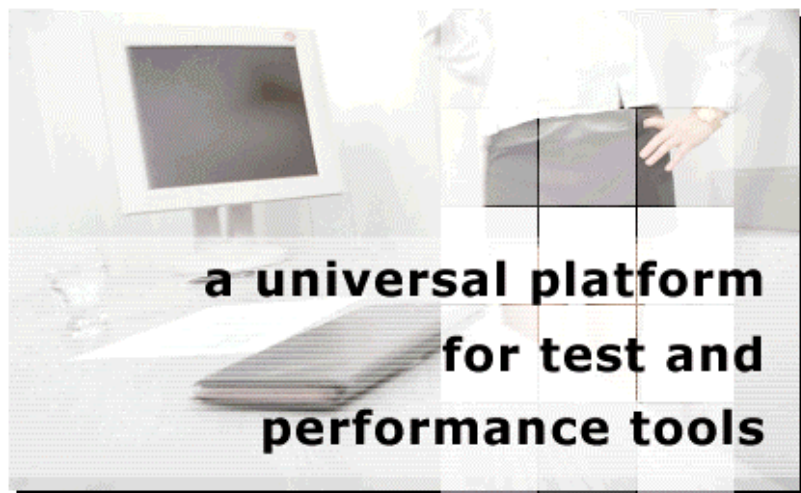


## Long Talk

# Testing Applications Using the Eclipse Test and Performance Tools Platform (TPTP)



**Paul Slauenwhite**  
IBM Rational Software

# Agenda

- Introduction
  - Speaker
  - Participants
- TPTP Project
  - Mandate
  - Participants
  - Composition
  - Architecture
- Agent Controller
  - Architecture
- Eclipse Modeling Framework
- TPTP EMF Data Models
  - Testing Profile
  - Behavioral
  - Execution History

# Agenda

- TPTP Test Tools
  - Test Perspective
  - Datapools
  - Execution Environment
  - Test Logs
  - Reports
  - Manual Testing
  - JUnit Testing
  - URL Testing
  - Automated GUI Testing
- Conclusion
  - Project Extensibility
  - Future work
  - Resources
- Questions, Answers and Discussion
- TPTP Sessions at EclipseCon 2006

# Introductions - Speaker

- Paul Slauenwhite
  - Software Developer
  - TPTP Project Committer
  - IBM Rational Software
  - Toronto Software Lab
  - [paules@ca.ibm.com](mailto:paules@ca.ibm.com)



Source: Google Maps  
(<http://maps.google.com/maps?f=q&hl=en&q=8200+warden+avenue,+markham&ll=43.849287,-90.338418&spn=0.002809,0.007381&t=k>)

## Introductions – Participants

- What is your level of experience with Eclipse?
  - a) First time user.
  - b) Casual user.
  - c) Frequent user.
  - d) Contributor/commmitter.
  
- What is your familiarity with TPTP?
  - a) What is TPTP?
  - b) Casual user.
  - c) Frequent user.
  - d) Trying to find a wireless signal to check my mail.

# TPTP Project - Mandate

- Started in 2002 as an Eclipse tools subproject named Hyades.
- Promoted in 2004 to an Eclipse project and renamed as the Test and Performance Tools Platform (TPTP) project.
- Open-source platform providing an extensible framework for Automated Software Quality (ASQ) tools including reference implementations for testing, tracing, tuning and monitoring software systems running on a range of target platforms.
- Goals:
  - Platform of choice for test, performance, and monitoring tools
  - Exemplary tools
  - Enables value-added third-party tooling through extensibility and high-quality APIs
- Principles:
  - Extension of the Eclipse Value Proposition
  - Vendor Ecosystem
  - Vendor Neutrality
  - Standards-Based Innovation
  - Agile Development
  - Inclusiveness & Diversity
- Mission:

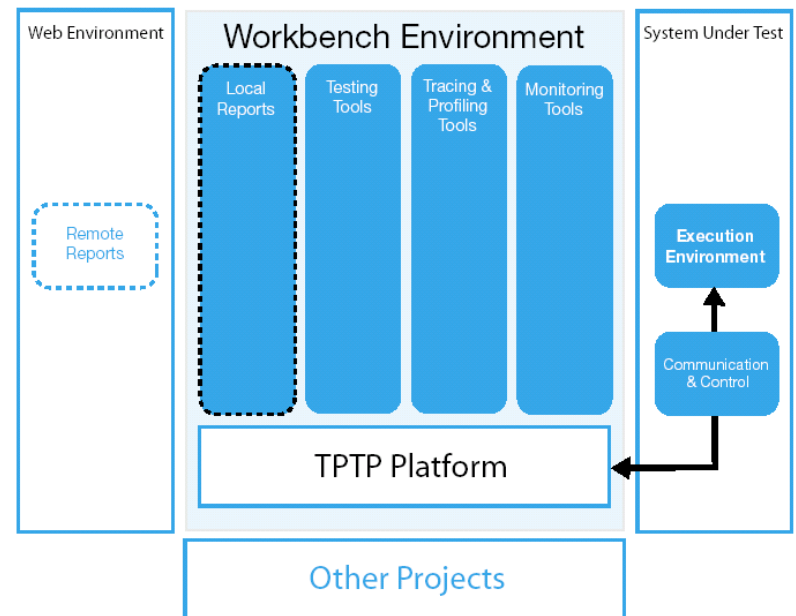
“to build **a generic, extensible, standards-based tool platform** upon which software developers can create specialized, differentiated, and interoperable offerings for world class **test and performance tools.**”

# TPTP Project - Participants

- Eclipse Strategic Developers:
  - Computer Associates
  - Compuware
  - IBM
  - Intel
  - Scapa Technologies
- Eclipse Strategic Consumers:
  - SAP
- Eclipse Add-in Providers:
  - FOKUS
  - OC Systems

