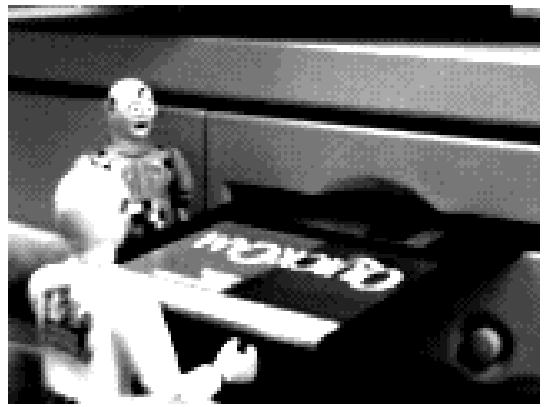
Rational. software

Functional Testing Overview from IBM Rational: What's New, What's Cool?

Neil Williams
Technical Consultant, Automated Software Quality
nwilliams@uk.ibm.com



IBM Rational Software Development Conference UK 2007

What keeps me **Rational**?

QM4

IBM

Agenda

- **The Current State of Manual Testing**
  - Challenges of Manual Testing
  - Rational Manual Tester
- Functional Testing Today
- What's new, what's cool?

What is Manual Testing?

- Using human labor to validate the business use cases and fitness for use of an application, system or device
- Includes labor to plan, design, execute and evaluate the status of the application or device under test
- Experience/research indicates that 75-80% of the testing done today is performed manually

Why Teams use Manual Testing?

Recruitment of subject-matter experts and end-use

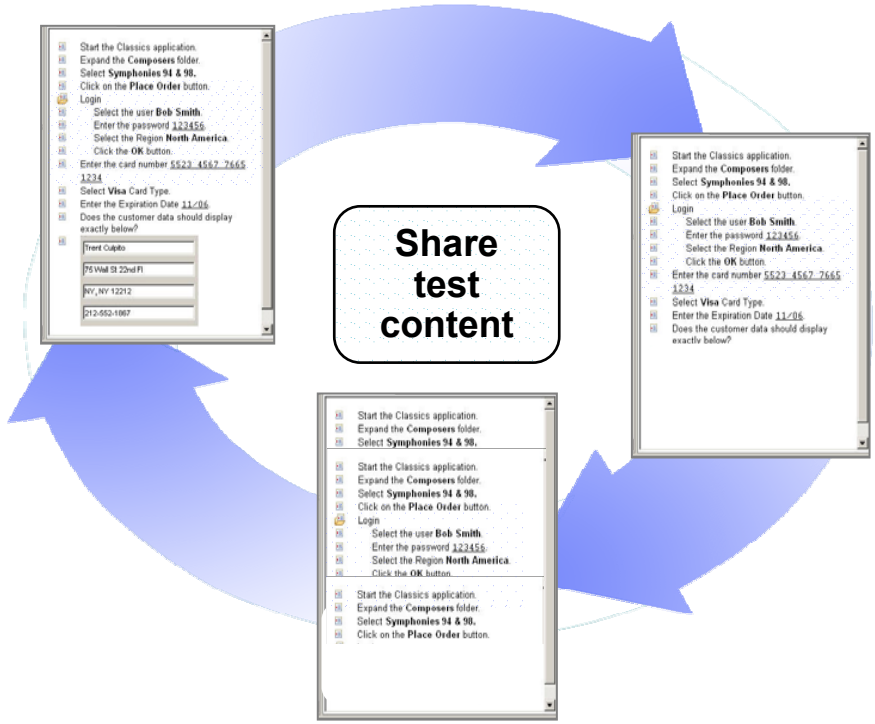
Strict regulatory or process requirements aren't met with automation tools

Poor or non-existent test automation-ready interfaces

Challenge 1: Labor Intensive

- **Challenge**
  - Creating and maintaining manual tests is labor intensive and difficult with tools such as Microsoft Word or Excel.

- **Resolution**
  - Implement a solution that promotes the best practice of modular tests which can share common content
  - Leverage existing Word or Excel test documents to quick-start test development



Challenge 2: Unreliable

- **Challenge**
  - Human error often skews tests results
- **Resolution**
  - Assist testers with the tasks most prone to error – data entry and data verification
  - Build manual tests that are clear, concise and easy for the tester to follow

Challenge 3: Resistant to Change

- **Challenge**
  - Each project has diverse processes, tester skill sets and types of applications and devices under test

- **Resolution**
  - Provide a solution that testers of all skills levels can leverage
  - Customize the tool to fit each project's process metrics and values
  - Leverage a solution that is independent of the software or hardware under test

Agenda

- **The Current State of Manual Testing**
  - Challenges of Manual Testing
  - Rational Manual Tester
- Functional Testing
- What's new, what's cool?

Transform Manual Testing with IBM Rational Manual Tester

From ad-hoc testing to controlled and efficient testing

- Support analysts, subject-matter experts and professional testers
- Develop detailed, easy to follow manual tests
- Minimize test maintenance and ensure test consistency
- Reduce human error during data entry and data verification
- Support diverse usage models and processes
- Built on open source Eclipse architectural framework and TPTP (Test and Performance Tools Platform)

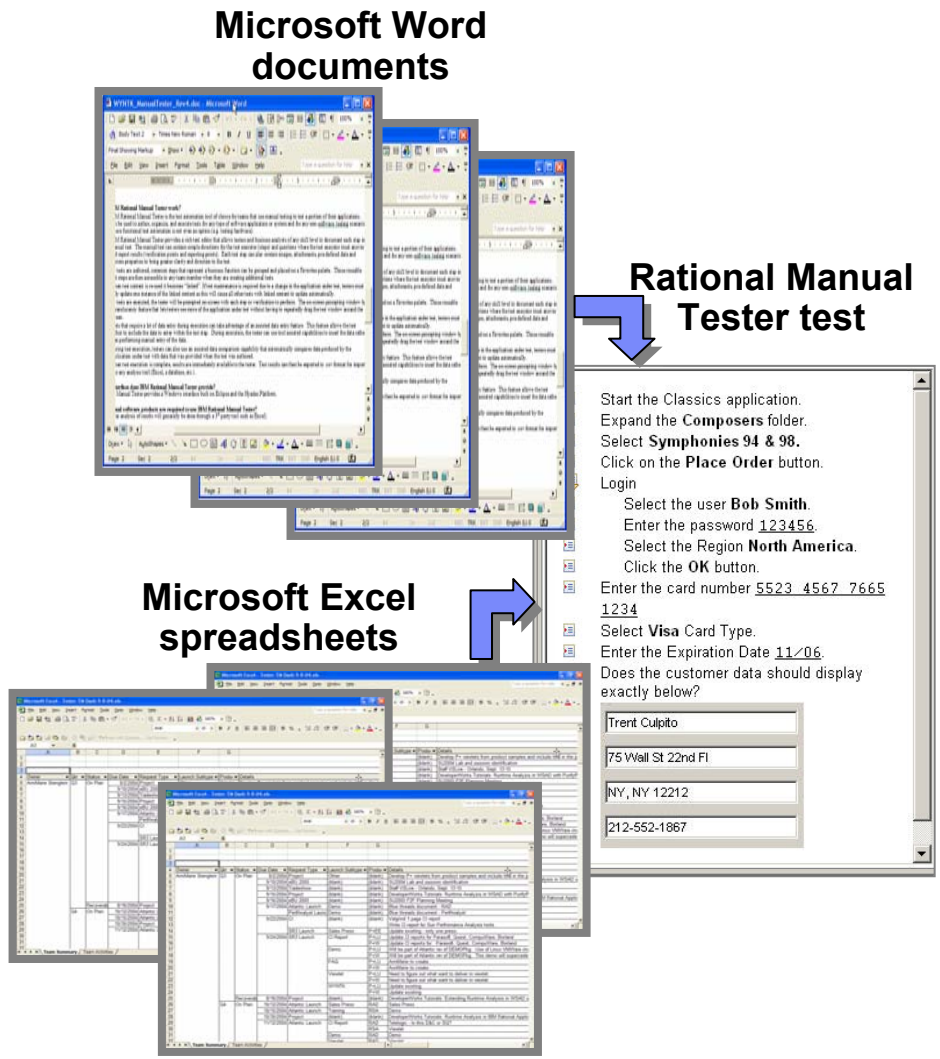
Analysts

Technical & non-technical Testers

Leverage Existing Test Documents

Import from Word or Excel

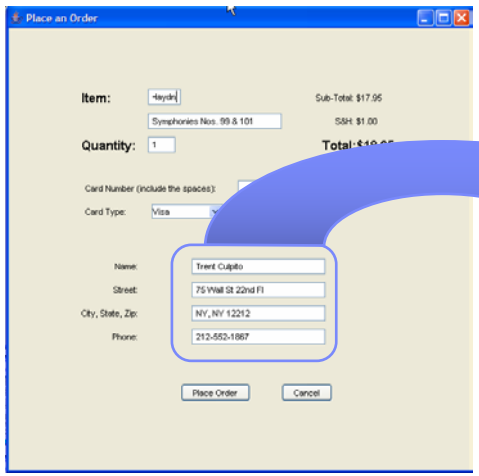
- Jump-start adoption of Rational Manual Tester by leveraging existing assets
- Bulk import manual tests documented in Word or Excel



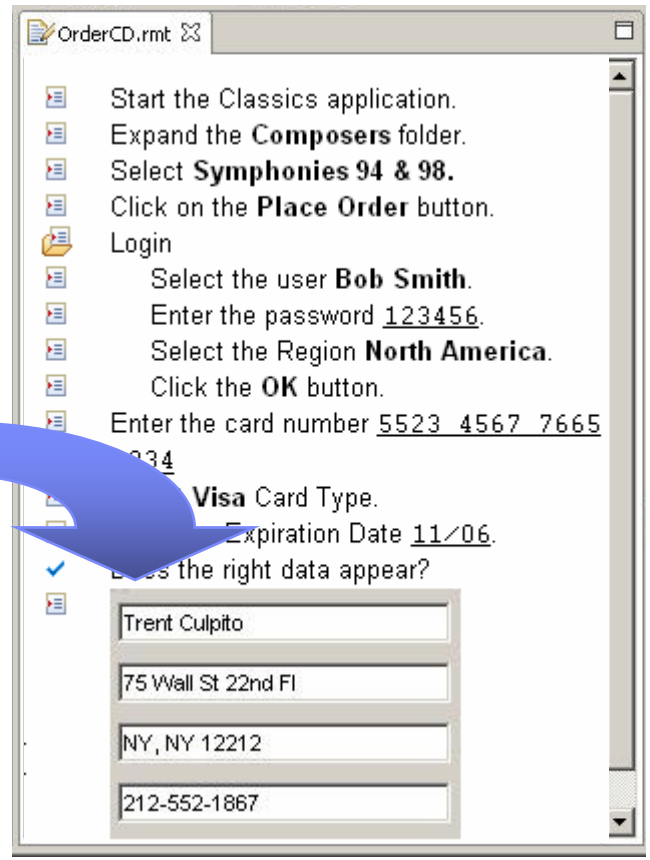
Create Robust, Easy to Follow Tests

Rich text editor

- Provides a rich text editing to document clear, easy to understand tests
- Allows embedding of images or attach files to add clarity and detail
- Uses granular and broad verification points to capture both detailed observations and higher level test results



