Performance Testing Web 2.0

David Chadwick
Rational Testing Evangelist
dchadwick@us.ibm.com

Dawn Peters
Systems Engineer, IBM Rational
petersda@us.ibm.com

Rational. software

© 2009 IBM Corporation

Go to IBM

WEB 2.0 what is it?

Web 2.0

- Started out as next generation internet, but not very well defined as to what that meant.
- Starting to settle down into collaboration. One page can now display from multiple sites.
- Moving away from the internet server doing the processing and more towards client side processing.

Web 2.0 Technology

Rational. software

QM06 – Performance Testing Web 2.0

© 2009 IBM Corporation

Technologies fueling Web 2.0

- AJAX – Asynchronous JavaScript and XML
- JSON – JavaScript Object Notation
- Dojo – Open Source JavaScript Toolkit

AJAX

- Asynchronous JavaScript XML
- Uses XMLHttpRequest which is a DOM API that allows java script to send a request directly to web server and load response back into scripting language
- Allows one part of a page to update without having to re-fetch entire page contents.

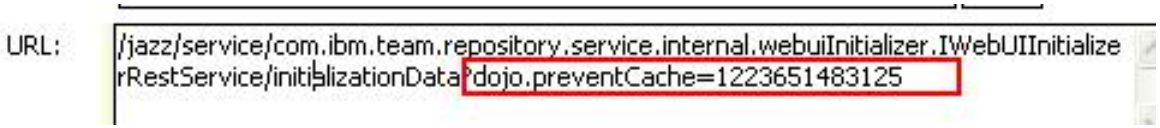
JSON

- Java Script Object Notation
- Data representation that is used to pass data between client and server
- Response header has Content-type: text/json
- Sample Data


```
},
"com.ibm.team.process.service.projectAreaWebUIInitializionData": {
  "projectAreas": [
    {
      "archived": true,
      "itemId": "_GRnCYJ8OEd2pzYTSVDgJ3A",
      "myProjectArea": false,
      "name": "JUnit Project"
    },
    {
      "archived": false,
      "itemId": "_i0FzgJpwEd2UCf8mttsadQ",
      "myProjectArea": true,
      "name": "RPT Development Project"
    },
    {
      "archived": true,
      "itemId": "_sV3hYKRYEd2sgtYO0cjvaA",
      "myProjectArea": false,
      "name": "RPT Development"
    }
  ]
},
```

Dojo

- Open source library of java script objects
- Aids in development of AJAX applications
- Executes the XMLHttpRequest and performs communication between client and server.
- Can often be identified by the dojo.preventCache in the URI



Testing Your Application

Rational. software

QM06 – Performance Testing Web 2.0

© 2009 IBM Corporation

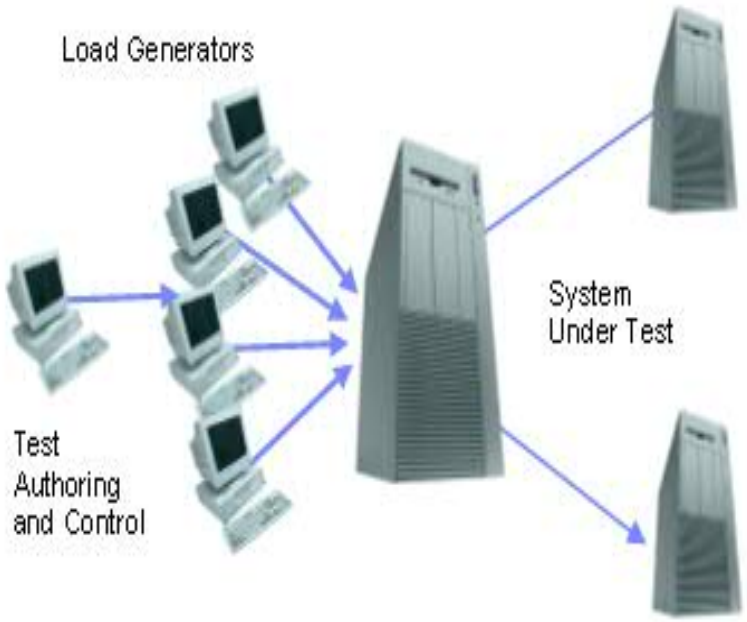
IBM Rational Performance Tester

- What is Performance Testing ?
  - The process of exercising an application by emulating actual users with a load generation tool for the purpose of finding system bottlenecks
- Why do we need Performance Testing ?
  - Because a break at any point in your system means your customers are not getting the service you think they are
- IBM Rational Performance Tester
  - Automates the performance and scalability testing of the SUT
  - Isolates the source of performance bottlenecks and optimizes resource utilization before deployment

Build Scripts **Schedule Workload** **Execute & Analyze**

Testing your application

- Define what you are testing
- Create your tests
- Create your workload
- Run your workload