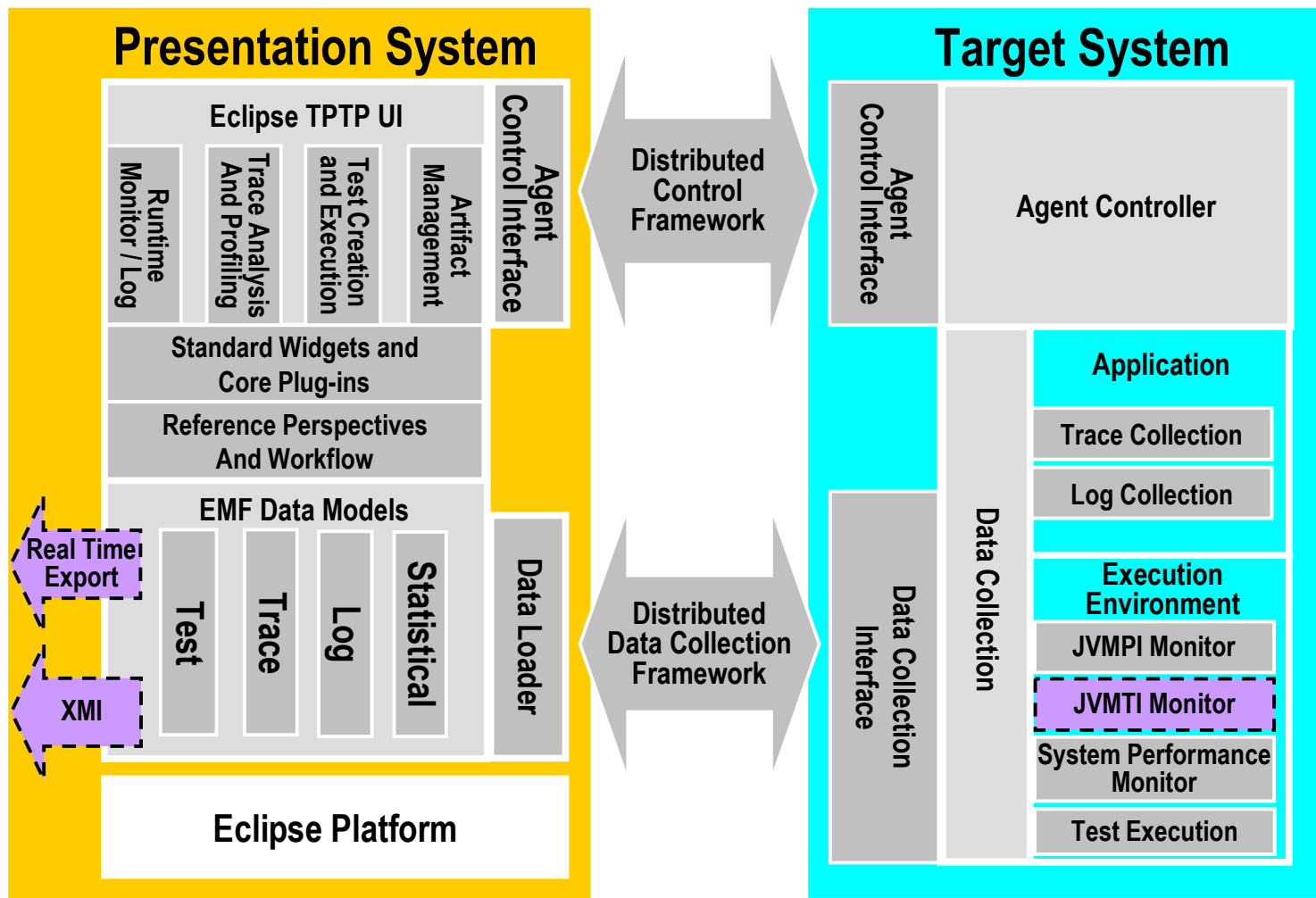
# TPTP Project - Composition

- TPTP is composed of 4 projects:
  - Platform
  - Test
  - Trace and Profiling
  - Monitoring
- Each project is divided into technology domains:
  - UI
  - Analysis
  - EMF Model
  - Collection Control
  - Communications
  - Execution
  - Agents/Events
  - Doc
- Each technology domain divided into components, each with an assigned lead committer.