Manual Test document

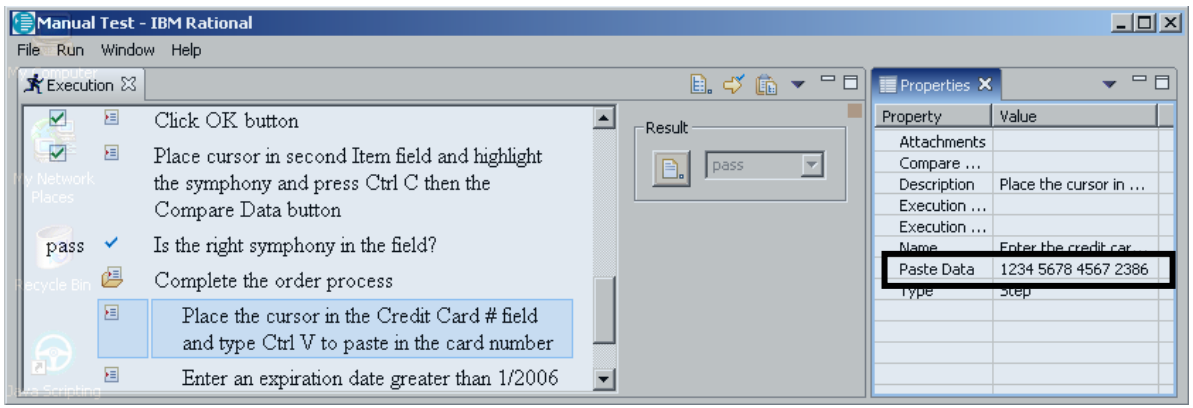


Assisted Data Entry and Data Verification

Reduces human error

- Store critical data used during the test as part of the test instructions

Test execution window

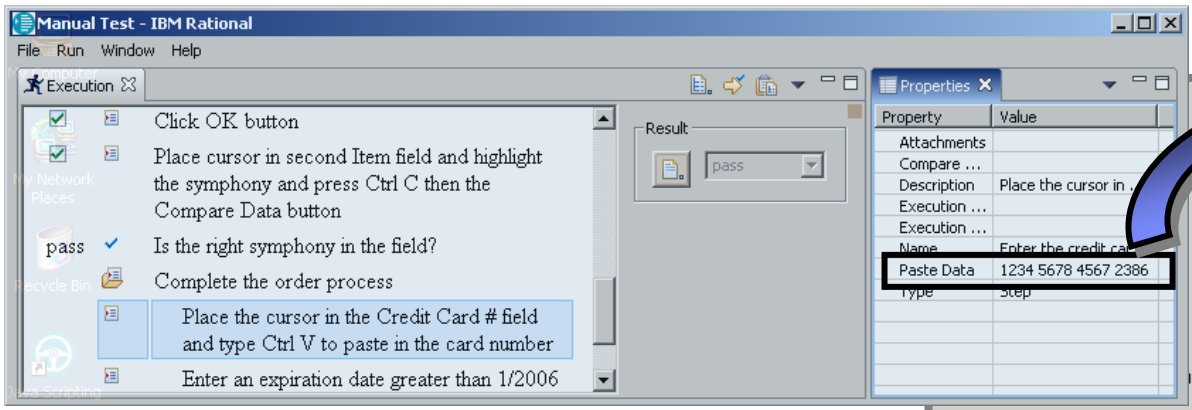


Assisted Data Entry and Data Verification

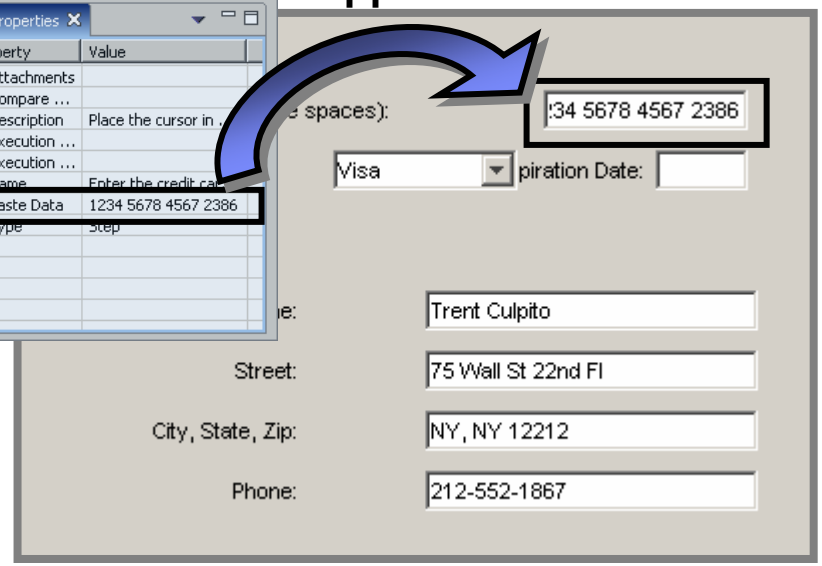
Reduces human error

- Store critical data used during the test as part of the test instructions
- Automate data entry during test execution to speed entry and reduce errors

Test execution window



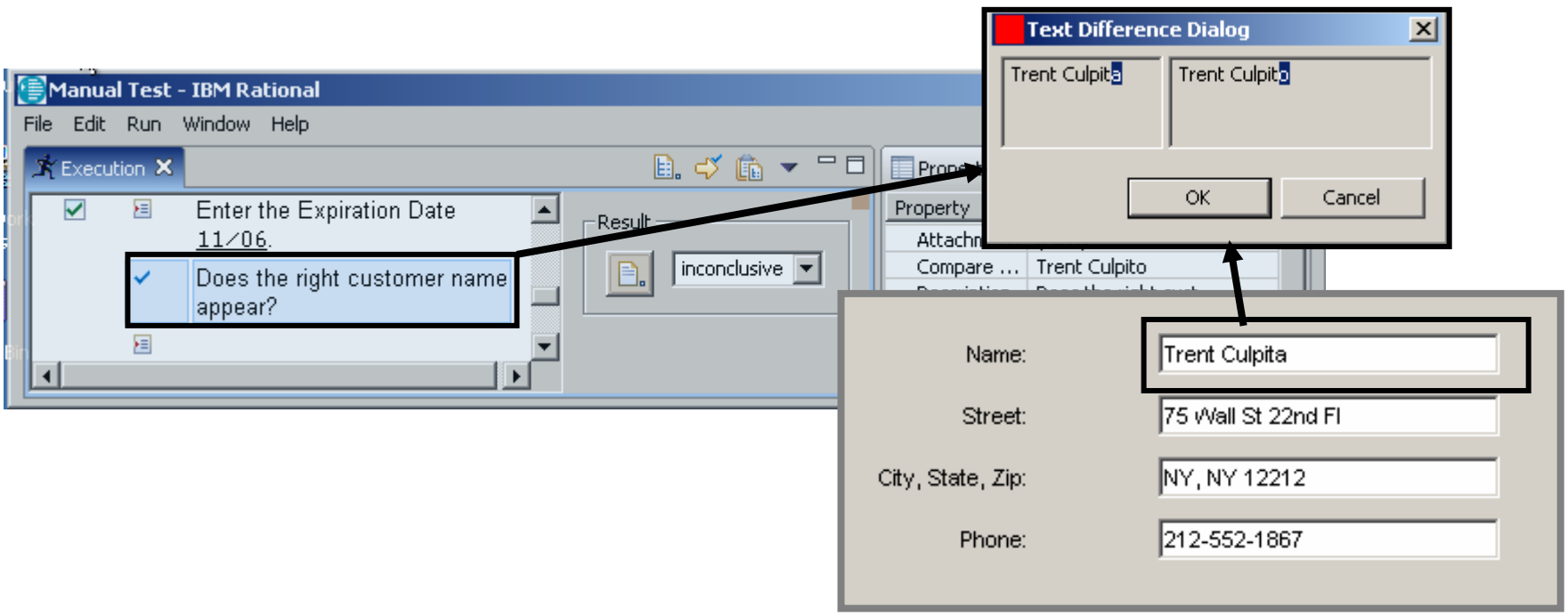
Application under test



Assisted Data Entry and Data Verification

Reduces human error

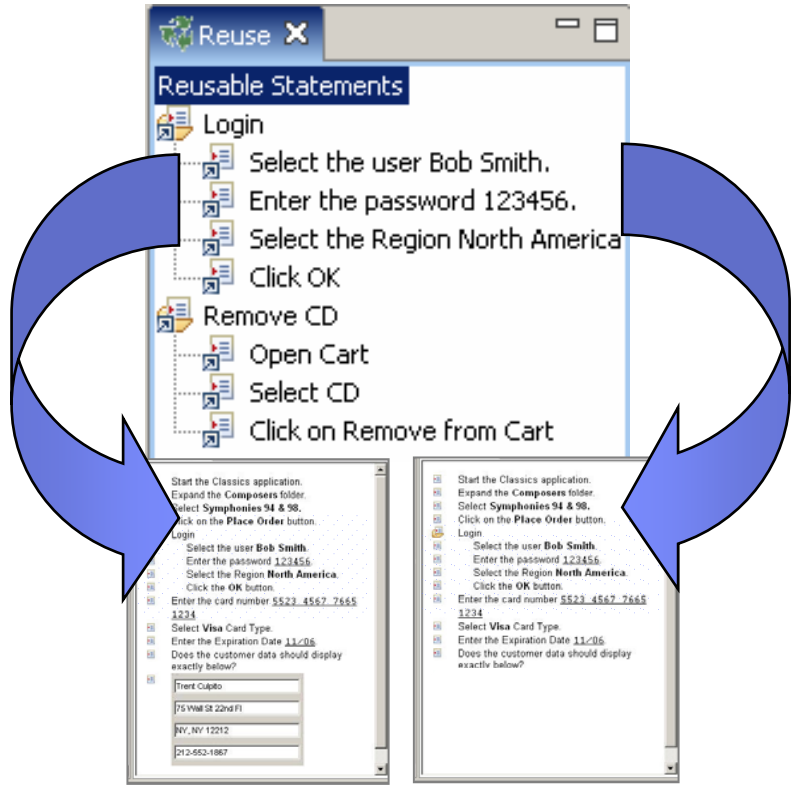
- Store critical data used during the test as part of the test instructions
- Automate data entry during test execution to speed entry and reduce errors
- Automate data validation during test execution to ensure accurate results