Define What You Are Testing

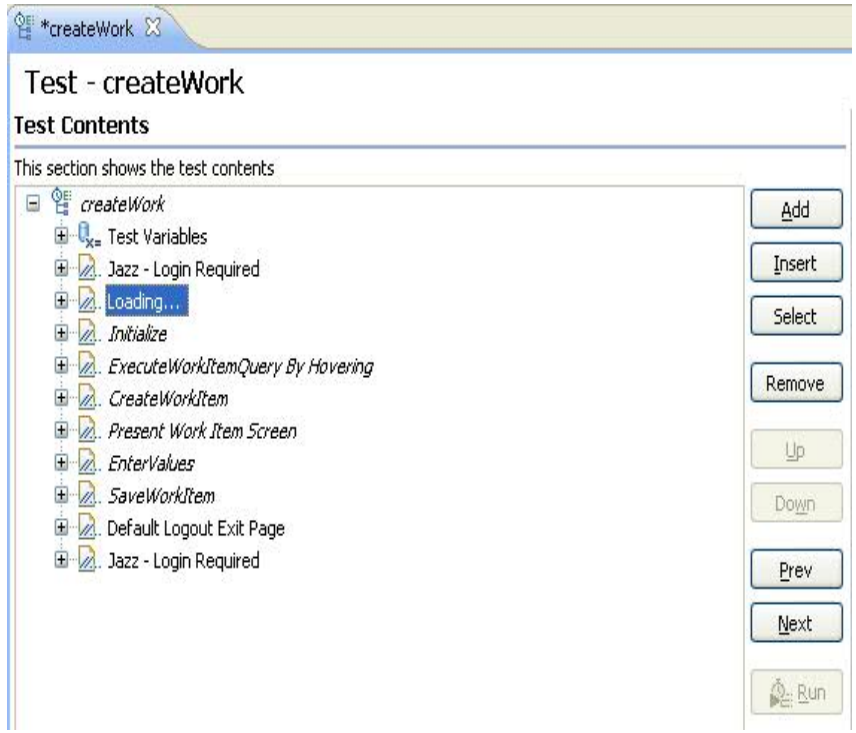
- Need to figure out how many users
- Define performance requirements
- What parts of application are you going to execute
- What is the weight for each part of the application
- What types of users are you testing

We will test Rational Team Concert – 10 user test

- Login
- Create Work Item – execute 30% of time
- Modify Work Item – execute 70% of time
- Logout

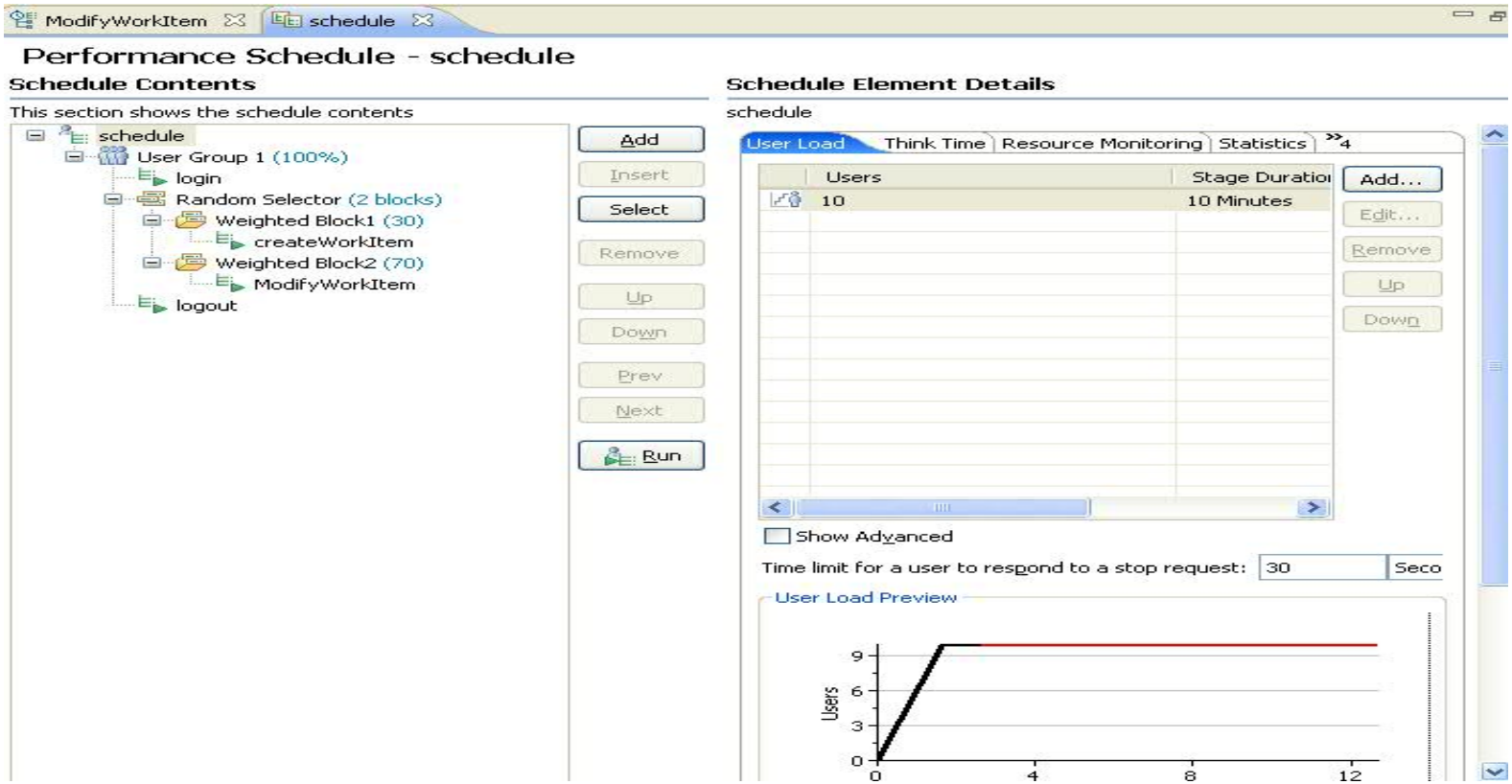
Creating Tests

- Use annotations while recording to help with test readability
- Use split test to create smaller more manageable pieces
  - Create performance requirements for important pages and transactions
- Ensure your tests are working correctly
  - Use verification points in tests