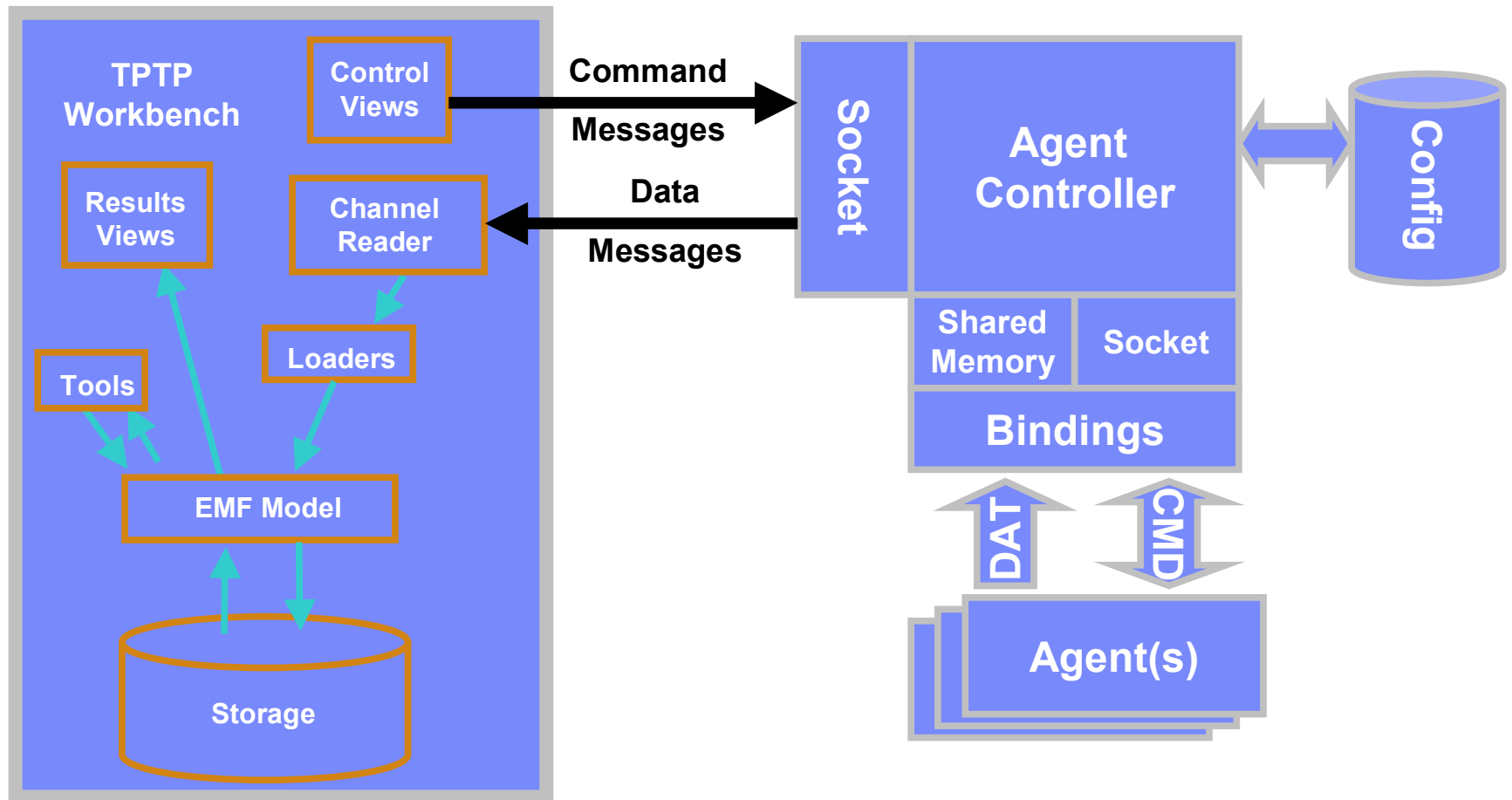
# TPTP Project - Architecture



# Agent Controller

- Standalone component independent of the TPTP workbench.
- Flexible service/daemon that resides on each local or remote deployment host.
- Used to launch new processes and attach to agents that coexist within processes.
- Provides extensible agent architecture for control capabilities (e.g. launch, pause and attach) and native data collection by monitoring clients (e.g. TPTP workbench).

# Agent Controller – Architecture



# Agent Controller – Architecture

- Supported platforms include:
  - AIX® (PPC)
  - HP-UX (RISC)
  - Linux™ (IA32)
  - Linux™ (PPC64)
  - Linux™ (zSeries)
  - OS/400® (iSeries)
  - Solaris (SPARC)
  - Windows™ (IA32)
  - Z/OS® (zSeries)

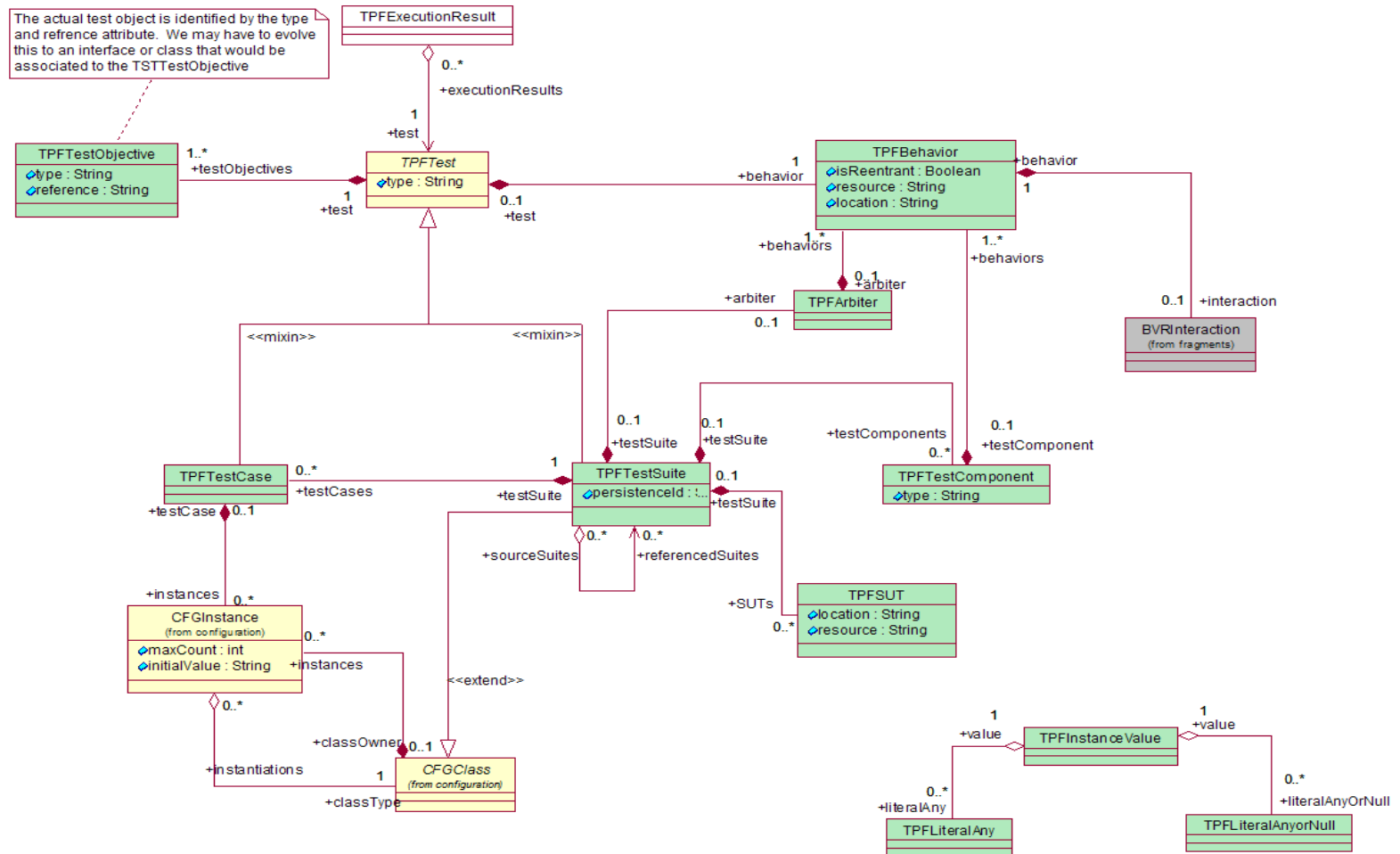
# Eclipse Modeling Framework (EMF)