Enables Shared Content Across Multiple Tests

Reduces test maintenance and ensure consistency

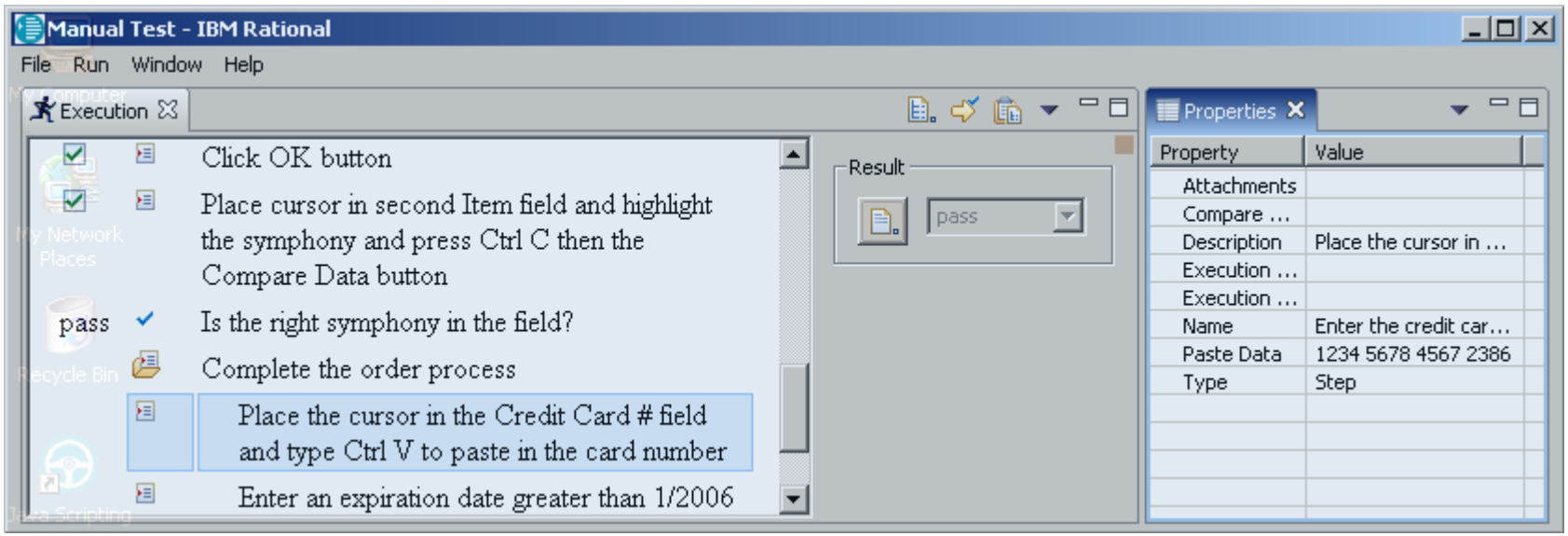
- Create commonly used test statements once
- Reuse statements across multiple tests through simple drag and drop
- Maintenance of shared content is efficient with single-point updating of shared content
- Ensures consistency and reduces test maintenance



On-Screen Prompted Test Execution

Immediate and accurate results collection

- On-screen prompting of test steps
- Immediately record test results and attach images or files for clarity
- Tool will assist with data entry and data verification
- Results recorded in permanent log



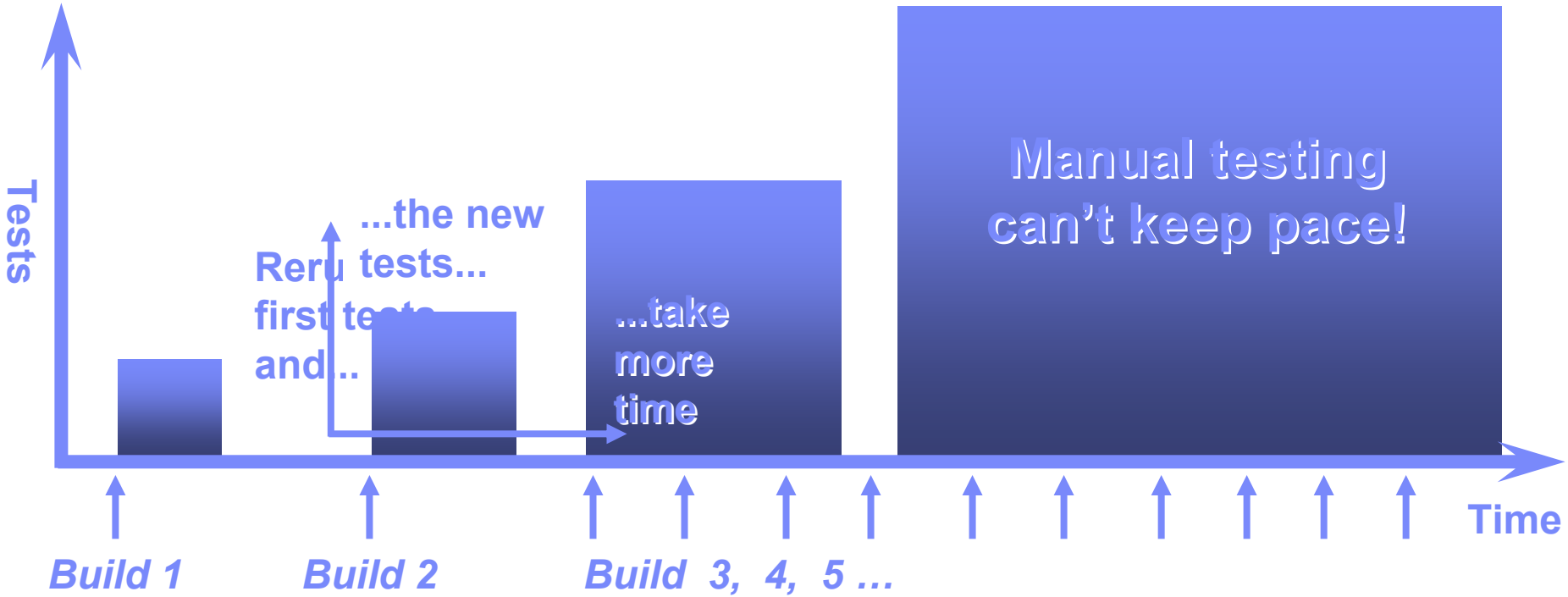
Summarizing Key Features in Manual Tester

- **Advances manual testing**
  - Rich text editor for test authoring
  - Test step reuse palette
  - Test script import from Microsoft Word or Excel
- **Satisfies diverse needs**
  - Supports usage by distributed teams
  - Generates spreadsheet-ready results data
  - Multiple validation points
  - Custom data fields
- **Simplifies through assistance**
  - Assisted data entry and data verification during test execution

Agenda

- The Current State of Manual Testing
- **Functional Testing**
  - Challenges of Functional Testing
  - Rational Functional Tester
- What's new, what's cool?

Challenge of Manual Testing with Short Test Cycles

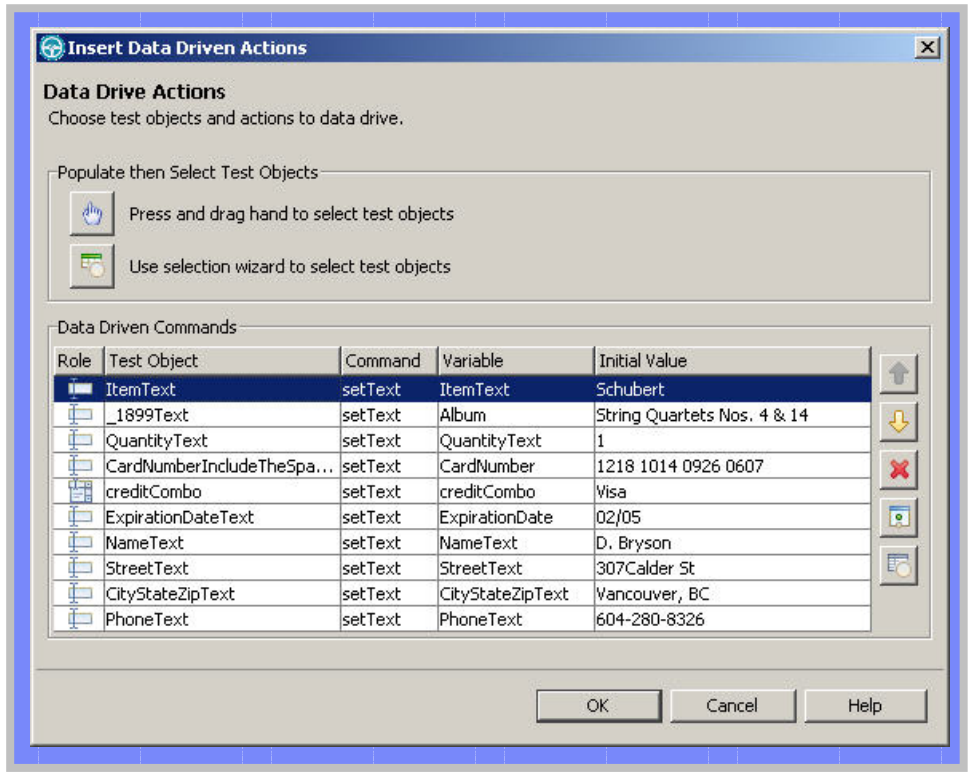


- Manual testing alone can't keep pace with the rapid development of applications
  - Quality will be sacrificed to keep on schedule
- Automation will help reduce costs

Challenge 1: Controlling the cost of Automation

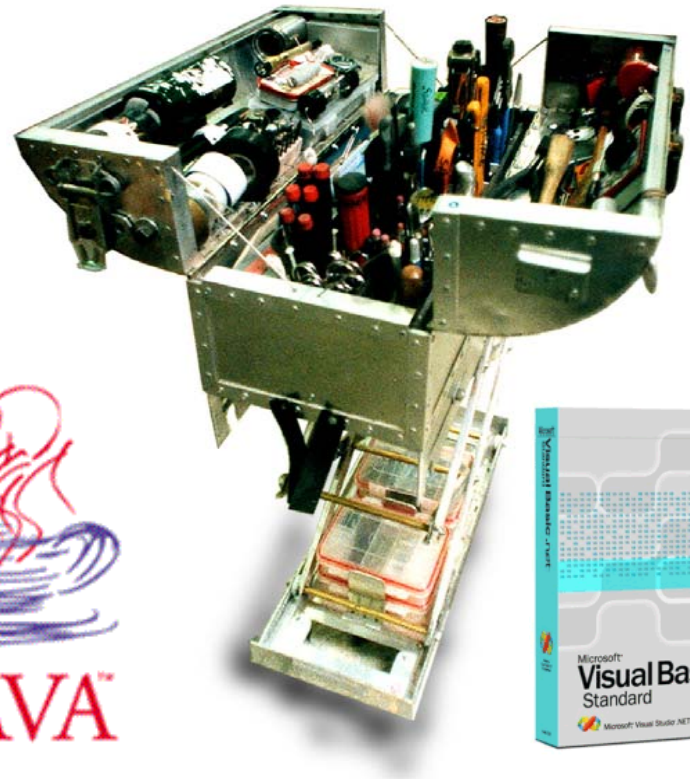
- **Challenge**
  - Overcome the complexity of test automation to ensure teams spend more time on high value quality assessment rather than on low return test maintenance.

