



ORACLE®

Application Testing Suite: A fully Java-based software testing platform for testing Oracle E-Business Suite and other web applications

Murali Iyengar, Principal Sales Consultant, Oracle



Agenda

- What: Introduction to Oracle Application Testing Suite
- How: a Worked Example
- Testing EBS & Web Services Applications
- Conclusion



- **What: Introduction to Oracle Application Testing Suite**
- How: a Worked Example
- Testing EBS & Web Services Applications
- Conclusion

Oracle Application Quality Management

High quality testing for all tiers of your application stack

Testing Application Changes

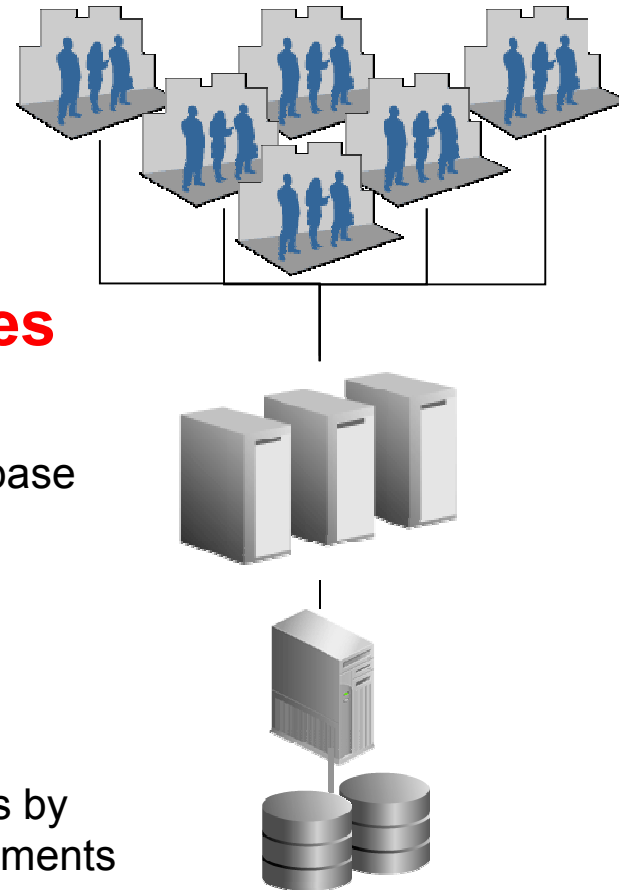
- *Application Testing Suite*
 - Ensure quality and performance with end-to-end application testing

Testing Infrastructure Changes

- *Real Application Testing*
 - Designed and optimized for testing database tier infrastructure changes

Test Data Management

- *Data Masking Pack*
 - Achieve security & compliance objectives by obfuscating sensitive data in test environments
 - Enables secure, production-scale testing



