IBM Innovate 2011
Session Track Template

Rajani Vatikunta and Deepak Muniram

Performance Test 35000 Users for 18 Hours ? No Problem

rvatikun@in.ibm.com,deepak.muniram@in.ibm.com

Session Track Number : QM-2139

IBM Software
Innovate2011
The Premier Event for Software and Systems Innovation

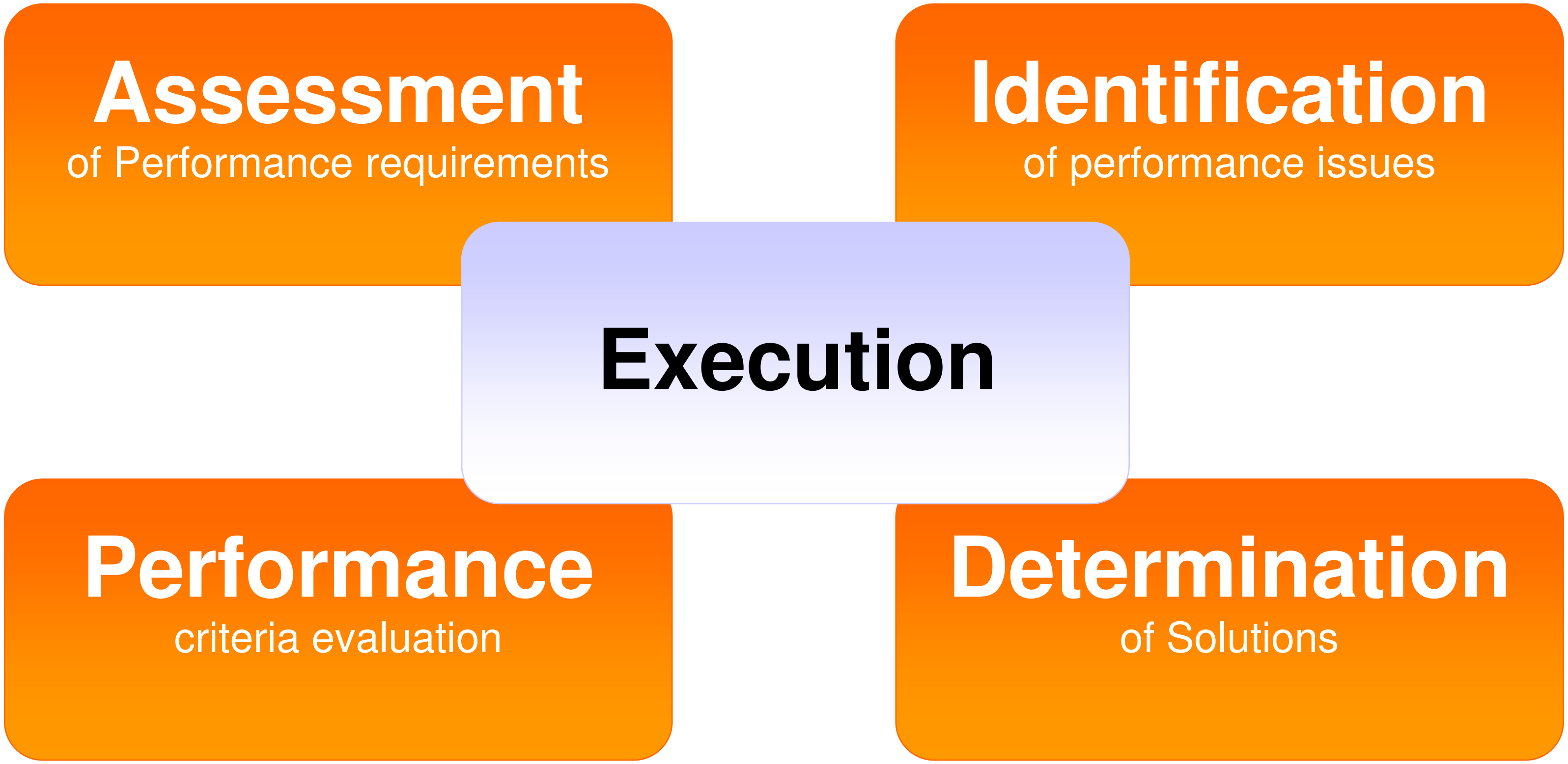
Software. Everyware.

June 5–9 Orlando, Florida

Agenda

- Introduction
- Performance Testing for large loads
- Simple Thumb rules
- Custom Codes for Rescue
- Custom Reports
- Optimum Settings
- Schedule and Agents Settings

Targets of performance testing



Assessment
of Performance requirements

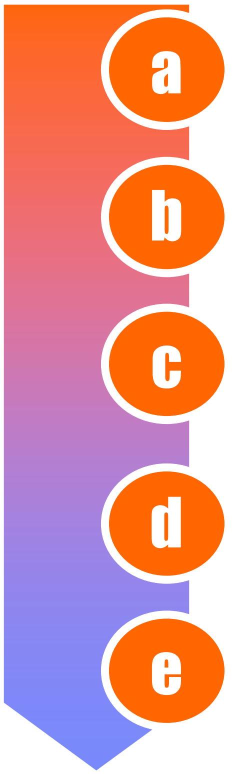
Identification
of performance issues

Execution

Performance
criteria evaluation

Determination
of Solutions

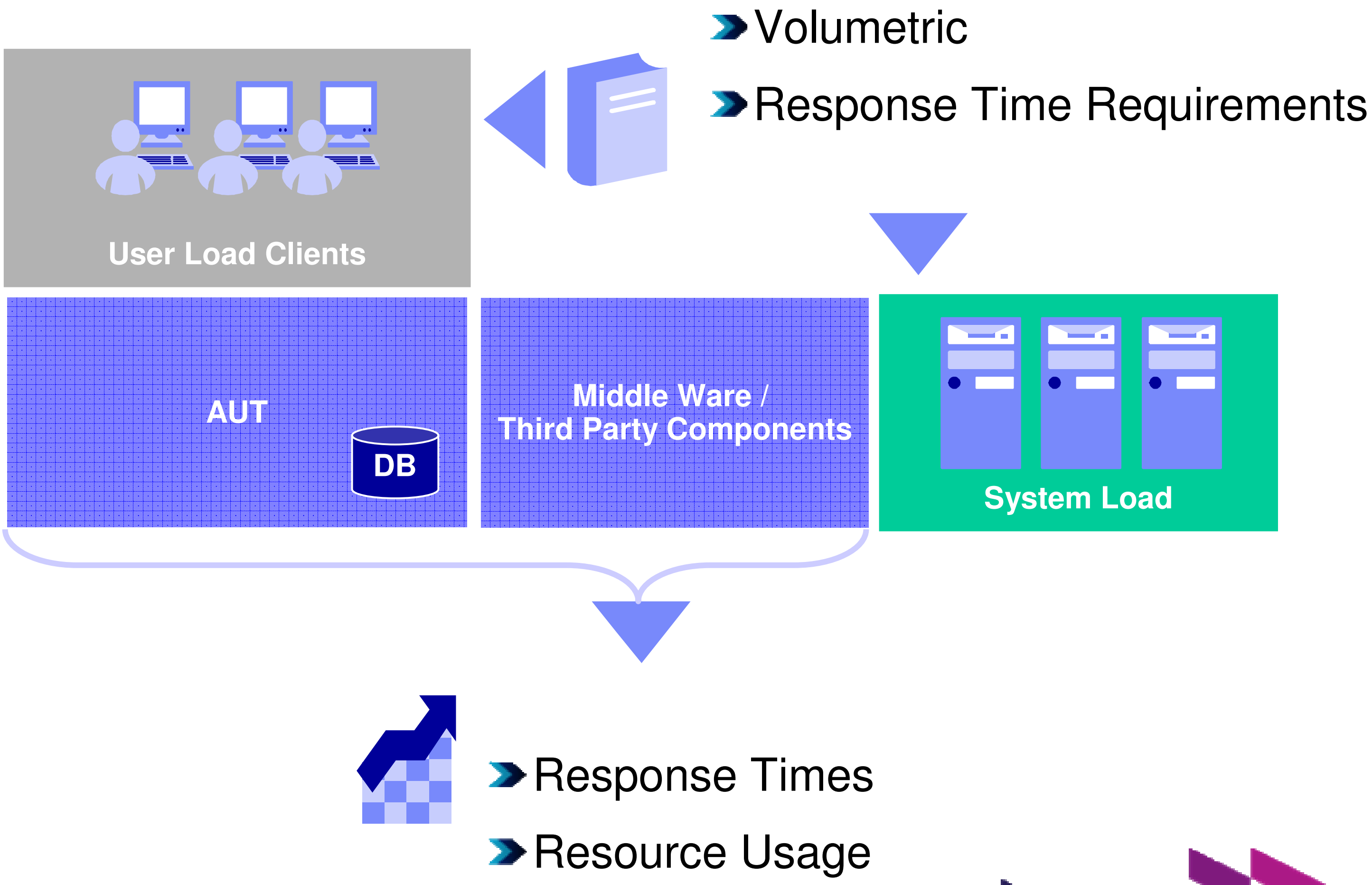
Performance testing types



- a Benchmark testing
- b Contention test
- c Performance profiling
- d Load testing
- e Stress testing

How to run successful large load tests?