- Framework and code generation facility for designing and implementing applications based on a structured data model.
- Data models are used to describe data objects, attributes and relationships.
- Data models may be designed using annotated Java, XML, or UML class diagrams and converted to an EMF model specification defined in XML Metadata Interchange (XMI).
- The XMI model specification is used to generate the Java implementation classes for the data model, a set of adapter Java classes that enable viewing and command-based editing of the data model, and a basic editor UI.
- The generated Java implementation classes are used to instantiate, load and persist in-memory data models.
- A resource persistence layer is provided to permit file system abstractions, such as flat files or a relational database.
- Provides a framework for the mechanical creation of data models that are the foundation for interoperability with other EMF-based applications.

## TPTP EMF Data Models

- EMF data models are used in TPTP as an integration point.
- TPTP User Interface and Execution Environment are build around the TPTP EMF data models.
- TPTP contains four core EMF data models:
  - Trace
  - Log
  - Statistical
  - Test
    - Testing Profile
    - Behavioural
    - Execution History
- Data models are populated by model loaders consuming structured XML fragments originating from a variety of sources.

# TPTP Test Models



# TPTP Test Models – Testing Profile

- Reference implementation of the UML2 Test Profile's standalone MetaObject Facility (MOF) model.
  - Part of the finalization process in the OMG's acceptance of the UML2 Test Profile submission.
- Definition model for the creation, definition and management of test artifacts.
- Test artifacts include:
  - Test suites.
  - Test cases.
  - Datapools.



# TPTP Test Models – Behavioural

- Implementation of the UML2 Interactions meta model.
- Used for defining the behaviour of test suites including:
  - Loops.
  - Invocations:
    - Test cases within the test suite defining the behaviour.
    - Other test suites and/or test cases (hierarchical).
  - Synchronization.
- Façade provided:
  - Semantically familiar programming model (e.g. loops instead of combined interaction fragments).
  - Interoperability among tests from different test tool vendors (multiple ways of representing the same transaction).
  - Access to behavior that is not modeled (via another implementation of the same façade interface).

# TPTP Test Models – Execution History

- Definition model for the creation, definition, management and persisting of test executions over time.
- Test executions include:
  - Tests to be executed.
  - Deployments.
  - Locations.
  - Verdicts.
  - Attachments.
  - Messages and console output from the test execution.
- A collection of test execution traces and results, commonly referred to as a *test log*.
- Extensible with custom attributes.

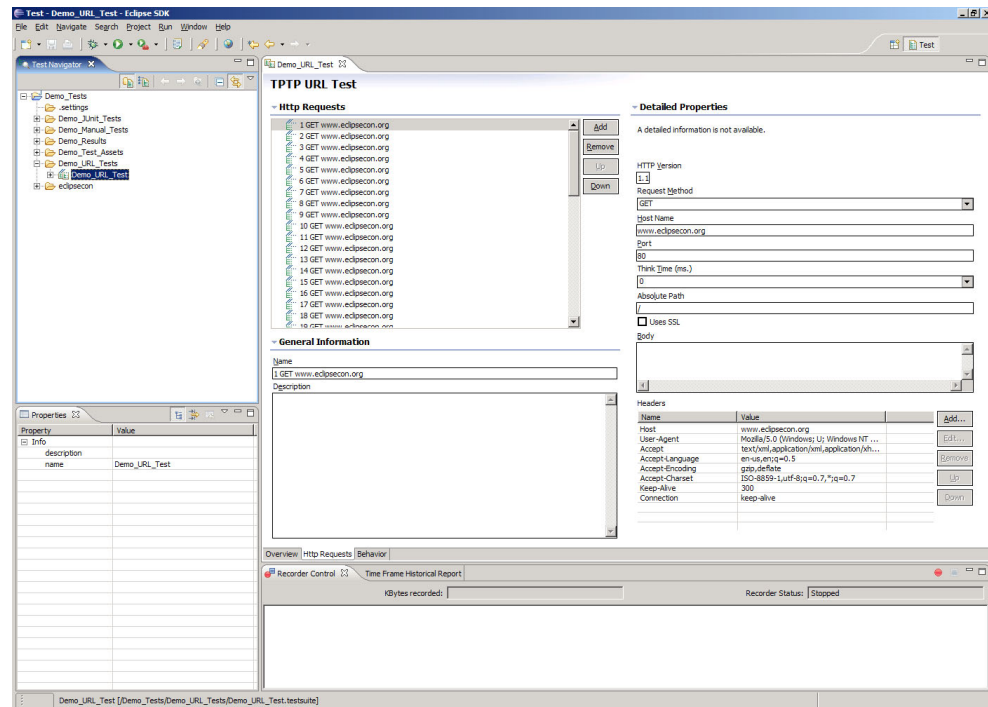
# TPTP Test Tools

- Common framework for testing tools thereby integrating disparate test types and execution environments.
- Extends the TPTP Platform:
  - Common perspectives and views for interacting with target systems and resources.
  - Reference navigators, viewers, editors and wizards through extension points.
  - Standard EMF data model, query framework and assets repository.
  - Common data collection and execution framework on local and remote targets.
- Includes several test tool reference implementations:
  - Manual testing.
  - JUnit and JUnit Plug-in testing.
  - URL testing.
  - Automated GUI Recording/Playback