Application & Mid-tiers

Infrastructure & DB-tiers

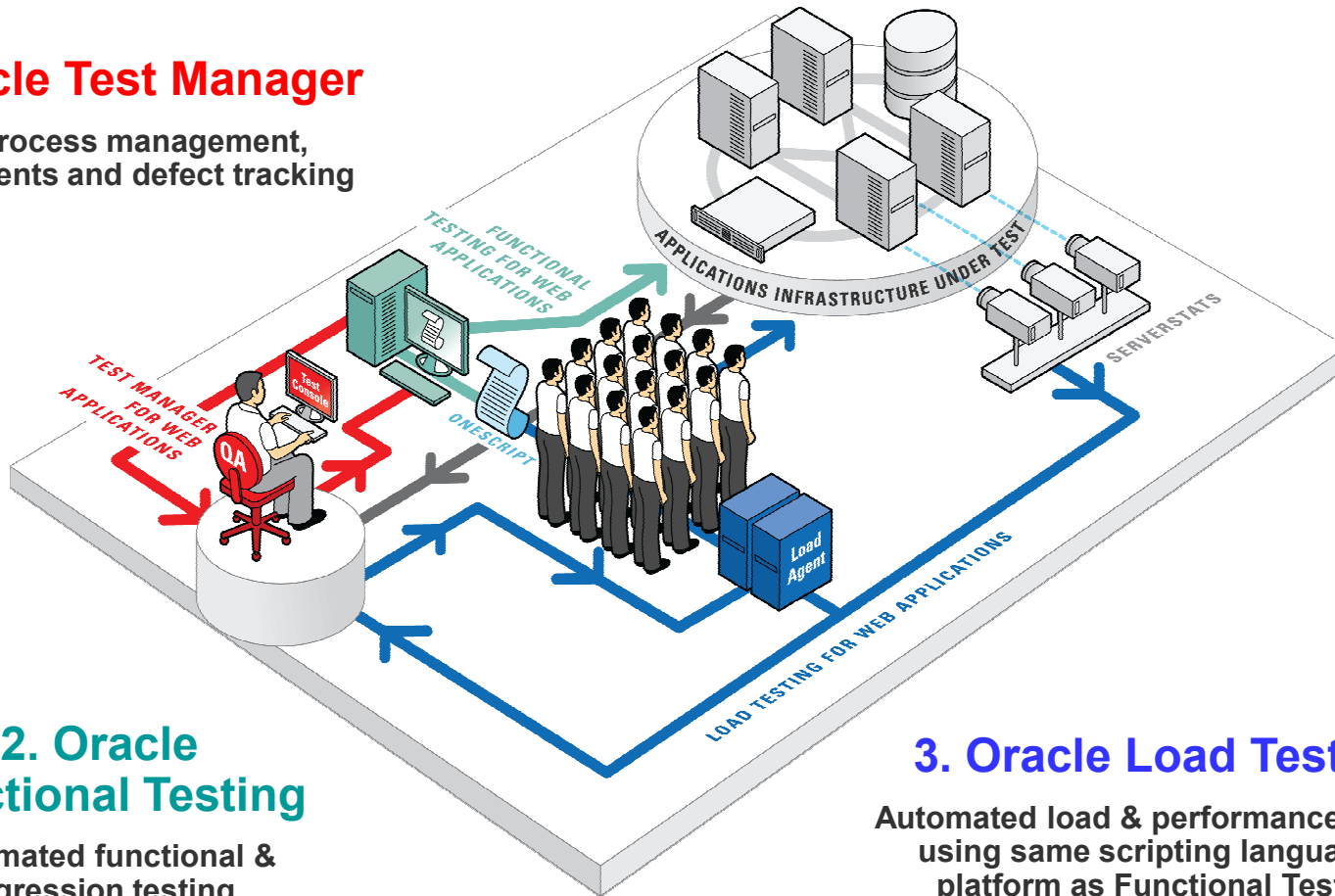
ORACLE®

Oracle Application Testing Suite

A powerful, integrated testing solution for ensuring application quality, performance and reliability

