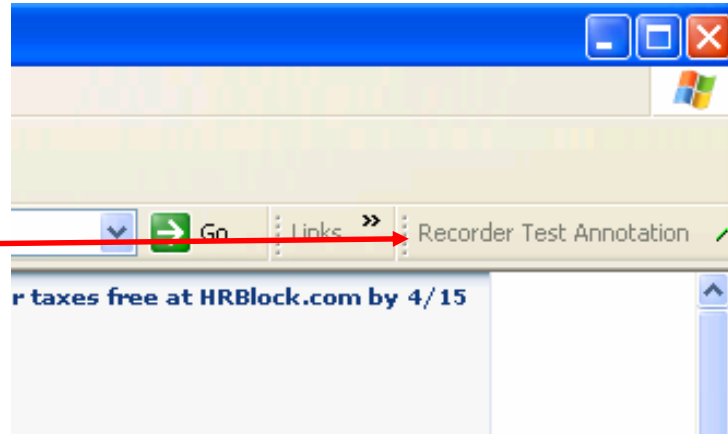
Creating a test

- Clear browser cache
- Pause between pages (5 seconds)

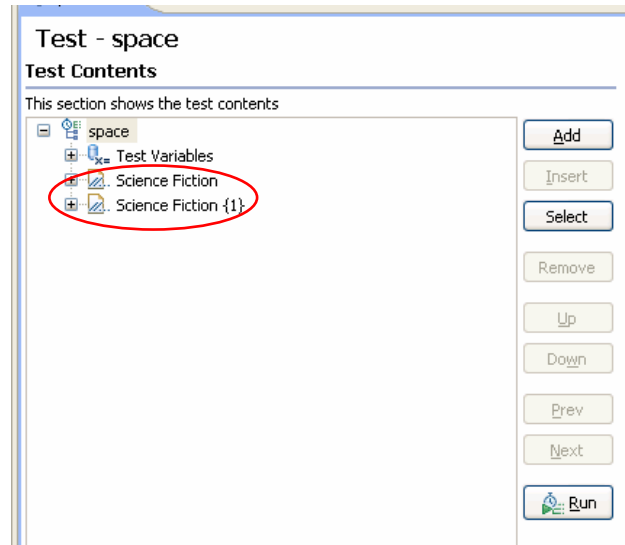


Creating a test

- Use comments

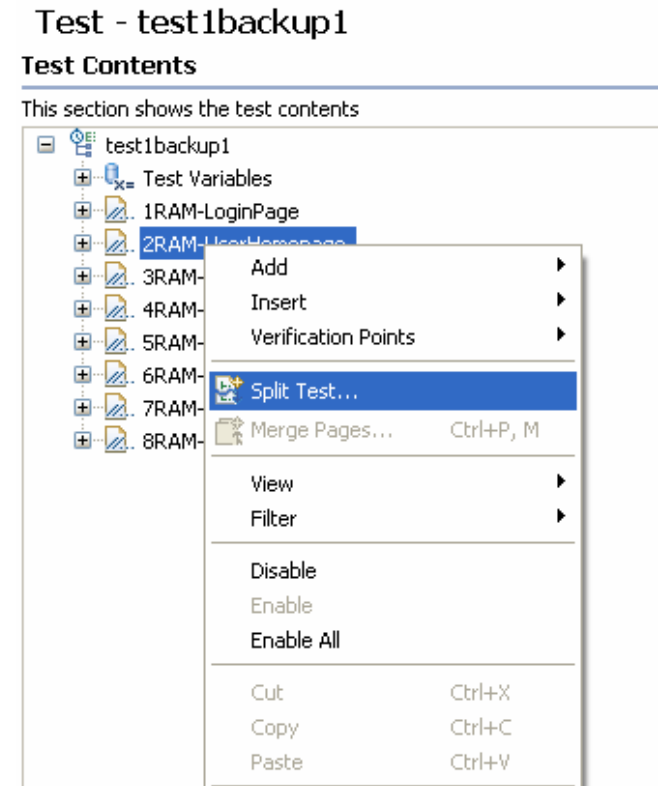


- Rename pages



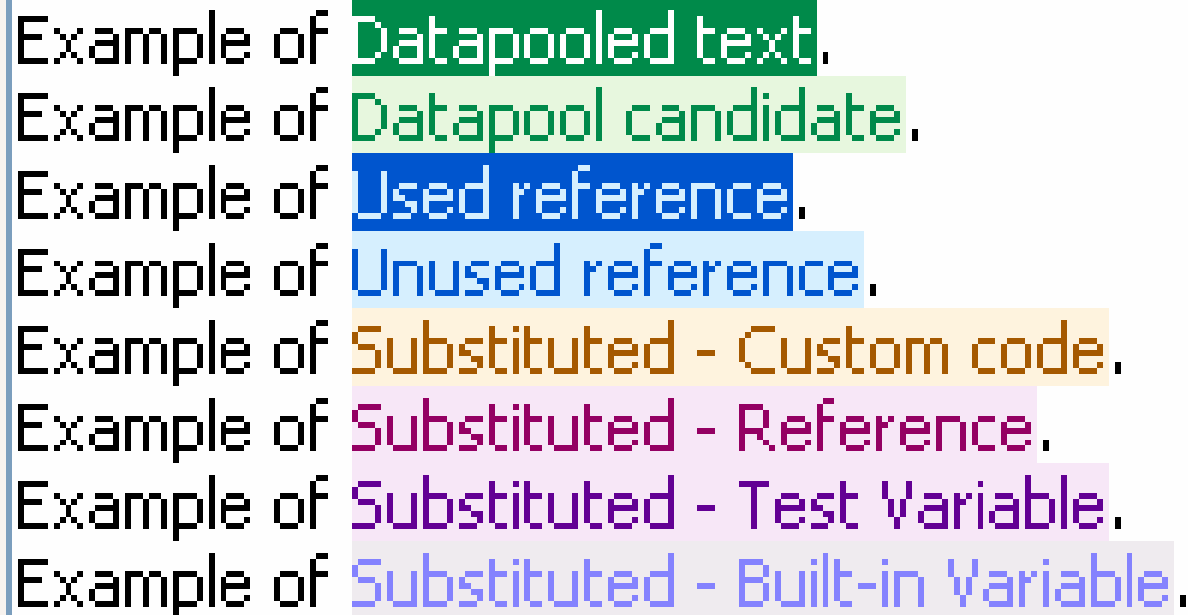
Split test

- Make multiple tests from one
- Separate login from work to repeat



Color

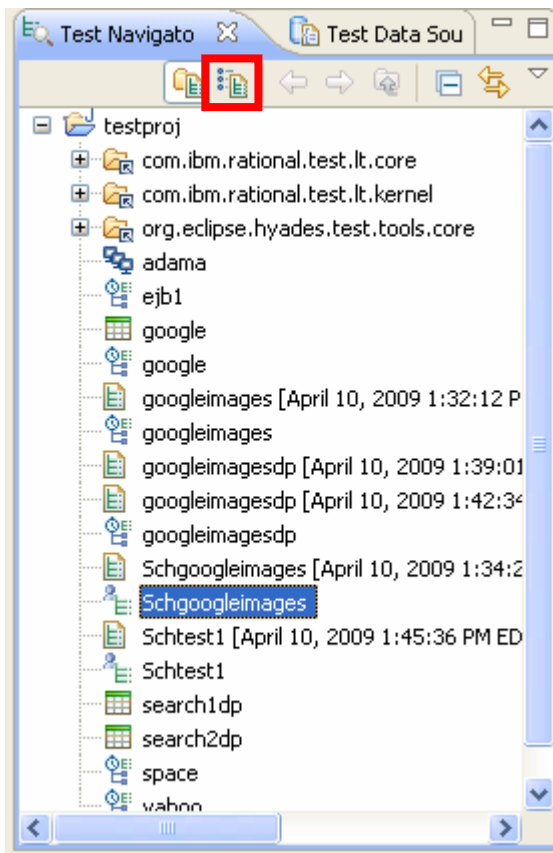
- Window > Preferences > Test > Test Editor

A screenshot of a text editor window with a white background and a blue border. The text is color-coded as follows: 'Datapooled text' is green, 'Datapool candidate' is light green, 'Used reference' is blue, 'Unused reference' is light blue, 'Substituted - Custom code' is orange, 'Substituted - Reference' is purple, 'Substituted - Test Variable' is pink, and 'Substituted - Built-in Variable' is light purple. A vertical scrollbar is on the right side of the editor.