Load and performance Test



- Volumetric
- Response Time Requirements

User Load Clients

AUT

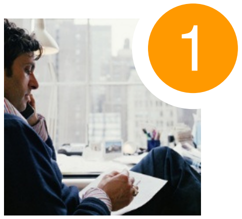
DB

Middle Ware / Third Party Components

System Load

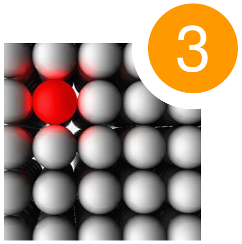
- Response Times
- Resource Usage

Steps of Large Load Tests



1 Evaluating the results for a single actor or use-case instance

2 Gradually increase the load and then Evaluate the results



3 Examining summary statistics and actual data values for specific actor or use-case instance

4 Detailed analysis takes the low-level data and uses statistical methods

Identification of performance acceptance criteria

1 Response Time

2 Throughput

3 Resource Utilization

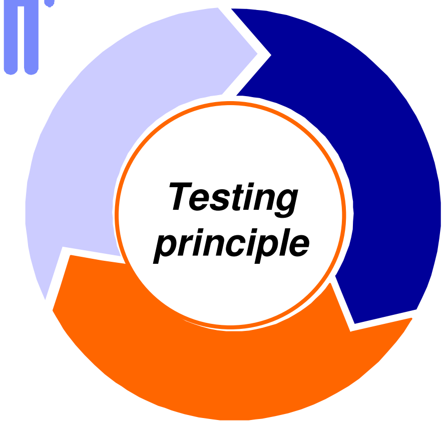
The right team for success

Analysis

requires a skilled team

Design

to non functional
requirement.

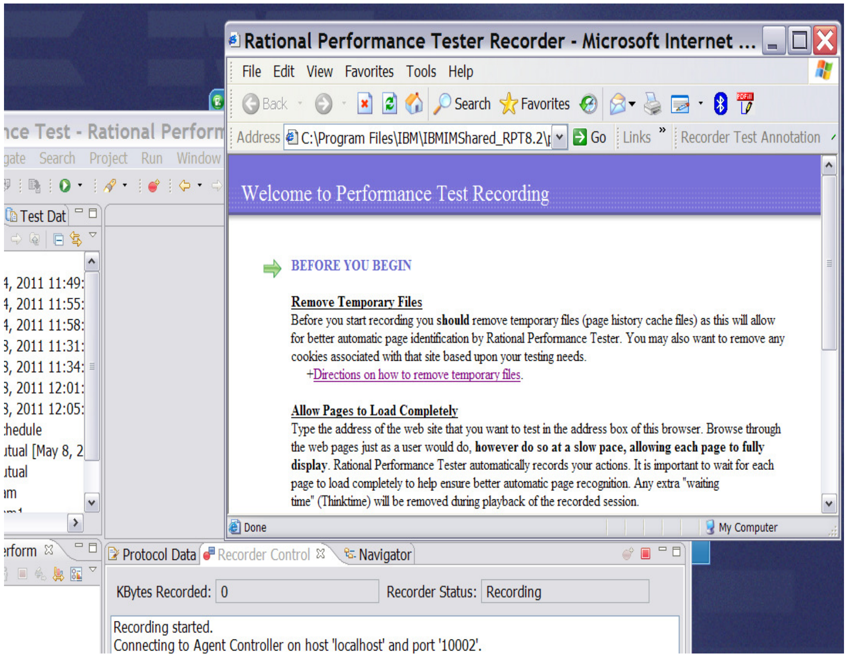


Test

conducted properly

Auto Scripting through RFT

- RFT can be used for scripting in RPT
- Record scripts in RFT and baseline it. Playback RFT scripts on a browser which is currently opened by RPT for recording
- Manual intervention is limited
- Comes very handy when Application is undergoing some changes for fixes or when application needs to be benchmark on multiple test environments



Welcome to Performance Test Recording

BEFORE YOU BEGIN

Remove Temporary Files

Before you start recording you **should** remove temporary files (page history cache files) as this will allow for better automatic page identification by Rational Performance Tester. You may also want to remove any cookies associated with that site based upon your testing needs.

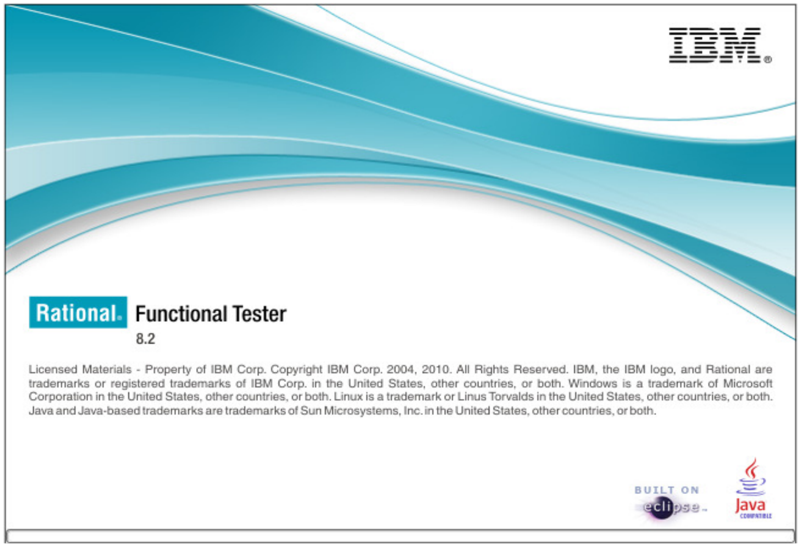
+Directions on how to remove temporary files.

Allow Pages to Load Completely

Type the address of the web site that you want to test in the address box of this browser. Browse through the web pages just as a user would do, **however do so at a slow pace, allowing each page to fully display**. Rational Performance Tester automatically records your actions. It is important to wait for each page to load completely to help ensure better automatic page recognition. Any extra "waiting time" (Thinktime) will be removed during playback of the recorded session.

KBytes Recorded: 0 Recorder Status: Recording

```
Recording started.
Connecting to Agent Controller on host 'localhost' and port '10002'.
```

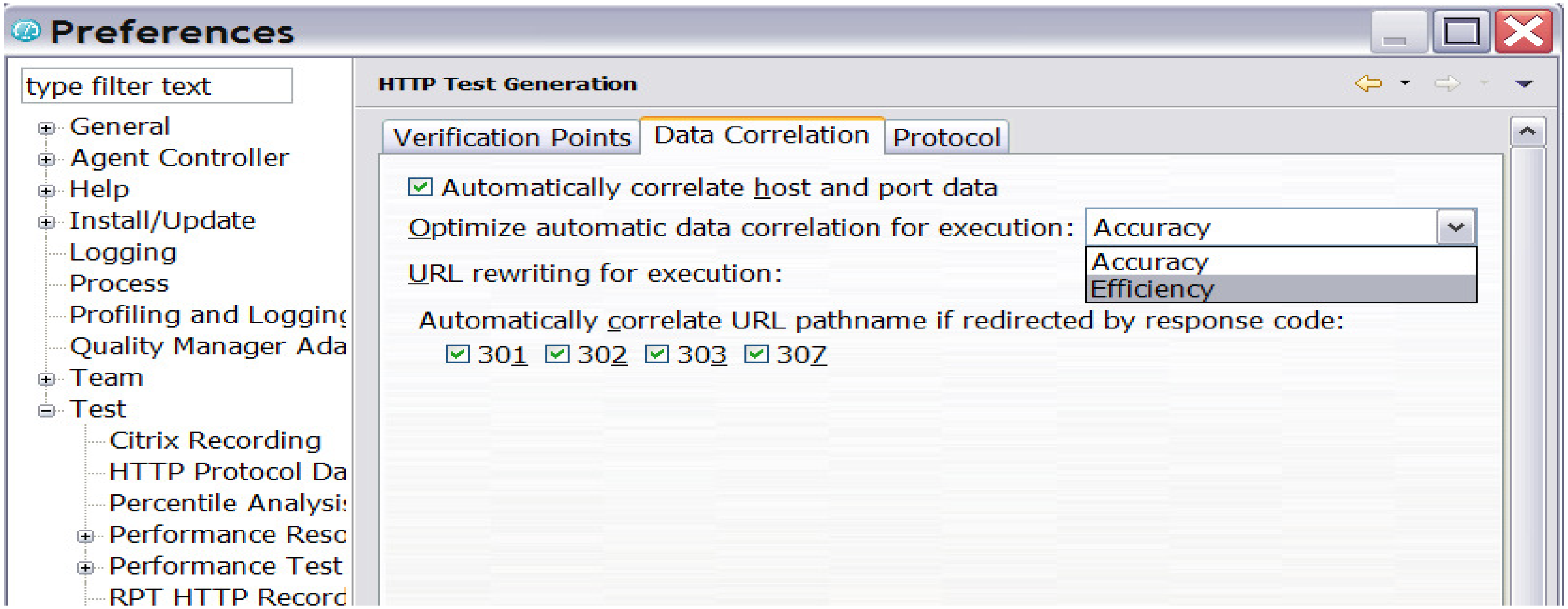


Rational. Functional Tester
8.2

Licensed Materials - Property of IBM Corp. Copyright IBM Corp. 2004, 2010. All Rights Reserved. IBM, the IBM logo, and Rational are trademarks or registered trademarks of IBM Corp. in the United States, other countries, or both. Windows is a trademark of Microsoft Corporation in the United States, other countries, or both. Linux is a trademark or Linus Torvalds in the United States, other countries, or both. Java and Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