1. Oracle Test Manager

Test process management, requirements and defect tracking



2. Oracle Functional Testing

Automated functional & regression testing

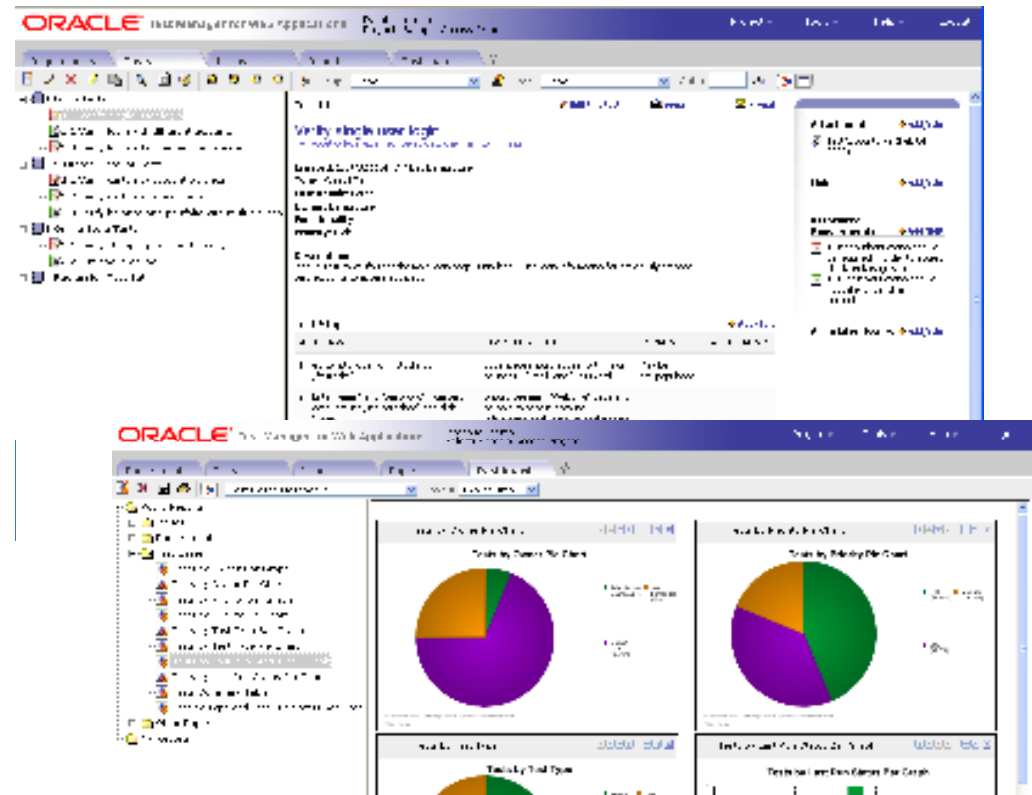
3. Oracle Load Testing

Automated load & performance testing using same scripting language & platform as Functional Testing

Oracle Test Manager:

Puts Test Process Management at every Developer's fingertips

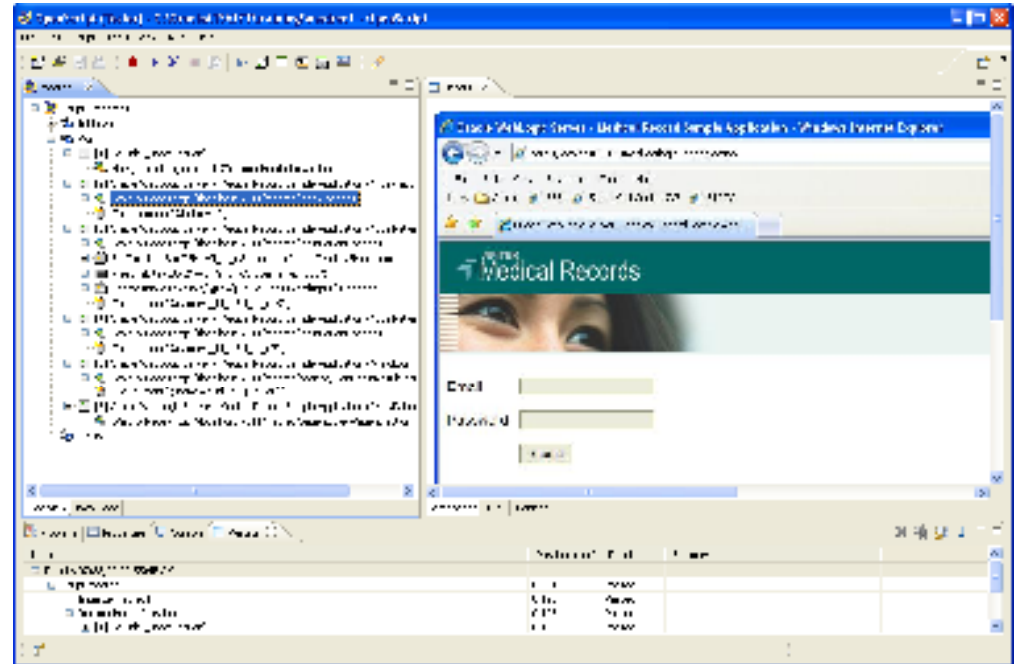
- Manage test process from a central, collaborative console
- Comprehensive traceability from test requirements, to test cases, to bugs & issues
- Document both manual & automated test cases and execute from your test plan
- Create reports to provide visibility into the test process
- Fully customizable to fit your development methodology
- Scalable enterprise architecture, yet easy to deploy & manage over web
- Includes Oracle DB & WLS



Oracle Functional Testing:

