Rational on Rational,how Rational Software use Rational test tools to test ClearCase and ClearQuest

Paul Weiss
IBM Sr. Development Manager Performance and Reliability Team
Bob Ryder
IBM Sr. Development Manager Functional Verification Testing

Presented today by:
Paul Murray, Technical Representitive
pmurray@uk.ibm.com

IBM Rational Software Development Conference UK 2007

What keeps me **Rational**?

© 2007 IBM Corporation

Agenda

- IBM Rational ClearCase and ClearQuest
- Execution of Build Triggered Smoke Tests using Build Forge Triggers
- Tracking test progress using CQTM
- Testing the CC and CQ Clients using RFT
- Performance testing using RPT
- Performance testing using RFT

IBM Rational ClearCase and ClearQuest

Rational uses Rational - In-House Deployment (IHD)

- Pre-release versions of the 7.0.1 ClearCase and ClearQuest installed and deployed prior to eGA
  - We practice what we preach - We use Rational Tools!
  - Prior to 7.0.1 GA, two in-house releases were deployed (CC and CQ)
  - ClearCase Deployment
    - 500 developers 15 Replicas across 8 Geographic regions
    - ClearCase Development uses Network Attached and CIFS software based storage solutions for Vob and View storage.
    - Full CC and CCRC users
  - ClearQuest Deployment
    - 3000 users and 12 replicas Full Eclipse Client and CQWeb

Running Build Triggered Smoke Tests

Build Triggered Smoke Tests

- Initial Smoke tests on Core Functionality of CC and CQ
  - Smoke tests are perl based command line driven
- After Core tests pass – Smoke Tests on Clients – using RFT
  - CCRC – ClearCase Remote client
  - CQEC – ClearQuest Eclipse client (replaces windows based client)
  - CQWeb – ClearQuest Web Client
  - CCWeb – ClearCase Web Client

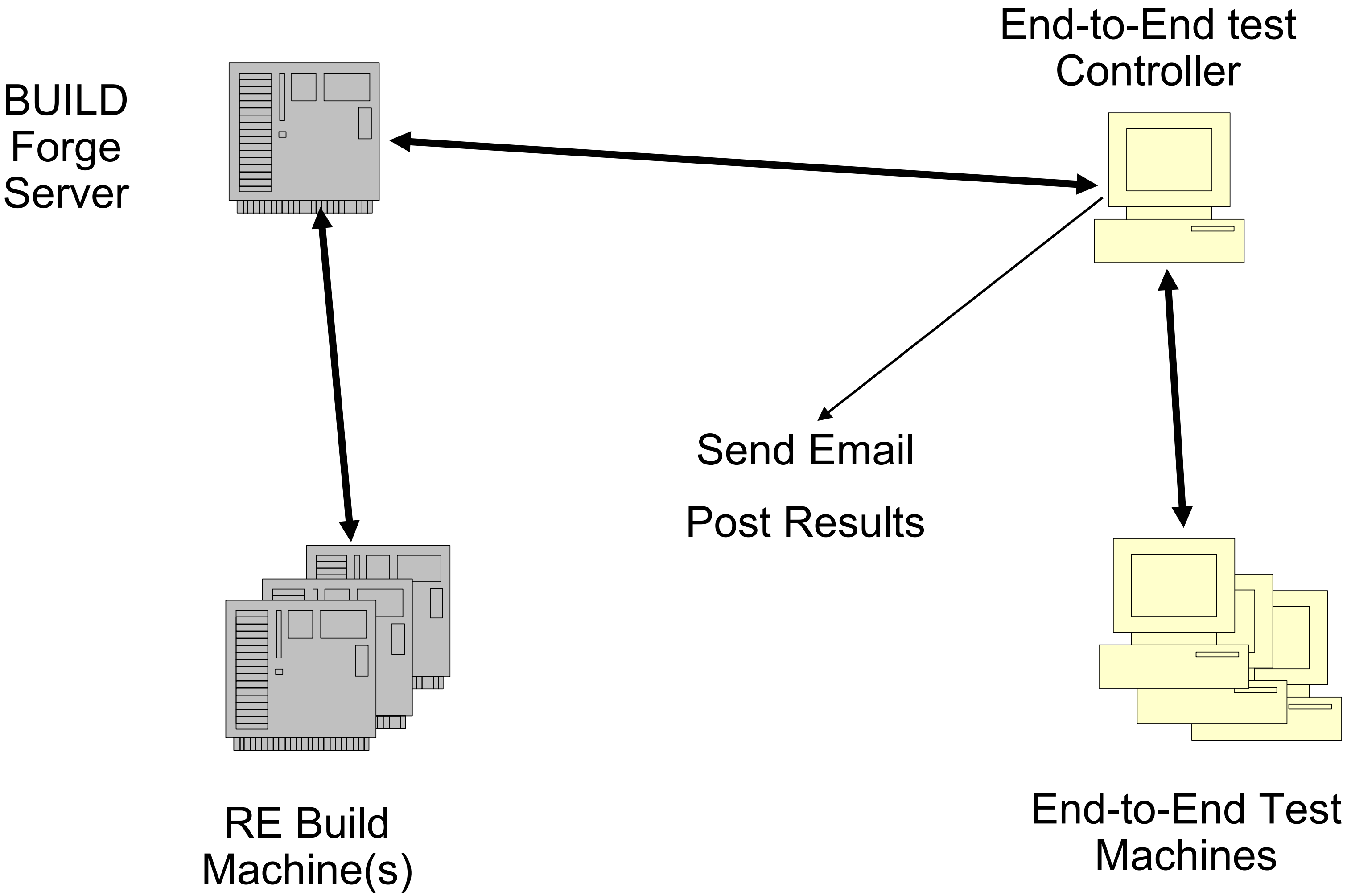
Use Build Forge to Trigger Smoke test

- Triggers are kicked off at the completion of an installable build
- Smoke tests are executed in series
- Triggers execute the primary core/server functionality tests first
- Triggers then kick off the RFT tests for ClearCase/ClearQuest Clients
- Analyzes and posts results to an internal website

Trigger Details

- Sends email to QE to announce tests have begun
- Checks for an available target machine
- Uninstalls previous versions of the product
- Installs product to be tested
- Configures product components for testing
- Executes smoke tests
- Analyzes results
- Posts results and sends email

Smoke Test Configuration



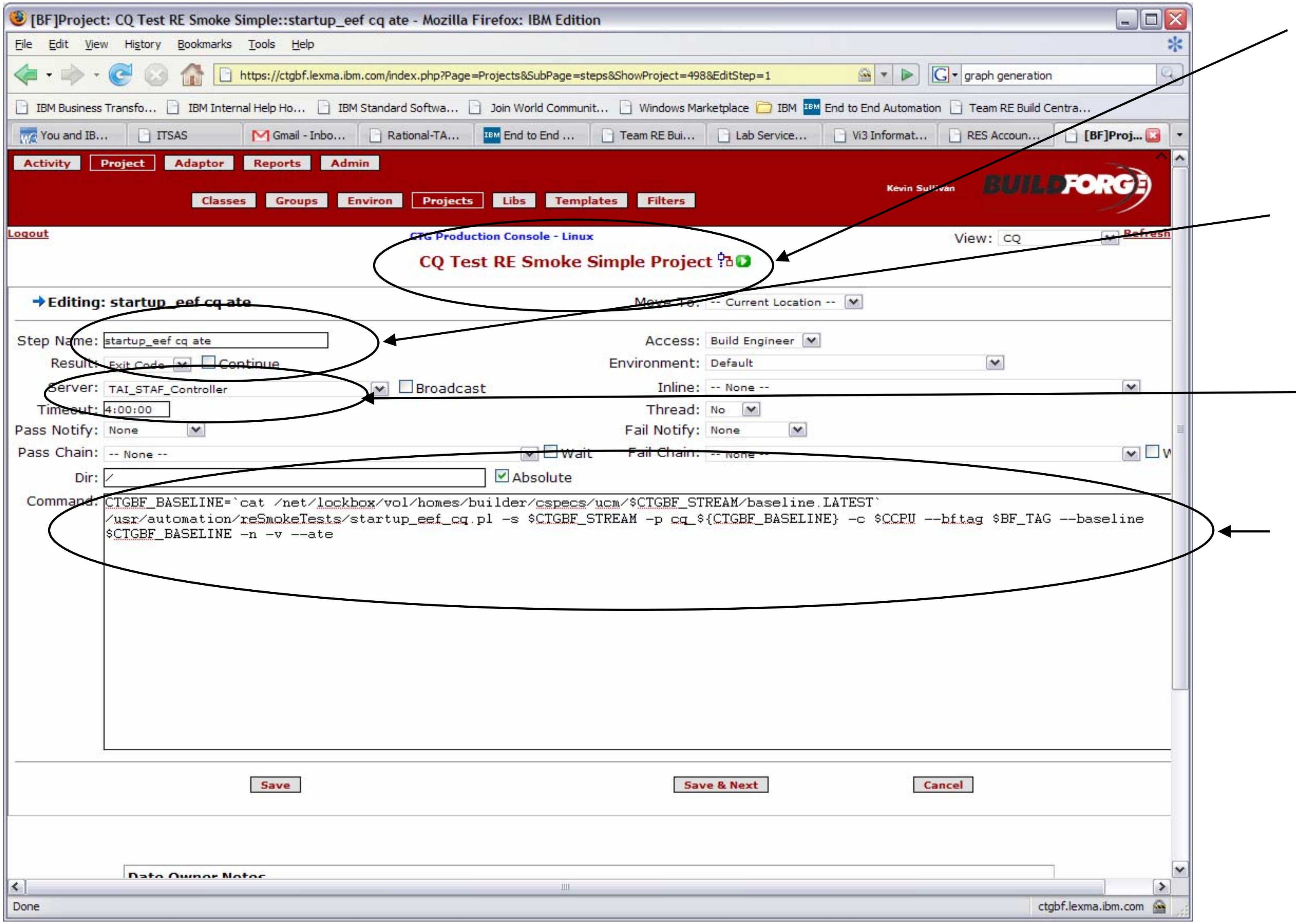
Setup Smoke Tests – End to End Test Controller

- Install the Build Forge client
- Set Build Forge client up as a service that accepts communication from Build Forge process on RE machine.

Smoke Test Setup – Build Forge Server

- Create a host communication profile in Build Forge process on RE machine.
- Verify Build Forge can log into E2E automation machine for proper communication.
- Create a project within Build Forge that points to E2E test controller
- Project contains command lines which will be executed on the E2E test controller
- Project needs to be included in the Build/Spin Project to be executed upon successful completion of the build/spin
- When the End to End automation process has completed the E2E test controller accepts the test results and notifies the Build Forge Server of the results.

Create a Build Forge Project


[BF]Project: CQ Test RE Smoke Simple::startup_eef cq ate - Mozilla Firefox: IBM Edition
File Edit View History Bookmarks Tools Help
https://ctgbf.lexma.ibm.com/index.php?Page=Projects&SubPage=steps&ShowProject=498&EditStep=1
graph generation
IBM Business Transfo... IBM Internal Help Ho... IBM Standard Softwa... Join World Communit... Windows Marketplace IBM End to End Automation Team RE Build Centra...
You and IB... ITSAS Gmail - Inbo... Rational-TA... End to End ... Team RE Bui... Lab Service... Vi3 Informat... RES Accoun... [BF]Proj...
Activity Project Adaptor Reports Admin
Classes Groups Environ Projects Libs Templates Filters
Kevin Sullivan
BUILDFORGE
Logout
CTG Production Console - Linux
Refresh
Save Save & Next Cancel
Date Owner Notes
Done
ctgbf.lexma.ibm.com

View: CQ

CQ Test RE Smoke Simple Project

➔ Editing: startup_eef cq ate

Move To: -- Current Location --

Step Name:	startup_eef cq ate	Access:	Build Engineer
Result:	Exit Code □ Continue	Environment:	Default
Server:	TAI_STAF_Controller □ Broadcast	Inline:	-- None --
Timeout:	4:00:00	Thread:	No
Pass Notify:	None	Fail Notify:	None
Pass Chain:	-- None -- □ Wait	Fail Chain:	-- None -- □ W
Dir:	/ ☑ Absolute		

Command:

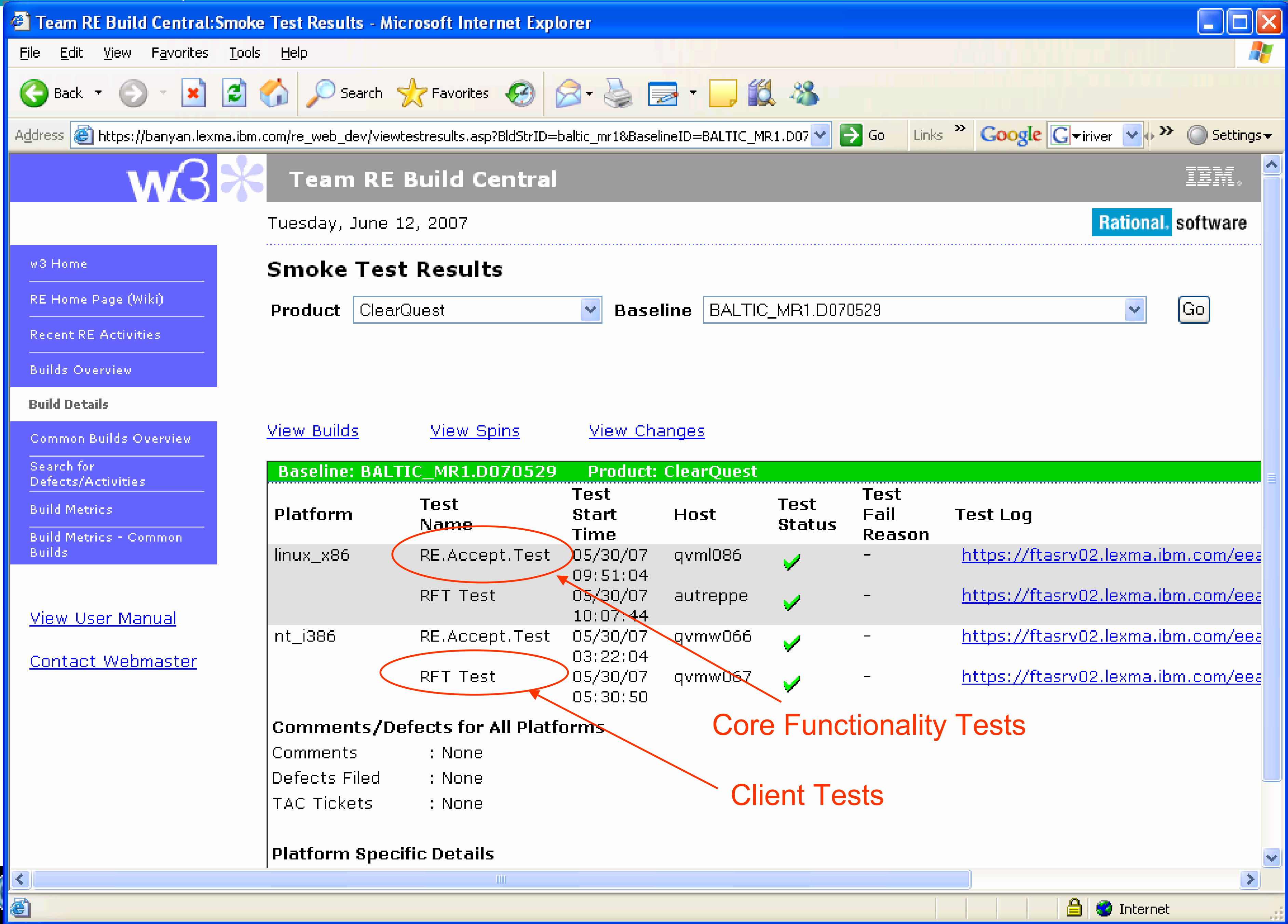
```
CTGBF_BASELINE=`cat /net/lockbox/vol/homes/builder/cspecs/ucm/$CTGBF_STREAM/baseline.LATEST`
/usr/automation/reSmokeTests/startup_eef_cq.pl -s $CTGBF_STREAM -p cq_${CTGBF_BASELINE} -c $CCPU --bftag $BF_TAG --baseline
$CTGBF_BASELINE -n -v --ate
```