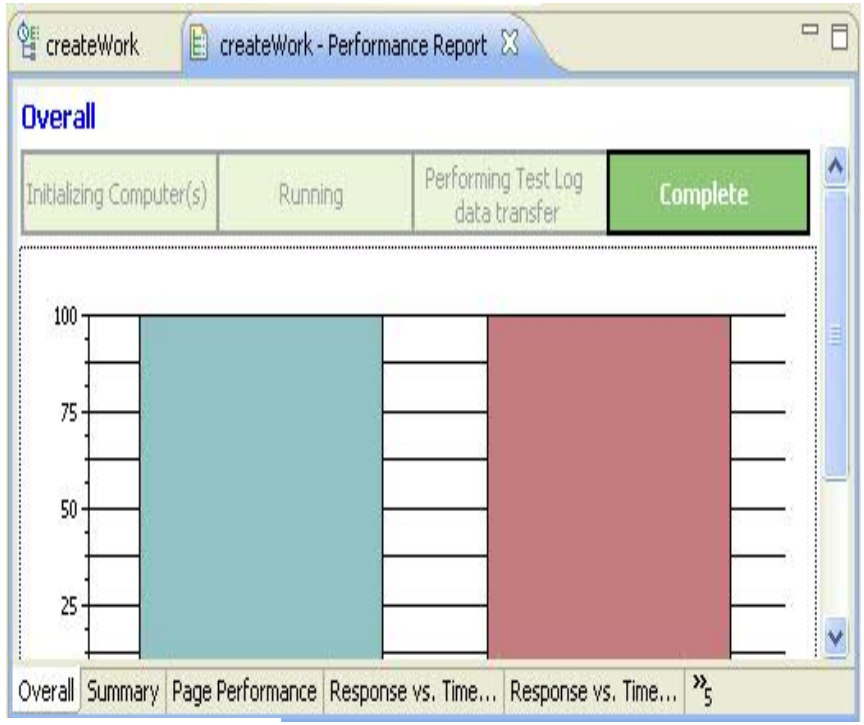
Create Workload

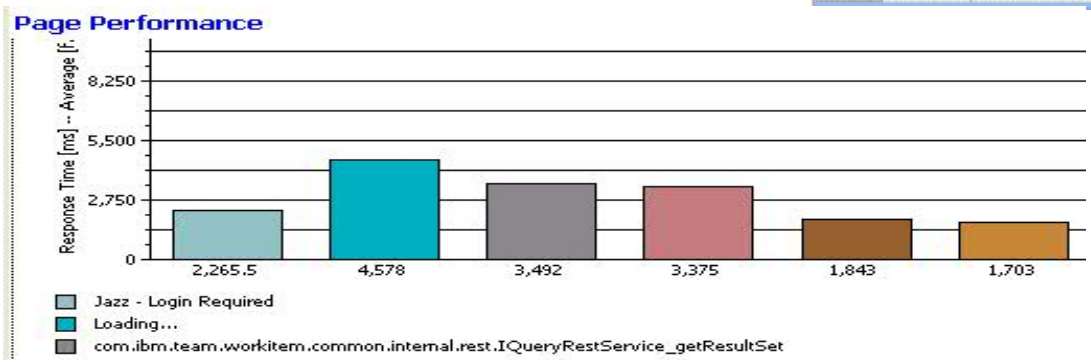
- Define the weight of different scripts you are running
- Number of users
- Length of run



Run Workload

- Monitor your server and your tests
- Use the real time protocol data viewer to monitor individual users
- Monitor response times shown
- Make sure performance requirements are passing





Web 2.0 Challenges

Rational. software

QM06 – Performance Testing Web 2.0

© 2009 IBM Corporation

Challenges of Testing Web 2.0

- More client-side processing is occurring
- Concept of a page is changing
- Data Correlation is not as easy

More Client Side Processing Occurring

- Large amounts of data returned which could lead to
  - Not enough memory to process data
  - Slow performance because of inefficient client side processing of data
- Client could have memory leaks, which would also lead to out of memory errors
- Capturing client side processing time
  - It can increase with the size of data being processed.