Automated Functional & Regression Testing

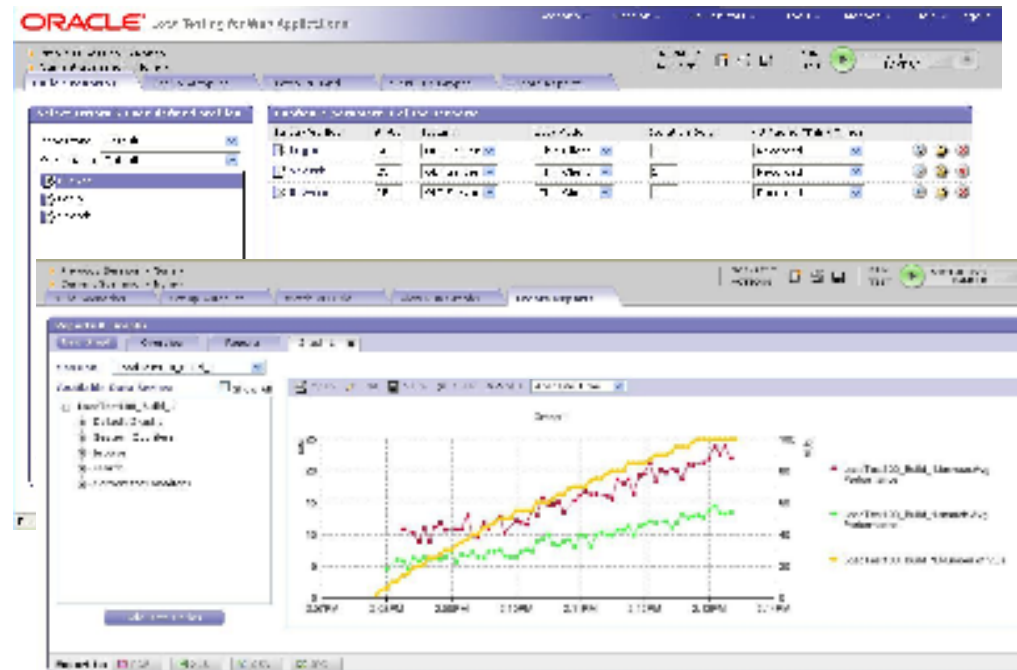
- Automates functional & regression testing - for Web, EBS, ADF, Siebel, SOA, Forms, packaged & custom applications
- OpenScript integrated platform for functional and load testing
- Intuitive visual scripting interface simplifies test process
- Powerful Java IDE provides superior scripting extensibility
- Custom accelerators for EBS, Siebel, Web Services, ADF & Oracle Database for efficient, optimized testing



Oracle Load Testing:

Load and Performance Testing and Tuning from Application down to DB tiers

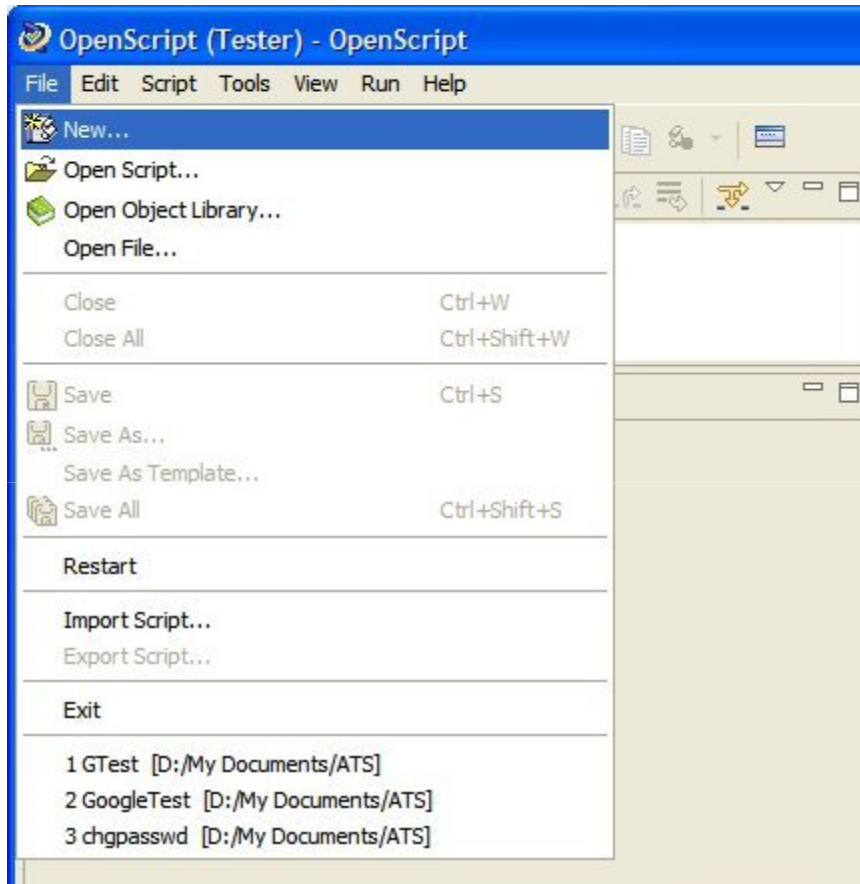
- Realistic load & performance testing for Web, EBS, Siebel, ADF, SOA, Forms, packaged & custom applications
- Scale to thousands of concurrent users to simulate peak production loads
- Delivers better accuracy by performing functional content validation under load
- Intuitive Web console promotes collaborative testing
- Server monitors ID & resolve performance bottlenecks
- Offers custom accelerators like Functional Testing has
- Includes Oracle DB & WLS