Project Name

Project Step Name

E2E test controller

Command to execute



https://banyan.lexma.ibm.com/re_web_dev/viewtestresults.asp?BldStrID=baltic_mr1&BaselineID=BALTIC_MR1.D07

Team RE Build Central

Tuesday, June 12, 2007

- w3 Home
- RE Home Page (Wiki)
- Recent RE Activities
- Builds Overview

Build Details

- Common Builds Overview
- Search for Defects/Activities
- Build Metrics
- Build Metrics - Common Builds

View User Manual

Contact Webmaster

Smoke Test Results

Product ClearQuest **Baseline** BALTIC_MR1.D070529 Go

View Builds View Spins View Changes

Baseline: BALTIC_MR1.D070529 Product: ClearQuest

Platform	Test Name	Test Start Time	Host	Test Status	Test Fail Reason	Test Log
linux_x86	RE.Accept.Test	05/30/07 09:51:04	qvml086	✓	-	https://ftasrv02.lexma.ibm.com/eea
	RFT Test	05/30/07 10:07:44	autreppe	✓	-	https://ftasrv02.lexma.ibm.com/eea
nt_i386	RE.Accept.Test	05/30/07 03:22:04	qvmw066	✓	-	https://ftasrv02.lexma.ibm.com/eea
	RFT Test	05/30/07 05:30:50	qvmw067	✓	-	https://ftasrv02.lexma.ibm.com/eea

Comments/Defects for All Platforms

Comments : None

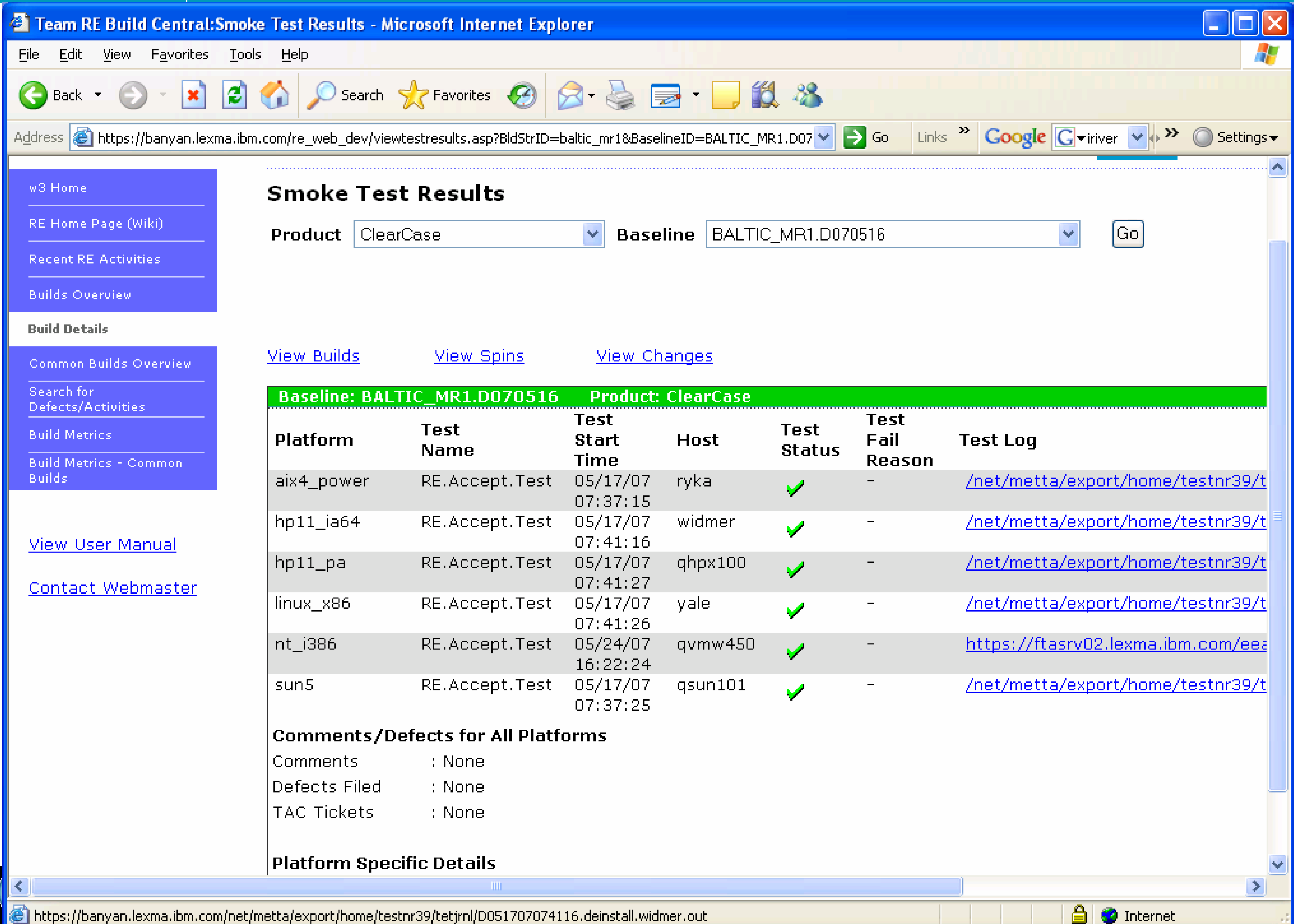
Defects Filed : None

TAC Tickets : None

Platform Specific Details

Core Functionality Tests

Client Tests


Team RE Build Central:Smoke Test Results - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Address
https://banyan.lexma.ibm.com/re_web_dev/viewtestresults.asp?BldStrID=baltic_mr1&BaselineID=BALTIC_MR1.D07
Go
Links
Google
iriver
Settings
w3 Home
RE Home Page (Wiki)
Recent RE Activities
Builds Overview
Build Details
Common Builds Overview
Search for Defects/Activities
Build Metrics
Build Metrics - Common Builds
View User Manual
Contact Webmaster
Internet

Smoke Test Results

Product ClearCase **Baseline** BALTIC_MR1.D070516 Go

View Builds View Spins View Changes

Baseline: BALTIC_MR1.D070516 Product: ClearCase

Platform	Test Name	Test Start Time	Host	Test Status	Test Fail Reason	Test Log
aix4_power	RE.Accept.Test	05/17/07 07:37:15	ryka	✓	-	/net/metta/export/home/testnr39/t
hp11_ia64	RE.Accept.Test	05/17/07 07:41:16	widmer	✓	-	/net/metta/export/home/testnr39/t
hp11_pa	RE.Accept.Test	05/17/07 07:41:27	qhpx100	✓	-	/net/metta/export/home/testnr39/t
linux_x86	RE.Accept.Test	05/17/07 07:41:26	yale	✓	-	/net/metta/export/home/testnr39/t
nt_i386	RE.Accept.Test	05/24/07 16:22:24	qvmw450	✓	-	https://ftasrv02.lexma.ibm.com/eea
sun5	RE.Accept.Test	05/17/07 07:37:25	qsun101	✓	-	/net/metta/export/home/testnr39/t

Comments/Defects for All Platforms

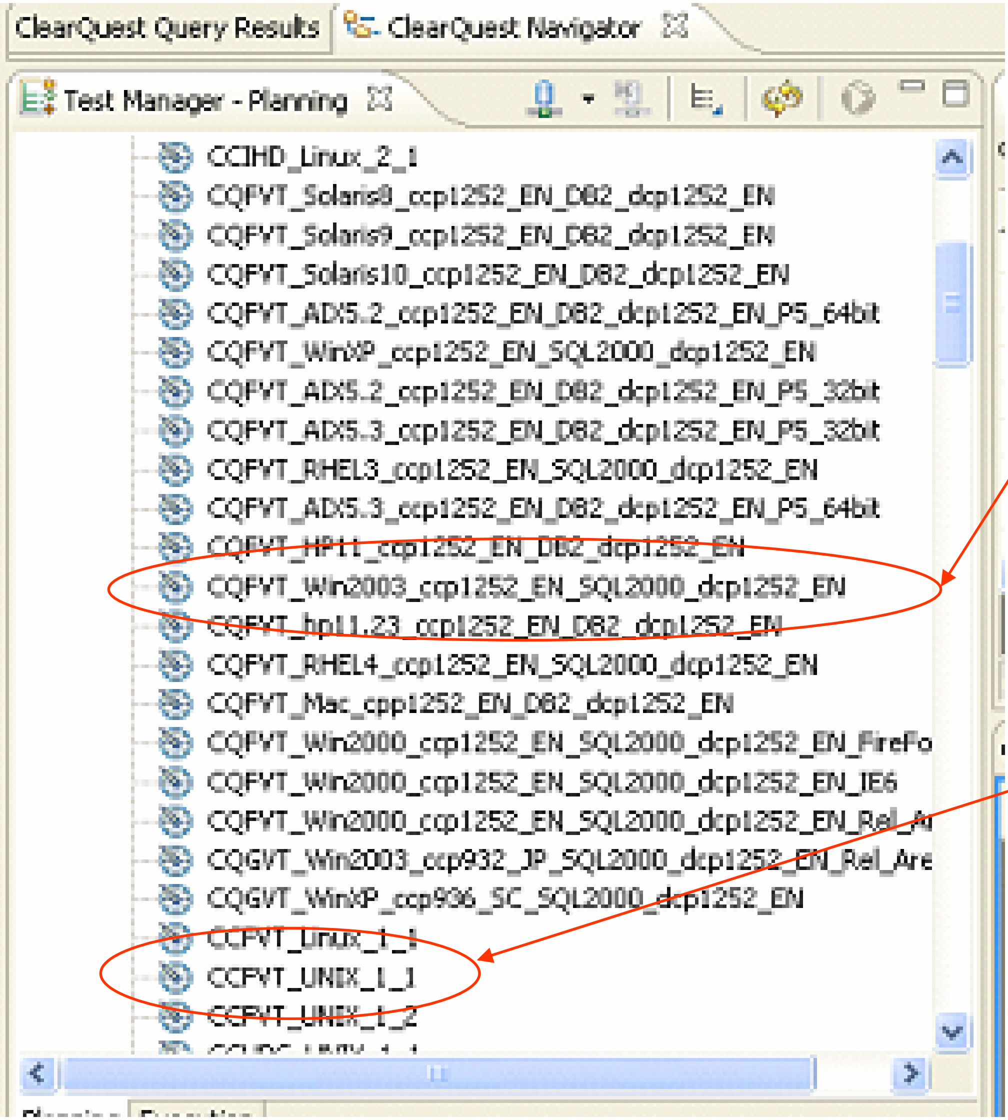
Comments : None
Defects Filed : None
TAC Tickets : None

Platform Specific Details

https://banyan.lexma.ibm.com/net/metta/export/home/testnr39/tetjrnl/D051707074116.deinstall.widmer.out

Tracking Progress with CQTM

CQTM – Setting up configurations



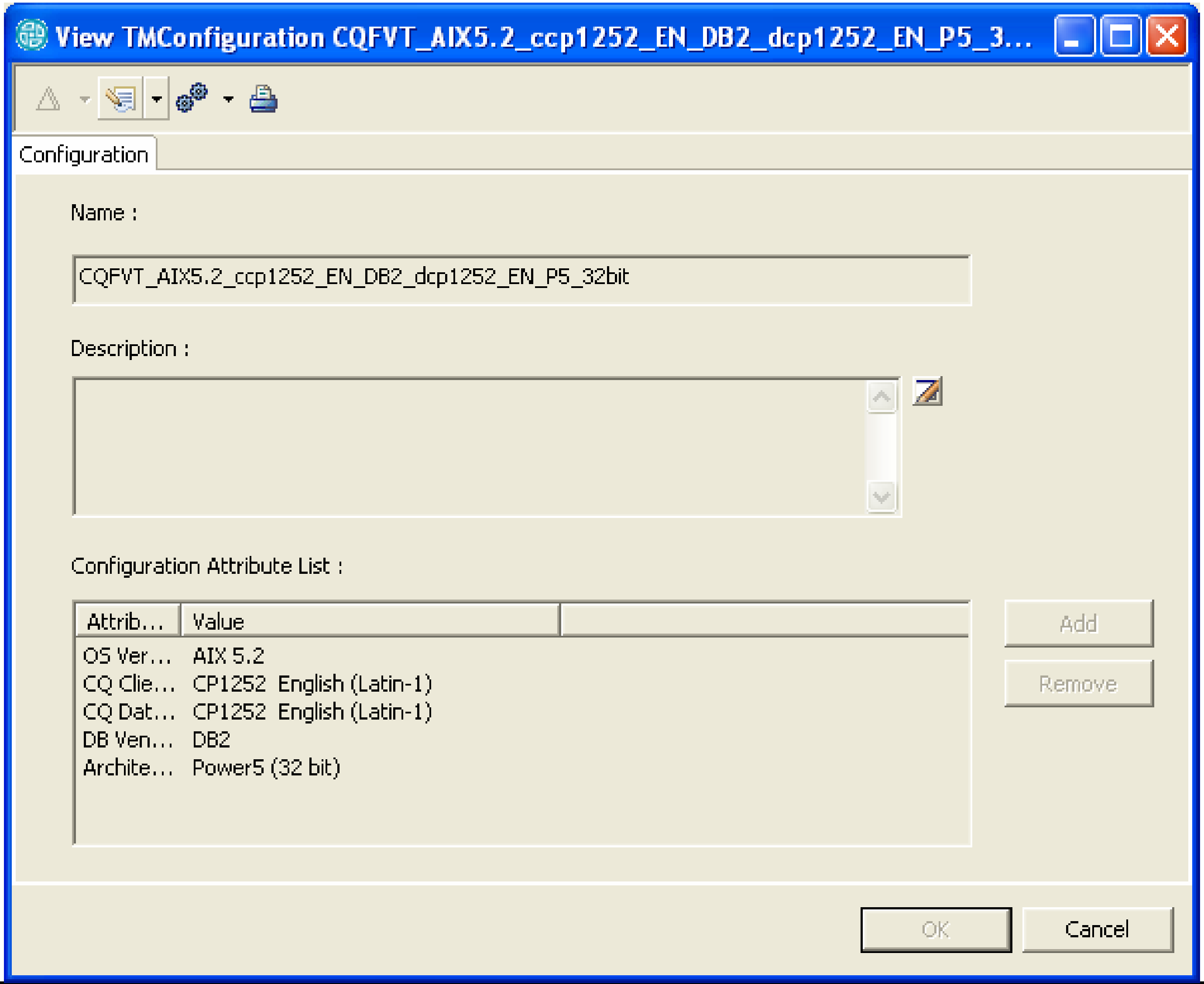
CCIHD_Linux_2_1
CQFVT_Solaris8_ccp1252_EN_DB2_dcp1252_EN
CQFVT_Solaris9_ccp1252_EN_DB2_dcp1252_EN
CQFVT_Solaris10_ccp1252_EN_DB2_dcp1252_EN
CQFVT_AIX5.2_ccp1252_EN_DB2_dcp1252_EN_P5_64bit
CQFVT_WinXP_ccp1252_EN_SQL2000_dcp1252_EN
CQFVT_AIX5.2_ccp1252_EN_DB2_dcp1252_EN_P5_32bit
CQFVT_AIX5.3_ccp1252_EN_DB2_dcp1252_EN_P5_32bit
CQFVT_RHEL3_ccp1252_EN_SQL2000_dcp1252_EN
CQFVT_AIX5.3_ccp1252_EN_DB2_dcp1252_EN_P5_64bit
CQFVT_HP11_ccp1252_EN_DB2_dcp1252_EN
CQFVT_Win2003_ccp1252_EN_SQL2000_dcp1252_EN
CQFVT_hp11.23_ccp1252_EN_DB2_dcp1252_EN
CQFVT_RHEL4_ccp1252_EN_SQL2000_dcp1252_EN
CQFVT_Mac_cpp1252_EN_DB2_dcp1252_EN
CQFVT_Win2000_ccp1252_EN_SQL2000_dcp1252_EN_FireFo
CQFVT_Win2000_ccp1252_EN_SQL2000_dcp1252_EN_IE6
CQFVT_Win2000_ccp1252_EN_SQL2000_dcp1252_EN_Rel_A
CQGVT_Win2003_ccp932_JP_SQL2000_dcp1252_EN_Rel_Are
CQGVT_WinXP_ccp936_SC_SQL2000_dcp1252_EN
CCFVT_Linux_1_1
CCFVT_UNIX_1_1
CCFVT_UNIX_1_2

What's in a Name?