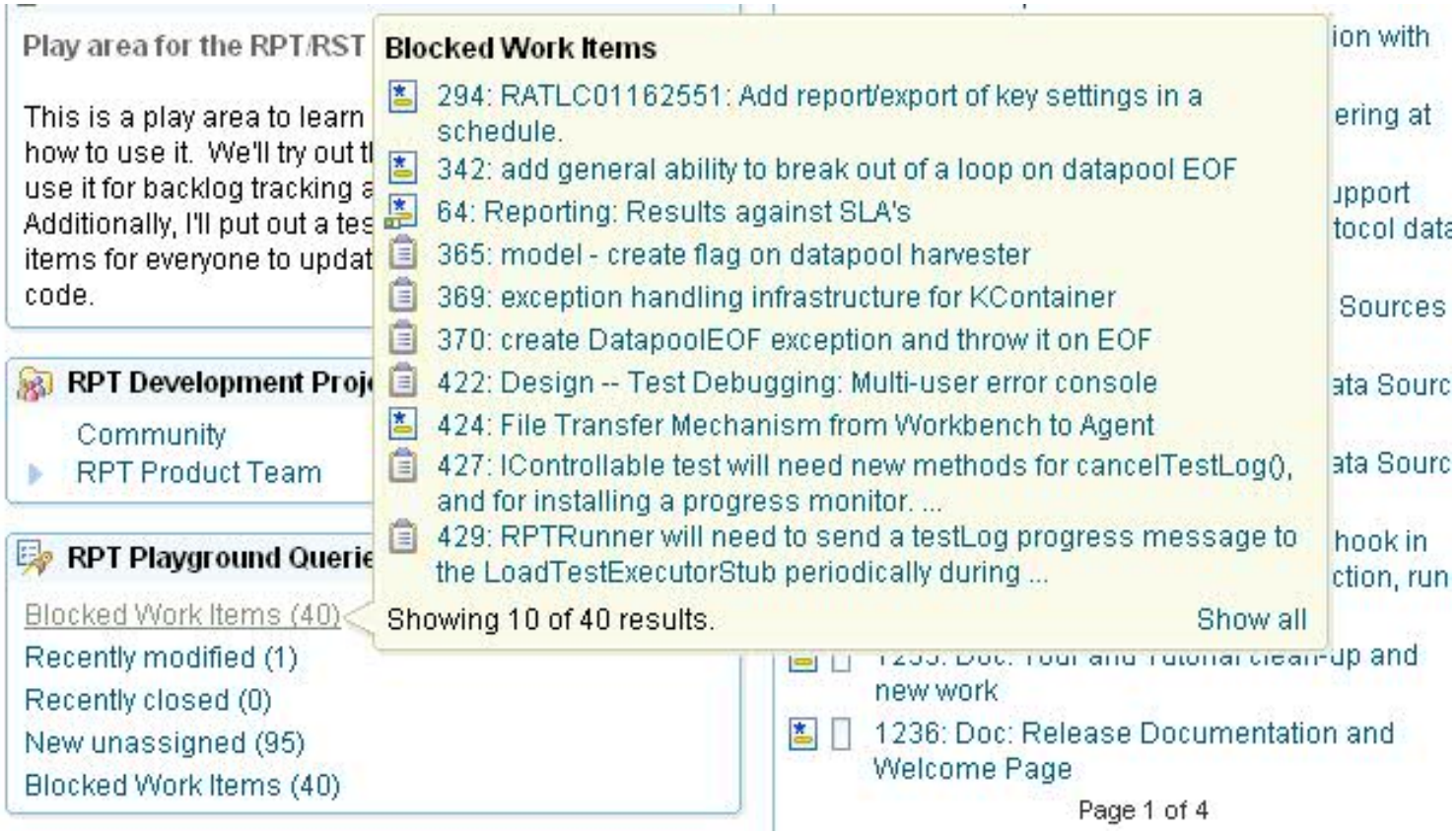
Concept Of A Page Changing

Each of these lines, if they are hovered over, will execute an http request. It will execute the query and return the list of items, without creating a new page.

This means you need to be careful where your mouse is and what you are doing or else you might be executing queries that are unintended as part of your test. If you do want those things to execute, then you need to make sure to wait after causing the query to execute so that RPT will know that that is supposed to be a page.

Concept Of A Page Changing

It’s like a page within a page



Concept Of A Page Changing

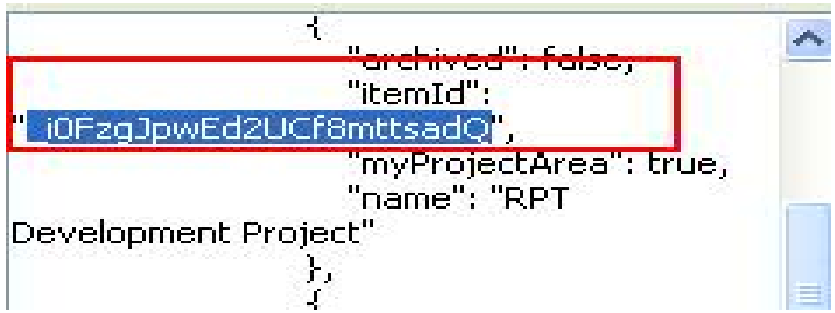
- Anatomy of an RPT HTTP Page
  - Each page has a primary request. This request is the main request that will send out all other requests on the page.
  - When the response from the primary request has been received, we start releasing requests.
  - Requests after the primary request go out in parallel with each other.
  - The number of requests that are executing in parallel is dependent on the number of connections that were used at record time.
  - This is helpful in replicating the asynchronous AJAX requests that go out.