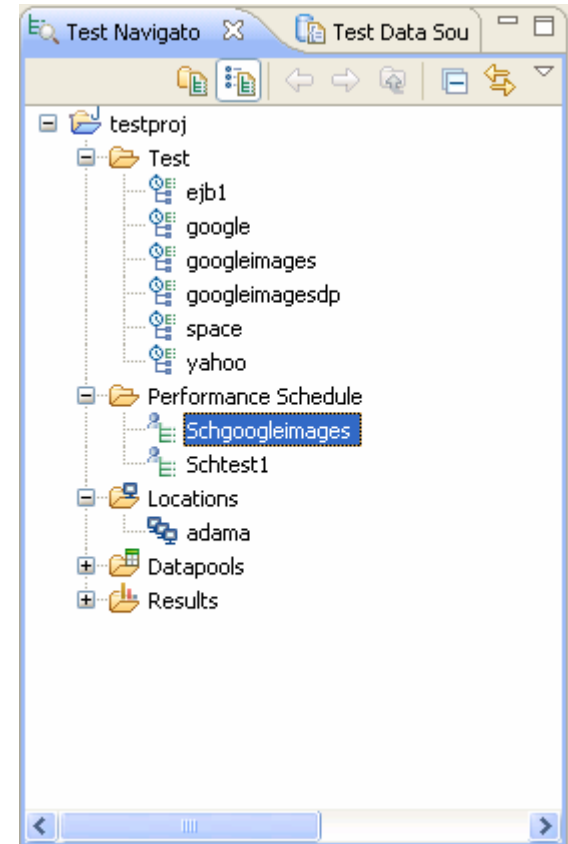
Example of Datapooled text.
Example of Datapool candidate.
Example of Used reference.
Example of Unused reference.
Example of Substituted - Custom code.
Example of Substituted - Reference.
Example of Substituted - Test Variable.
Example of Substituted - Built-in Variable.

Logical Test Navigator

- From this...



- To this...



Test Log

- Best Practice in Production
 - Large numbers of users
 - Best performance
- For Debug Only
 - Very small number of users
 - Great detail

Schedule Element Details

Schedule_Monday

Think Time | Resource Monitoring | Statistics | **Test Log** »»3

What to Log	Log Level
<input checked="" type="checkbox"/> Show <u>e</u> rrors and failures	All
<input checked="" type="checkbox"/> Also show <u>w</u> arnings	All
<input checked="" type="checkbox"/> And also show all <u>o</u> ther types	Primary Test Actions
<input checked="" type="checkbox"/> Only sample information from a subset of users	
<input checked="" type="radio"/> Fixed number of users: <input type="text" value="5"/> per user group	
<input type="radio"/> Percentage of users: _____ per user group	

Schedule Element Details

Schedule_Monday

Think Time | Resource Monitoring | Statistics | **Test Log** »»3

What to Log	Log Level
<input checked="" type="checkbox"/> Show <u>e</u> rrors and failures	All
<input checked="" type="checkbox"/> Also show <u>w</u> arnings	All
<input checked="" type="checkbox"/> And also show all <u>o</u> ther types	All
<input type="checkbox"/> Only sample information from a subset of users:	
<input type="radio"/> Fixed number of users: 5 per user group	
<input checked="" type="radio"/> Percentage of users: 100 per user group	



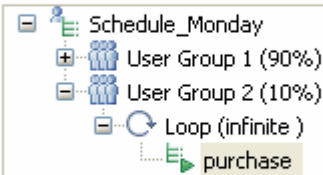
Loops

- **Schedule**
 - Start new connections on each iteration
 - Increased server workload

Performance Schedule - :

Schedule Contents

This section shows the schedule contents

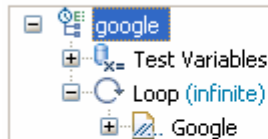


- **Tests**
 - Re-use connections on each iteration

Test - google

Test Contents

This section shows the test contents



SmartLoad

- **Stage**
 - Time period with specified number of users
- **Users**
 - How many users should run at the same time during the stage
 - It is not how many users to add or subtract
- **Stage Duration**
 - How long the stage should last
 - It is not the time at which the stage should begin
 - It begins AFTER all users have been started for the stage and AFTER any Settle Time
- **Change Rate**
 - How quickly users should start
 - Default is to start all users as fast as possible
 - All / 1 Minute means users for the stage are added or removed in a random uniform fashion over 1 minute
- **Settle Time**
 - How long to delay before beginning Stage Duration
 - Affects when the time range for the stage is created
 - Allows the System Under Test to “settle” after the disruption of changing the number of users running