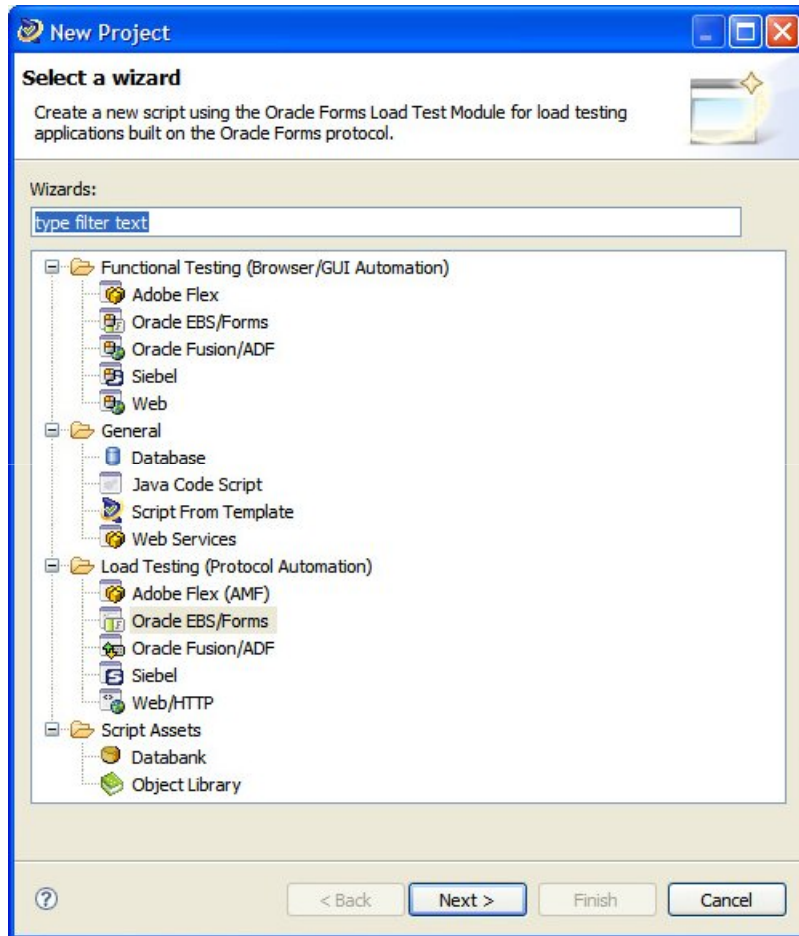
- What: Introduction to Oracle Application Testing Suite
- **How: a Worked Example**
- Testing EBS & Web Services Applications
- Conclusion