RPT Setting

- Min correlation by doing manual correlation
- This makes scripts more robust and light weight in terms on CPU and Memory consumption
- Use efficiency instead of Accuracy based on the application type and scenarios



☑ Automatically correlate host and port data

Optimize automatic data correlation for execution: Accuracy

Accuracy
Efficiency

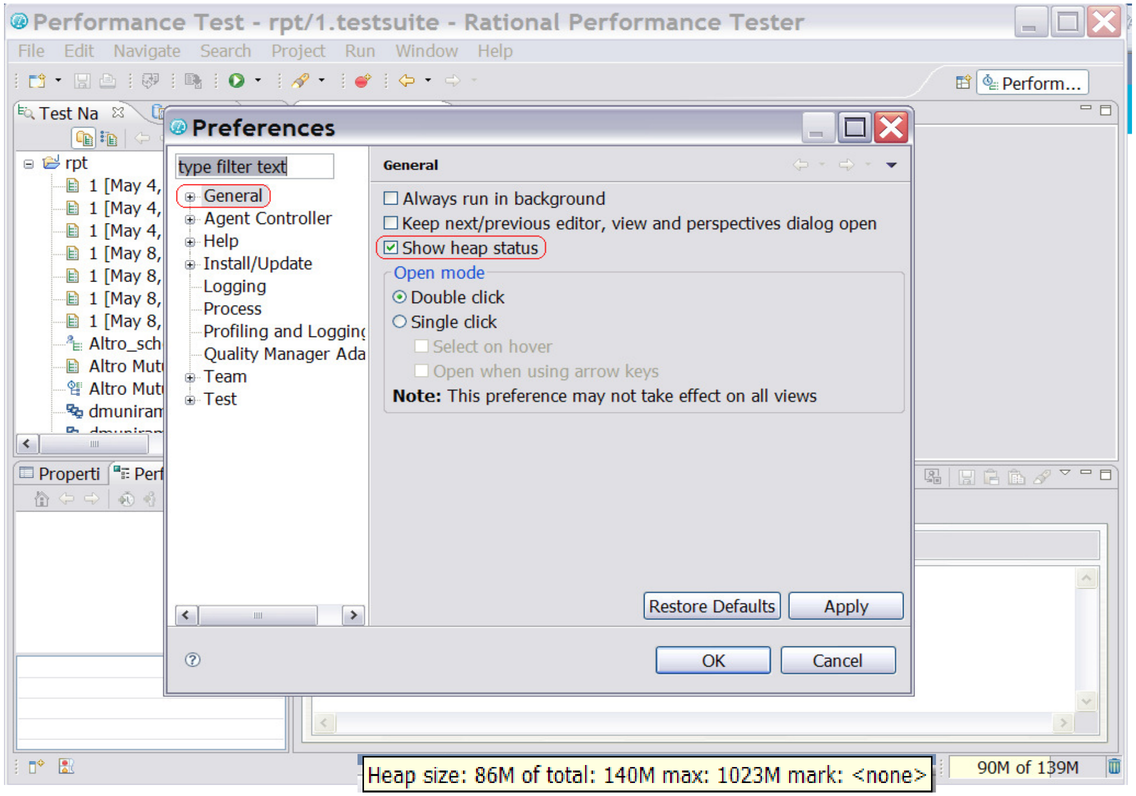
URL rewriting for execution:

Automatically correlate URL pathname if redirected by response code:

☑ 301 ☑ 302 ☑ 303 ☑ 307

RPT Fine tuning

- Add VM Parameters into Eclipse.ini to increase default memory.
- Enable Heap status to monitor Heap usage
- Optimum value would be to have a value -Xmx1550m



General

☐ Always run in background

☐ Keep next/previous editor, view and perspectives dialog open

☑ Show heap status

Open mode

● Double click

○ Single click

☐ Select on hover

☐ Open when using arrow keys

Note: This preference may not take effect on all views

Heap size: 86M of total: 140M max: 1023M mark: <none>

90M of 139M

TCP IP Tuning

- During Performance Testing, RPT consumes socket connections, in order to properly simulate a work load on a AUT.
- However, when the available socket connections from the Operating System have been exhausted, the test will likely experience failure.
- Two parameters in registry would help in fixing connection issues
  - TimedWaitDelay and MaxUserPort
  - Optimum Values for MaxUserPort = 65000 (Decimal) and TimedWaitDelay = 30 (Decimal)

Note :Modifying and adding appropriate value would overcome socket connection issues

TCP TimedWaitDelay

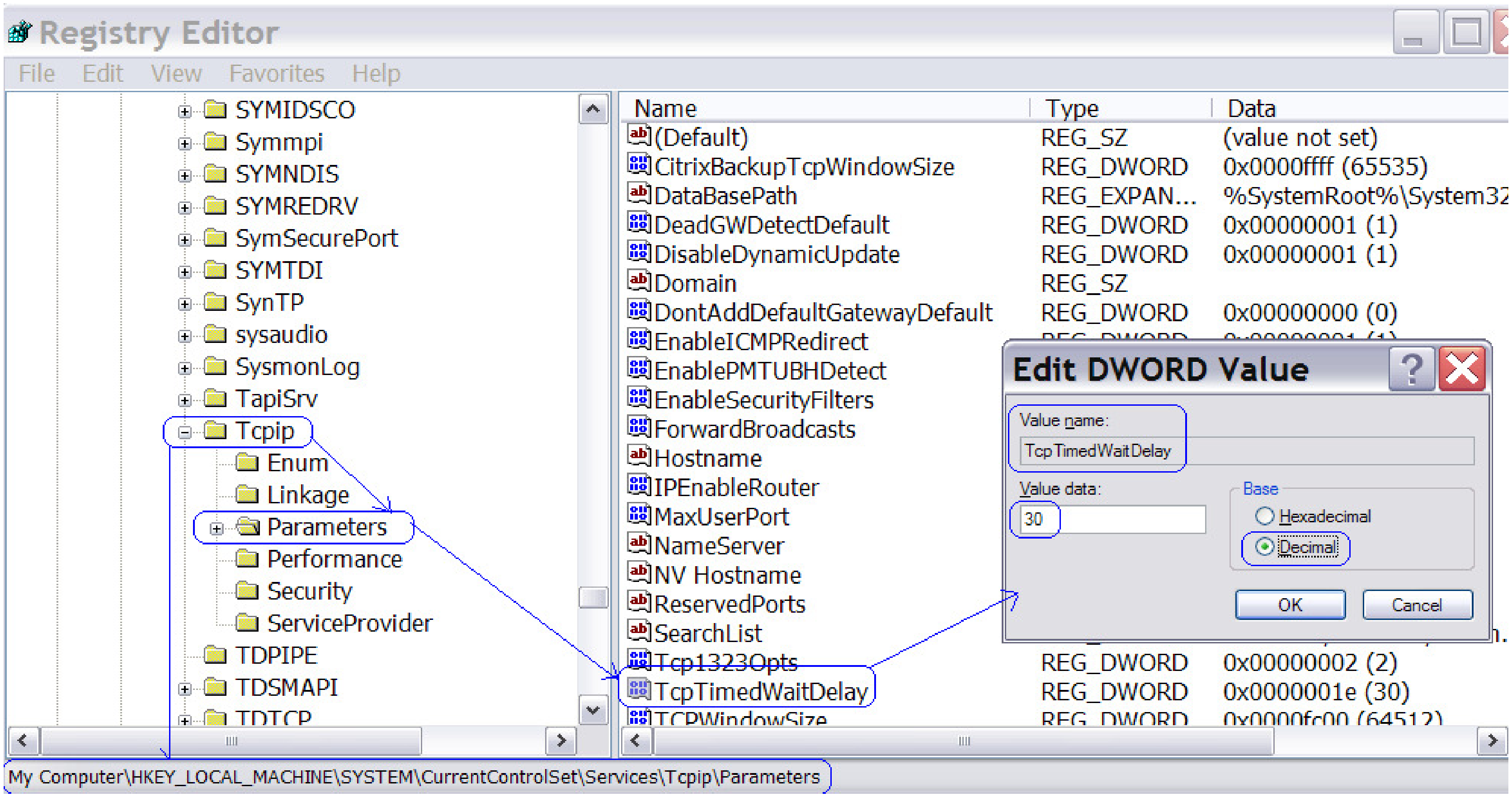
- This Parameter determines the time that must elapse before TCP can release a closed connection and reuse its resources.
- During this time, the connection can be reopened at much less cost to the client and server than establishing a new connection.
- Reducing the value of this entry allows TCP to release closed connections faster, providing more resources for new connections.