Schedule Element Details

Schedule_Monday

User Load Think Time Resource Monitoring Statistics Test Log »2

Users	Stage Duration	Change Rate	Settle Time
5	10 Minutes	All/1 Minute	30 Seconds
50	10 Minutes	All/1 Minute	30 Seconds
500	10 Minutes	All/1 Minute	30 Seconds

Show Advanced
 Time limit for a user to respond to a stop request: 30 Seconds

User Load Preview

Context Sensitive Help

- Whenever you forget...
 - Click in sub-window of interest
 - Press F1



dule_Monday

Schedule Element Details

Schedule_Monday

User Load Think Time Resource Monitoring >>4

Users	Stage Duration	Change	Add...
5	10 Minutes	All/1 M	Edit...
50	10 Minutes	All/1 M	Remove
500	10 Minutes	All/1 M	Up
			Down

Show Advanced

Time limit for a user to respond to a stop request: 30

User Load Preview

The graph shows the number of users over a 30-minute period. The y-axis is labeled 'Users' and ranges from 0 to 500. The x-axis is labeled 'Run duration [min]' and ranges from 0 to 30. The line starts at 0, remains at 0 until approximately 10 minutes, then rises to 50, and finally rises to 500 at approximately 25 minutes, remaining at 500 until the end of the 30-minute period.

Help

Related Topics

number of users that start a run, to stagger the users when the run starts, and optionally to run the schedule in stages, with a different number of users at each stage.

Right-click in the table, and select **Add** to add a stage. To modify a stage, select the row and then click **Edit** or click the user icon in the first column.

- **Users:** Enter the total number of users active in the stage (not the number of users to add or subtract to those currently running).
- **Run for specified period of time:** Enter the length of time (and the time units) for the stage to run. Once the number of **Users** is achieved, the users will run for up to this time. When the time expires, the users continue to run if they are needed for the next stage, or, if not, they are stopped gracefully.

Click **Show Advanced** to set further options to prepare the system-under-test before the users actually enter the stage:

- **Change Rate:** Enter a number to set a delay between adding or removing each user, rather than adding them or subtracting them all at once. Staggering users avoids overloading the system, which can cause connection timeouts. The **User Load Preview** shows this time in black.
- **Settle Time:** A system might react to a sudden change in user population. Defining a settle time, which starts when the target number of users is reached, allows the system to settle into a steady state.

Go To:

- All Topics
- Search
- Bookmarks
- Index

Enabled Writable

Tips for Top Performance

- **Never drive load from the RPT User Interface workbench machine**
- **Make sure agent computers have sufficient TCP/IP ports for Windows**
 - HKEY_LOCAL_MACHINE > SYSTEM > CurrentControlSet > Services > TCPIP > Parameters > **MaxUserPort**
 - Does not exist by default (5000 port maximum is standard)
 - DWORD
 - Increase to 65000
 - Reboot!
- **Stay below 1500m workbench heap**
 - Default is half physical memory up to 1024m
 - Set in eclipse.ini with maximum 1200-1400m
 - Exceeding can ironically introduce out of memory error
- **Agent computer maximum heap**
 - Up to 1500m Windows
 - Up to 3000m Linux
 - Set using RPT_VMARGS general property in the location editor
- **Increase Linux per process file limit**
 - Default is 1024. Increase to 10,000 – 30,000.
- **Increase Statistics Interval for longer runs**
 - Default is 5 seconds. Increase to 30 or 60 seconds.



Trick: Multiple engines on one box

- How can I take advantage of a hot box with lots of memory?
- RPT can run more than one engine as long as the *hostname* is different
- Create another location for that box using the IP address for the hostname
- Example (scooby IP is 9.42.117.101)
 - Name: scooby1 Name: scooby2
 - Hostname: scooby.rtp.ibm.com Hostname: 9.42.117.101



Custom Code

- Execute your Java code during the test
- Manage loops (break, continue)
- Perform processing on response from the server and return results as references for future use
- Run external programs



Stop test & Stop user

```
// Stop the user  
throw new RPTStopUserEvent("User Stopped");
```

```
// Stop the test  
RPTStopTestEvent stopTest =  
    new RPTStopTestEvent("Test Stopped");  
// Default is ERROR  
stopTest.setSeverity(MessageEvent.WARNING);  
// No stack trace in the test log  
stopTest.setShowStackTrace(false);  
throw stopTest;
```