Data Correlation Is Not As Easy

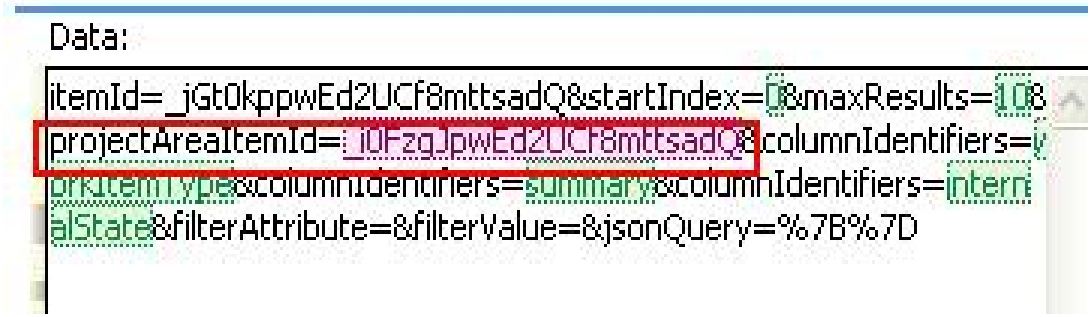
- Data returned from the server does not have the same name as data that is sent to the server.
  - Caused by java script.
- Testers have to do manual work to fix up these data correlations
  - Use Data Source View and Find More Substitution Sites

Mismatched Response And Request Data

Server response returns data that looks like



Request sends data that looks like



Web 2.0 Tricks and Tips

Rational. software

QM06 – Performance Testing Web 2.0

© 2009 IBM Corporation

Web 2.0 Tips and Tricks

- Enhancements made to RPT 8.1 to aid in Web 2.0 testing
- Recording
- Determining “health” of test
- Fixing “health” of test

Enhancements to RPT 8.1 That Aid Web 2.0 Testing

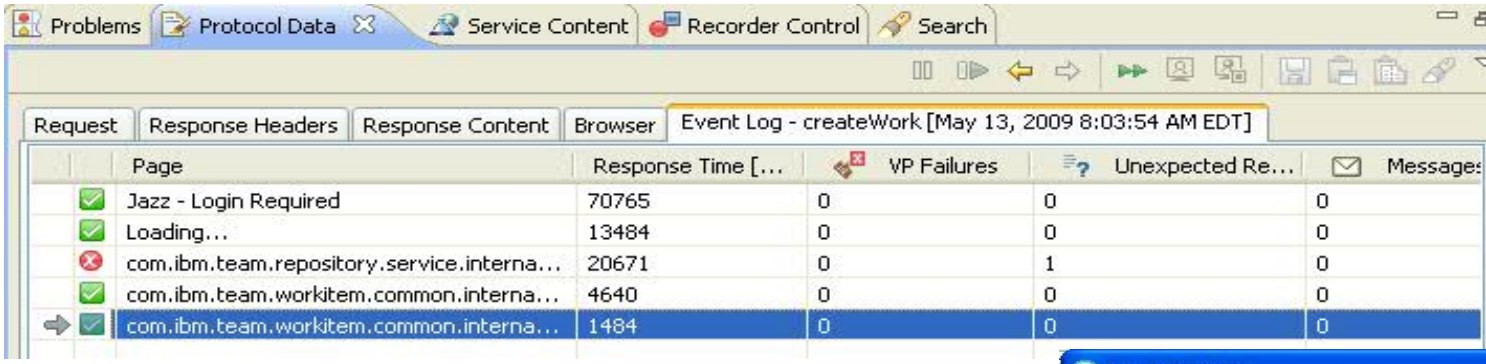
- Improved recognition of JSON strings
  - RPT used to only look for name=value pairs, and can now match “name”:”value”
  - Improved regular expressions to harvest data out of JSON
- Annotations – Allows user to give direction at record time
  - User can input comments that will be inserted into the test to show what was happening when a request was generated
  - Pages can be named
  - Transactions can be created
- Better page recognition
  - RPT can now identify new response data types and use them to create pages
- Split test
  - Users can now record smaller pieces at a time and mix and match them

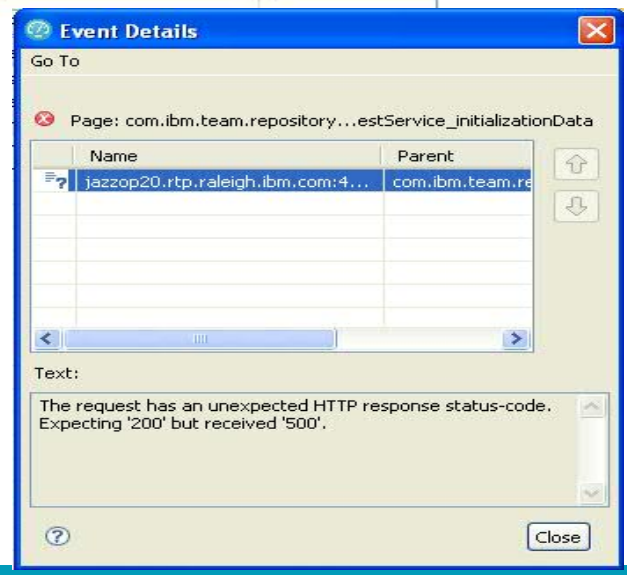
Hints to Record Web 2.0

- Go slow. Wait 5 seconds after a page comes up before performing another action
- Name your pages while recording.
  - Requests are not for entire pages, but just a small action.
  - Entering a name as you go for each action will help with test readability
- Start with a simple transaction in the your product so you can learn you application
- Be purposeful with your mouse movements and actions
- Make sure server is set up the way all users will be accessing it
  - Seed your data
  - Record as “type” of user you will play back with