- Win 2003 Client
- Client Code Page 1252
- SQL Server 2000 Database
- Data Base Code Page 1252

- Unix Client/Server
- Configuration 1

CQTM – Setting up configurations



View TMConfiguration CQFVT_AIX5.2_ccp1252_EN_DB2_dcp1252_EN_P5_3...
Configuration

Name :

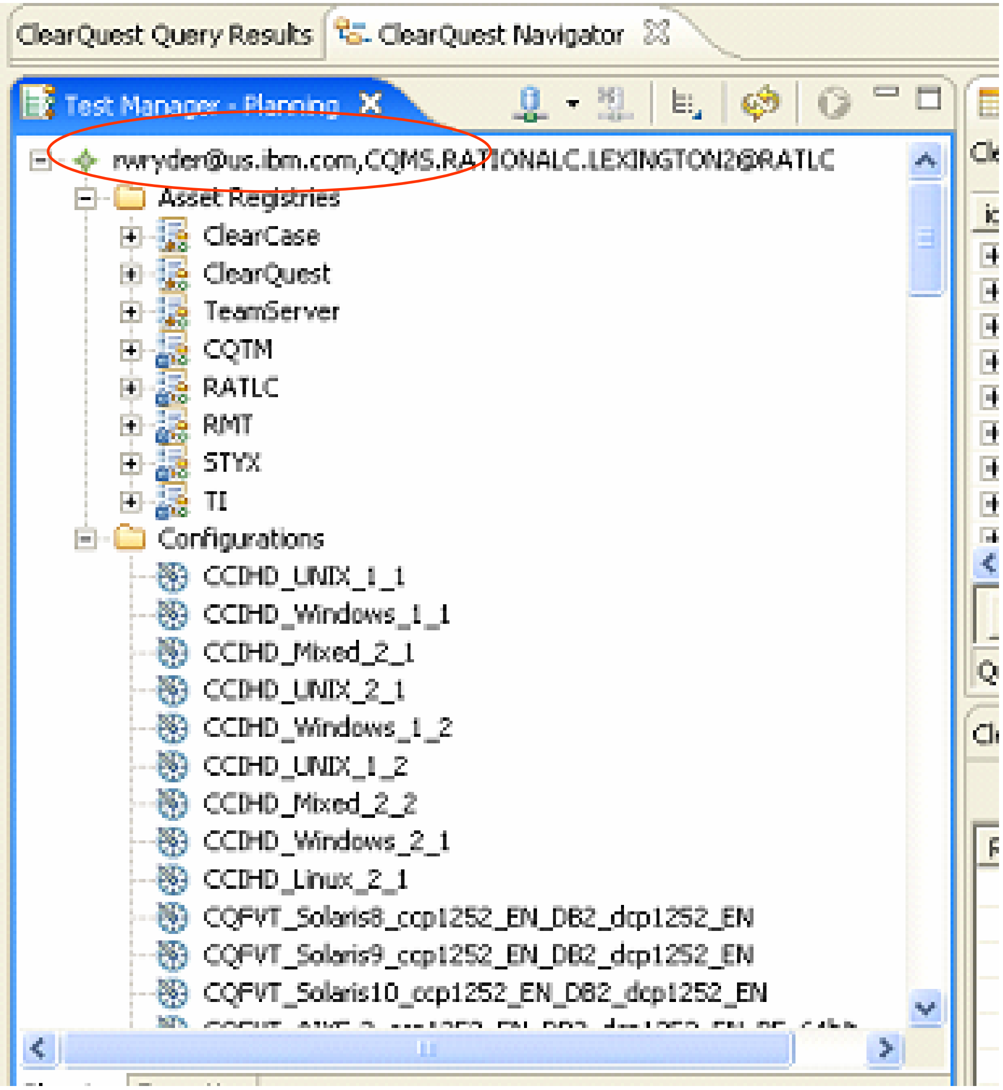
CQFVT_AIX5.2_ccp1252_EN_DB2_dcp1252_EN_P5_32bit

Description :

Configuration Attribute List :

Attrib...	Value
OS Ver...	AIX 5.2
CQ Clie...	CP1252 English (Latin-1)
CQ Dat...	CP1252 English (Latin-1)
DB Ven...	DB2
Archite...	Power5 (32 bit)

Adding Test Cases

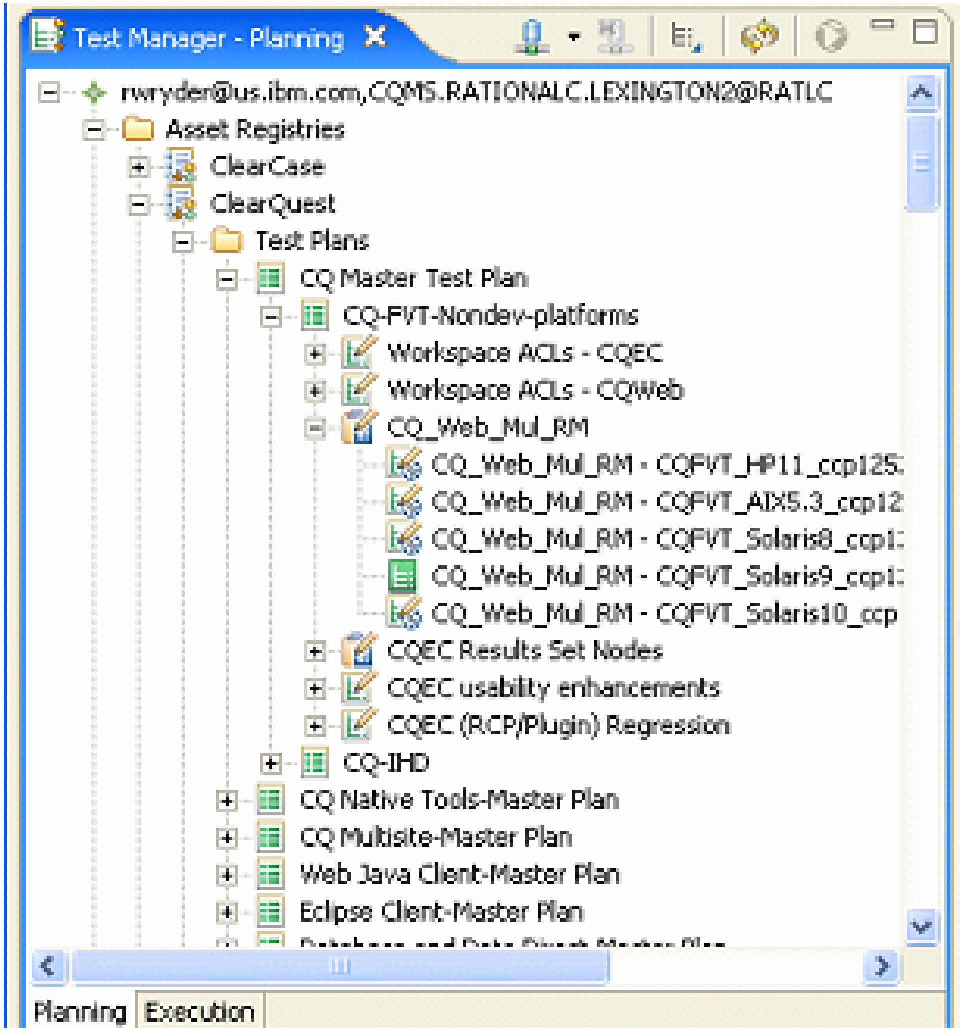


- rwryder@us.ibm.com,CQMS.RATIONALC.LEXINGTON2@RATLC
  - Asset Registries
    - ClearCase
    - ClearQuest
    - TeamServer
    - CQTM
    - RATLC
    - RMT
    - STYX
    - TI
  - Configurations
    - CCIHD_UNIX_1_1
    - CCIHD_Windows_1_1
    - CCIHD_Mixed_2_1
    - CCIHD_UNIX_2_1
    - CCIHD_Windows_1_2
    - CCIHD_UNIX_1_2
    - CCIHD_Mixed_2_2
    - CCIHD_Windows_2_1
    - CCIHD_Linux_2_1
    - CQFVT_Solaris8_ccp1252_EN_DB2_dcp1252_EN
    - CQFVT_Solaris9_ccp1252_EN_DB2_dcp1252_EN
    - CQFVT_Solaris10_ccp1252_EN_DB2_dcp1252_EN

Example

- Using CQTM to track new feature testing
- On AIX, Sun, and Solaris Platforms

Drilling down on test plans – Configured test cases

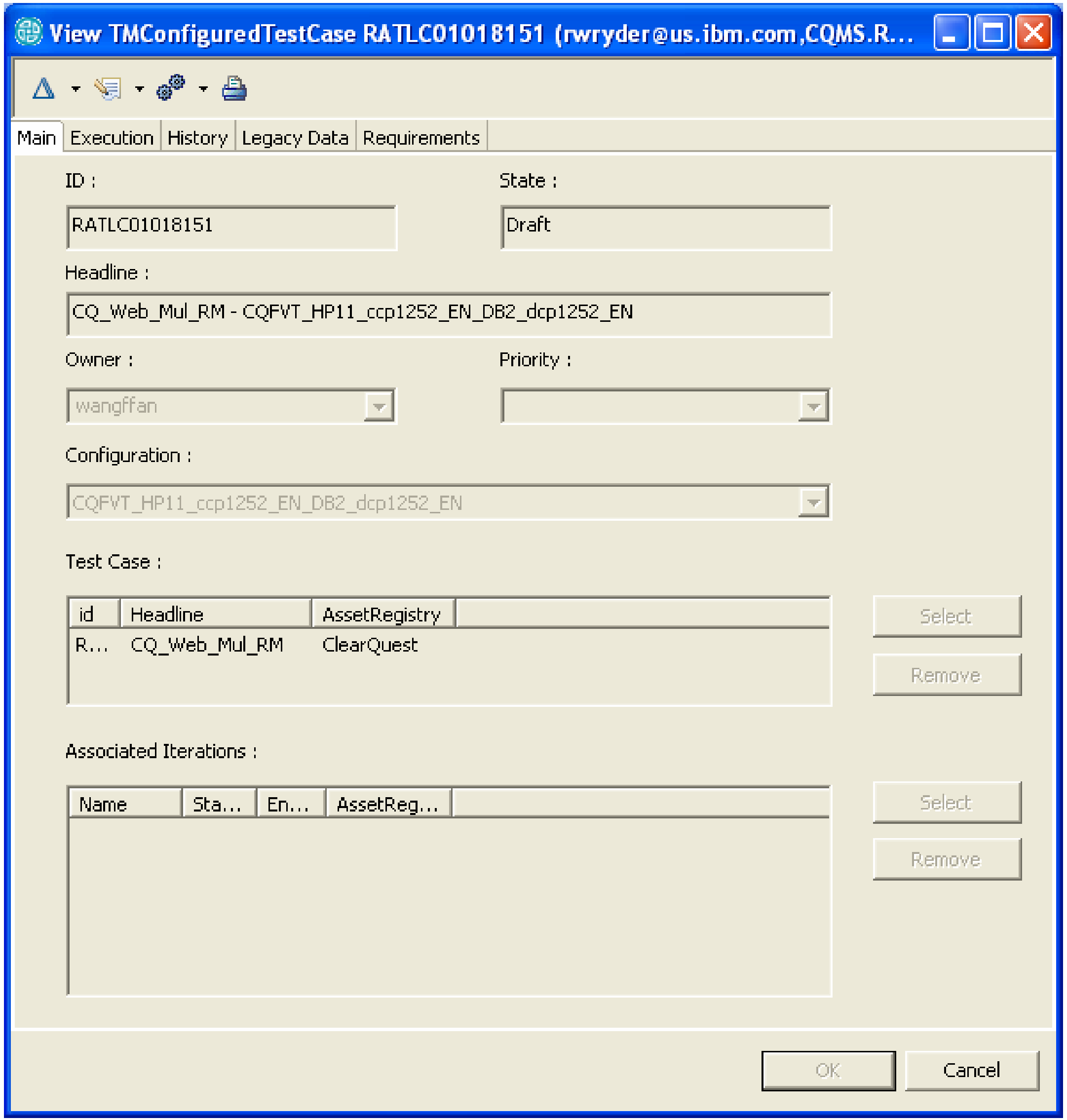


```
Test Manager - Planning
rwryder@us.ibm.com,CQMS.RATIONALC.LEXINGTON2@RATLC
  Asset Registries
    ClearCase
    ClearQuest
      Test Plans
        CQ Master Test Plan
          CQ-FVT-Nondev-platforms
            Workspace ACLs - CQEC
            Workspace ACLs - CQWeb
            CQ_Web_Mul_RM
              CQ_Web_Mul_RM - CQFVT_HP11_ccp125:
              CQ_Web_Mul_RM - CQFVT_AIX5.3_ccp12
              CQ_Web_Mul_RM - CQFVT_Solaris8_ccp1:
              CQ_Web_Mul_RM - CQFVT_Solaris9_ccp1:
              CQ_Web_Mul_RM - CQFVT_Solaris10_ccp
            CQEC Results Set Nodes
            CQEC usabilty enhancements
            CQEC (RCP/Plugin) Regression
          CQ-IHD
        CQ Native Tools-Master Plan
        CQ Multisite-Master Plan
        Web Java Client-Master Plan
        Eclipse Client-Master Plan
Planning | Execution
```