Debug Custom Code

- **Copy location to create a “debug” location**
- **Pass debug enablement string**
 - `RPT_VMARGS=-Xdebug -Xnoagent -Djava.compiler=NONE`
`-Xrunjdwp:transport=dt_socket,server=y,suspend=y,address=8000`



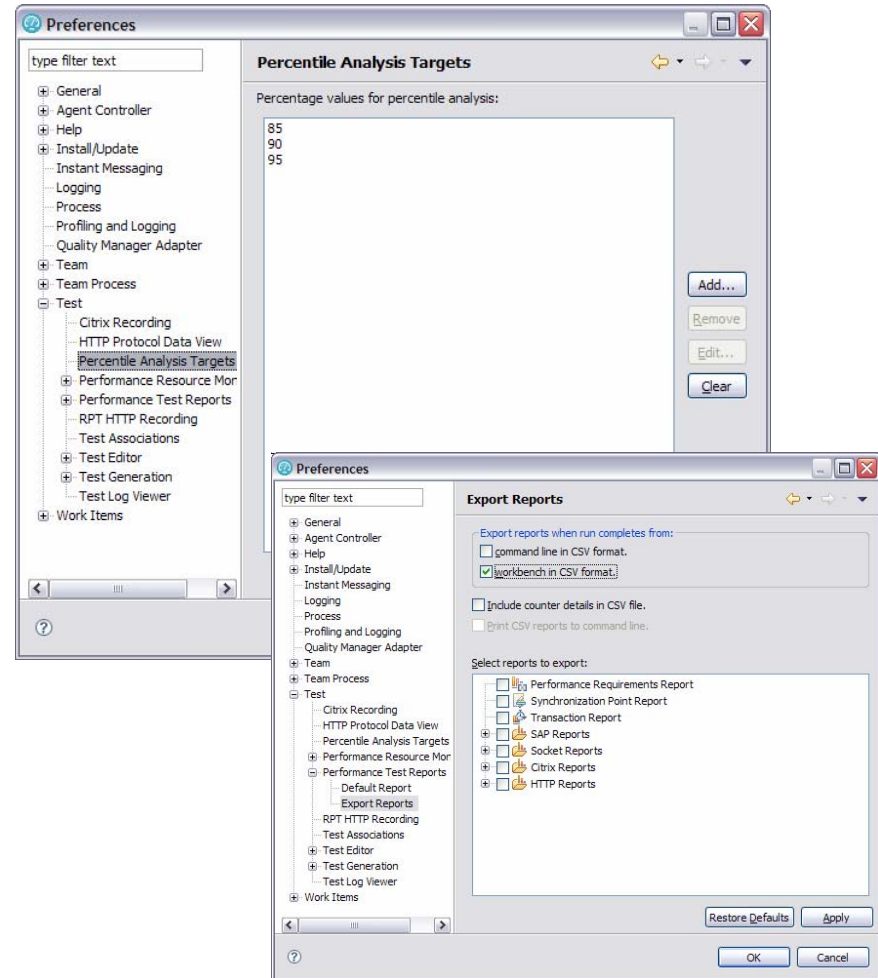
How Do I Troubleshoot?

- Catastrophic Recovery without Reboot
 - Try again
 - Delete stray processes: java, vmstat, typeperf, ACServer
 - Use Process Explorer and look at command line to identify processes.
 - Stop and restart all Agent Controllers and RPT
 - Linux uses RAStop.sh and RASstart.sh
 - Control Panel on Windows
 - Windows command line:
 - Windows stop command is net stop “ibm rational agent controller”
 - Windows start command is net start “ibm rational agent controller”



Thorough Analysis with RPT

- Page Percentile Report
- Load level performance comparison
- Determine problems
 - Resource Data correlation
 - Response Time Breakdown analysis



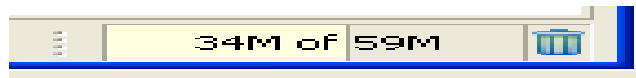
Agenda

- Let me know What is IBM Rational Performance Tester first
- Hmm...Do you have a list of features !!!
- Great... is there more ?
- How about some tips and techniques.
- Any Guidance to our team ?
- I have few questions.