# TPTP Test Tools – Test Perspective

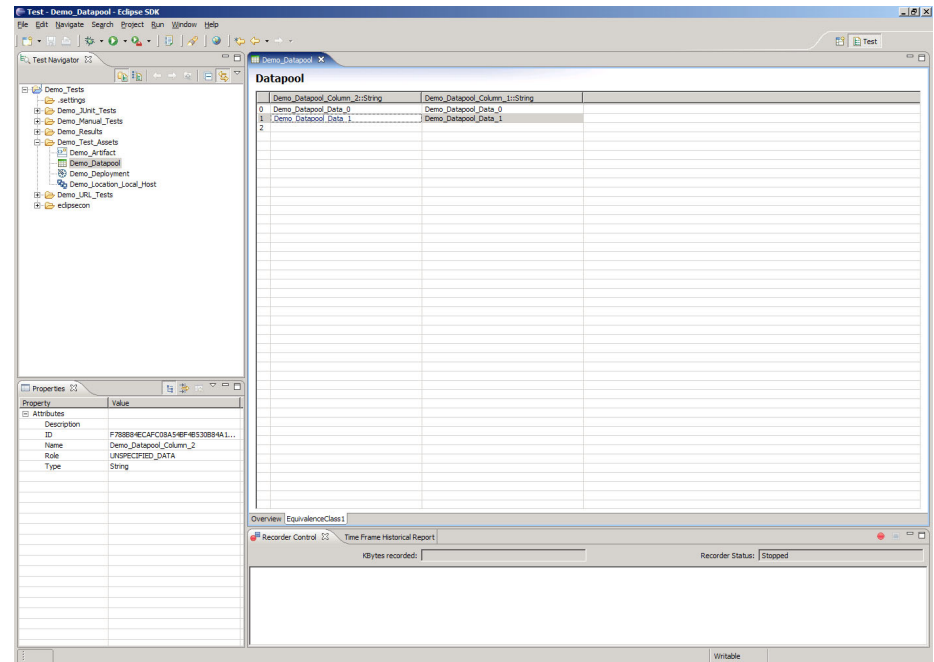
- Set of navigators, viewers, editors and wizards for creating, viewing, editing and navigating tests and test assets including:

- Artifacts:
  - Test suites.
  - Test cases.
  - Datapools.
- Recordings.
- Deployments.
- Locations.
- Execution histories.
- Reports.



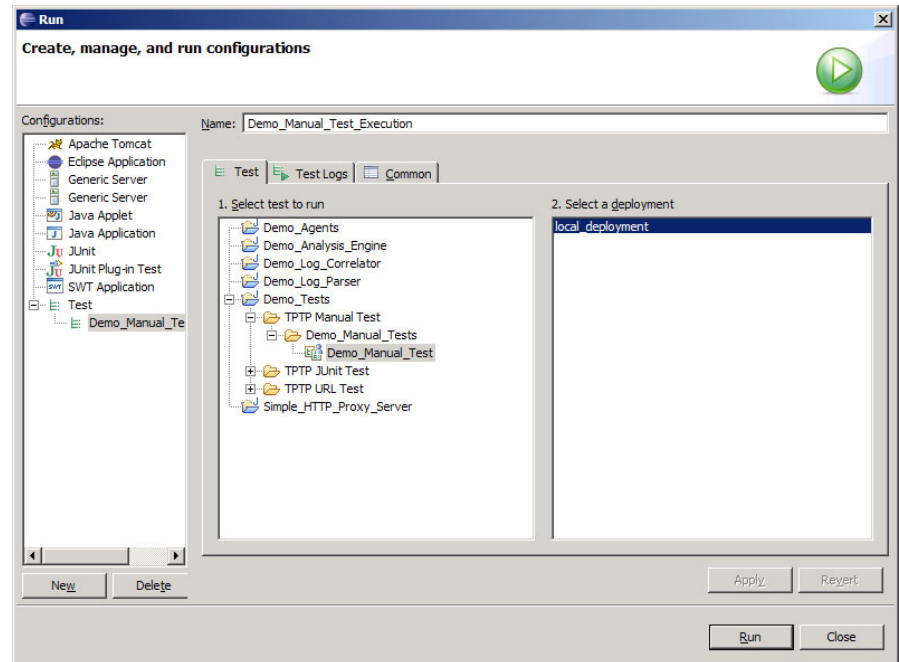
# TPTP Test Tools – Datapools

- Provides variable data to tests during execution:
  - Equivalence Class:** Subset of data that produces the equivalent pass/fail result.
  - Variable:** Named and typed data (column) within an equivalence class.
  - Record:** Slice of data (row) of one or more variables.
- Datapool variables are associated with test input and output for substitution of equivalence classes during execution.
- Support for importing large volumes of two-dimensional data from CSV files.



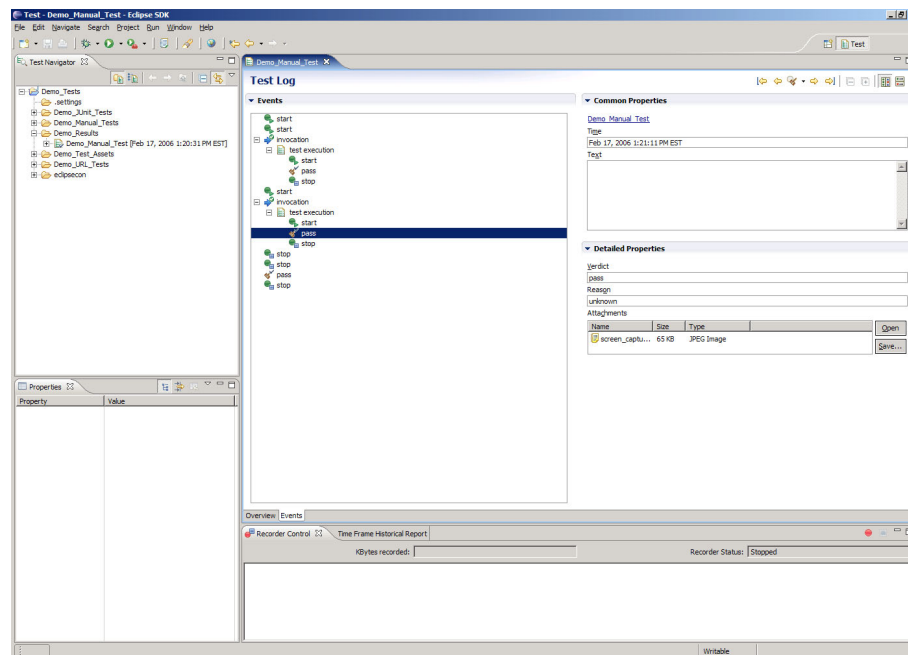
# TPTP Test Tools – Test Execution

- Tests are *typically* executed from the TPTP workbench using a launch configuration.
- Execution includes:
  - Test assets deployed to the target environment through the Test Execution Harness.
  - Typed test runner invoked to execute the test.
  - Execution events routed through the test agent back to the TPTP workbench.
  - Execution events loaded into the execution history model.
- An automatable test execution service is provided for launching tests programmatically via scripts (e.g. shell and ANT) and external applications (e.g. Java).