CQWEB Multiple Request Managers

- Testing on 5 additional configurations
- HP 11
- AIX
- 3 Solaris

Configured Test Cases



View TMConfiguredTestCase RATLC01018151 (rwryder@us.ibm.com,CQMS.R...
Main
Execution
History
Legacy Data
Requirements

ID : RATLC01018151

State : Draft

Headline : CQ_Web_Mul_RM - CQFVT_HP11_ccp1252_EN_DB2_dcp1252_EN

Owner : wangffan

Priority :

Configuration : CQFVT_HP11_ccp1252_EN_DB2_dcp1252_EN

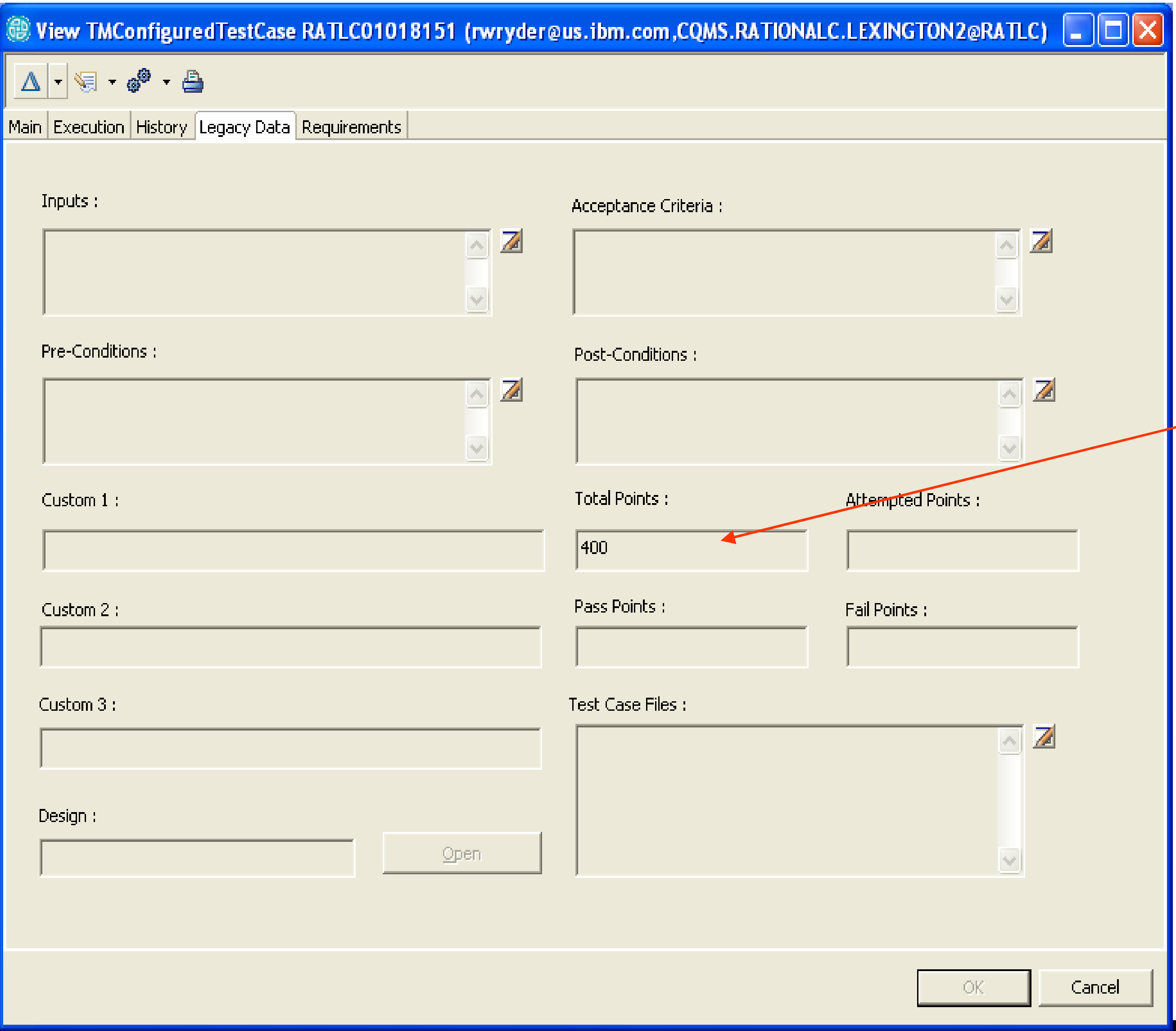
Test Case :

id	Headline	AssetRegistry
R...	CQ_Web_Mul_RM	ClearQuest

Associated Iterations :

Name	Sta...	En...	AssetReg...

Test Case Progress



View TMConfiguredTestCase RATLC01018151 (rwryder@us.ibm.com,CQMS.RATIONALC.LEXINGTON2@RATLC)

Inputs :

Acceptance Criteria :

Pre-Conditions :

Post-Conditions :

Custom 1 :

Total Points : 400

Attempted Points :

Custom 2 :

Pass Points :

Fail Points :

Custom 3 :

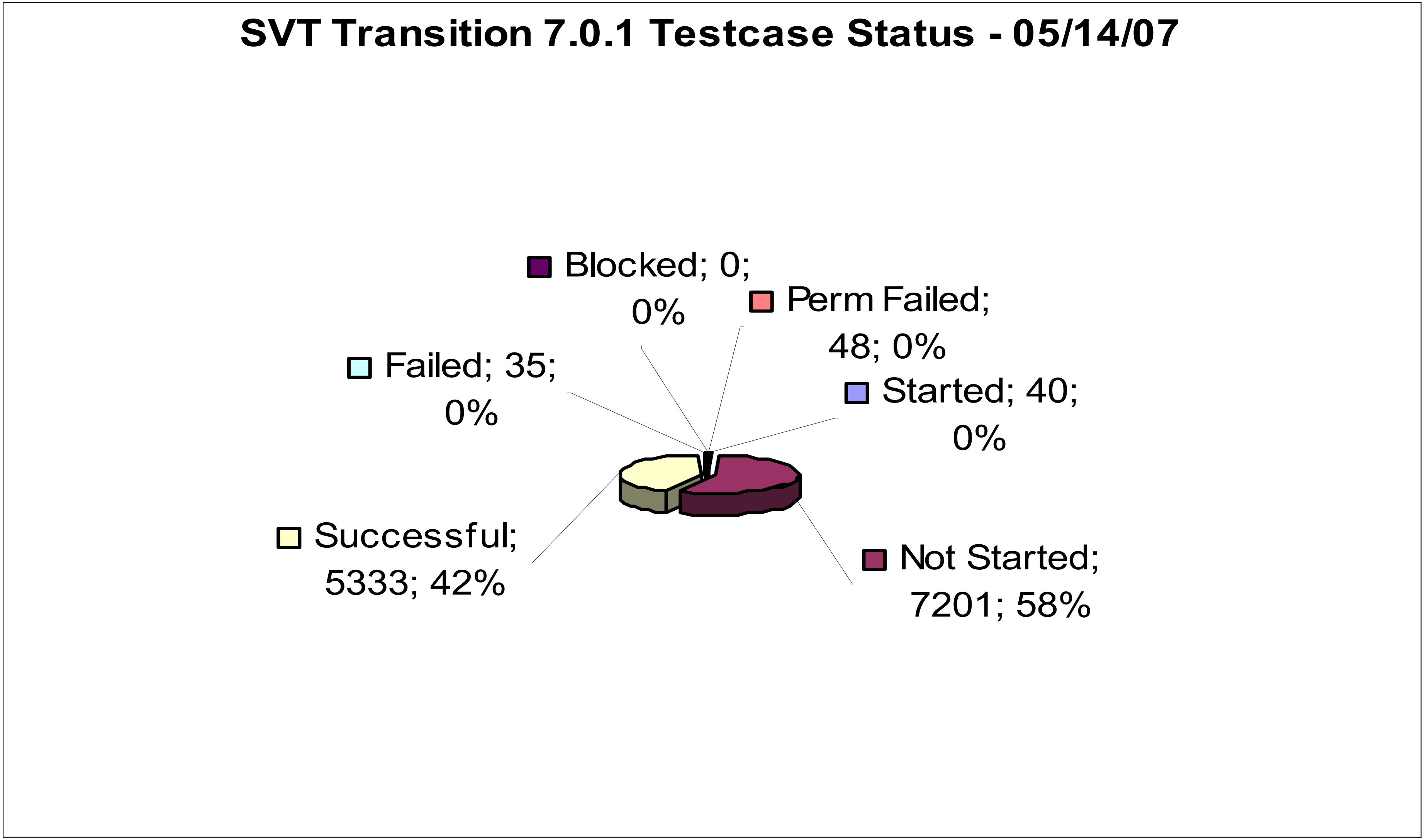
Test Case Files :

Design :

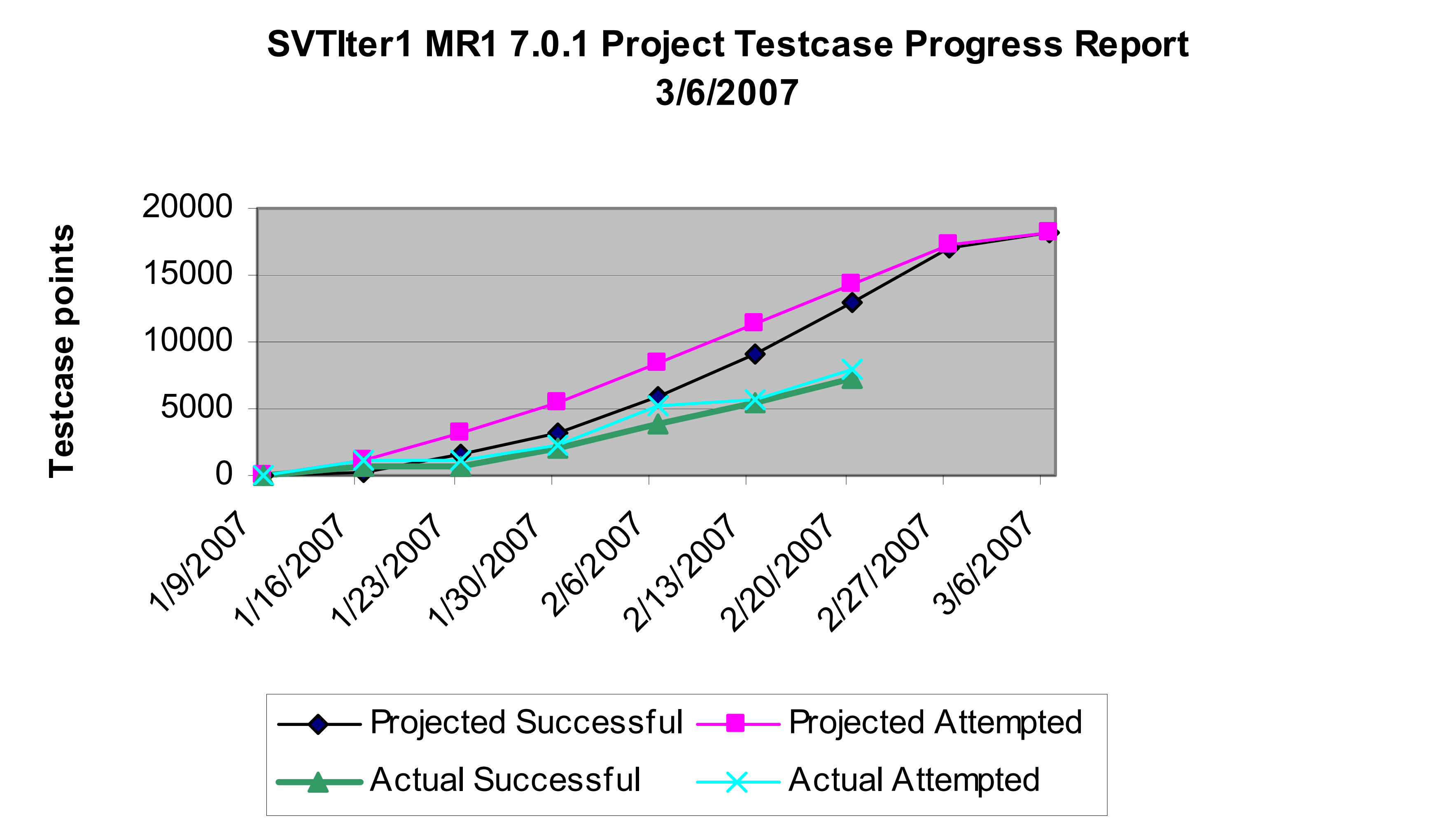
Used the points system to track progress