- **Resolution**
  - Provide technologies to enable novice testers to craft complex tests with no programming
    - Data driven wizard
    - Dynamic data verification
  - Reduce script maintenance
    - Wizard driven global updates to centralized object map



Challenge 2: Tools Lack Flexibility

- **Challenge**
  - Fulfill both standard test needs as well as expert requirements to ensure usage limited only by the imagination of the project team.

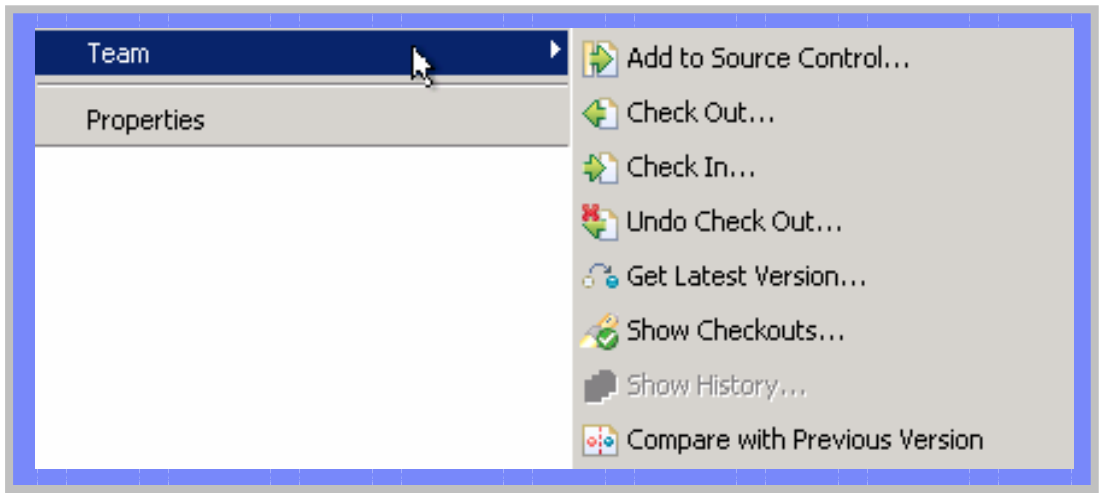


- **Resolution**
  - **Provide two test script development languages**
    - **Java**
    - **Visual Basic .NET**
  - **Give professional testers the flexibility to overcome testing challenges in the language of their choice**

Challenge 3: Test Team Integration

- **Challenge**
  - Accommodate the requirements of both dedicated, independent QA teams as well as integrated project teams to ensure high operational integrity.

- **Resolution**
  - Provide version control support in the box with ClearCase LT
  - Provide integrations for standard version control features
    - Check In/Out
    - Compare with Previous Versions
    - Show change history



Agenda

- The Current State of Manual Testing
- **Functional Testing**
  - Challenges of Functional Testing
  - Rational Functional Tester
- What's new, what's cool?

Functional Testing with IBM Rational Functional Tester

Test automation for the novice and the professional

- **IBM Rational Functional Tester**
  - Automated regression testing for Web, J2EE, .Net, Terminal, Siebel SAP and mySAP applications
- **Key Benefits**
  - Minimize test maintenance with scripts resilient to application changes
  - Wizard enhanced automation to speed test creation for the new user
  - Powerful scripting language and IDE for the professional tester
  - Supports Team oriented parallel development

What is Automated Testing?

- Using automation tools to record actions taken against an application.
- Test scripts are derived from these recordings which are then used for subsequent test execution.

Effective Test Automation

Creating an automated test is a three steps process

Record

- Script Recording
  - Capture user interactions with the system under test

Effective Test Automation

Creating an automated test is a three steps process

Record **Enhance**

- Script Recording
  - Capture user interactions with the system under test
- Script Enhancing
  - Modify scripts for situation specific challenges

Effective Test Automation

Creating an automated test is a three steps process

Record **Enhance** **Execute**

- Script Recording
  - Capture user interactions with the system under test
- Script Enhancing
  - Modify scripts for situation specific challenges
- Script Execution
  - Ensure reliable playback for local or remote execution

Effective Test Automation

Recording Scripts

- Create a Test Script that captures system-user interaction
  - Test scripts are recorded on the fly as user navigates application
  - Verification points are inserted to validate system response

- Key considerations when recording scripts
  - Broad environment support
  - Create data driven tests without coding
  - Static data and properties verification
  - Dynamic data validation without coding
  - What can't or shouldn't be automated?

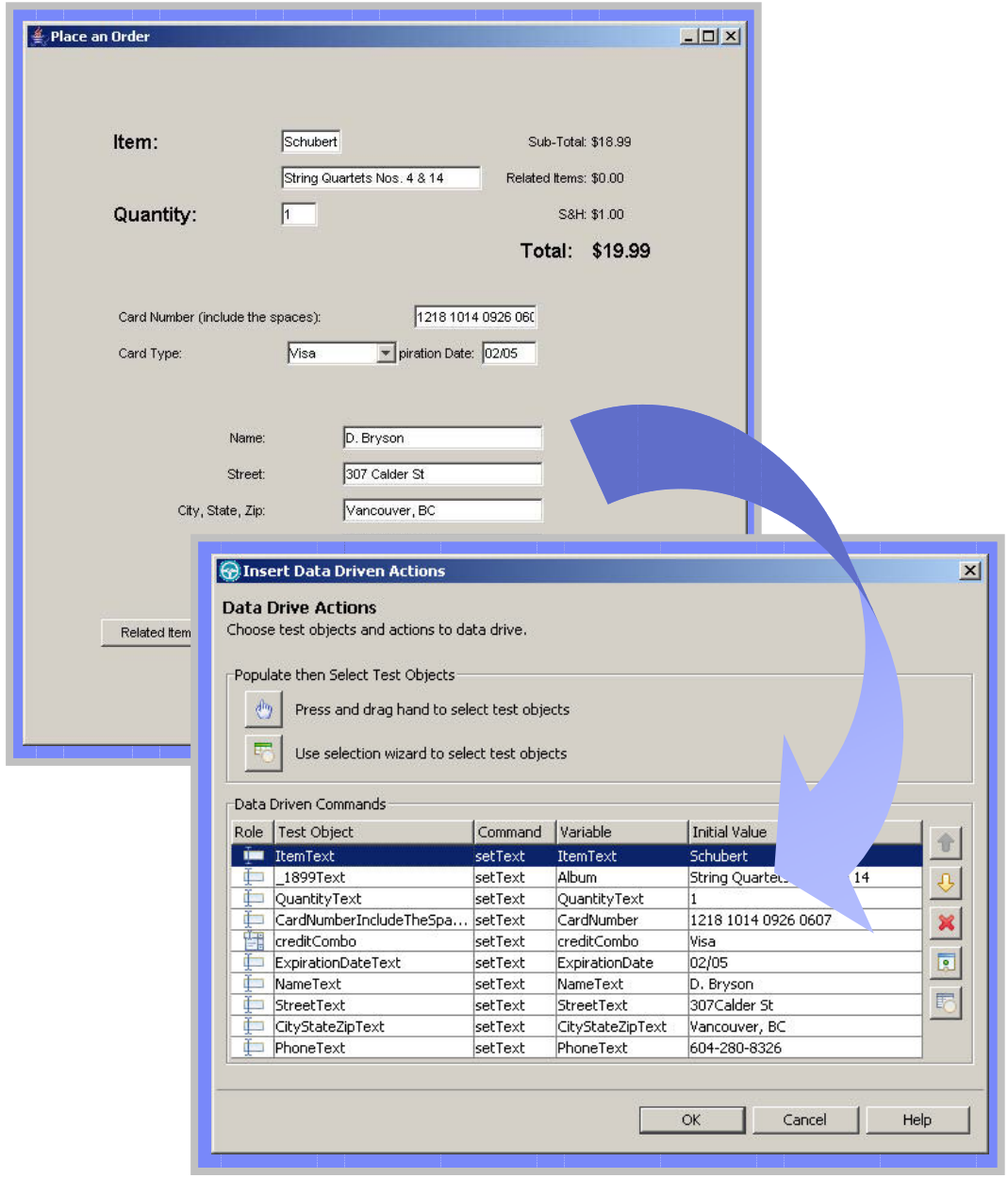
Recording Scripts
Environment Support

- IBM Rational Functional Tester supports the following environments
  - Web
  - Java
  - Any VS.NET application running under the .NET Framework
  - Siebel Applications
  - SAP applications
  - Infragistics controls (available soon)
- Terminal Emulation
  - Add-on support for 3270/5250 terminal-based applications

Recording Scripts

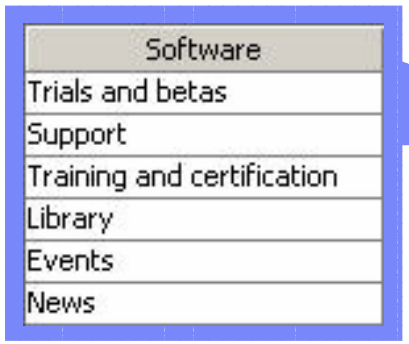
Data Driven Testing

- **Data Driven Testing**
  - Separates test data from test script
  - Enables a single script to run multiple tests by using multiple data sets
- **Wizard driven process**
  - No programming involved
  - Import data from external sources

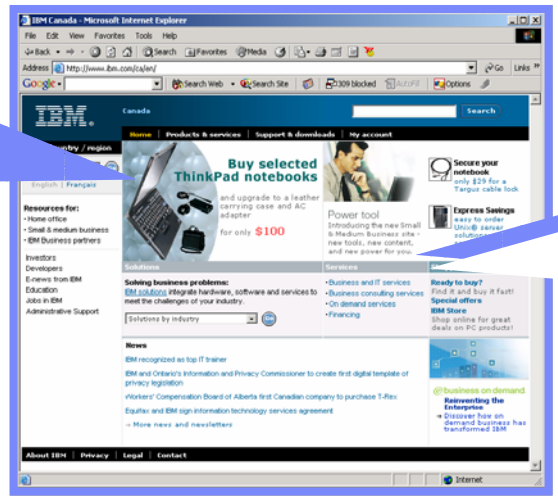


Recording Scripts

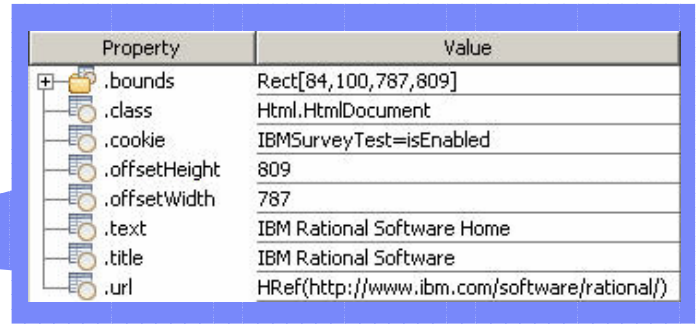
Verification Points



Functional Tester Sees Data



You See…



Functional Tester Sees Properties

- **Automated Validation**
  - Functional Tester captures data and properties that can be invisible to users
  - During script execution, current results are compared to stored baselines
  - Discrepancies are flagged and reported to user in an HTML based test log