Determining “health” of your test

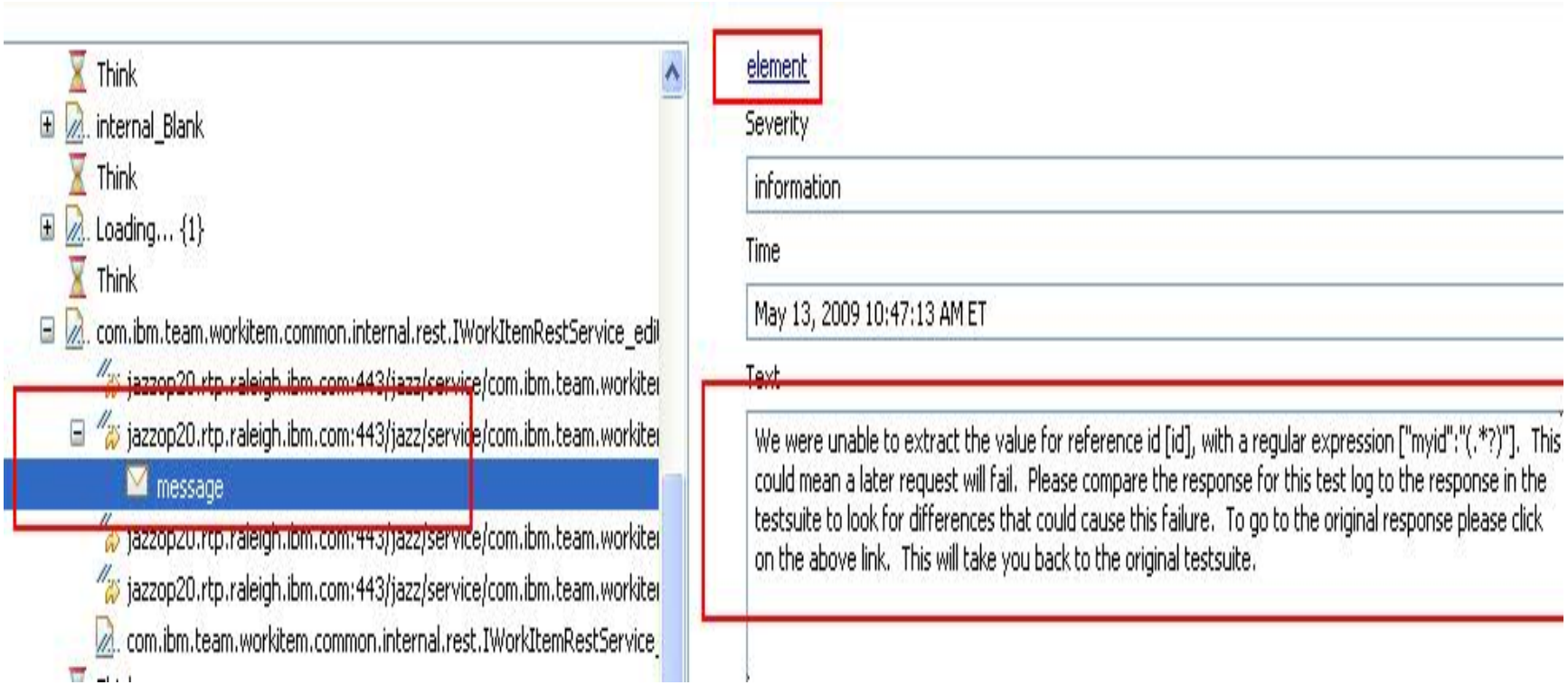
- Use the Real Time Protocol Data View





Determining “health” of your test

- Look in the test log
  - Look at message in test log.
  - Text is saying data correlation failed.
  - Click on Element to back to test to debug.



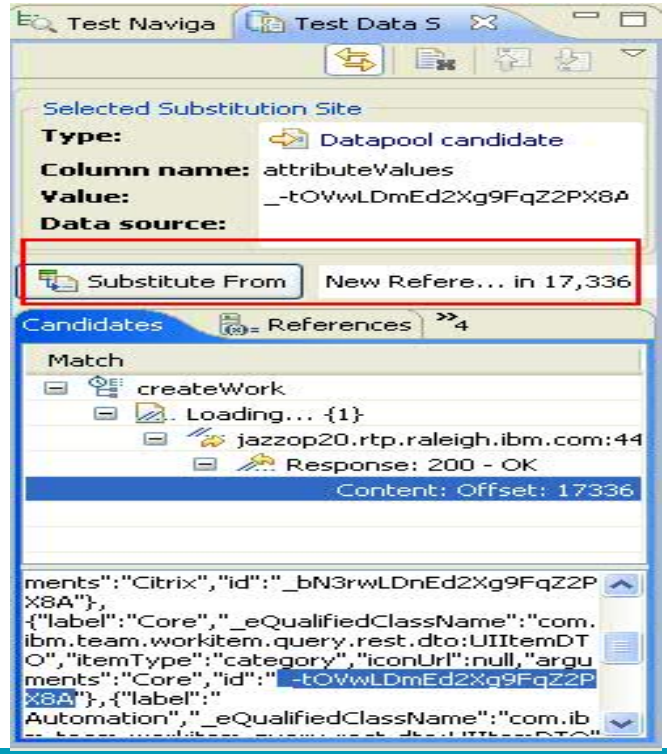
Fixing “Health” Of Test

- Find failed data correlations
  - Web 2.0 applications do not match their response data to the request data. This means a lot of applications will have missing correlations in the URI and POST data.
  - Use test log and real time protocol data viewer to hunt them down
- Use Protocol Data Viewer compare feature to compare logs and tests
- Use test editor and data source view to fix failed data correlations
  - Find More Substitution sites to find all places that need to be correlated