NOTE : if the value is too low, TCP might release connection resources before the connection is complete, requiring the server to use additional resources to re-establish the connection.

To Configure :

- Open the Windows program **regedt32** and go to subkey: "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters"
- Create a new REG_DWORD value named TcpTimedWaitDelay and set it to "0x1E" (30 seconds), as an example, to effect a quick turn around time for socket reuse (the minimum is 30 seconds).

TCP TimedWaitDelay



- SYMIDSCO
- Symmpi
- SYMNDIS
- SYMREDRV
- SymSecurePort
- SYMTDI
- SynTP
- sysaudio
- SysmonLog
- TapiSrv
- Tcpip
  - Enum
  - Linkage
  - Parameters
  - Performance
  - Security
  - ServiceProvider
- TDPIPE
- TDSMAPI
- TDTCP

Name	Type	Data
(Default)	REG_SZ	(value not set)
CitrixBackupTcpWindowSize	REG_DWORD	0x0000ffff (65535)
DataBasePath	REG_EXPAN...	%SystemRoot%\System32
DeadGWDetectDefault	REG_DWORD	0x00000001 (1)
DisableDynamicUpdate	REG_DWORD	0x00000001 (1)
Domain	REG_SZ	
DontAddDefaultGatewayDefault	REG_DWORD	0x00000000 (0)
EnableICMPRedirect	REG_DWORD	0x00000001 (1)
EnablePMTUBHDetect		
EnableSecurityFilters		
ForwardBroadcasts		
Hostname		
IPEnableRouter		
MaxUserPort		
NameServer		
NV Hostname		
ReservedPorts		
SearchList		n.
Tcp1323Opts	REG_DWORD	0x00000002 (2)
TcpTimedWaitDelay	REG_DWORD	0x0000001e (30)
TCPWindowSize	REG_DWORD	0x0000fc00 (64512)

Edit DWORD Value

Value name: TcpTimedWaitDelay

Value data: 30

Base: ○ Hexadecimal ● Decimal

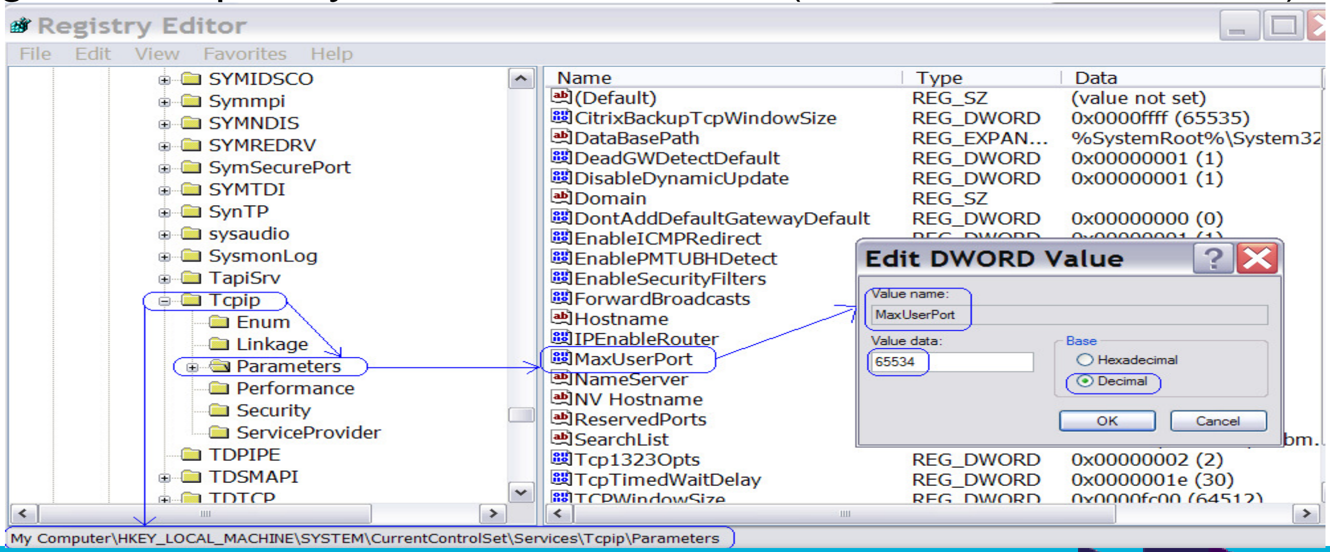
My Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters

MaxUserPort

- Determines the highest port number TCP can assign when an application requests an available user port from the system.

To Configure :

- Open the Windows program **regedt32** and go to subkey: "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters ".
- Create a new REG_DWORD value named MaxUserPort and set it to "0x8000 (32768)", as an example, so TCP will have a much larger range of port numbers to assign for temporary socket connections (the maximum is 65534).



```
SYMIDSCO
Symmpi
SYMNDIS
SYMREDRV
SymSecurePort
SYMTDI
SynTP
sysaudio
SysmonLog
TapiSrv
Tcpip
    Enum
    Linkage
    Parameters
    Performance
    Security
    ServiceProvider
TDPIPE
TDSMAPI
TDTCP
```

Name	Type	Data
(Default)	REG_SZ	(value not set)
CitrixBackupTcpWindowSize	REG_DWORD	0x0000ffff (65535)
DataBasePath	REG_EXPAN...	%SystemRoot%\System32
DeadGWDetectDefault	REG_DWORD	0x00000001 (1)
DisableDynamicUpdate	REG_DWORD	0x00000001 (1)
Domain	REG_SZ	
DontAddDefaultGatewayDefault	REG_DWORD	0x00000000 (0)
EnableICMPRedirect	REG_DWORD	0x00000001 (1)
EnablePMTUBHDetect		
EnableSecurityFilters		
ForwardBroadcasts		
Hostname		
IPEnableRouter		
MaxUserPort		
NameServer		
NV Hostname		
ReservedPorts		
SearchList		bm.
Tcp1323Opts	REG_DWORD	0x00000002 (2)
TcpTimedWaitDelay	REG_DWORD	0x0000001e (30)
TCPWindowSize	REG_DWORD	0x0000fc00 (64512)

Edit DWORD Value

Value name: MaxUserPort

Value data: 65534

Base: ○ Hexadecimal [✓ ● Decimal]

My Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters

Tuning Parameters on Other Operating Systems

OS	Parameter	Value	Implementation details
Linux	net.ipv4.ip_local_port_range	2048 65000	Edit /etc/sysctl.conf with the following parameters and save the file. Run the command – sysctl –p
	net.core.rmem_default	262144	
	net.core.wmem_default	262144	
	net.core.wmem_max	33554432	
	net.core.rmem_max	33554432	
	net.core.netdev_max_backlog	5000	
	net.ipv4.tcp_no_metrics_save	1	
	net.ipv4.tcp_rmem	4096 16777216 33554432	
	net.ipv4.tcp_wmem	4096 16777216 33554432	
	net.core.optmem_max	40960	
AIX	sb_max	6192000	no –p –o <parameter> = <value> chdev -l sys0 -a maxuproc=<value>
	clean_partial_conns	1	
	tcp_timewait	1	
	tcp_keepidle	600	
	tcp_keepinit	40	
	tcp_nodelayack	1	
	tcp_keepintvl	10	
	Maxuproc	8192	
	tcp_sendspace	4096000	
	tcp_recvspace	4096000	
	tcp_ephemeral_low	1024	
	rfc1323	1	

Custom Code

- Custom Codes comes to rescue many times for complex and secured enterprise applications.
- Extensively used custom codes in projects implementing advanced features ,functionality's and client side activities

Examples of the custom codes used

- SSO Login and SSO Logout
- Support for Large Datapools
- Customized transactions reporting
- Fast Content VP with Custom Counters

Custom Code : SSO Login and SSO Logout

There are instances where Application methods needs to be called directly from RPT to execute some methods when security logic is implemented at client side.

- Example : SSO login is a method used to generate security tokens at client machines and later these security tokens are used to validate the user id and password.
- Security Tokens are also used in generating session id, the generated session id will be used for further transactions.
- As Security Tokens are generated on client machines ,Custom Code were used to connect to set of API's for login and get the session id.
- Captured Session_ID is later used to correlate this value in subsequent requests for the transaction.

Custom Code : SSO Login and SSO Logout

Steps to implement Custom Code for calling Java methods directly in RPT.

- Copy the required files from your SSO setup and save it in your local machine

  SSOLoginConfig.conf

  SSOClientConfig.conf

  ssoclient.jar

  ssocommon.jar

- Import the below packages in custom code for SSO_Login

  ```
  import java.util.HashMap;
  import com.sso.client.SSOLoginApp;
  ```

- Import required jar files in Project Properties->Java BuildPath->Libraries-> Add External Jar Files.

  ssoclient.jar

  ssocommon.jar

- Copy the SSO login contents to custom code of the script to get the required parameters like Session_id and SSO_ServiceToken.