Recording Scripts

Validating Dynamic Data

- Dynamic Data/Content Matching
  - Use pattern matching technique to verify dynamic data and create robust tests
  - E.g. Instead of validating “Order ID 230”, validate “Order ID ###” or Order ID 2##, etc.
  - This allows for a wide variety of acceptable responses as well as restrictions on acceptable responses when validating the application’s behavior

Effective Test Automation

Enhancing Scripts

- Enhancing Scripts with basic coding extends their value and reach
  - VB.net or Java code is added to perform a variety of functions
  - Typical Modifications: Conditional branching, datapooling, refactoring
- Key considerations when enhancing scripts
  - Flexible coding language
  - Powerful, professional debugger
  - Object map editing flexibility
  - Version control of scripts

Enhancing Scripts

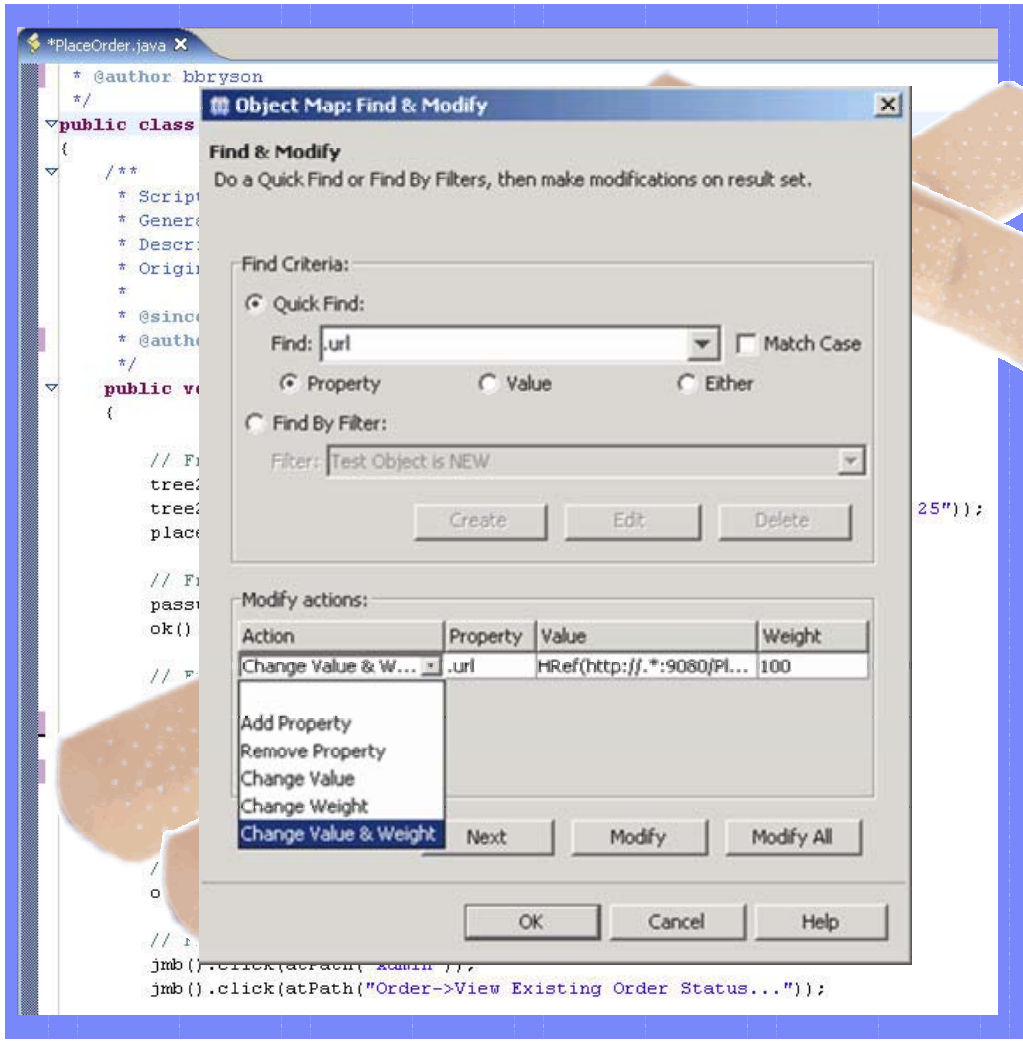
Flexible Coding Language

- IBM Rational Functional Tester uses Java or VB.net for scripting
  - Standard language syntax
    - Not a custom version of Java or VB.net
  - Augments language commands with test specific functions
    - Click, Verify, Select, etc…
- Flexible power to enhance scripts
  - Programmatic access to all GUI objects
  - Datapool facility enables data driven tests
  - Leverage existing code and resources from a variety of sources
    - Books, Internet, developerWorks, etc…

Enhancing Scripts

Object map editing flexibility

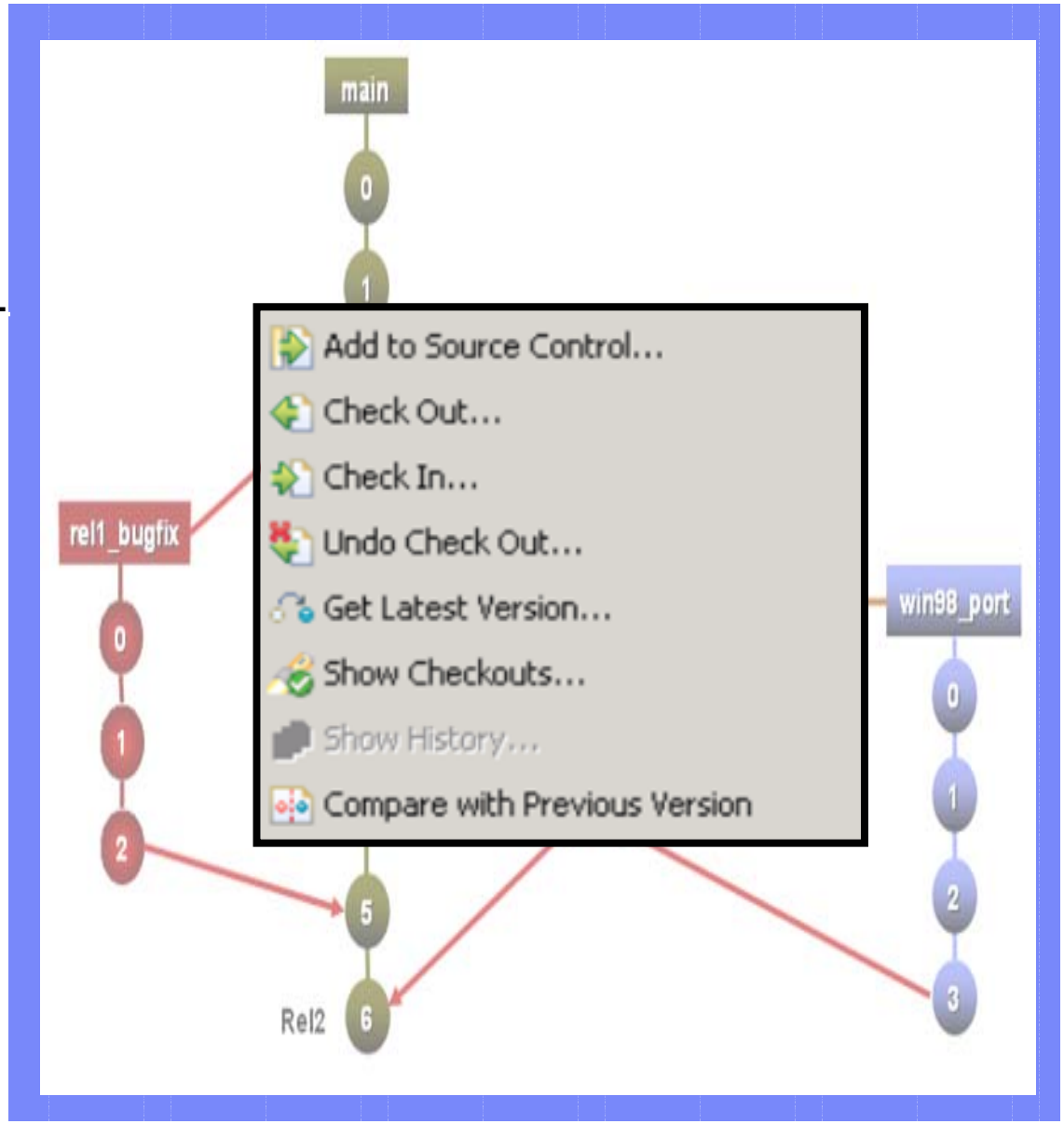
- Script Maintenance can outpace script development as the volume of tests grows
- Functional Tester includes an Object Map update tool which enables batch updates to a centralized object map
  - Reduces time spent fixing individual scripts
  - Frees up more time for script development



Enhancing Scripts

Version Control of Scripts

- Functional Tester includes ClearCase LT for test script version control
- ClearCase LT enables parallel script development
  - Work simultaneously on multiple releases
  - Coordinated workflows across the team
  - Secure, versioned software test assets