Find More Substitution Sites

Right click on value that needs to be correlated. Substitute From -> Data Source View

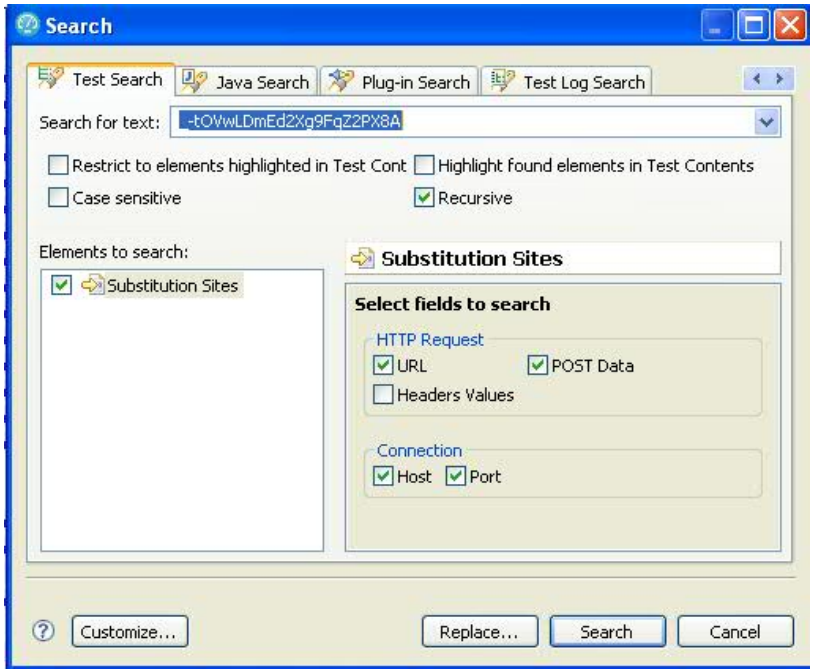
Go to Test Data Source View and click on Candidates tab. Find your match and click Substitute From.



Find More Substitution Sites continued

- Go back to substitution site, right click on it, and say Find->More Substitution Sites. Note substitution site is now pink to show it is correlated.

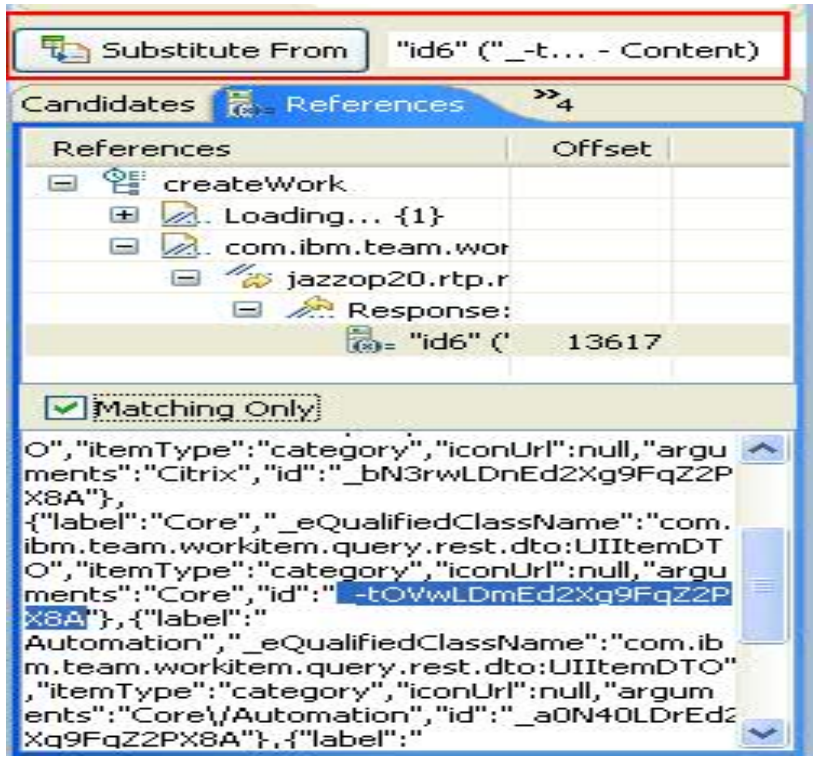
attributeIdentifiers=internalPriority&attributeIdentifiers=internalSeverity&attribute
attributeIdentifiers=tags&attributeIdentifiers=teamArea&attributeIdentifiers=time
s=workItemTyp&attributeValues=Create%20demo%20work%20item.&attributeV
es=_-tOVwLDmEd2Xg9FqZ2PX8A&attributeValues=_vA6oMJrdEd2UCf8mttsadQ&al
wEd2UCf8mttsadQ&attributeValues=This%20item%20needs%20to%20be%20dor