OpenScript



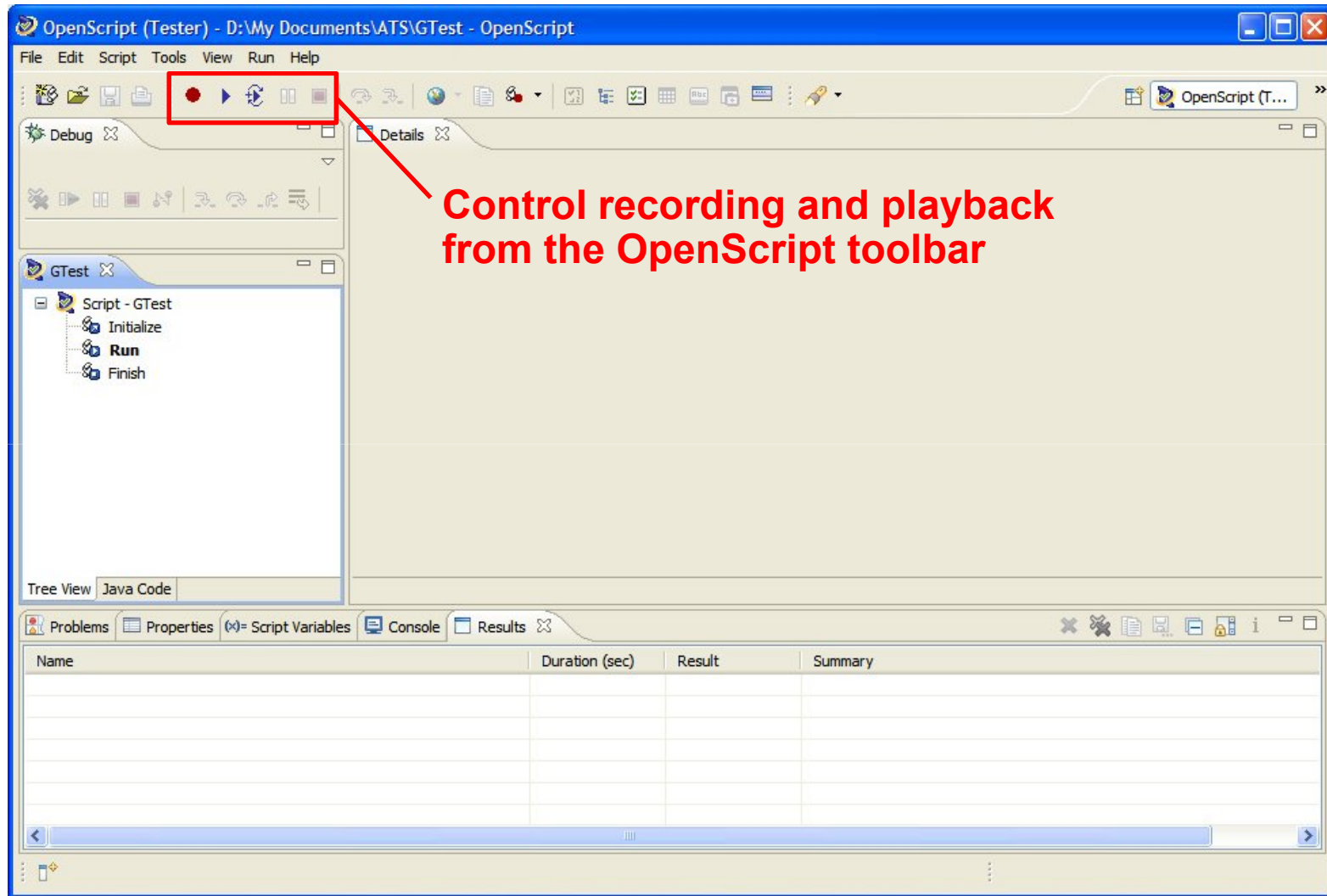
- Easy-to-use Eclipse GUI
- Create functional and load tests using intuitive “recording and playback” paradigm
- Already familiar to majority of Java developers
- Use as little or as much Java code customization as desired