10 points = 1 hour

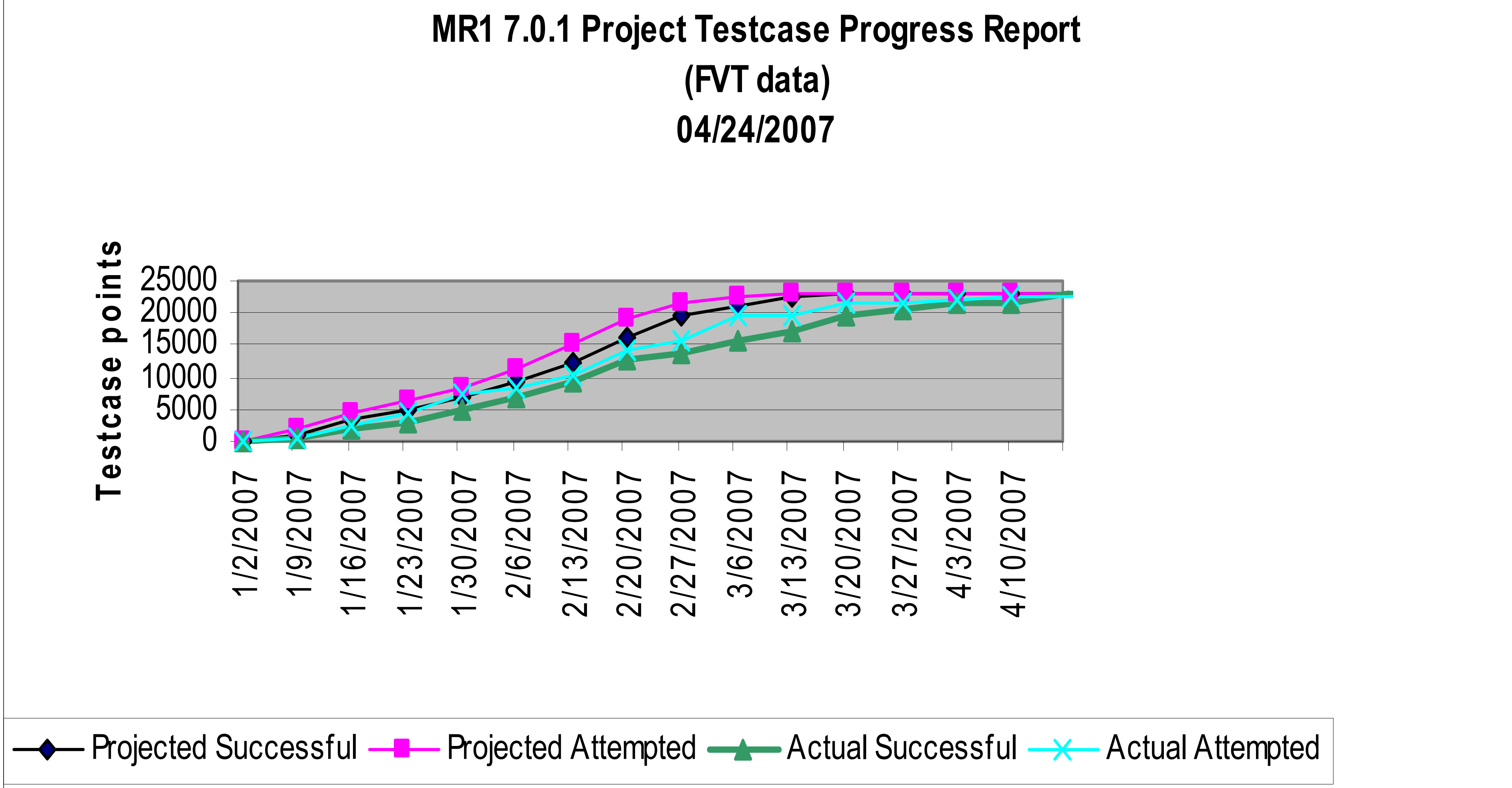
Reporting using CQTM



Reporting examples using CQTM

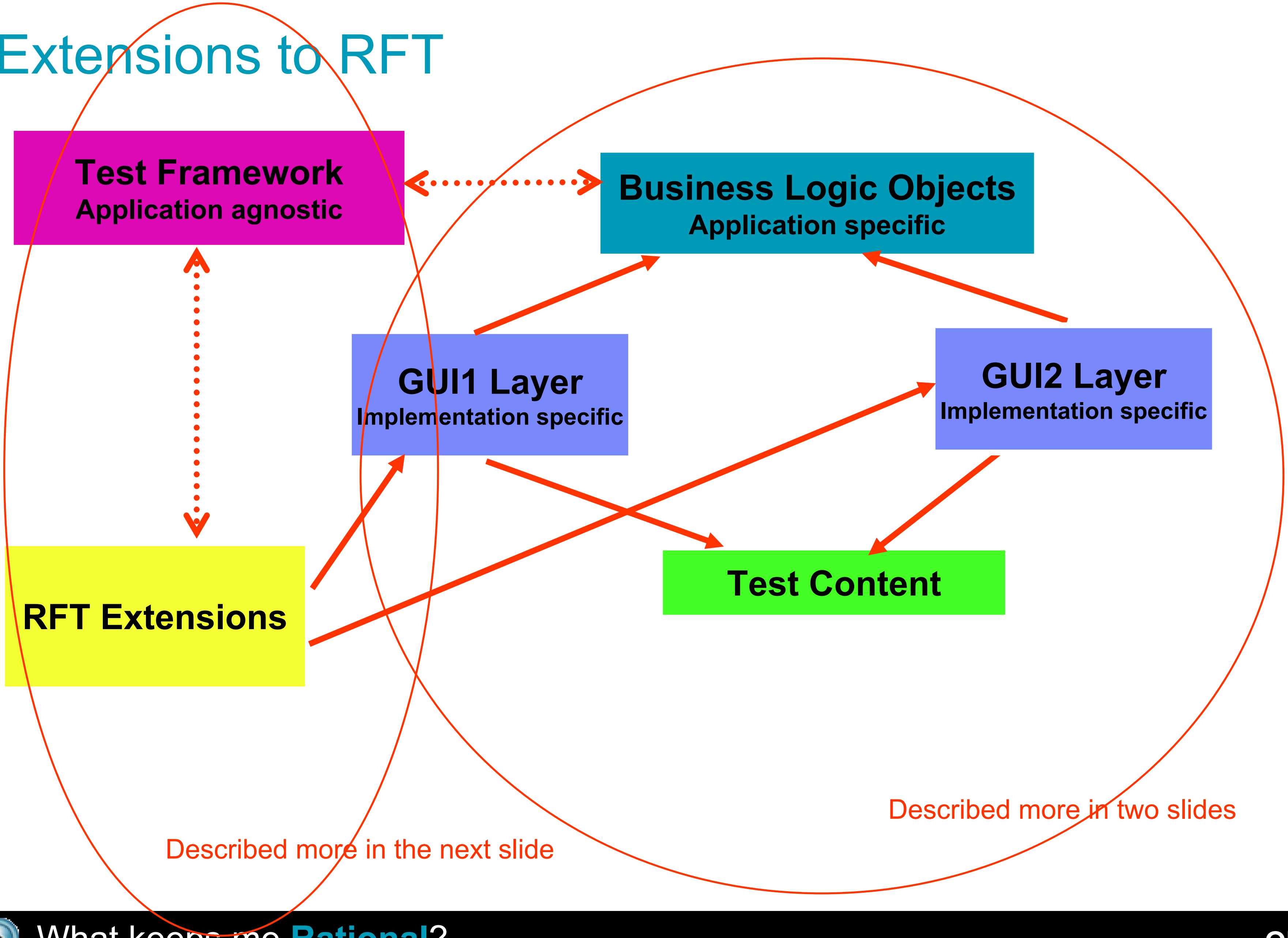


Reporting Examples using CQTM

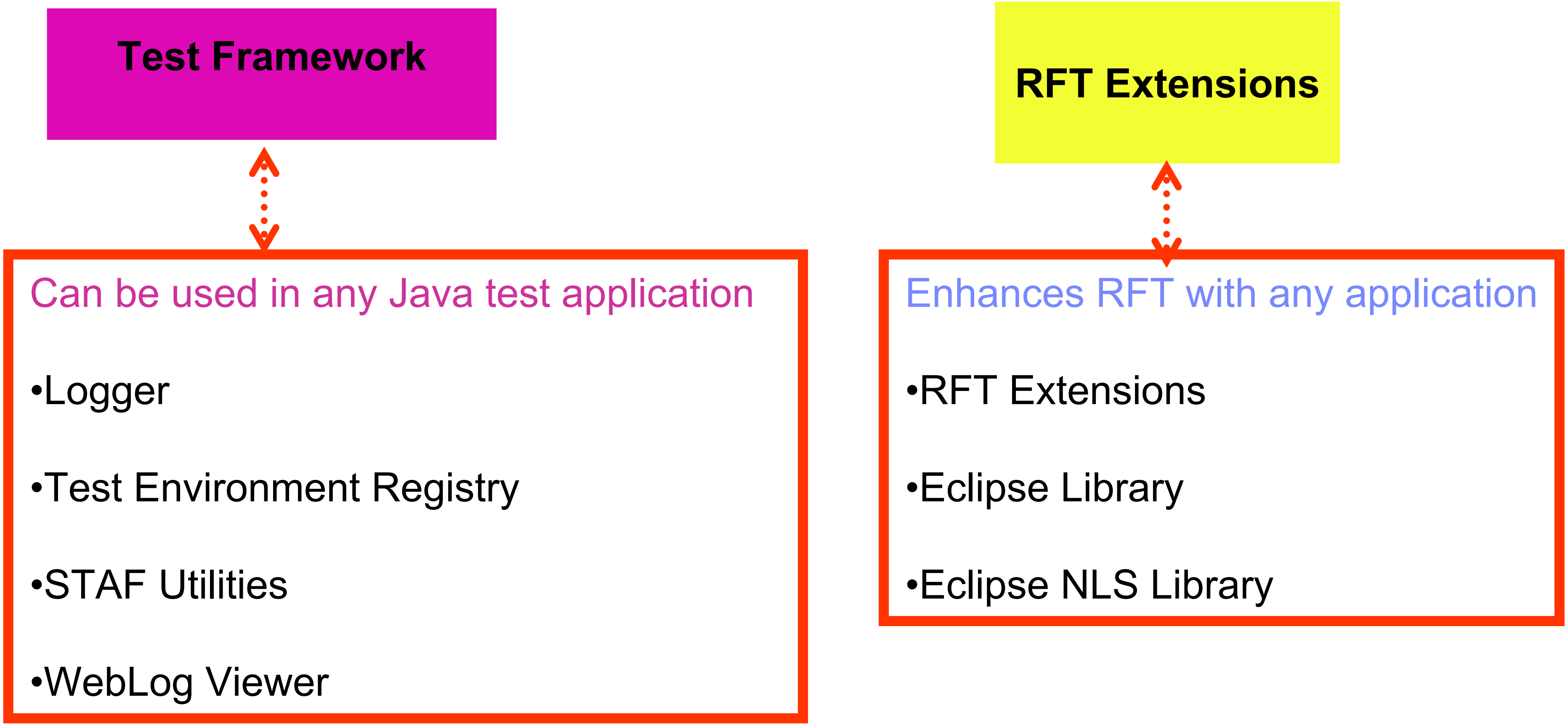


Testing CC and CQ Clients Using RFT

Extensions to RFT



Common RFT Components



Test Framework

Can be used in any Java test application

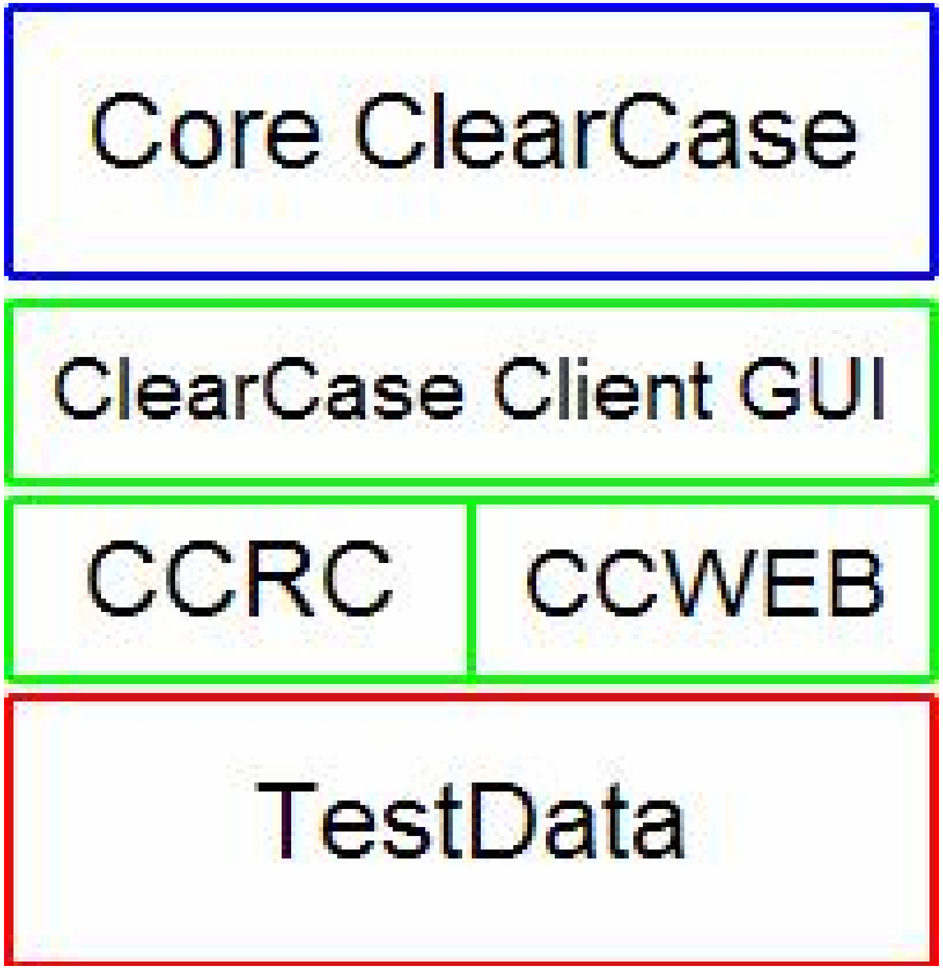
- Logger
- Test Environment Registry
- STAF Utilities
- WebLog Viewer

RFT Extensions

Enhances RFT with any application

- RFT Extensions
- Eclipse Library
- Eclipse NLS Library

Application Specific Components



Core ClearCase

ClearCase Client GUI

CCRC

CCWEB

TestData

- **Business Objects - Core ClearCase** can be used in any Java test environment,
- **GUI Layer - ClearCase** extends the business logic of Core ClearCase for GUI specific operations
- **CCWeb & CCRC**: teach the tool how to do an operation in the GUI
- **Test Data** works on any GUI front end

More on Test Data

- Test Data is shared across to clients to the extent the clients allow.
- We use a factory implementation to determine which GUI Layer class to instantiate
- Business Logic Java objects are initialized with ‘hard’ test data
- Operations are called on the Java objects
- Scripts look like manual test cases in pseudo-code
- Data Driven Test Data uses XML to drive multiple sets of data through one use case
- Test Data scripts can also be ‘hard-coded’ day in the life
- Test Data uses ‘runtime XML’ to determine what to run for each client, suite, down to individual test cases
- Configuration is managed from a flat ASCII file: runtime.properties (allows for unattended testing)
- Test Content scripts are easy for domain experts to write – *no Java programming skills required*

Performance Testing Using RPT

Performance and Reliability Testing

- Rational Performance Tester is used to test ClearCase and ClearQuest products
  - Performance and Reliability Team posts reports to new web site

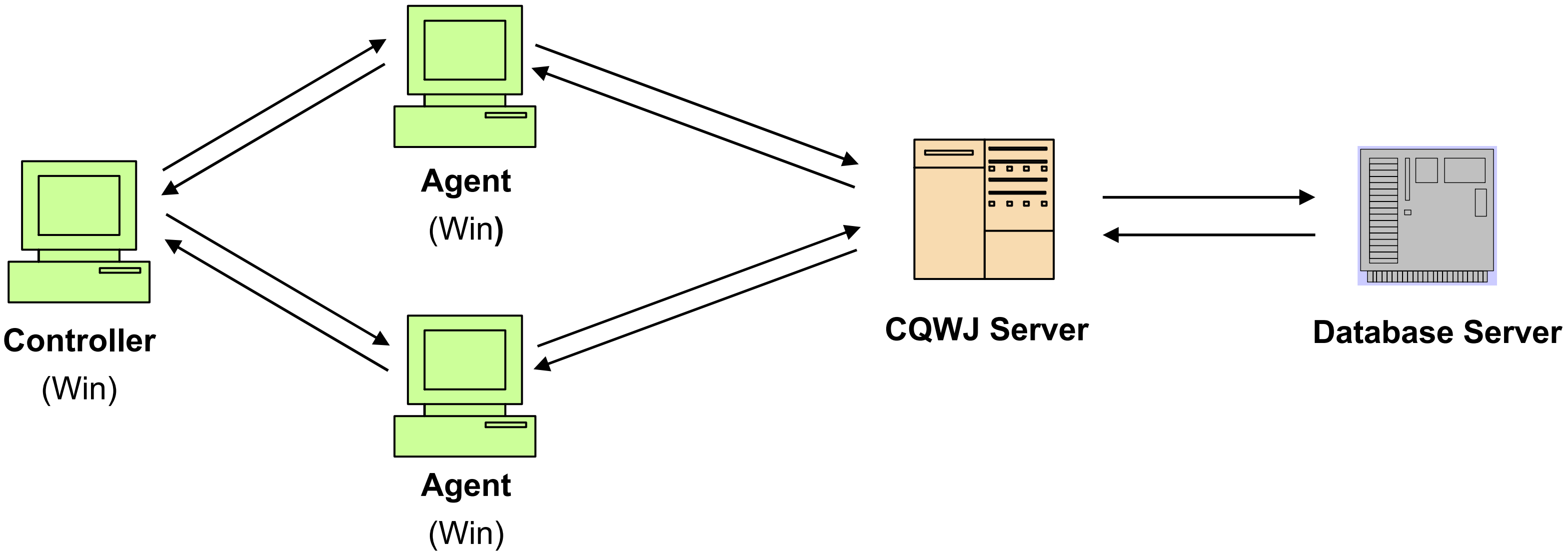
    www.ibm.com/developerworks/rational/performance
  - CQWeb testing - use record/playback capabilities of RPT (HTTP/HTTPS)
  - CCRC testing – use Custom code feature within RPT to drive internal API to benchmark server response under load.
  - Benchmark tool available for CQ community

- Rational Functional Tester used to measure response of GUIs.