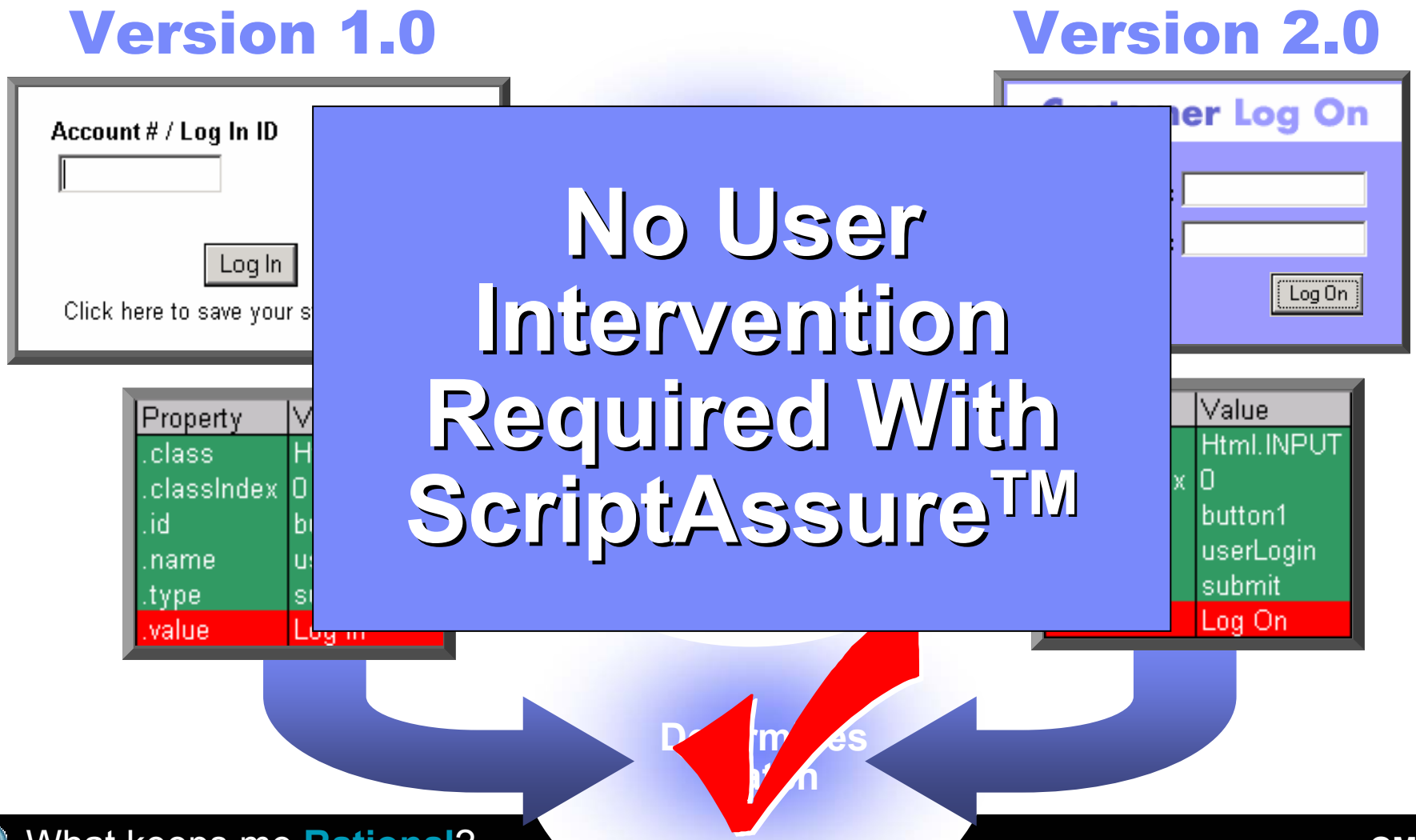
Effective Test Automation

Executing Tests

- Scripts are executed, discrepancies are noted
  - Scripts are executed and test logs created
  - Test logs are highlight differences between actual and expected results
- Key considerations when executing scripts
  - Reliable playback with ScriptAssure

Reduce Test Script Maintenance

Reliable Playback with Script Assure



Agenda

- The Current State of Manual Testing
- **Functional Testing**
  - Challenges of Functional Testing
  - Rational Functional Tester
- What's new, what's cool?

What are our customers telling us?

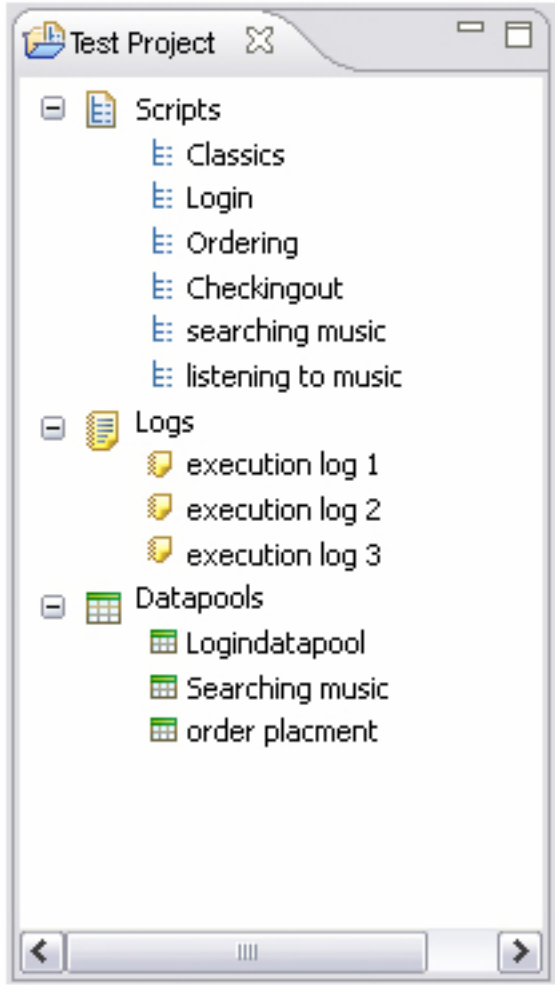
- **I need help automating my manual tests**
- **Our tools are too complex**
- **Our tools are not extensible**
- **Additional control & domain support**

What's new in RMT 7.0.1?

- **Improvements to the editor**
  - Improved control of fonts and styles
  - Spell check and auto-complete (tentative)
- **Project based development**
  - RMT inheriting the project concept similar to other desktop products
- **Datapool support for RMT**
  - Datapool can be shared between RMT and RFT
- **Automation of manual statements**
  - Keyword testing integration with RFT

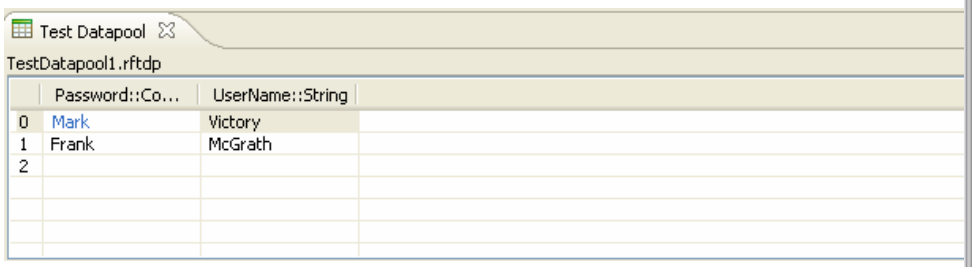
RMT Updates

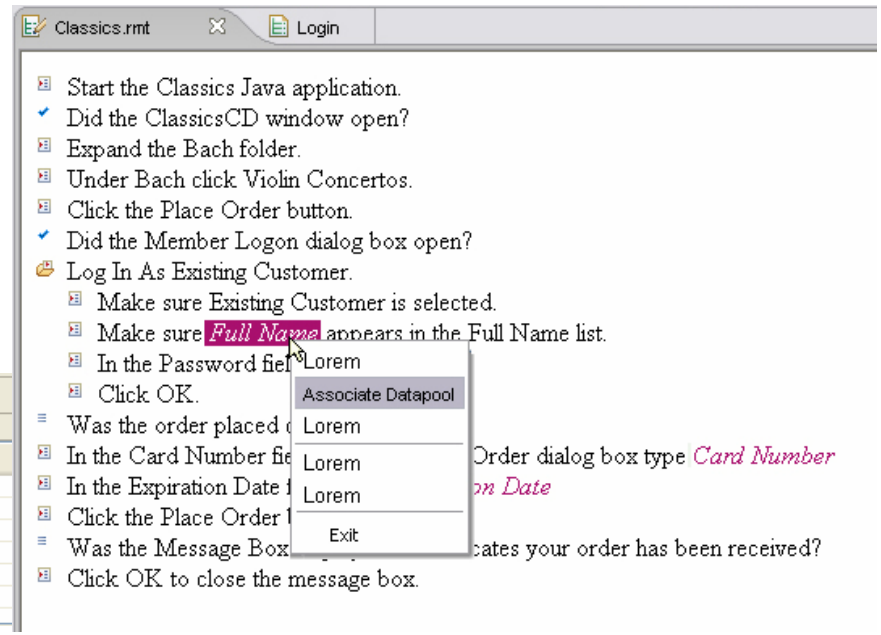
- Projects
  - The project will be a logical grouping of Manual and Functional test scripts
  - Users that desire the keyword capabilities will be required to utilize the new project concepts



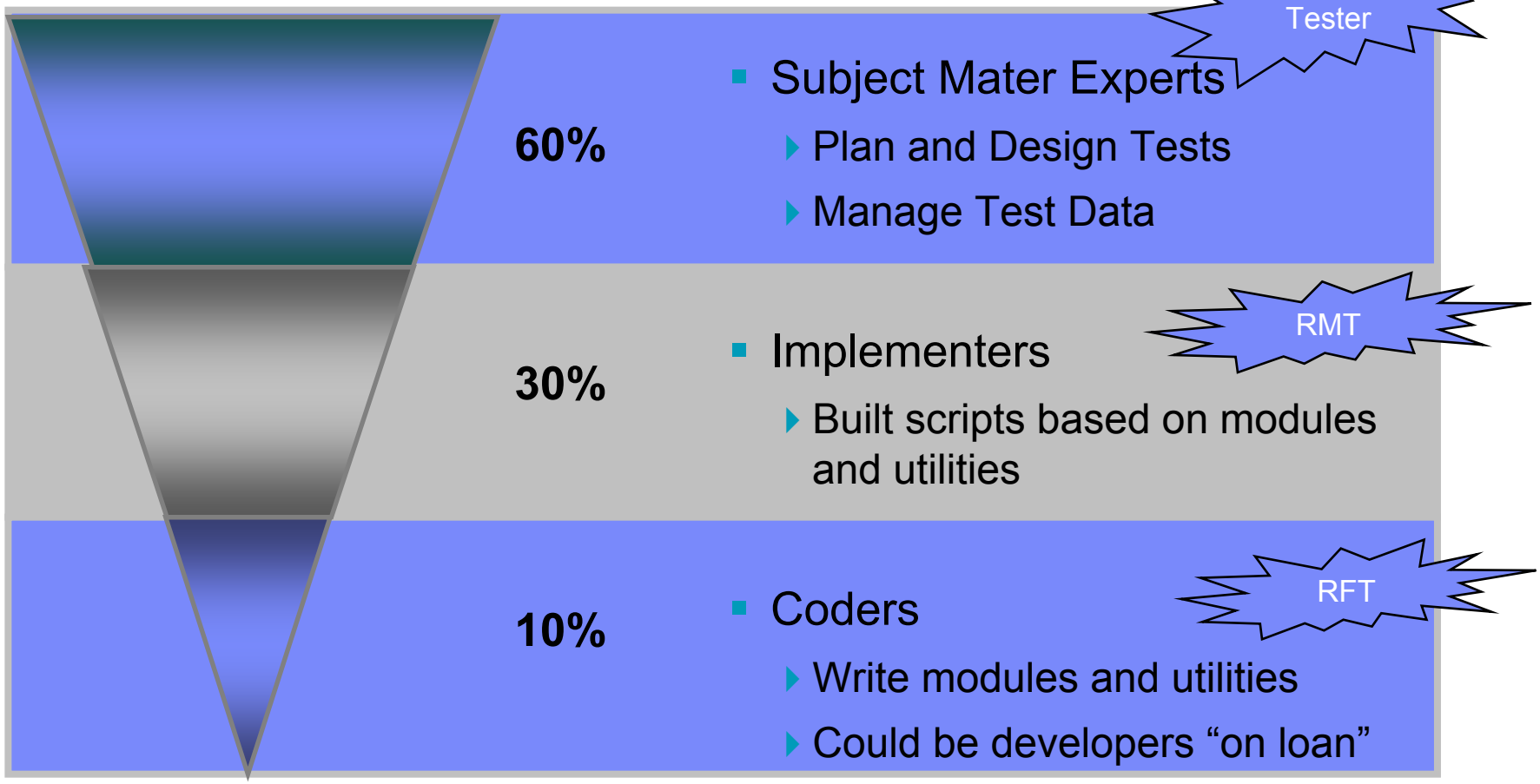
RMT Updates

- Datapools
  - Users will be able to associate a datapool to a manual test script in RMT
  - RMT scripts will be to associate variables (datapool rows) to a script and upon execution the datapool is traversed.
  - Parameterized Data easily visualized in the RMT Scripts