Find More Substitution Sites continued

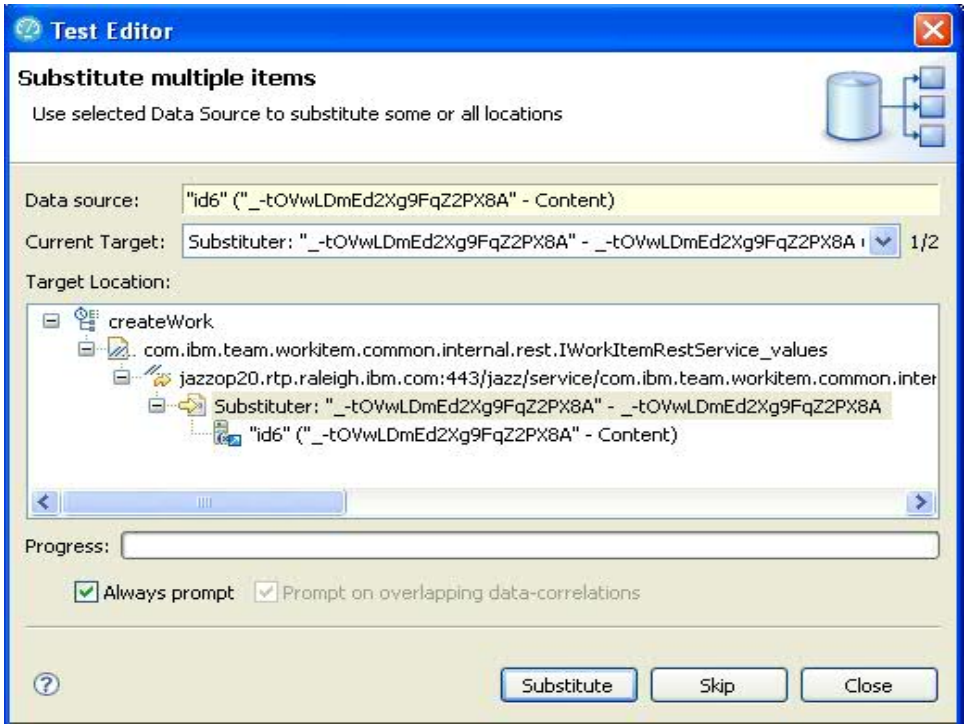
Select all matches that show up in Search Window, right click, Substitute In Data Source View

In the Test Data Sources Window, select the References tab, and select the reference you just created. Click Substitute From.



Find More Substitution Sites continued

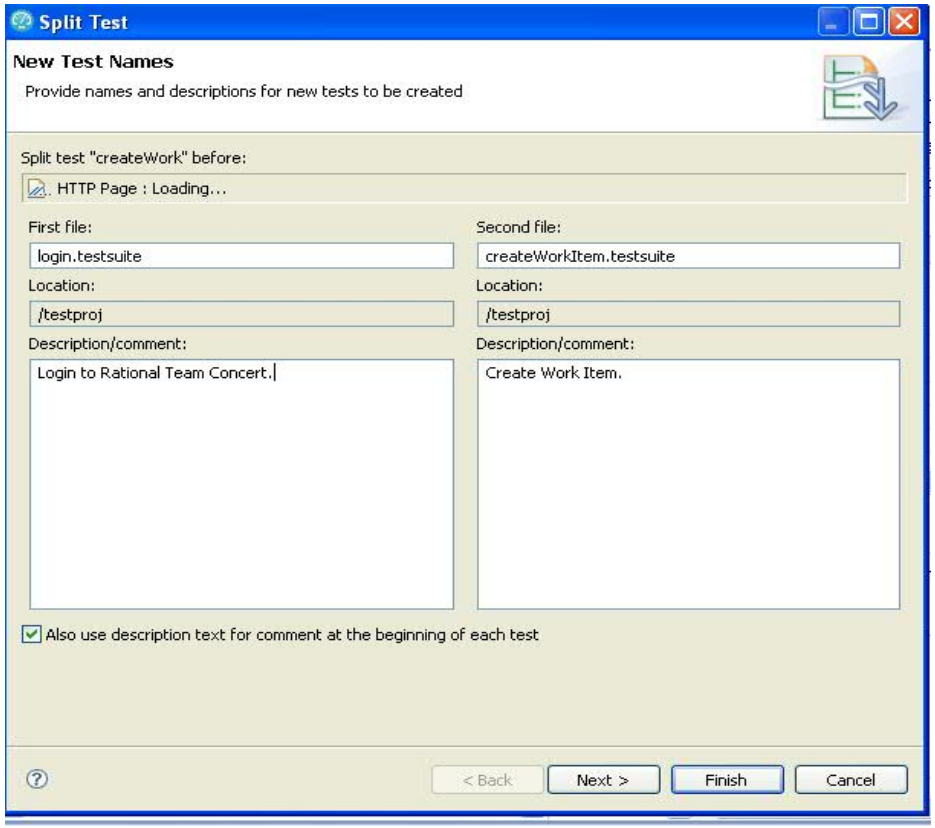
- When you are substitution more than one site at a time, you will be presented with a dialog that will let you control what you do for each substitution site. Finish this dialog and you're done! You've correlated all missing correlations for the value.



DEMO

Split Login and Logout from CreateWorkItem

- Right Click on Login page, Split Test. Name your two tests, Click Finish. Repeat for logout script.



Questions

© **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.