RPT's Eclipse Workbench JVM – Process Size

- Workbench JVM (javaw.exe) is “overhead” independent of # of users in a running test
- Size starts at 50-100 MB and can grow significantly when viewing/editing of large assets (tests, reports, datapools)
- However, javaw.exe's process memory consumption will then typically stabilize at maximum heap size (plus overhead)
- Garbage collection potentially will reduce unused heap memory over time, but can be invoked manually as well.
 - TIP: Set Eclipse's General Preference: Show Heap Status



- Shows Current Heap Size Info (*Used of Allocated*) – not Max Heap
- Click on the Garbage Can to invoke manual garbage collection (hint, not guarantee)

RPT's Execution Engine's "Driver" JVM – Memory Factors

- **Memory usage relates mostly to the # of users & workload**
- **Mostly proportional to # of users (allowing for fixed overhead)**
- **Higher for workloads that have:**
 - Large GIFs, large posts (high I/O requires more memory)
 - Many page elements per page (stored concurrently)
 - Long response times (memory held longer)
 - Small think times (high activity increases "density")
 - Users that run in lock-step (low "dispersion" leads to higher memory "peaks")
- **Determines # of driver machines you need *if memory bound***



Some Real World Examples

- As always, *mileage will vary* especially if you accelerate “per user” activity
- Examples of what is *achievable* for light-to-medium workloads:
 - Not recommended for sizing without workload validation, preferably including memory and CPU calibration – otherwise use previous more conservative guidance!

Server Hardware	CPU Speed	Main Memory	Capacity
xSeries 330 (2 CPUs)	800 MHz (PIII)	1 GB memory	1300 VTs
xSeries 346 (2 CPUs)	3.4 GHz DualCore	3.2 GB memory	2400 VTs
AIX JS21 (2 CPUs)	2.7 GHz (64-bit)	16 GB memory	4100 VTs



Learn more about IBM® Rational® Performance Tester

- **Workshop on Rational Performance Tester 8.1**
 - Tomorrow, from 2:00 to 4:00





Thank You

© Copyright IBM Corporation 2009. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, Rational, the Rational logo, Telelogic, the Telelogic logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.



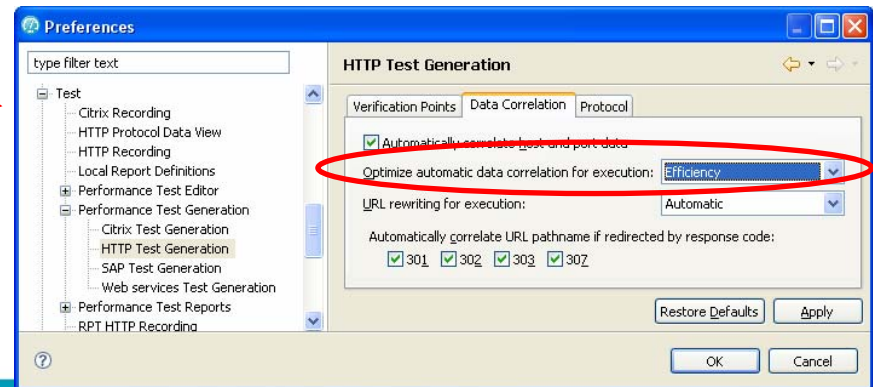
Driver (Agent) Configuration Recommendations (1 of 4)

- **Dedicate Drivers to performing the load test**
- **Use dual or quad processor systems with at least 2GB memory**
- **Fewer faster/bigger Driver systems are better than more smaller ones**
 - Fewer Drivers means less communication and startup/cleanup time
 - This also reduces stats model aggregation overhead on the Workbench *if* you choose to keep individual statistical data for each driver machine – this isn't the default, but if you uncheck the "Only store All Hosts statistics" on the Schedule box then:
 - Stats model size grows proportional to $(1 + \# \text{ of Drivers})$
- **Avoid looping within schedule over a short test – move loop inside the test instead**
 - More efficient and allows re-use of HTTP connections across loops – otherwise the default is to close connections at the end of the test
 - A huge gain in "blasting" tests!