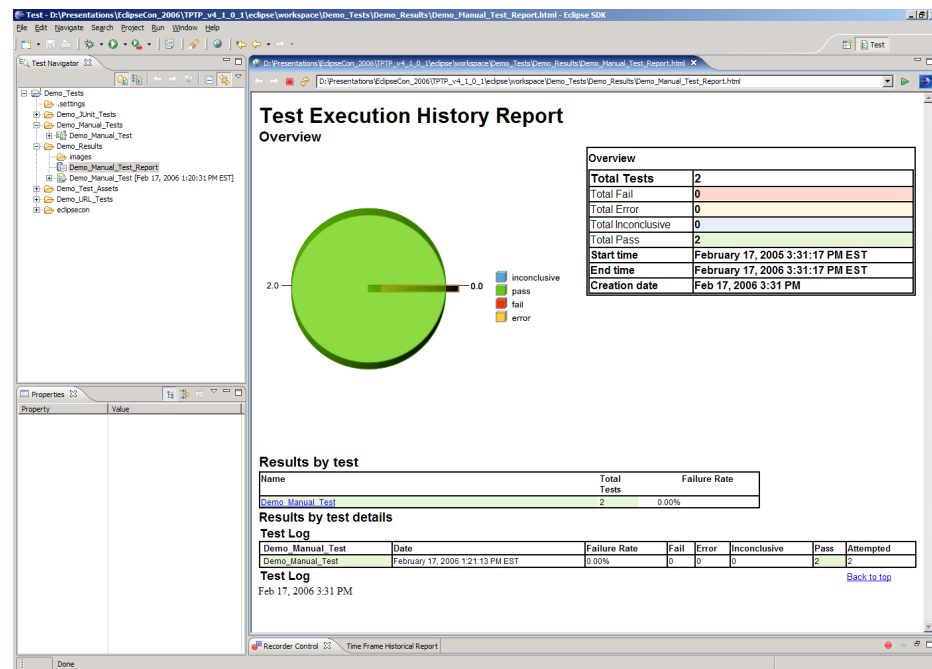
# TPTP Test Tools – Test Logs

- Executed tests produce test logs for persistence, reporting and historical analysis.
- Test logs include:
  - Tests to be executed.
  - Deployments.
  - Locations.
  - Verdicts.
  - Attachments.
  - Messages and console output from the test execution.
- Viewer to summarize, view, navigate and filter test logs.



# TPTP Test Tools –Reports

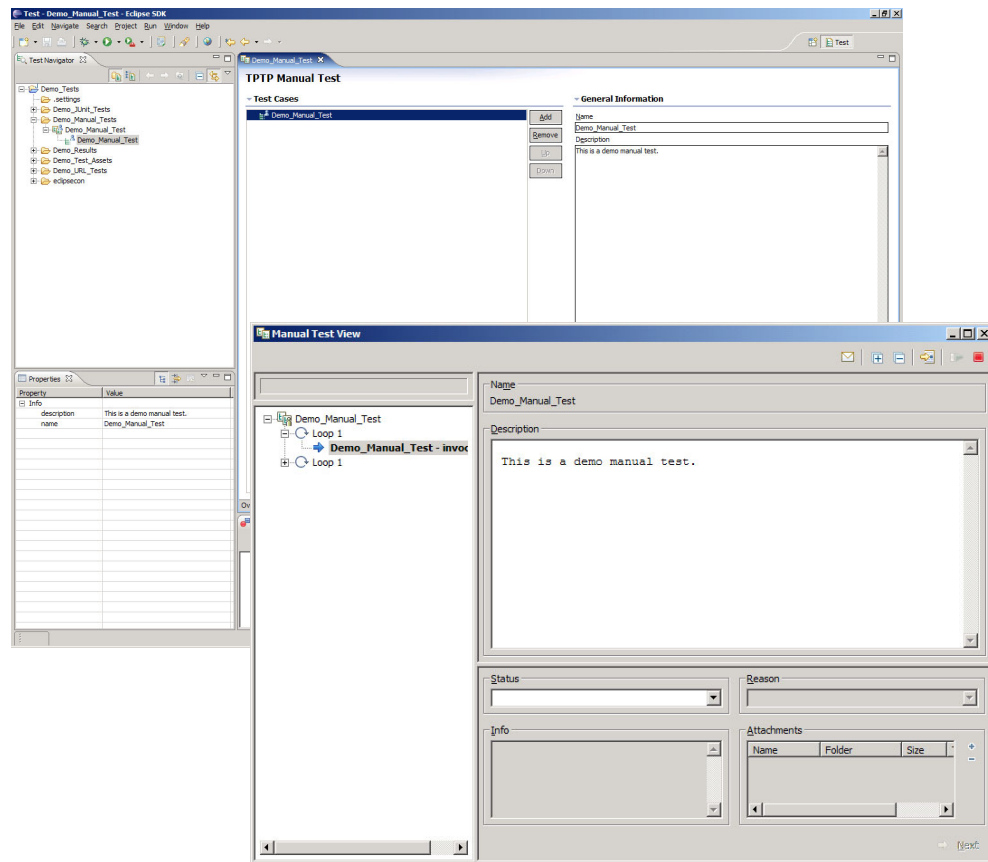
- Aggregates and summarizes numerous test executions over a period of time (report window).
- Integrated with BIRT for defining custom report templates.
- Extension points and preferences to provide custom report generators to create custom content.
- Existing report generators include:
  - HTML Test Pass (BIRT)
  - HTTP Page Hit Rate
  - HTTP Page Response Time
  - Test Pass
  - Time Frame Historic
  - XML Test