RMT updates - Keyword Testing support

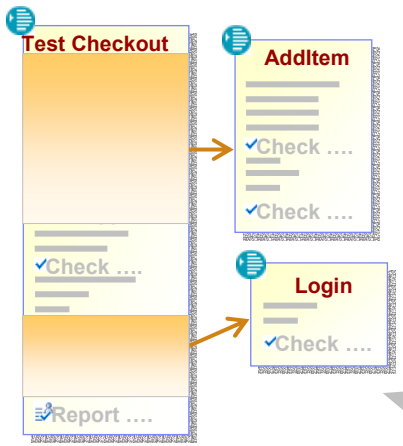


Objectives for the Keyword tester support

- Ease of Use
- Keyword testing must enable testing earlier in the lifecycle.
- Keywords shall make manual testing faster and easier
  - Keywords can be used to construct manual test
  - Keywords are reusable components
- Keywords shall make test automation faster and easier
- Keywords should be defined by any user type
  - Keywords should be able to be defined and used at any time.
  - Keywords can be queried
- Leverage existing Technologies
  - Rational Manual Tester and Rational Functional Tester

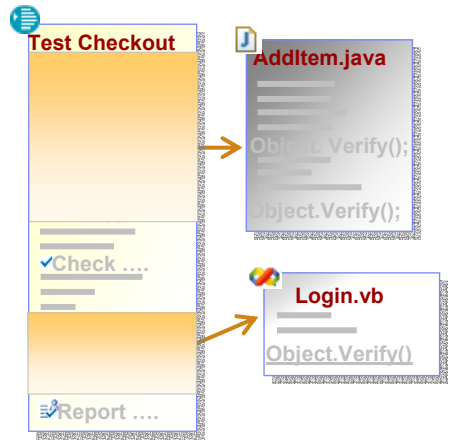
Three Pieces to the Solution

RMT



- Manual tests and Keyword tests created
- Keyword description is the text in the Manual step
- While Creating tests new keywords can be created
- Execution of the test
- Information for generation of Automation Script stored in RMT.

RFT

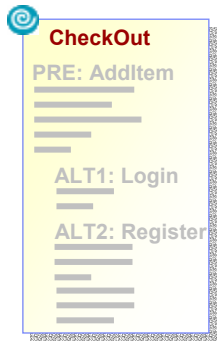


Steps tied to keywords

Keywords realized by Automation

- Keywords associated to Manual test Steps
- Keywords defined
- Keywords associated to automation script
- Information on Keyword captured

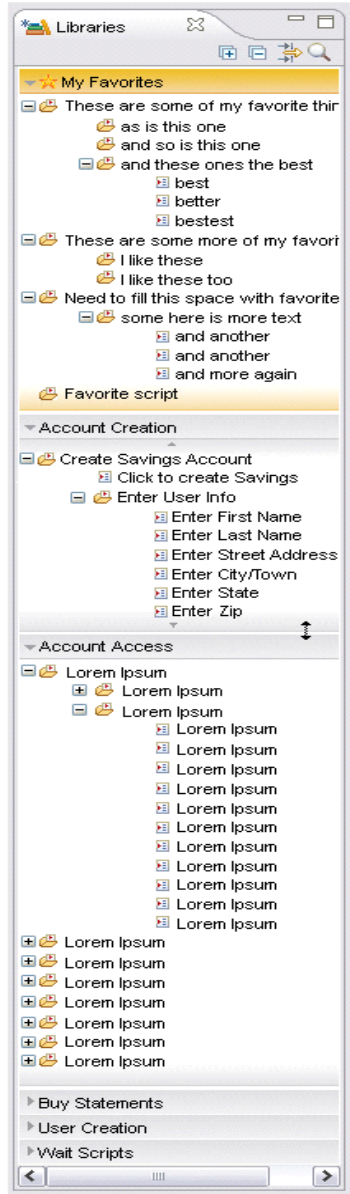
RMT Keywords



- Automation Framework defines Scripts to map to the keywords
- Changes to scripts may not effect the keywords

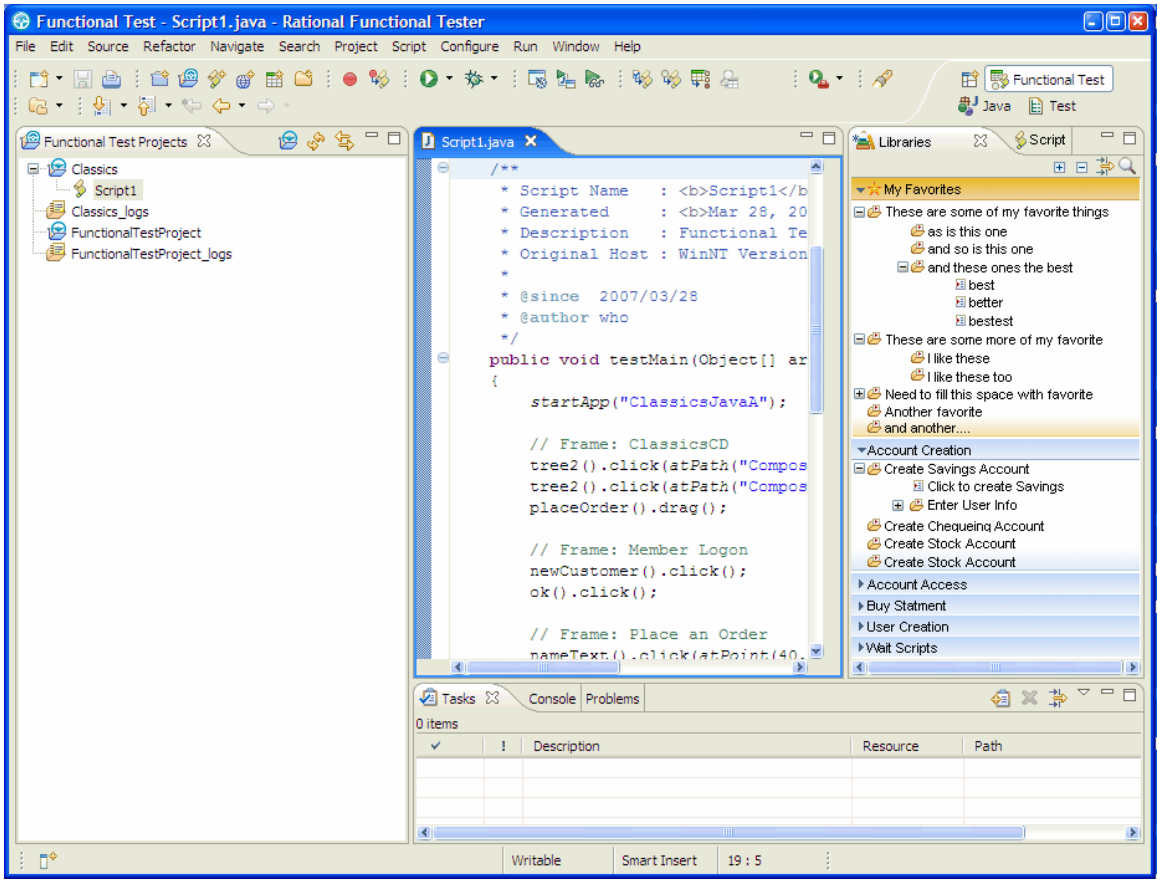
RMT Updates

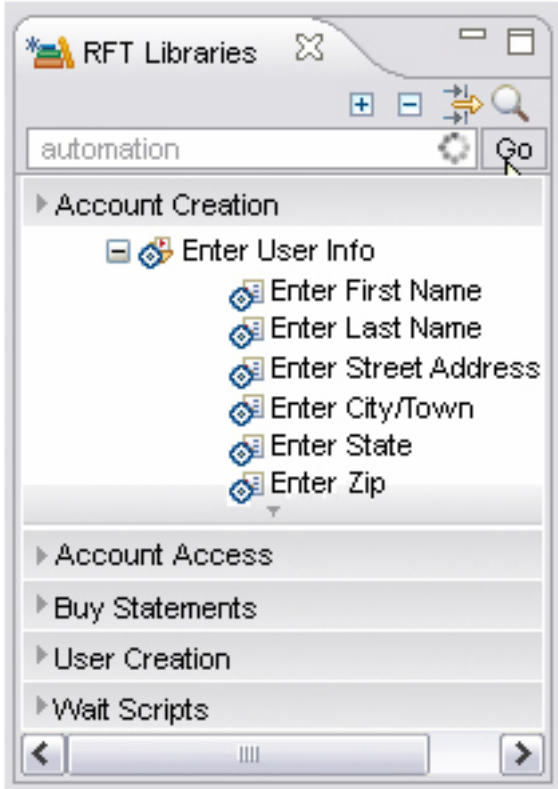
- Keywords as a type of reusable statements
  - Keyword is any RMT statement type
  - A statement can be reused across RMT scripts
- Statement Libraries
  - Reuse statements will be maintained in Libraries
  - Libraries are shared
  - Project can contain 1..m libraries
  - Used as a way to logically organize reuse statements



RFT Updates

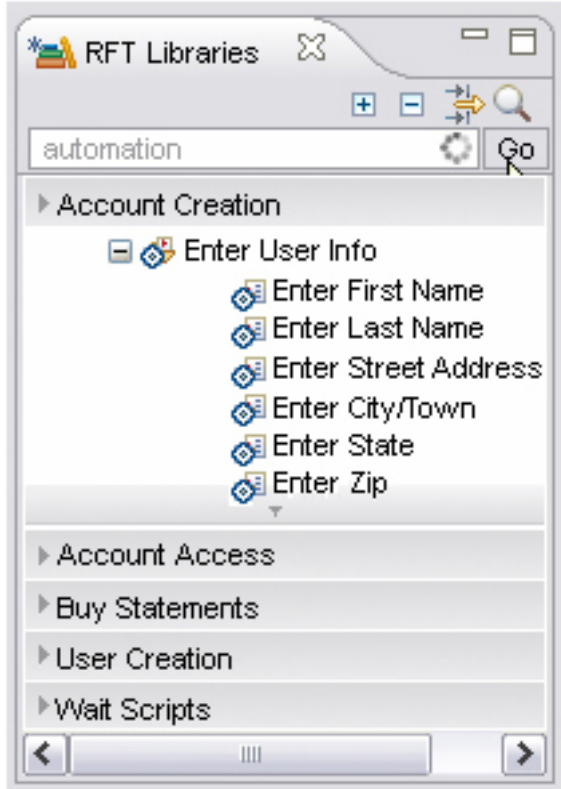
- Keyword Libraries Accessible in RFT
  - Able to create keywords
  - Associate Automation scripts to keywords