Driver (Agent) Configuration Recommendations (2 of 4)

- **Beware of running out of CPU before memory (keep average utilization < 70%)**
- **Allocate 75% or more of the system's available memory to the JVM heap (capped at 1400 – 1500 MB)**
- **“Tune” Data Correlations to reduce CPU overhead:**
 - Remove “Over Correlations” (unnecessary correlations)
 - Remove redundant reference gathering
by optimizing data correlation for
Efficiency vs. Accuracy



Driver (Agent) Configuration Recommendations (3 of 4)

- Use 1Gb NIC cards (but match benchmark configuration if specified)
- Make sure you have sufficient # of TCP/IP ports
 - On Windows drivers the limit is typically 5000
 - Use the netstat -a command to observe port usage during run
 - If the largest number you see is 5000 then you have a problem!
 - You can increase it by adding a registry entry (need to reboot for it to take effect!)
HKEY_LOCAL_MACHINE > SYSTEM > CurrentControlSet > Services > Tcpip > Parameters > MaxUserPort
(dword, default is 5000, max is 65000)
- Linux drivers need per process open file limit higher than 1024
 - As root, ulimit -n 30000 (or appropriate value) before starting agent controller
- Treat Windows and Linux as roughly equivalent for sizing for RPT 8.x



Driver (Agent) Configuration Recommendations (4 of 4)

- How to leverage big iron – large 64-bit multi-processor, multi-core servers ?
 - RPT is typically CPU-bound for most workloads, however large multi-processor, multi-core servers may often provide enough CPU processing capacity such that RPT's 32-bit JVM max heap limit (1.5 GB on Windows, ~3 GB on Linux) becomes the bottleneck...
 - To fully utilize large driver systems (*only needed if memory-bound*)
 - Run multiple engines by defining separate locations that map to the same system this will allow utilization of up to 1500 MB memory per engine
 - You must use network aliases to supply unique hostnames to accomplish this, e.g. `/etc/hosts` (on Windows: `C:\WINDOWS\system32\drivers\etc\hosts`)
`x.x.x.x driverhostname, driver_2, driver_3, driver_4`
 - By adding as many RPT engines as needed, RPT can scale up to fully utilize the fastest multi-core machines available
 - Hot AIX boxes could be the most scalable...
 - IBM Bladecenter: JS23 (4-core) or JS43 (8-core) 4.2GHz POWER6 processors
 - IBM Power 595: up to 64-core 4.2 or 5 GHz POWER6 processors



Workbench Configuration Recommendations (1 of 3)



- **Use 2GB workbench with max heap set to 1200MB (up to 1400+ MB if possible, but < 1500 MB)**
- **Don't run any users locally on Workbench**
- **Dual processor is normally not important, but is useful for**
 - Very long runs
 - Better interactivity during run
 - Use of response time breakdown (this data is loaded into workbench during the run)
 - Use of live rendering during run

Workbench Configuration Recommendations (2 of 3)

- **Use sampling options on the Schedule to reduce CPU, Memory, and Disk overhead *on the Workbench***

- **Statistics**

- If you don't need response times or server health for page elements (i.e., Page statistics are sufficient)
 - Set the *Statistics log level* to **Primary Test Actions** (=Pages for HTTP)
 - » This can easily save 5 – 50X or more (based on average # of page elements per page)
- Consider using less frequent sampling by increasing the *Statistics sampling interval* beyond the default of **5** seconds:
 - 10 – 60 second range is reasonable, especially for longer running tests
 - General advice for typical workloads:
 - » If run < 1 hour, leave at 5 seconds
 - » If run > 4 hours, use 30 seconds (or more for multi-day runs)
 - If you have a very large number of unique page elements, you may want to compensate accordingly with longer sampling intervals
- For long runs realize that packing more than one data point per display pixel isn't useful! 😊
 - » 24 hours / 1600 pixels = 54 seconds sample interval per pixel – additional samples from shorter intervals aren't even visible at run completion
 - » But consider that you may want to “zoom in” by creating shorter Time Ranges for analysis



Workbench Configuration Recommendations (3 of 3)

Test Log

- Still be careful with the Test Log settings
 - Volume of Test Log data can easily become excessive without very low sampling rates unless you log only errors & failures and warnings
 - Default for *Test Log log level* is Primary Test Actions (=Pages for HTTP) for successes in addition to errors & failures and warnings
 - Default sampling is for 5 *Fixed number of users* per user group
- Once you're past debugging consider switching to only log Errors & failures and warnings
 - » Turn sampling off to detect errors and warnings on ALL users
 - » If you need HTTP percentile reports, you must add in Primary Test Actions for other events* (you don't need Secondary or All)
- Ensure that you have available disk space before you collect using the All setting for log level, and do so only when you really want all the data.