- Rational Log Parser for CQWeb environment.
  - Web Access log parser
  - Utilizes Eclipse and BIRT Technology

Rational Performance Tester Environment

- Rational Performance Tester 7.x is now used for all performance load testing
  - Ability to drive performance and scalability testing in the latest release
    - Published scalability data - CCRC and CQ

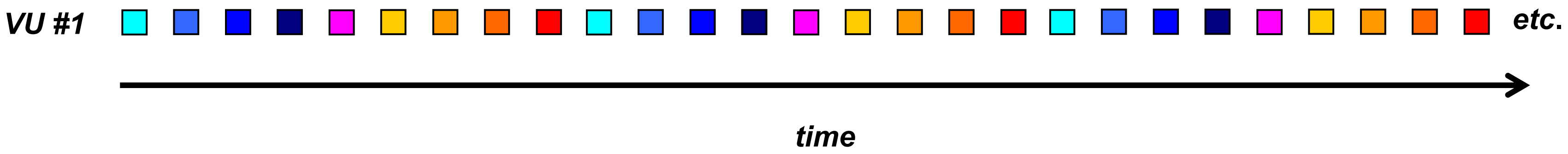


RPT Testing Strategy : Use Cases ↔ Transactions

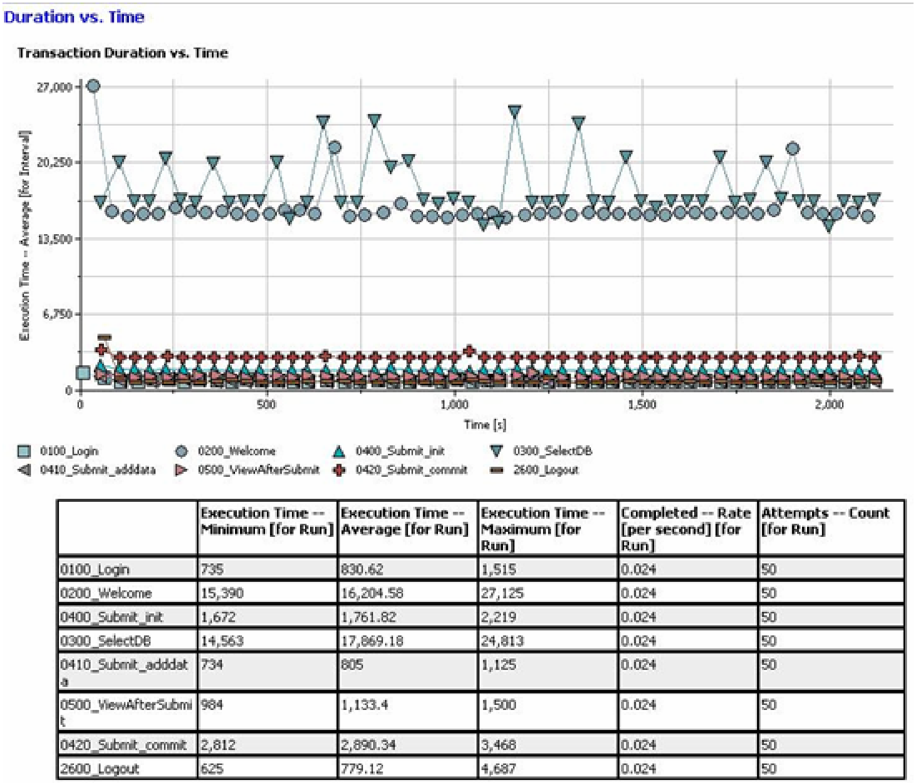
- We translate customer use cases into **Transactions**.
  - Transactions mimic user experience
  - Response times are captured for each transaction
- Transactions separate major interactions.
  - Transaction A may be predominantly database-centric.
  - Transaction B may be predominantly web-application-centric.

Use Cases	Transactions	Primary Interactions
Login	🟦 0100_login	web-app; db (read)
Create a defect	🟦 0300_ent_cre_new_init	web-app; db (read)
	🟦 0313_ent_cre_new_fill	user
	🟦 0315_ent_cre_new_commit	db (write)
View the defect	🟪 0410_ent_find	db (read)
Modify the defect	🟨 0510_ent_mod_init	web-app; db (read)
	🟧 0513_ent_mod_fill	user
	🟧 0515_ent_mod_commit	db (write)
Logout	🟥 9000_logout	web-app

Sequential Transactions Can Generate Baselines



- Transactions run in sequence generate baselines.
  - Response times are captured for each transaction.
  - We run a full set of transactions 50 times, collect the response times, then calculate mean and standard deviation.



Duration vs. Time

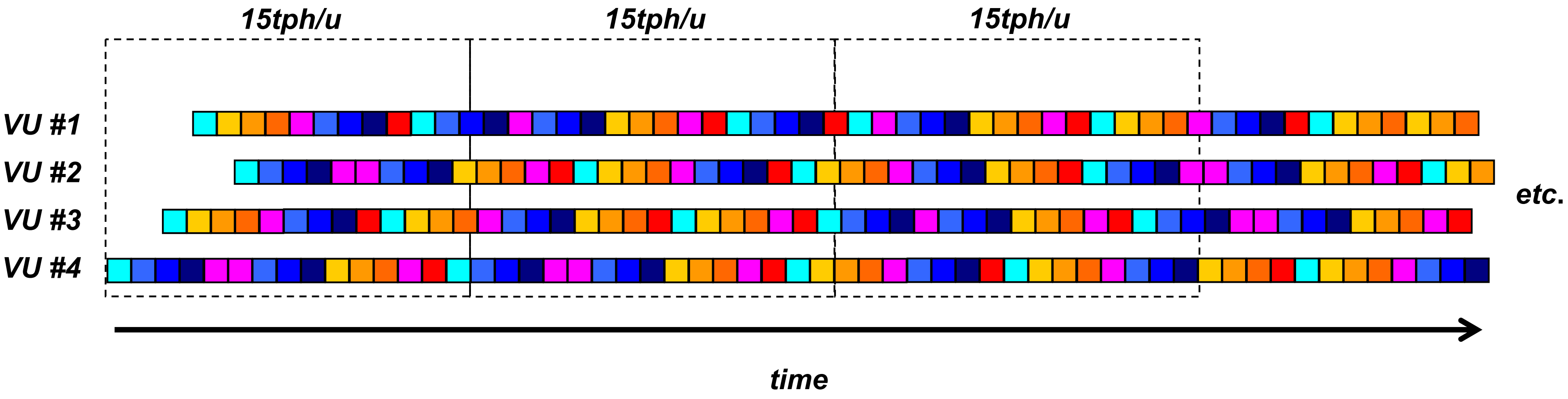
Transaction Duration vs. Time

	Execution Time -- Minimum [for Run]	Execution Time -- Average [for Run]	Execution Time -- Maximum [for Run]	Completed -- Rate [per second] [for Run]	Attempts -- Count [for Run]
0100_Login	735	830.62	1,515	0.024	50
0200_Welcome	15,390	16,204.58	27,125	0.024	50
0400_Submit_init	1,672	1,761.82	2,219	0.024	50
0300_SelectDB	14,563	17,869.18	24,813	0.024	50
0410_Submit_adddata	734	805	1,125	0.024	50
0500_ViewAfterSubmit	984	1,133.4	1,500	0.024	50
0420_Submit_commit	2,812	2,890.34	3,468	0.024	50
2600_Logout	625	779.12	4,687	0.024	50

Transactions-per-hour/user (tph/u) Load Model

- Ideal Model: Constant tph/u rate reflects customer usage.
  - 15tph/u represents 1 transaction every 4 minutes per user.
  - Use case mapping accurately reflects user experience.
  - Increase load by adding more users.
  - Differences between workload models are easier to abstract:

    "A system supports 1500tph, or 100users at 15tph/u"



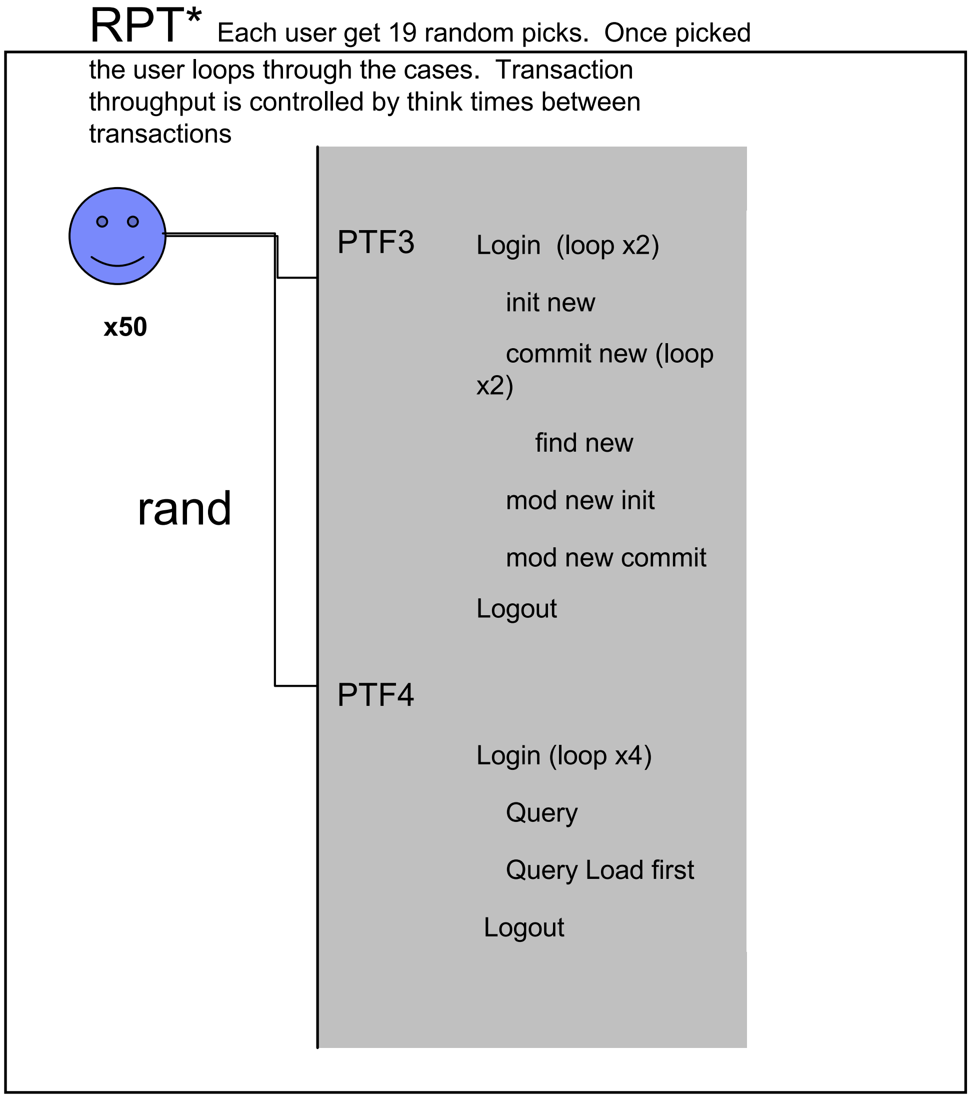
Transactions Improve Iterative Development

- Transaction response times are tracked during product lifecycle.
  - Improvements and regressions can be observed.
  - Development team can investigate anomalies.
- Useful in customer situations.
  - Can isolate problems.
  - Can demonstrate improvement.
  - Sometimes customers realize that response times values are acceptable or identify problems outside of the system.

		build 1	build 2	build 3	build 4	build 5	build 6	build 7	build 8	build 9	build 10	build 11	build 12	build 13	build 14	build 15
🟦	0100_login	1.27	1.24	1.28	1.32	1.53	1.41	1.51	1.67	1.60	1.48	1.81	12.27	2.00	1.91	2.19
🟦	0310_initate_new_defect	0.94	0.95	0.92	0.91	0.94	1.08	1.22	1.36	1.33	1.12	1.11	1.52	1.58	1.35	1.37
🟦	0315_commit_new_defect	0.88	0.89	0.87	0.86	0.94	1.29	1.50	1.67	1.55	1.44	1.46	1.82	1.54	1.51	1.83
🟦	0410_find_by_rand_id	1.03	1.06	1.34	1.23	1.28	1.09	1.10	1.32	1.26	1.05	1.14	1.61	1.23	1.25	1.92
🟦	0430_find_new	1.15	1.20	1.15	1.15	1.20	1.09	1.13	1.35	1.28	1.12	1.09	1.58	1.50	1.36	1.96
🟪	0510_initiate_modify_new_defect	0.95	0.96	0.89	0.96	0.90	1.01	1.17	1.27	1.22	1.11	1.13	1.56	1.23	1.21	2.11
🟨	0515_commit_modify_new_defect	0.47	0.49	0.48	0.48	0.48	0.84	1.01	1.03	1.00	0.83	0.84	1.20	0.88	0.86	1.01
🟧	0550_initiate_modify_rand_defect	0.94	0.95	1.15	1.12	1.12	1.20	1.26	1.38	1.35	1.21	1.26	1.54	1.36	1.20	2.08
🟧	0555_commit_modify_rand_defect	0.53	0.56	0.65	0.64	0.65	0.97	1.11	1.20	1.16	0.94	0.96	1.22	0.99	0.99	1.09
🟧	1110_query_load_results	0.52	0.68	1.65	1.49	1.38	1.32	1.18	0.81	0.75	0.78	0.66	0.82	1.26	0.70	0.81
🟥	1115_query_load_first	0.74	0.75	0.75	0.71	0.72	0.68	0.71	0.83	0.80	0.69	0.68	0.90	0.92	0.85	1.53