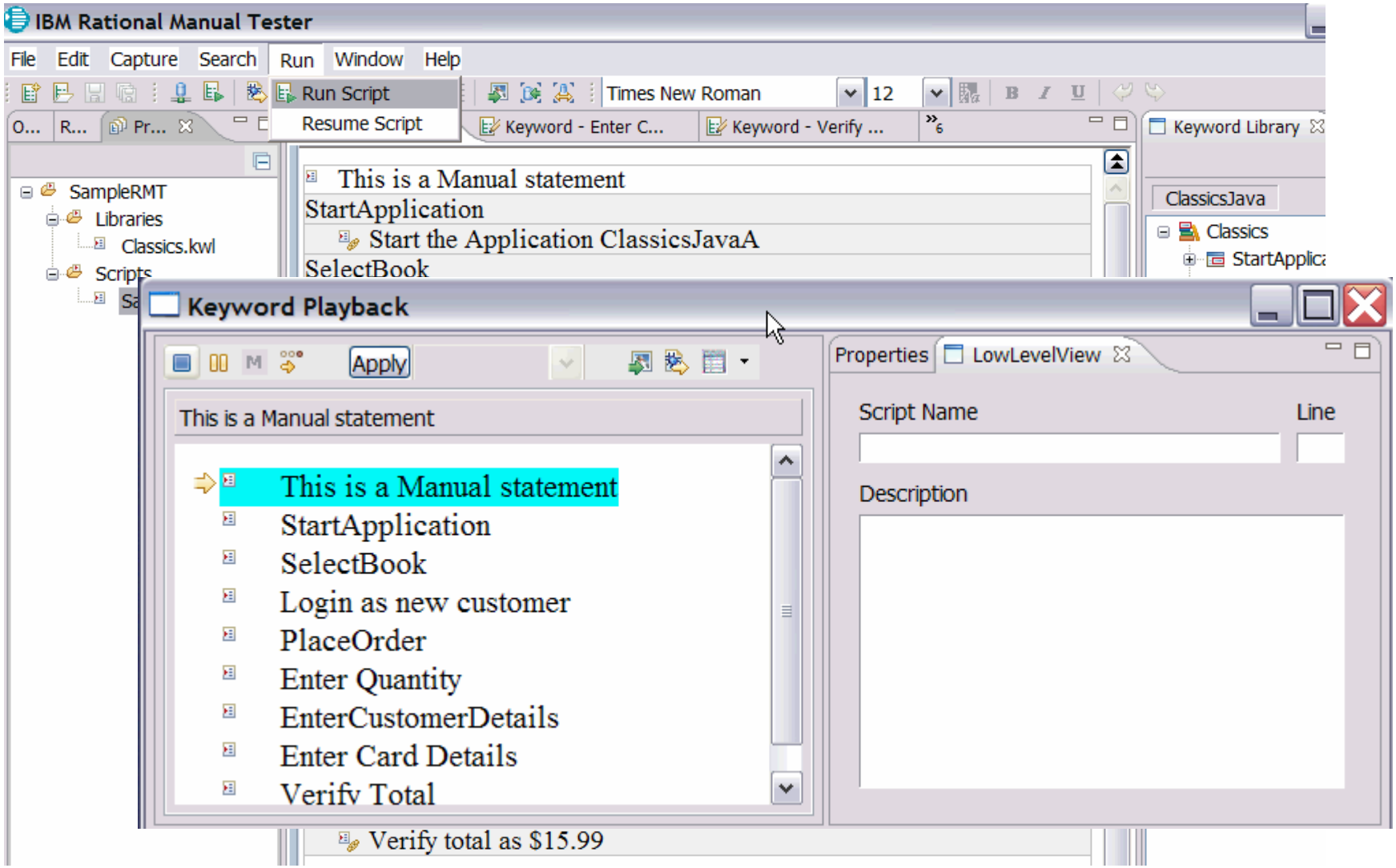


RFT Updates

- Keyword Libraries Accessible in RFT
  - Able to create keywords
  - Associate Automation scripts to keywords



Execution of keyword-enabled tests…



What's is new in RFT?

- **7.0.0.1 (Jan 2007)**
  - IE 7.0 support, Firefox 2.0 support
- **7.0.0.2 (April 2007)**
  - Object Map Properties support
  - Object library support
  - .Net v2.0 controls
  - Vista support
  - Firefox 2.0 support for Linux
  - Ajax support
  - Eclipse 3.2 controls
  - Usability improvements
  - Improved Windows platform support - activeX, MFC

What's coming?

- **7.0.1**
  - Keyword Tester (Eclipse IDE)
  - Proxy SDK (Java/Win domains)
  - Infragistics control support
  - Adobe controls - Flash, Flex (IE browser)
  - Object library UI
  - CCRC support – checkin/checkout
  - Image verification/comparison feature
  - XML logging
  - Full Accessibility
  - SAP certification

Proxy SDK

RFT customers have been requesting a capability to:

- Enable Modifying or Enhancing a RFT supported controls
  - Add new data/properties verification test
  - Change the recording behavior
  - Modify the recognition properties – Object Library could do this as well!
  - Adding new "sub-item"
  - Add new methods to a supported control
- Enable Adding support for an unsupported control
- Enable Adding support for new Domain

Proxy SDK is the answer

- API (Build / Runtime files) for proxy writing are already being shipped
- Documentation on “How to do?”
- Cleaning up of Internal APIs
- Deployable Sample proxy projects with control samples
  - with various difficulty levels of proxy writing challenges
- Lab exercises and Tutorials

Developer Prerequisites for writing proxies

- Programming knowledge with Java or .NET (C#)
- Knowledge of the Control being supported
- Understanding of RFT Framework
- Understanding of existing Proxies and their Hierarchy
- Understanding of existing TestObjects and their Hierarchy

Flex/Flash control support – example of writing a new proxy

- Increasingly Internet applications are becoming Rich
  - Expressiveness
  - Rich media integration
  - Performance/Responsiveness
  - Real time
- Web 2.0 technologies like AJAX are being used to develop Rich internet applications
- Adobe's Flex Development Environment is another platform which is being used to develop Rich internet applications
- Flex is one of the fastest growing platform for developing Rich internet applications

Object library support

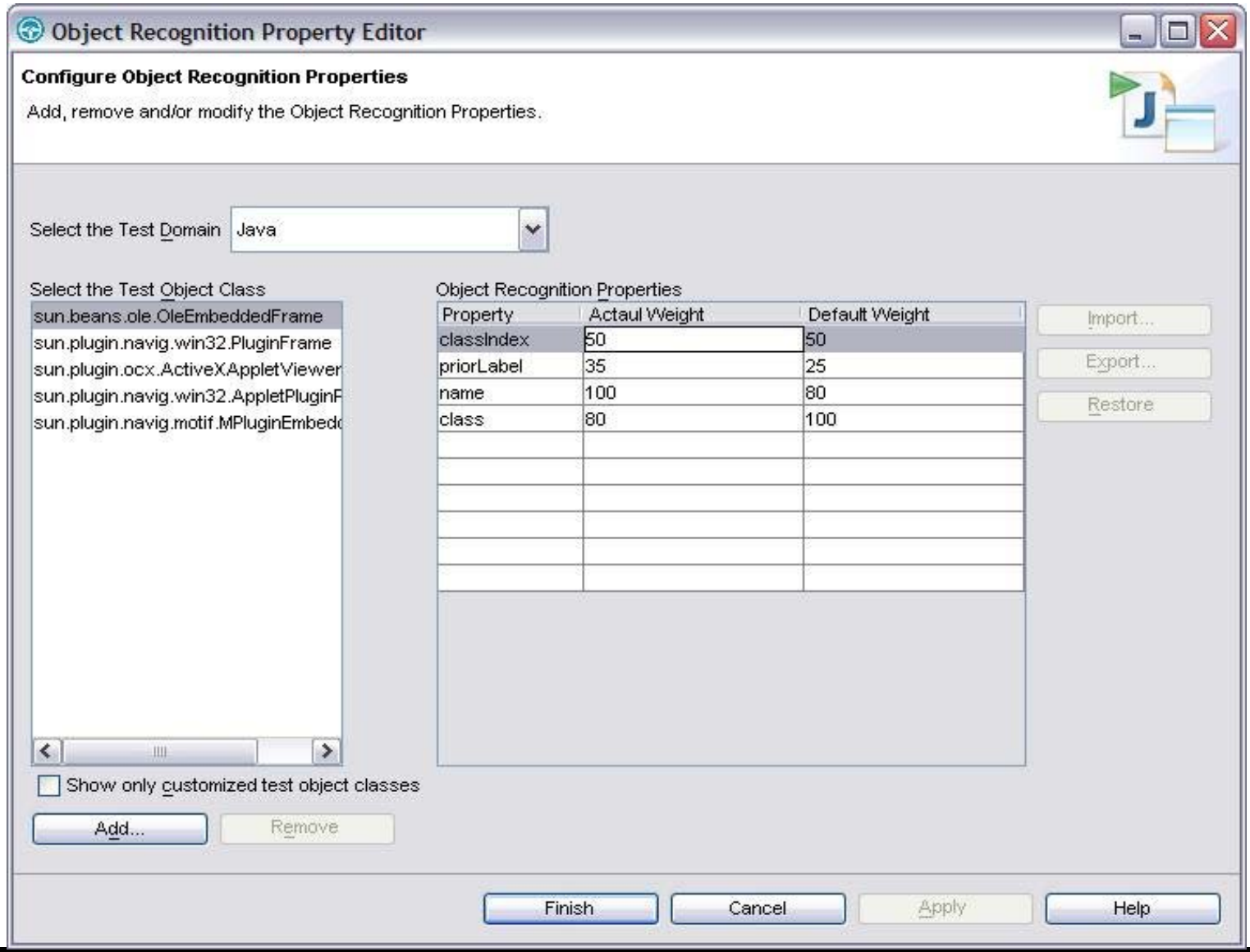
- Allows customers to customize object recognition properties for a class of objects
  - RFT proxy code defines the default recognition properties to be used by the recorder
  - This feature provides flexibility to change the recognition properties for that particular object.

- Example:
  - For “button”, ***text*** is a default property. But user can decide to use ***name*** instead. The RFT recorder will use the new property as the recognition property for any kind of button going forward

- Provides for more flexibility to users – improves robustness and reliability of scripts.

- In 7.0.0.2, customization done thru a configuration file. In 7.0.1, a UI is provided to make this easy

Object library UI in 7.0.1



Other key features in 7.0.1 release

- XML Logging
  - RFT provides already HTML & TPTP output logs today.
  - Now users can easily parse RFT logs
  - They can also extend to have the logs in the syntax they wish

- Image Comparator
  - Verification of images and text of images
  - Verification of controls that RFT doesn't support

Questions

Thank You

Neil Williams

nwilliams@uk.ibm.com