Custom Code : SSO Login and SSO Logout

Sample Code :

```
public String exec(ITestExecutionServices tes, String[] args) {
String Session_Service_id="";
synchronized(this)  {
 Session_Service_id="";
String ssoServiceToken = "";
String userID =args[0];
String password=args[1];
loginObj = new SSOLoginApp(path1, path2);
String session_id="";
try {
session_id=loginObj.login(userID, password, "INFENG");
        ssoServiceToken=loginObj.getServiceToken(session_id,"SSODeskTopServlet");
if  (SSO_Login_TEST.SSLoginAppList == null) {
    SSO_Login_TEST.SSLoginAppList= new HashMap();  }
SSO_Login_TEST.SSLoginAppList.put(session_id, loginObj); }
catch (Exception e) {
// TODO Auto-generated catch block
tes.getTestLogManager().reportMessage(e.getMessage());
e.printStackTrace(); }
Session_Service_id = session_id + "~" + ssoServiceToken;
}
return Session_Service_id;  } }
```

Custom Code : Support for Large Datapools

- The large datapool support classes are available to be used with manually deployed CSV files that are placed on the test agents in exactly the same path for every machine. That full pathname is then used to access the file from the custom code library.

  There are the following custom code classes available for large datapool support.

  OpenDatapoolCSV(pathname, dpname, accessType, rowsPerVU)

  CloseDatapool (dpname)

  GetDatapoolValue(dpname, columnName)

  IncrementRowCursor(dpname)

  GetDatapoolRowCursor(dpname)

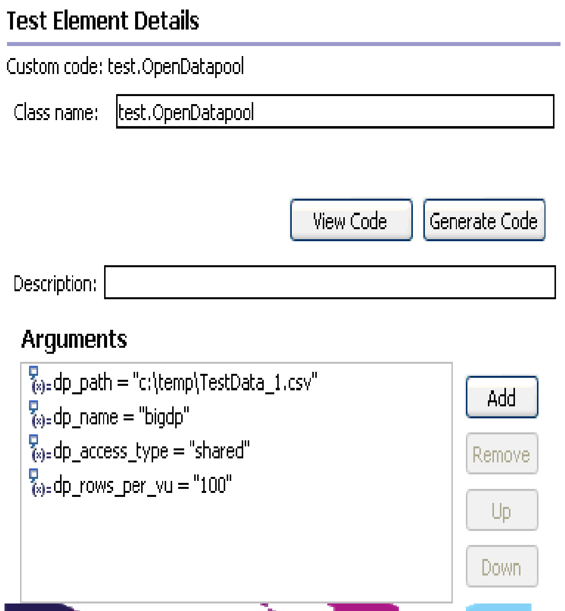
  SetDatapoolRowCursor(dpname, rowCursorValue)

Custom Code : Support for Large Datapools

OpenDatapoolCSV()

- The first VU to call this custom code class reads the datapool into the playback engine memory and sets it up for access by any number of VUs. Each successive VU must also perform the OpenDatapoolCSV() operation to set its context for access and to register itself as a user of the datapool. Any calls to the other operations will cause a test ending error if they are done without a preceeding OpenDatapoolCSV() call.
- ✓ Returned value : String datapool name of the datapool opened
- Custom Code Exec Call Content

```
public String exec(ITestExecutionServices tes, String[] args) {
String pathName = args[0];
String dpName = args[1];
String access = args[2];
int rowsVU = Integer.parseInt(args[3]);
LargeDatapool dp = new LargeDatapool();
return dp.openDatapool(pathName, dpName, access, rowsVU, tes); }
```



Test Element Details

Custom code: test.OpenDatapool

Class name: test.OpenDatapool

Description:

Arguments

dp_path = "c:\temp\TestData_1.csv"
dp_name = "bigdp"
dp_access_type = "shared"
dp_rows_per_vu = "100"

Custom Code : Support for Large Datapools

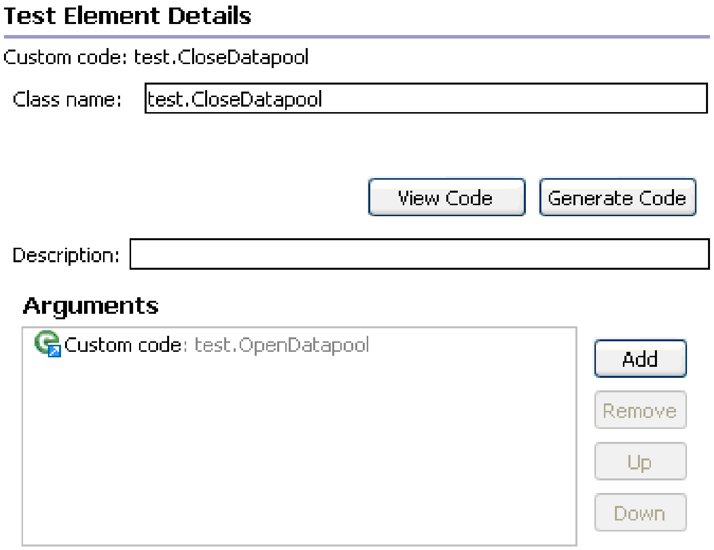
CloseDatapool()

- This custom code class closes the datapool for this virtual user and decrements the active user count. If the active user count reaches zero, the datapool is removed from the engineDataArea so the storage can be garbage collected. If the same virtual user reopens the datapool and the access method is segmentPerVU, the same range of rows previously assigned to that VU. This assumes that all VUs never all close the datapool at the same time and it has been removed from the engineDataArea.

✓ Returned value: Null string

- Custom Code Exec Call Content

```
public String exec(ITestExecutionServices tes, String[] args) {
String dpName = args[0];
LargeDatapool dp = new LargeDatapool();
return dp.closeDatapool(dpName, tes); }
```



Test Element Details

Custom code: test.CloseDatapool

Class name: test.CloseDatapool

Description:

Arguments

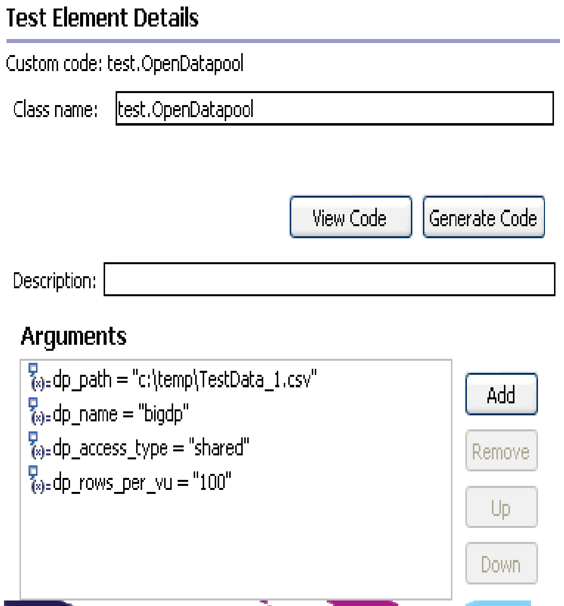
Custom code: test.OpenDatapool

Custom Code : Support for Large Datapools

GetDatapoolValue()

- This custom code class returns the column String value for the column requested on the current row for this virtual user. Note that the row index is not altered by this method. Repeated calls of this method may be used to retrieve the values of the same or different columns on this row of the datapool.
  - ✓ Returned value
- String that contains the column value using the current row index
- Custom Code Exec Call Content

```
public String exec(ITestExecutionServices tes, String[] args) {
String pathName = args[0];
String dpName = args[1];
String access = args[2];
int rowsVU = Integer.parseInt(args[3]);
LargeDatapool dp = new LargeDatapool();
return dp.openDatapool(pathName, dpName, access, rowsVU, tes); }
```



Test Element Details

Custom code: test.OpenDatapool

Class name: test.OpenDatapool

Description:

Arguments

dp_path = "c:\temp\TestData_1.csv"
dp_name = "bigdp"
dp_access_type = "shared"
dp_rows_per_vu = "100"

Custom Code : Support for Large Datapools

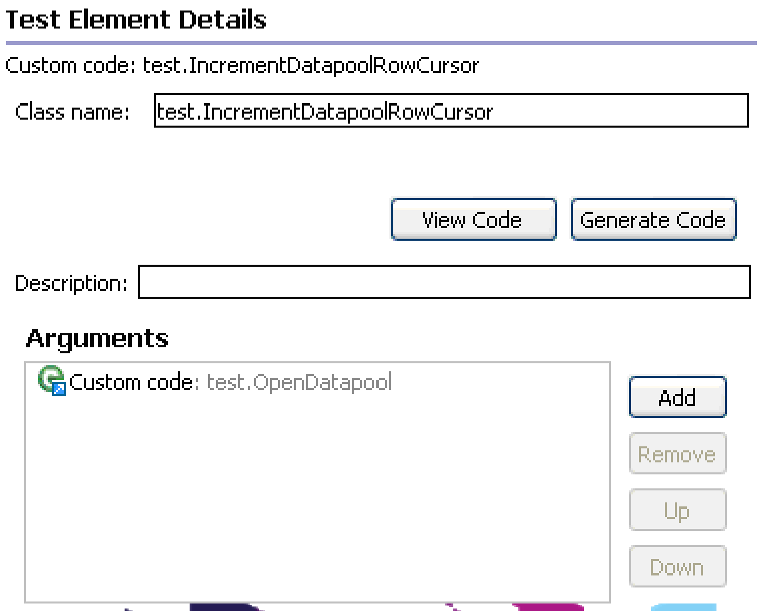
IncrementDatapoolRowCursor()

- This custom code class increments the datapool row cursor according to the access method of the datapool. If shared, the shared cursor next available row is used. If privatePerVU, the private VU cursor value is incremented by one (modulo datapool size in rows). If segmentPerVU, the local VU cursor is incremented by one (modulo segment size + base of segment offset). After this method is called in the case of shared access, the row is considered consumed and will not be allocated to a different VU.