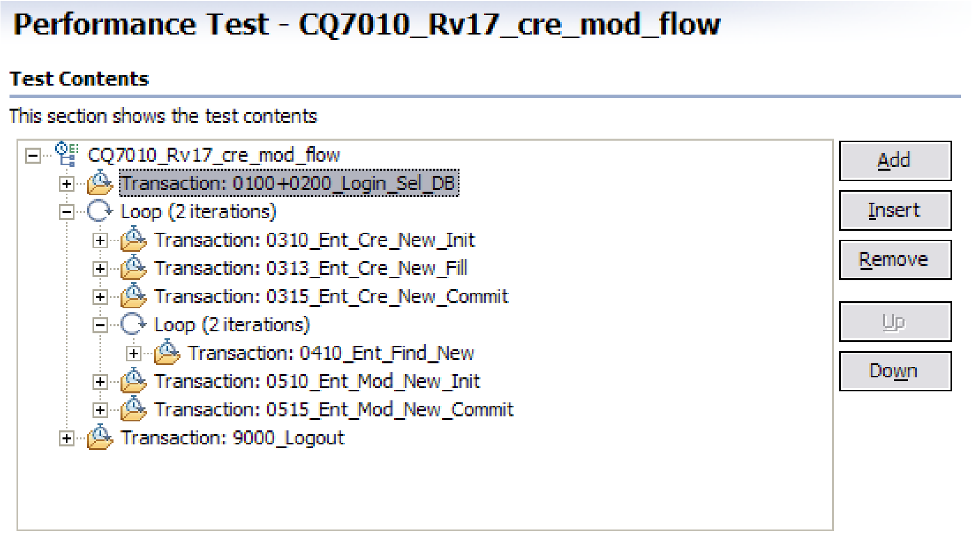
ClearQuest Web Performance and Reliability Testing

- Performance benchmarks
  - 1, 5 users as fast as possible (AFAP)
  - execute PTF3 and PTF4 uses cases (AFAP)
  - Response time avg. over 50 iterations.
  - Reliability tests (7.0.1*)
    - 50 Virtual Users AFAP
    - 30 sec delay between the start of each VU
    - ~16 hr. test duration
    - trans total



RPT* Each user get 19 random picks. Once picked the user loops through the cases. Transaction throughput is controlled by think times between transactions

Transaction Use Case within RPT

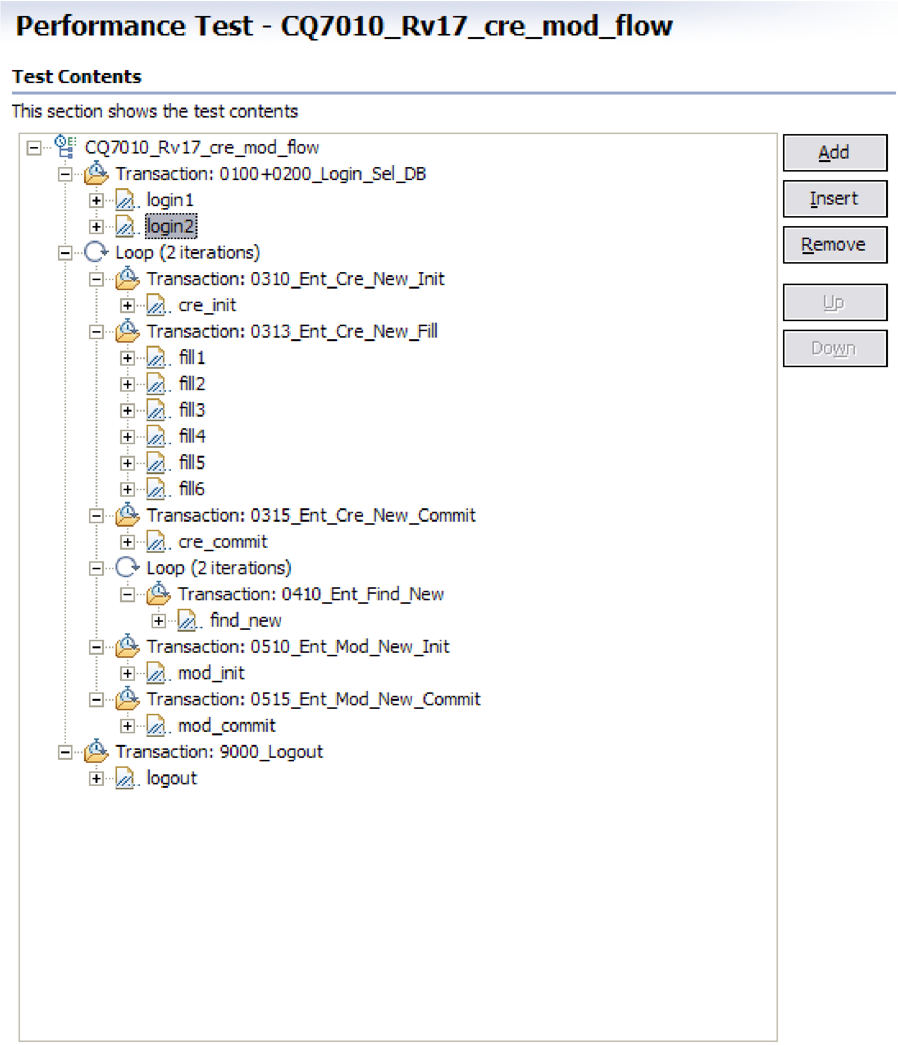


Performance Test - CQ7010_Rv17_cre_mod_flow

Test Contents

This section shows the test contents

- CQ7010_Rv17_cre_mod_flow
  - Transaction: 0100+0200_Login_Sel_DB
  - Loop (2 iterations)
    - Transaction: 0310_Ent_Cre_New_Init
    - Transaction: 0313_Ent_Cre_New_Fill
    - Transaction: 0315_Ent_Cre_New_Commit
    - Loop (2 iterations)
      - Transaction: 0410_Ent_Find_New
    - Transaction: 0510_Ent_Mod_New_Init
    - Transaction: 0515_Ent_Mod_New_Commit
  - Transaction: 9000_Logout



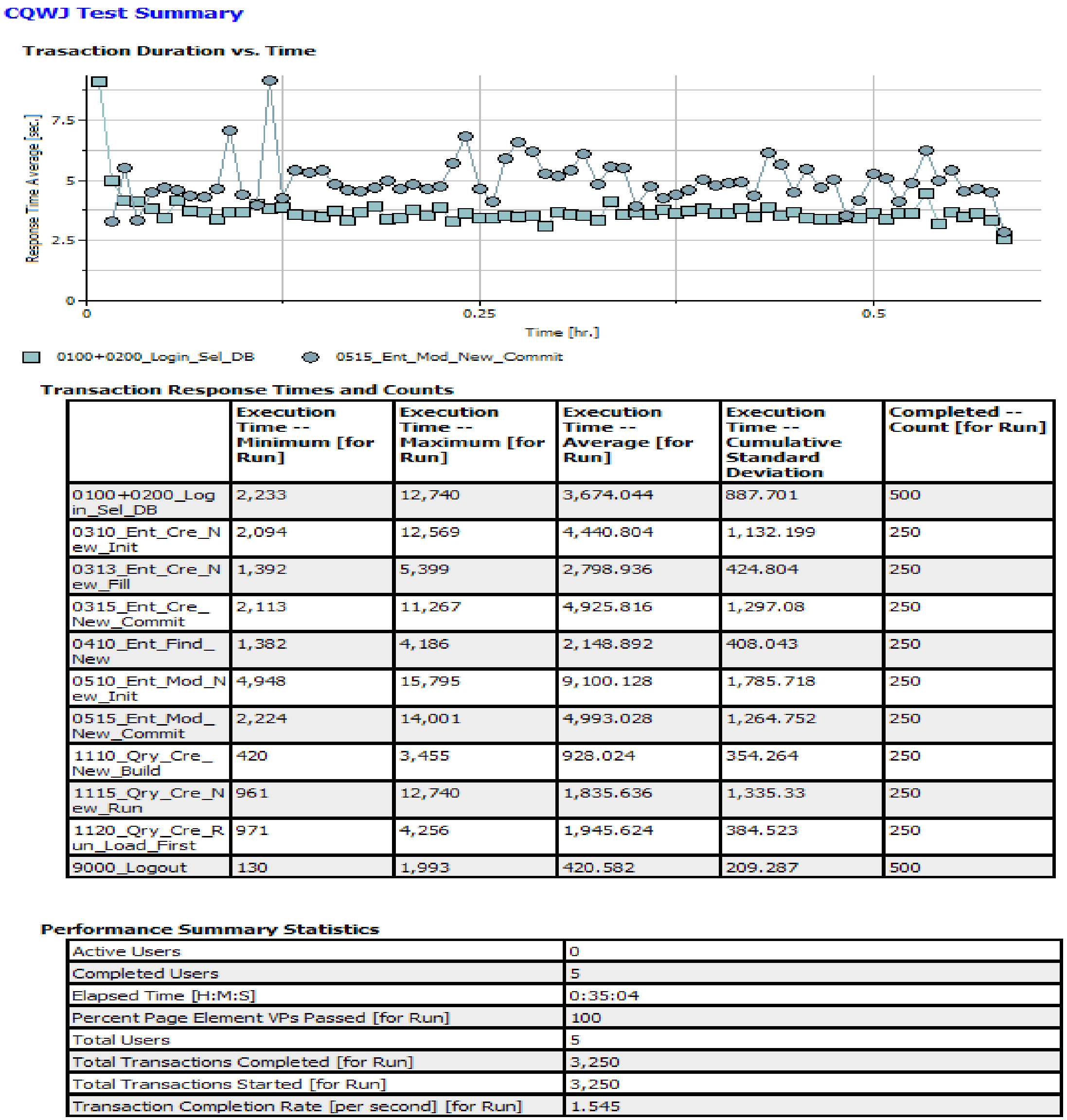
Performance Test - CQ7010_Rv17_cre_mod_flow

Test Contents

This section shows the test contents

- CQ7010_Rv17_cre_mod_flow
  - Transaction: 0100+0200_Login_Sel_DB
    - login1
    - login2
  - Loop (2 iterations)
    - Transaction: 0310_Ent_Cre_New_Init
      - cre_init
    - Transaction: 0313_Ent_Cre_New_Fill
      - fill1
      - fill2
      - fill3
      - fill4
      - fill5
      - fill6
    - Transaction: 0315_Ent_Cre_New_Commit
      - cre_commit
    - Loop (2 iterations)
      - Transaction: 0410_Ent_Find_New
        - find_new
    - Transaction: 0510_Ent_Mod_New_Init
      - mod_init
    - Transaction: 0515_Ent_Mod_New_Commit
      - mod_commit
  - Transaction: 9000_Logout
    - logout

RPT Transaction Report - CQWEB Performance Test



CQWJ Test Summary

Trasaction Duration vs. Time

Transaction Response Times and Counts

	Execution Time -- Minimum [for Run]	Execution Time -- Maximum [for Run]	Execution Time -- Average [for Run]	Execution Time -- Cumulative Standard Deviation	Completed -- Count [for Run]
0100+0200_Login_Sel_DB	2,233	12,740	3,674.044	887.701	500
0310_Ent_Cre_New_Init	2,094	12,569	4,440.804	1,132.199	250
0313_Ent_Cre_New_Fill	1,392	5,399	2,798.936	424.804	250
0315_Ent_Cre_New_Commit	2,113	11,267	4,925.816	1,297.08	250
0410_Ent_Find_New	1,382	4,186	2,148.892	408.043	250
0510_Ent_Mod_New_Init	4,948	15,795	9,100.128	1,785.718	250
0515_Ent_Mod_New_Commit	2,224	14,001	4,993.028	1,264.752	250
1110_Qry_Cre_New_Build	420	3,455	928.024	354.264	250
1115_Qry_Cre_New_Run	961	12,740	1,835.636	1,335.33	250
1120_Qry_Cre_Run_Load_First	971	4,256	1,945.624	384.523	250
9000_Logout	130	1,993	420.582	209.287	500

Performance Summary Statistics

Active Users	0
Completed Users	5
Elapsed Time [H:M:S]	0:35:04
Percent Page Element VPs Passed [for Run]	100
Total Users	5
Total Transactions Completed [for Run]	3,250
Total Transactions Started [for Run]	3,250
Transaction Completion Rate [per second] [for Run]	1.545

How can we help our customers with Rational Tools?

Performance Benchmarking Kit (PBK) - CQWJ

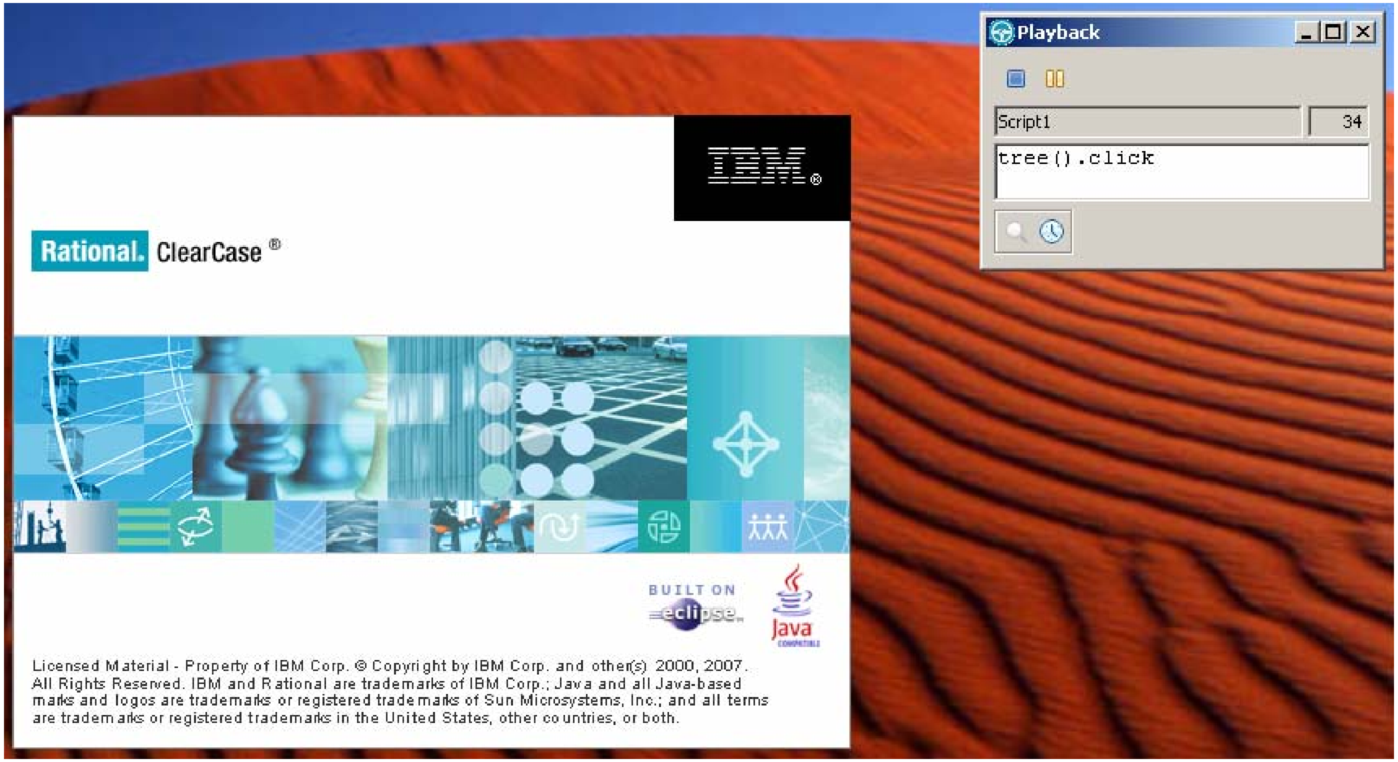
- Available for Download*
- Customer-distributable schema with corresponding performance test scripts.
- Schema derived from CQ-sample Enterprise Schema.
- A way to map existing benchmark data to customer environment.
- Customers can objectively evaluate performance in their environment.