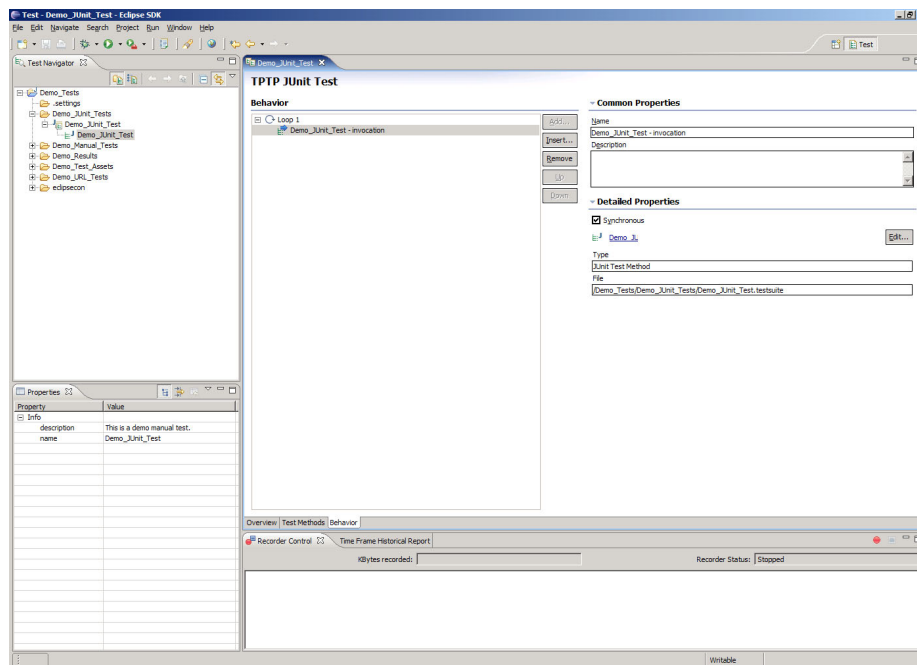
# TPTP Test Tools - Manual Testing

- Manual testing of applications by human testers based on textual test descriptions (plain text or HTML).
- Editor for creating manual test suites and defining tests and their behavior.
- Manual test runner for local and remote execution.
- Manual Test View for executing a manual test on the target environment by allowing the tester to step through the manual test and capture messages and results:
  - Verdicts.
  - Reasons.
  - Information.
  - Attachments.



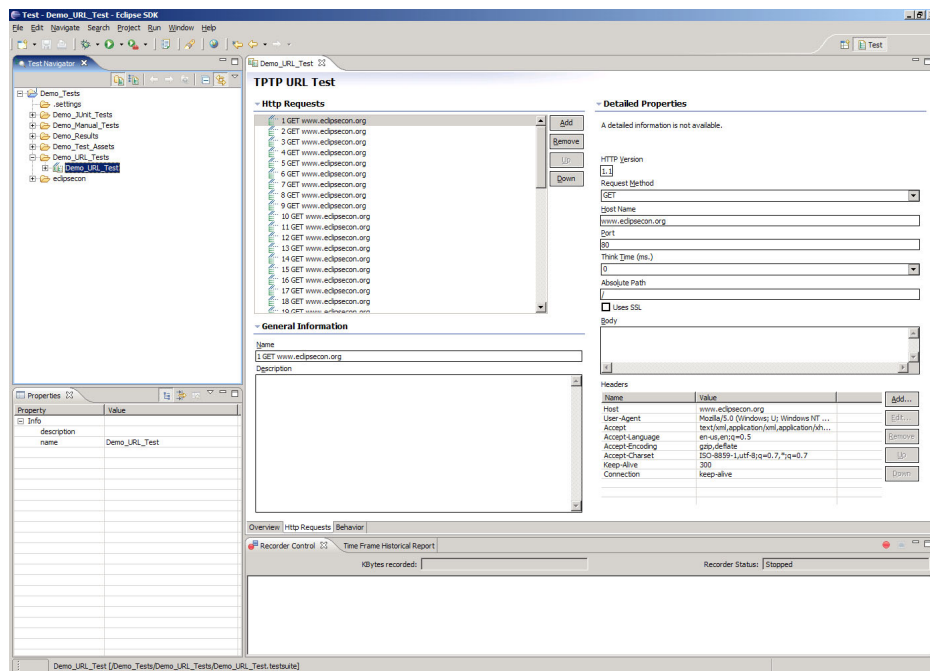
# TPTP Test Tools - JUnit and JUnit Plug-in Testing

- Automated unit testing of applications and plug-ins based on JUnit test classes.
- Editor for creating JUnit and JUnit plug-in test suites and defining metadata, tests and their behavior.
- JUnit test classes generated and synchronized automatically from the test suite while preserving user modifications.
- Wizard for creating TPTP JUnit and JUnit plug-in tests from existing JUnit and JUnit plug-in test classes without source modification.
- JUnit test runner for local and remote execution.