- ✓ Returned value : Null string

- Custom Code Exec Call Content

```
public String exec(ITestExecutionServices tes, String[] args)
String dpName = args[0];
LargeDatapool dp = new LargeDatapool();
return dp.incrementDatapoolRowCursor(dpName, tes); }
```



Test Element Details

Custom code: test.IncrementDatapoolRowCursor

Class name: test.IncrementDatapoolRowCursor

Description:

Arguments

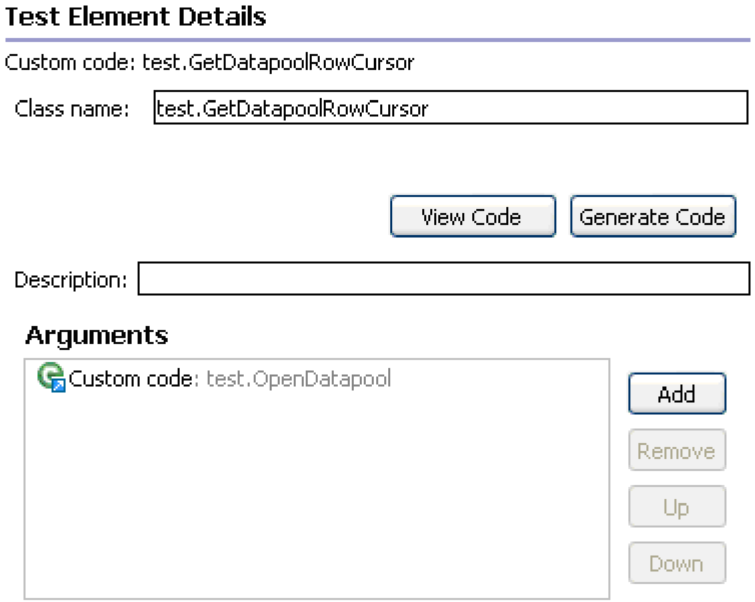
Custom code: test.OpenDatapool

Custom Code : Support for Large Datapools

GetDatapoolRowCursor()

- If you want to get the datapool row cursor value for either debugging purposes or to reset it later (like after the last iteration of a loop). Note that this is not often needed except for debugging purposes.
- Returned value : String that contains the current integer row cursor value
- Custom Code Exec Call Content

```
public String exec(ITestExecutionServices tes, String[] args) {
String dpName = args[0];
LargeDatapool dp = new LargeDatapool();
return dp.getDatapoolRowCursor(dpName, tes); }
```



Test Element Details

Custom code: test.GetDatapoolRowCursor

Class name: test.GetDatapoolRowCursor

Description:

Arguments

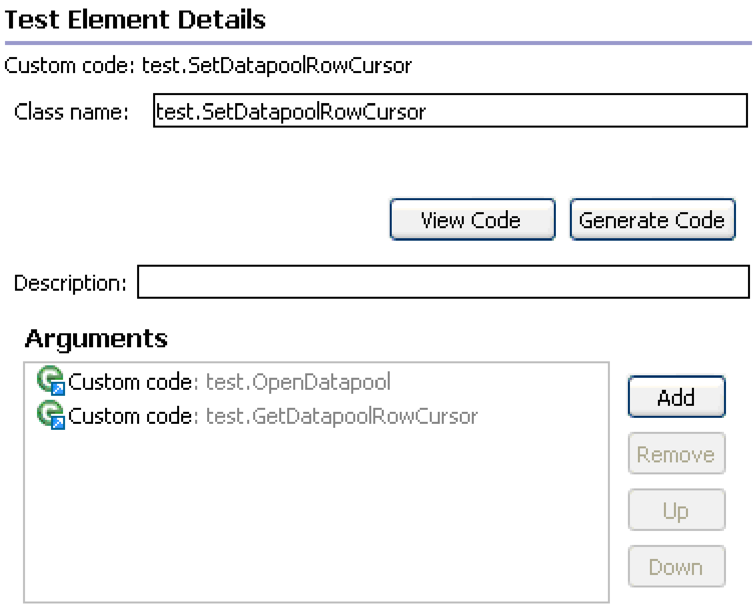
Custom code: test.OpenDatapool

Custom Code : Support for Large Datapools

SetDatapoolRowCursor()

- If you want to set the datapool row cursor to a specific value, this custom code class provides the way to do so. Note that this is a rare event and not generally used.
- ✓ Returned value : Null string
- Custom Code Exec Call Content

```
public String exec(ITestExecutionServices tes, String[] args) {
String dpName = args[0];
int rowIndex = Integer.parseInt(args[1]);
LargeDatapool dp = new LargeDatapool();
return dp.setDatapoolRowCursor(dpName, rowIndex, tes); }
```



Test Element Details

Custom code: test.SetDatapoolRowCursor

Class name: test.SetDatapoolRowCursor

Description:

Arguments

Custom code: test.OpenDatapool
Custom code: test.GetDatapoolRowCursor

RPT Workbench : Memory Optimization

Add following setting in RPT eclipse.ini

"-DRPTPageOutThresh=50",

"-DheapWatchFrequency=1", and

"-DcaptureStats=c:\stat_cap.xml"

RPTPageOutThresh will cause the workbench to page out (write to disk) data more frequently; this will cause the workbench to use slightly more CPU but substantially less heap space.

The **heapWatchFrequency** setting will add a counter for workbench heap size to the playback's statistics; this could be helpful in debugging.

captureStats=c:\stat_cap.xml is also a debug setting; it will cause one file per agent to be written to C:\. The files will have names like stat_cap_Agent1.xml, stat_cap_Agent2.xml

If the playback works successfully with these settings you could remove

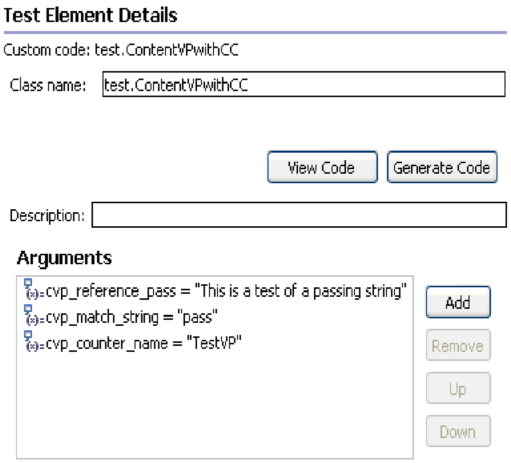
"**-DheapWatchFrequency=1**" and "**-DcaptureStats=c:\stat_cap.xml**" lines from the eclipse.ini but keep the "**-DRPTPageOutThresh=50**".

Verification Point : Custom Content VP

ContentVPwithCC()

- This custom code class searches the reference string for an exact instance of a string value. It increments the <counterName>_VPtotal counter every time it is called. It increments the <counterName>_VPpass every time a match is found and the <counterName>_VPfail every time a match is not found. These counters may be added to an existing or newly created custom report.
- ✓ Returned value : String value of true or false depending on whether a string match occurred
- Custom Code Exec Call Content

```
public String exec(ITestExecutionServices tes, String[] args) {
IStatisticsManager sm = tes.getStatisticsManager();
IStatTree st = sm.getStatTree();
IStatTree tranStat = st.getStat("Transactions", StatType.STRUCTURE);
IStatTree tvpStat = tranStat.getStat("Verification Points", StatType.STRUCTURE);
IStatTree myTranStat = tvpStat.getStat(args[2], StatType.STRUCTURE);
IScalar mySuccessVP = (IScalar)myTranStat.getStat("PASS", StatType.SCALAR);
IScalar myFailureVP = (IScalar)myTranStat.getStat("FAIL", StatType.SCALAR);
IScalar myTotalVP =   (IScalar)myTranStat.getStat("TOTAL", StatType.SCALAR);
myTotalVP.increment();
if (args[0].indexOf(args[1]) < 0)
myFailureVP.increment();
else
mySuccessVP.increment();
return null;}
```