* PBK is available for download – www.ibm.com/developerworks/rational/performance

Using RFT to baseline GUI performance

Rational Functional Tester – CCRC GUI Response

- RFT provides a mechanism for creating a performance baseline for Single User GUI Performance*
- RFT enables has playback/record functionality .



Script1 34

```
tree().click
```

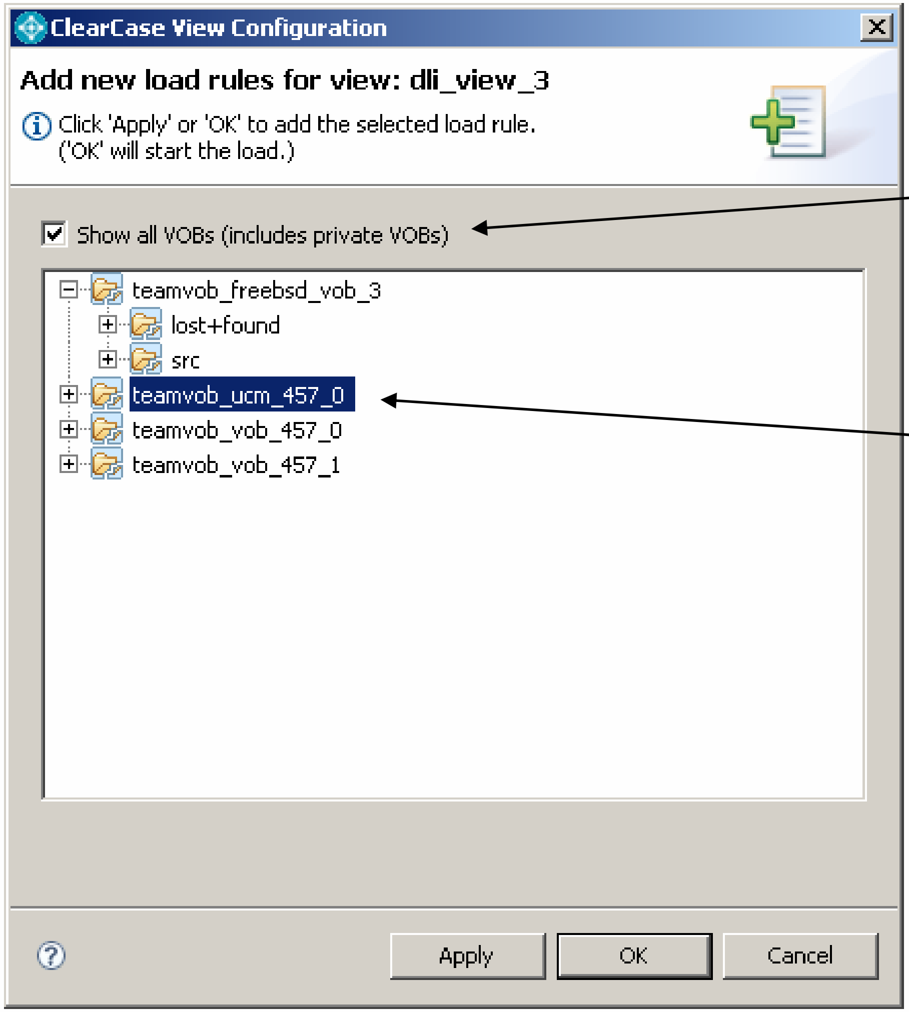
Licensed Material - Property of IBM Corp. © Copyright by IBM Corp. and other(s) 2000, 2007. All Rights Reserved. IBM and Rational are trademarks of IBM Corp.; Java and all Java-based marks and logos are trademarks or registered trademarks of Sun Microsystems, Inc.; and all terms are trademarks or registered trademarks in the United States, other countries, or both.

*RFT performance numbers can vary release to release of RFT

Timing Initial Login – RFT Sample code

```
// Frame: ClearCase Remote Client
        startApp("ccrc");
        timerStart("Connect");
        tree().click(RIGHT, atPath("dli_view_2_1 [http://x.x.x.x/ccrc,
  disconnected]"));

        contextMenu().click(atPath("Connect..."));
        // Frame: Connect to ClearCase Web server – Enter Password
        connectToClearCaseWebServer().inputKeys("<Password>{ENTER}");
        // Frame: Retrieving contents of dli_view_2_1....
        ok().click();
        timerStop("Connect");
```

CCRC Screen during Script Creation



ClearCase View Configuration

Add new load rules for view: dli_view_3

Click 'Apply' or 'OK' to add the selected load rule.
('OK' will start the load.)

☑ Show all VOBs (includes private VOBs)

- teamvob_freebsd_vob_3
  - lost+found
  - src
- teamvob_ucm_457_0
- teamvob_vob_457_0
- teamvob_vob_457_1

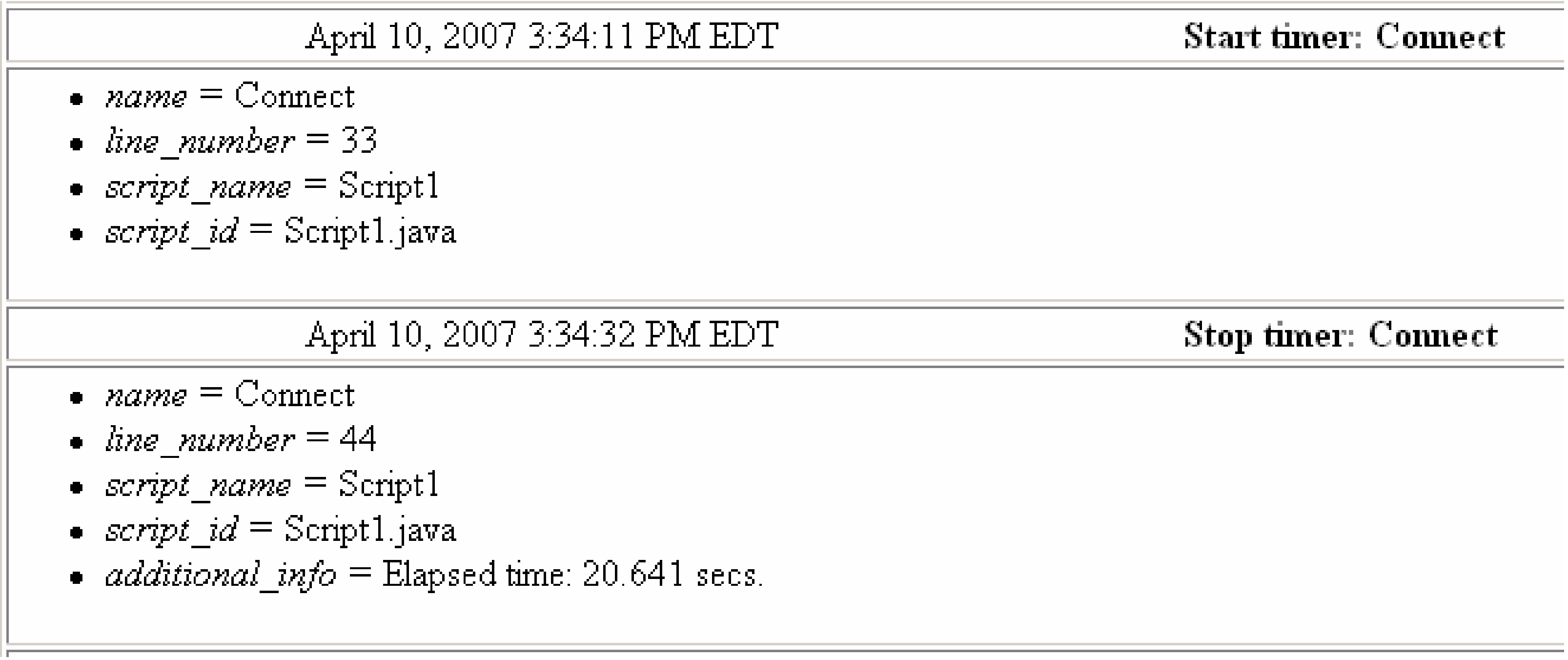
showAllVOBsIncludesPrivateVOBs()

tree2().click

RFT Load Vob Timing Script

```
timerStart("LoadVOB");

        // Frame: ClearCase Create View
        menu().click(atPath("Environment"));
        menu().click(atPath("Environment->Create ClearCase View"));
        // Frame: Create Base ClearCase View
        _Next().click();
        text().click(atPoint(96,14));
        createBaseClearCaseView().inputChars("view details");
        _Finish().click();

        // Frame: Create Base ClearCase View
        createBaseClearCaseView2().click(atPoint(319,187));
        _Yes().click();

        // Frame: ViewLoad –
        //Define one second delay
        // ClearCase View Config Screen is displayed  (see next slide)
        DelayEvent delay = new DelayEvent(1000);

        showAllVOBsIncludesPrivateVOBs().clickToState(SELECTED);
        tree2().click(atRow(atPath("teamvob_freebsd_vob_3")), atPoint(14,13));

        // Instantiate one second delay -
        delay.emit();
        tree2().click(atPath("teamvob_vob_457_0"));
        ok2().click();
        // Frame: Set view configuration...
```

Sample RFT Timing Output



April 10, 2007 3:34:11 PM EDT **Start timer: Connect**

- *name* = Connect
- *line_number* = 33
- *script_name* = Script1
- *script_id* = Script1.java

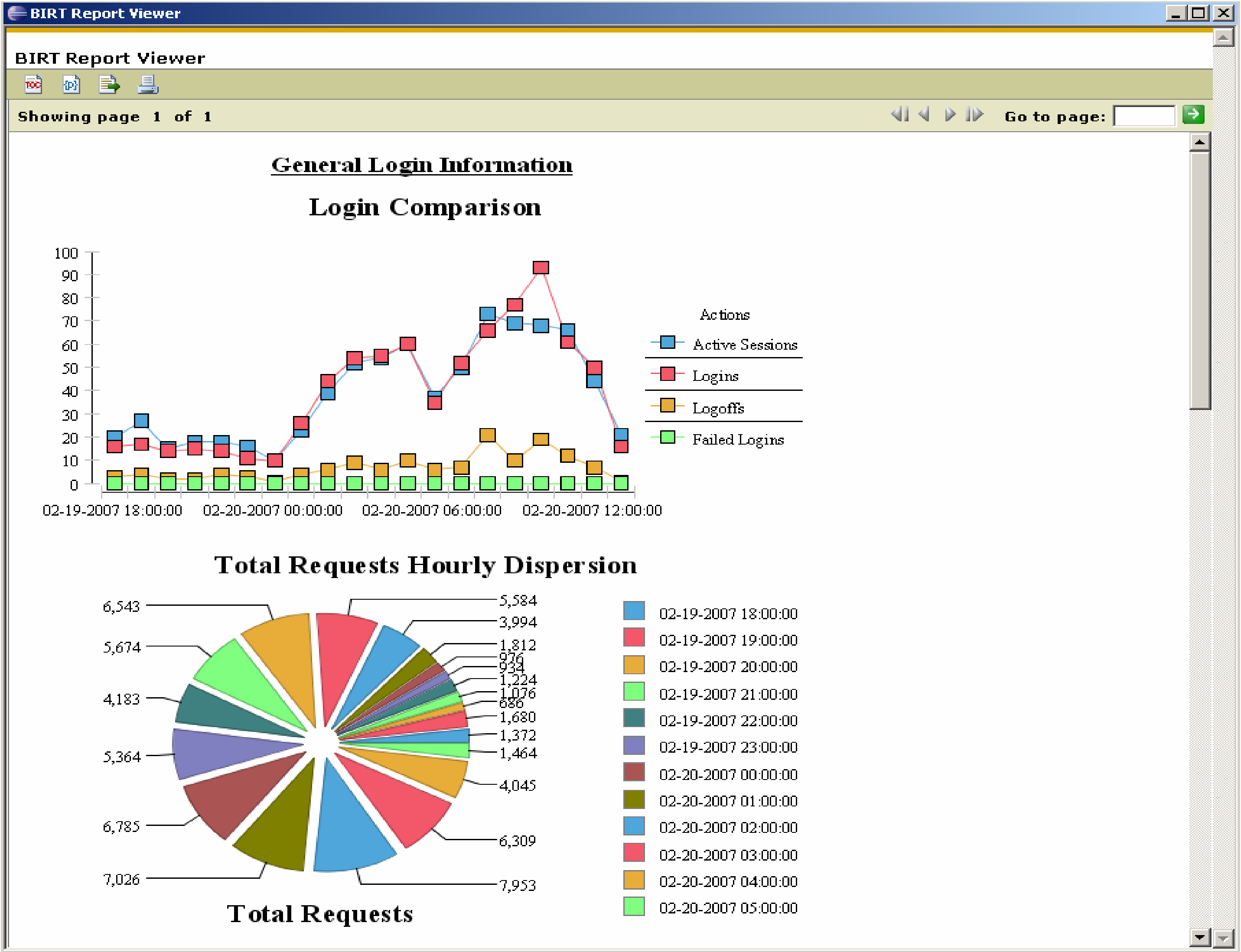
April 10, 2007 3:34:32 PM EDT **Stop timer: Connect**

- *name* = Connect
- *line_number* = 44
- *script_name* = Script1
- *script_id* = Script1.java
- *additional_info* = Elapsed time: 20.641 secs.

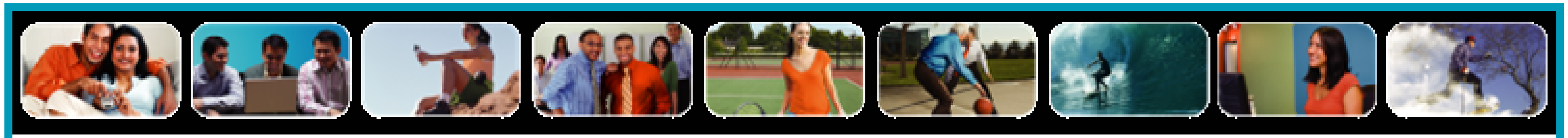
CQ Weblog Analyzer

- **Enables measurement of CQWeb Workload.**
- **Specifications of Analyzer.**
  - **Analyzer coded in Java 2.0**
    - Log parser written in Perl 5.6 and outputs into *.CSV (comma separated value) file
  - **Eclipse 3.2.2**
  - **Business Intelligence and Reporting Tools (BIRT) 2.1.2**
    - http://download.eclipse.org/birt/downloads/

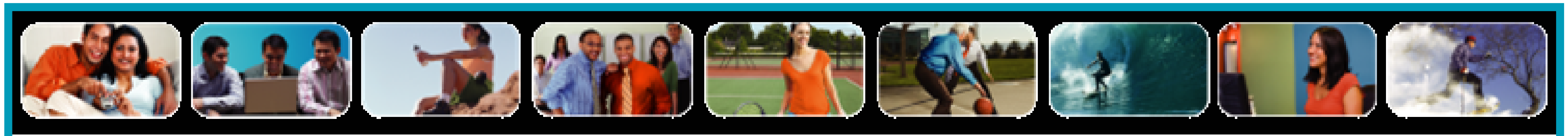
CQWEB Parser Sample Report



General Login Information



Questions



Thank You

Paul Murray

pmurray@uk.ibm.com