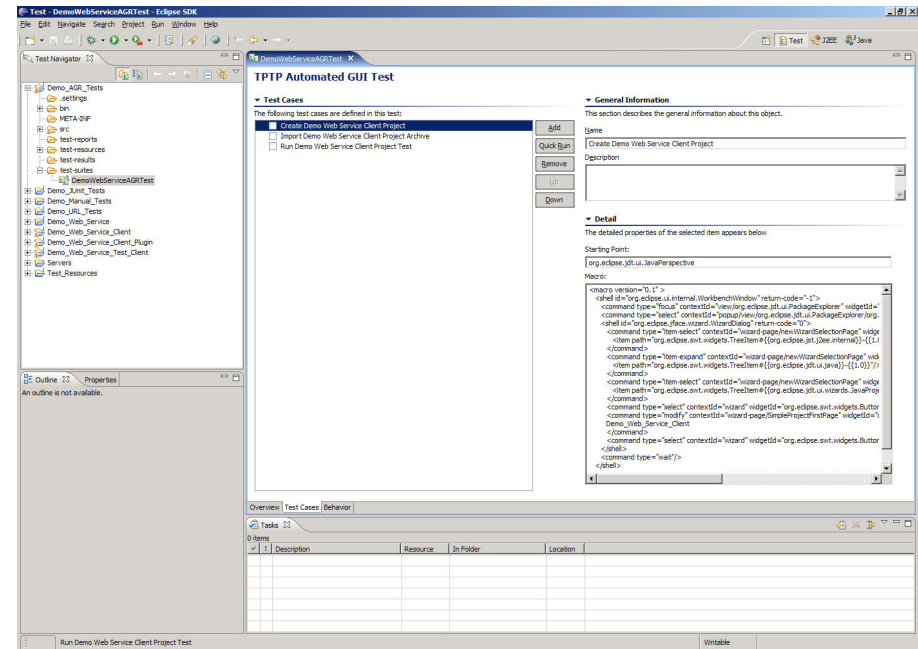
# TPTP Test Tools - URL Testing

- Automated HTTP performance testing of Web applications and plug-ins based on JUnit test classes.
- Editor for creating URL test suites and defining tests (HTTP requests) and their behavior.
- HTTP proxy recorder to intercept and record HTTP requests from user interactions with Web applications to create an URL test suite.
- URL test runner for local and remote execution.



# TPTP Test Tools – Automated GUI Testing

- Automated GUI Recorder (AGR) for testing Eclipse-based UIs.
- Editor for creating automated GUI test suites and defining tests and their behavior.
- Positional and object-based GUI recorder to intercept and record user interactions with UI widgets in the workbench to create an automated GUI test suite with datapools and verification points.
- Automated GUI test runner for local and remote execution.



## Conclusion

- TPTP Test Tools provide a centralized, open-source, flexible and extensible framework for testing tools.
- Framework provides tools for creating and managing tests, deployments, datapools, execution histories and reports.
- Reference implementations for manual, JUnit, URL and automated GUI testing of applications.

## Conclusion - Project Extensibility

- Extensible architecture to allow users to define vendor and product specific:
  - Test Creation Wizard
    - Extension point for defining wizard to create specific types of tests.
  - Test Editor
    - Extension point provided for associating editor with specific type of test.
  - Test Recorder
    - Extension point provided for registering a custom test recorder for a type of test.
  - Code Generator
    - Extension point for registering a custom code generator for a type of test.
  - Analyze Results
    - Extension point to open Log views used to analyze results of test run.
  - Publish Reports
    - Extension point for custom report types with custom content.
- Several existing industry test products built on TPTP:
  - IBM® Rational® Functional Tester, Manual Tester, Application Developer and Software Architect (<http://www-306.ibm.com/software/rational/>)
  - Scapa Test and Performance Platform ([http://www.scapatech.com/products\\_services/intro.html](http://www.scapatech.com/products_services/intro.html))

# Conclusion - Future Work

- 4.2 Planned Work
  - Increase platform support for the Agent Controller including 64 bit architectures (Linux, Windows, Z/OS).
  - Improved documentation, defect fixes and performance improvements.
  - New Agent Controller implementation supporting dynamic deployment of agents, agent metadata, agent discoverability, communication across firewalls, optional encryption, multiple client connections and pluggable communication protocols.
- Future Work
  - Generic Java API recorder.
  - Web Service test type, wizard, editor, execution and recorder.
  - Support for mixed test types.

## Conclusion - Resources

- Eclipse Modeling Framework (EMF), <http://www.eclipse.org/emf>.
- TPTP Documentation (Help >> Help Contents).
- TPTP Examples (File >> New >> Example...).
- TPTP Project, <http://www.eclipse.org/tptp>.
  - Project architecture, organization and plans.
  - Downloads.
  - Defects.
  - CVS.
  - Developer Resources.
  - Documentation:
    - Common Base Event v1.0.1 specification, schema and API documentation.
  - Tutorials.
  - Mailing Lists.
  - Newsgroup.



## Questions, Answers and Discussion

- Is there a *real* requirement in industry for open-source test tools?
- What are some common testing activities and obstacles?
- Are there viable proprietary extensions to the TPTP test tools?
- Possible suggestions for enhancements for the existing TPTP test tools.

# TPTP Sessions at EclipseCon 2006

- Test and Performance (TPTP) Track:
  - <http://www.eclipsecon.org/2006/Sub.do?id=521>
- Tutorials:
  - **(30) Profile your Java application using the Eclipse Test and Performance Tools Platform (TPTP)**
    - Eugene Chan and Guru Nagarajan, Monday, 8:30, | Cypress
  - **(61) Achieving Continuous Integration with the Eclipse Test and Performance Tools Platform**
    - Joe Toomey and Scott E. Schneider , Monday, 13:30 | Bayshore
- Long Talks:
  - **Honey, I shrunk Eclipse!**
    - Chris Laffra and Wing Hong Ho, Tuesday, 15:15 | Ballroom CD
  - **Probing Java application behavior with Probekit**
    - Guru Nagarajan and Navid Mehregani, Wednesday, 9:45 | Theater
  - **Testing Applications Using the Eclipse Test and Performance Tools Platform (TPTP)**
    - Paul Slauenwhite, Thursday, 9:45 | Ballroom EF
  - **Static Analysis Using the Eclipse Test and Performance Tools Platform (TPTP)**
    - Steven Gutz and Orlando Marquez, Thursday, 15:15 | Room 203&204
  - **Web Services component level tracing with TPTP**
    - Valentina Popescu, Thursday, 15:15 | Ballroom EF
- Short Talks:
  - **Extending Eclipse TPTP using the Generic Recorder Framework**
    - Mark D. Dunn and Jeff Nevicosi , Wednesday, 16:24 | Ballroom ABGH
  - **Leveraging the TPTP Data Collection Framework**
    - Qiyang Li , Thursday, 13:00 | Room 203&204

# Legal Notice

- AIX, OS/400, IBM, Rational, and Z/OS are registered trademarks of International Business Corp. in the United States and other countries.
- Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Linux is a trademark of Linus Torvalds in the United States, other countries, or both.
- Other company, product, or service names may be trademarks or service marks of others.