Test Element Details

Custom code: test.ContentVPwithCC

Class name: test.ContentVPwithCC

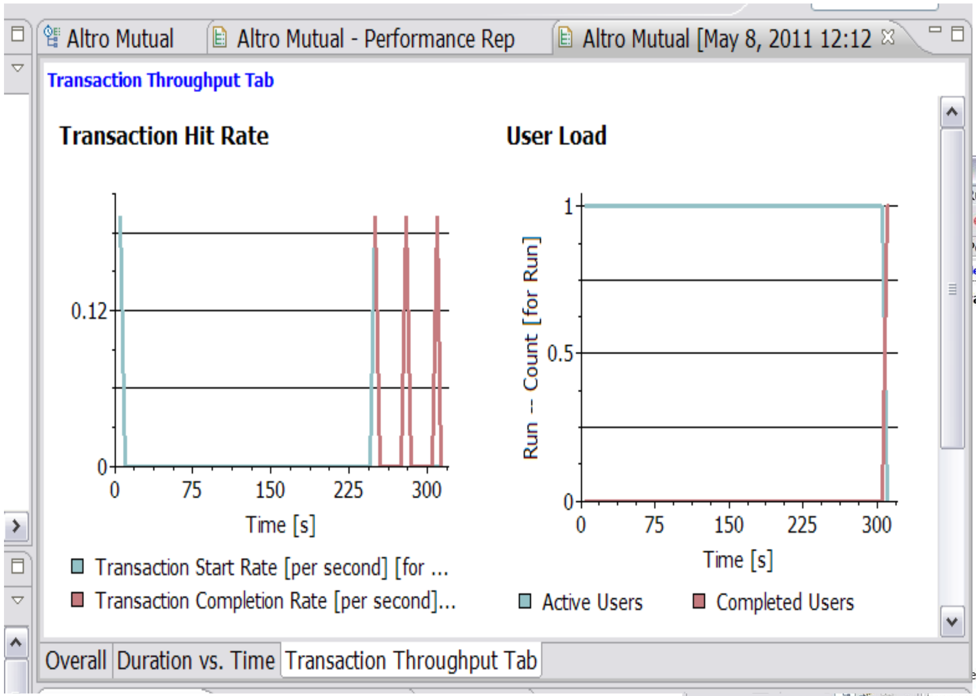
Description:

Arguments

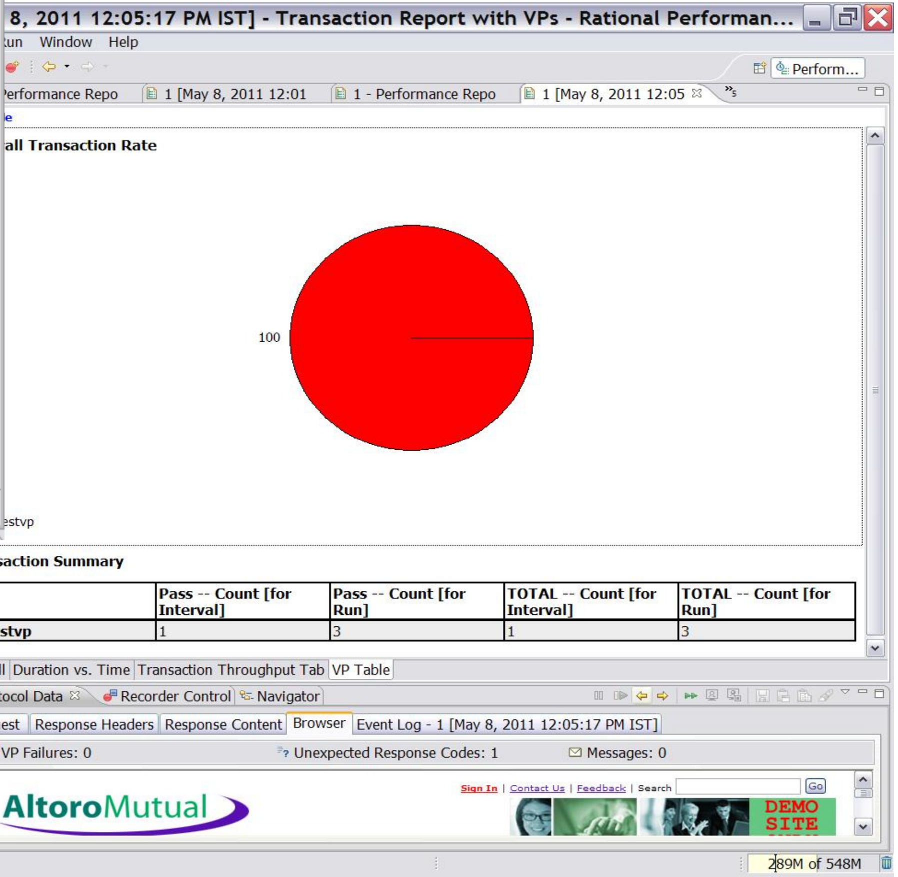
cvp_reference_pass = "This is a test of a passing string"
cvp_match_string = "pass"
cvp_counter_name = "TestVP"

Custom Report : Content VP

Customized Transaction Report with new report – VP Table


Altro Mutual
Altro Mutual - Performance Rep
Altro Mutual [May 8, 2011 12:12
Transaction Throughput Tab
Transaction Hit Rate
0.12
0
0
75
150
225
300
Time [s]
□ Transaction Start Rate [per second] [for ...
□ Transaction Completion Rate [per second]...
User Load
Run -- Count [for Run]
1
0.5
0
0
75
150
225
300
Time [s]
□ Active Users
□ Completed Users
Overall
Duration vs. Time
Transaction Throughput Tab

Out of Box Transaction Report



Transaction Summary

	Pass -- Count [for Interval]	Pass -- Count [for Run]	TOTAL -- Count [for Interval]	TOTAL -- Count [for Run]
testvp	1	3	1	3

VP Failures: 0 Unexpected Response Codes: 1 Messages: 0

DB Flash point

Have Multiple flash points on your DB for

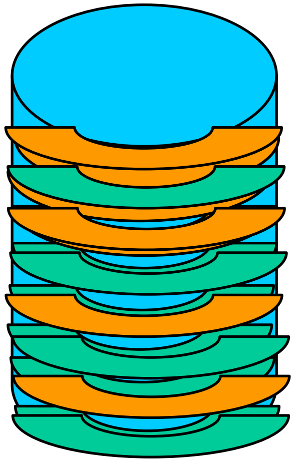
Persistence : Application Deployment

Preparation : Scripting and Smoke test

Measurement : Load Testing

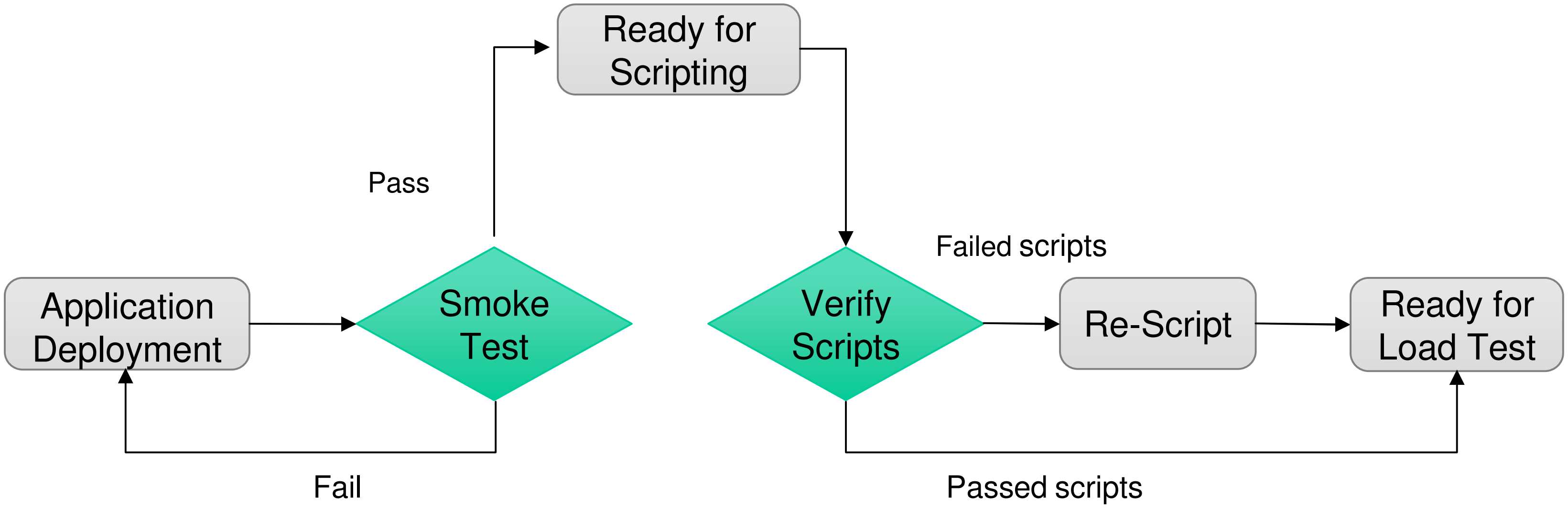
Test Data Preparation

Create testdata like customer, address, phone in DB



- Insert data as per the requirements
- Shuffle data within the database

Test Execution Procedure



Application Deployment

Smoke Test

Pass

Fail

Ready for Scripting

Verify Scripts

Failed scripts

Passed scripts

Re-Script

Ready for Load Test

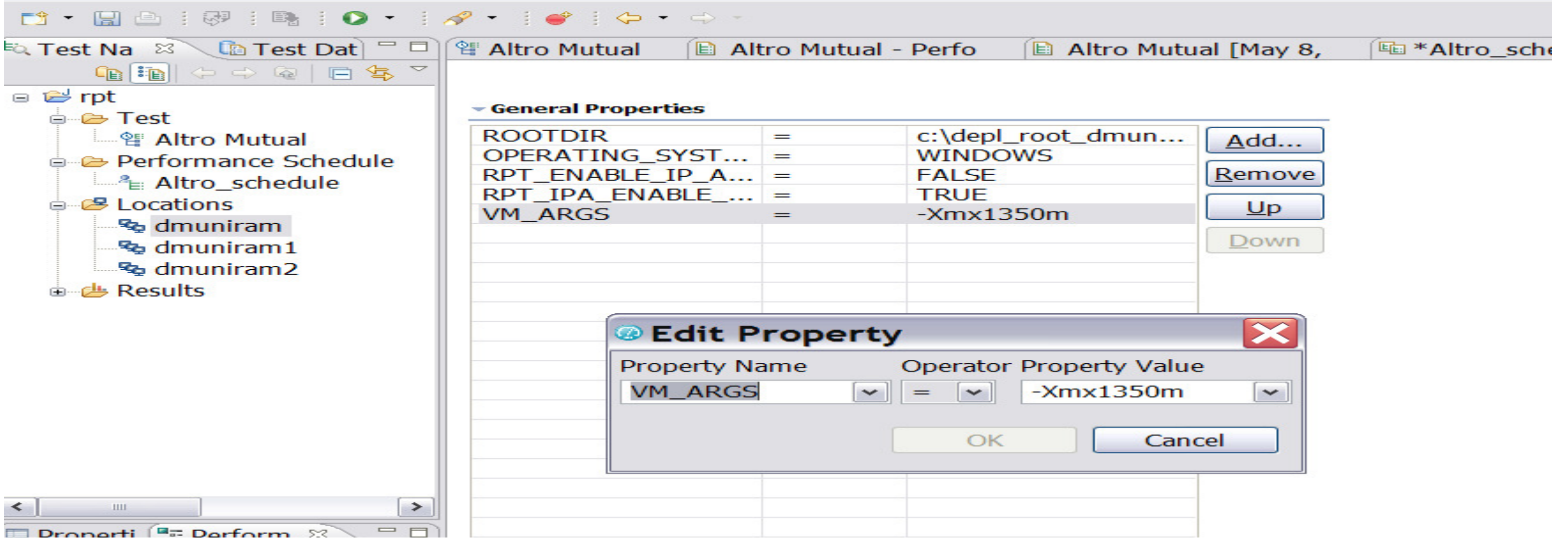
DB state: Persistent

DB state: Preparation

DB state: Measurement

Agent : Effective use of Agents

- Add VMARGS to increase default memory
- Depending on available physical Memory and load to be generated , determine how many agents are required
- Do a Run to check how many users each agent can support for a given workload with a target of 60-70% of CPU and Memory utilisation
- Optimum Value for VM_ARGS = -Xmx1350m



General Properties

ROOTDIR	=	c:\depl_root_dmun...
OPERATING_SYST...	=	WINDOWS
RPT_ENABLE_IP_A...	=	FALSE
RPT_IPA_ENABLE_...	=	TRUE
VM_ARGS	=	-Xmx1350m

Edit Property

Property Name	Operator	Property Value
VM_ARGS	=	-Xmx1350m

Agent : Vertical Scaling

- Use vertical scaling of agents to generate more load per box and utilize system resources more effectively.
- Create multiple locations with different host names or similar name and have different unique deployment root directory.
- In the Host file, map all the host names to same IP address of the actual box.

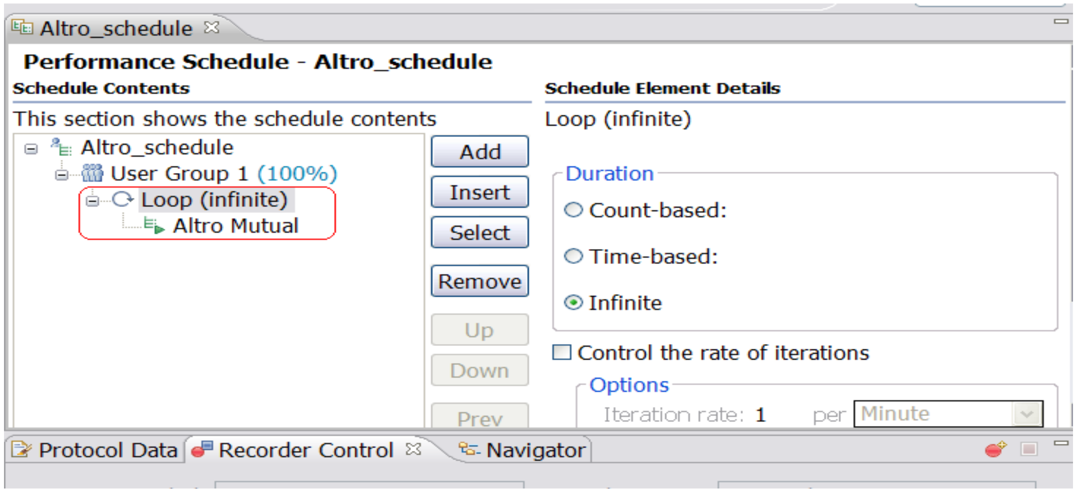
Example:

- Assume dmuniram is the host name of a physical windows machine
- Create multiple locations with similar hostnames like

dmuniram,dmuniram1,dmuniram2 etc however ensure the deployment root directory is different for each.

Schedule

- Always have outer loops in schedule rather than in Test itself.
- This will ensure connections are re-used and hence results would be more accurate
- Set log level to minimum or turn it off



Altro_schedule

Performance Schedule - Altro_schedule

Schedule Contents

This section shows the schedule contents

```
Altro_schedule
  User Group 1 (100%)
    Loop (infinite)
      Altro Mutual
```

Schedule Element Details

Loop (infinite)

Duration

○ Count-based:

○ Time-based:

● Infinite

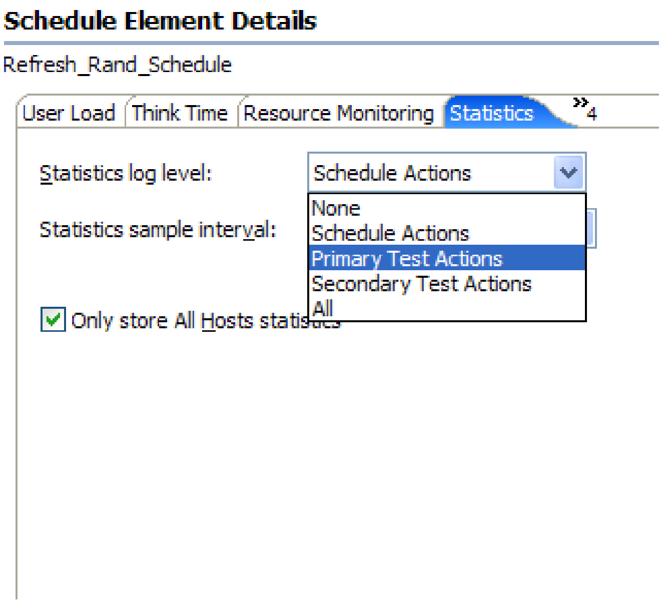
□ Control the rate of iterations

Options

Iteration rate: 1 per Minute

Schedule

- Set Statistics log level to “Primary Test Actions”
- Set sampling interval > 15 sec or appropriately based on user load



Schedule Element Details

Refresh_Rand_Schedule

Statistics log level: Schedule Actions

None
Schedule Actions
Primary Test Actions
Secondary Test Actions
All

Statistics sample interval:

☑ Only store All Hosts statis

Lessons learned?

Lessons learned around performance testing and tuning based on our experience

1 UnderStand Performance requirements and Goals clearly

2 Identify Issues and develop solutions at the initial phase

3 Repeat test multiple times after every single change

4 Performance test team and development team MUST work closely together to unearth and fix performance issues

Let‘s get started ...

www.ibm/software/rational

Daily iPod Touch giveaway

- □ Complete your session surveys online each day at a conference kiosk or on your Innovate 2011 Portal!
- □ Each day that you complete all of that day’s session surveys, your name will be entered to win the daily IPOD touch!
- □ On Wednesday be sure to complete your full conference evaluation to receive your free conference t-shirt!

SPONSORED BY

AllianceTech
Intelligent EVENTS

www.ibm/software/rational

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.