Support Multiple Application Models

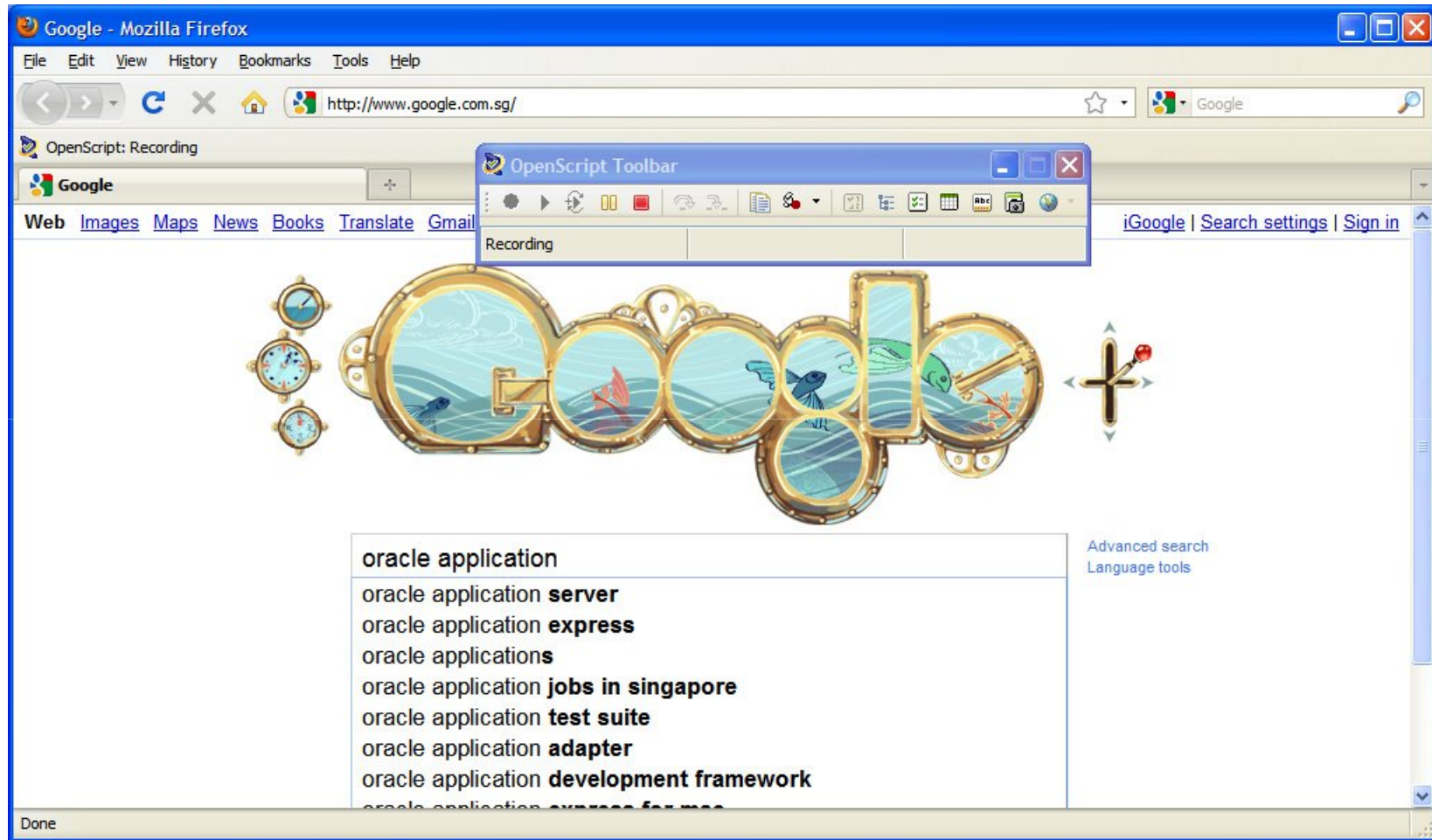


- Robust support for Adobe Flex, Oracle EBS/Forms, Oracle Fusion/ADF, Siebel HI, and Web applications
- Also provides database load testing, SOA/web services testing, and generic Java tests for non-web applications (e.g. sockets-based)

Familiar Eclipse GUI



Record or Play Back in Firefox or IE



Recorded Script and Results

The screenshot displays the OpenScript (Tester) application interface. The main window shows a recorded script for a Google search. The script steps are:

- Initialize
- Run
 - [1] Google (/www.google.com.sg/)
 - Navigate google.com
 - WaitForPage (http://www.google.com.sg/)
 - SetText textBox("@name='q'") oracle application testing

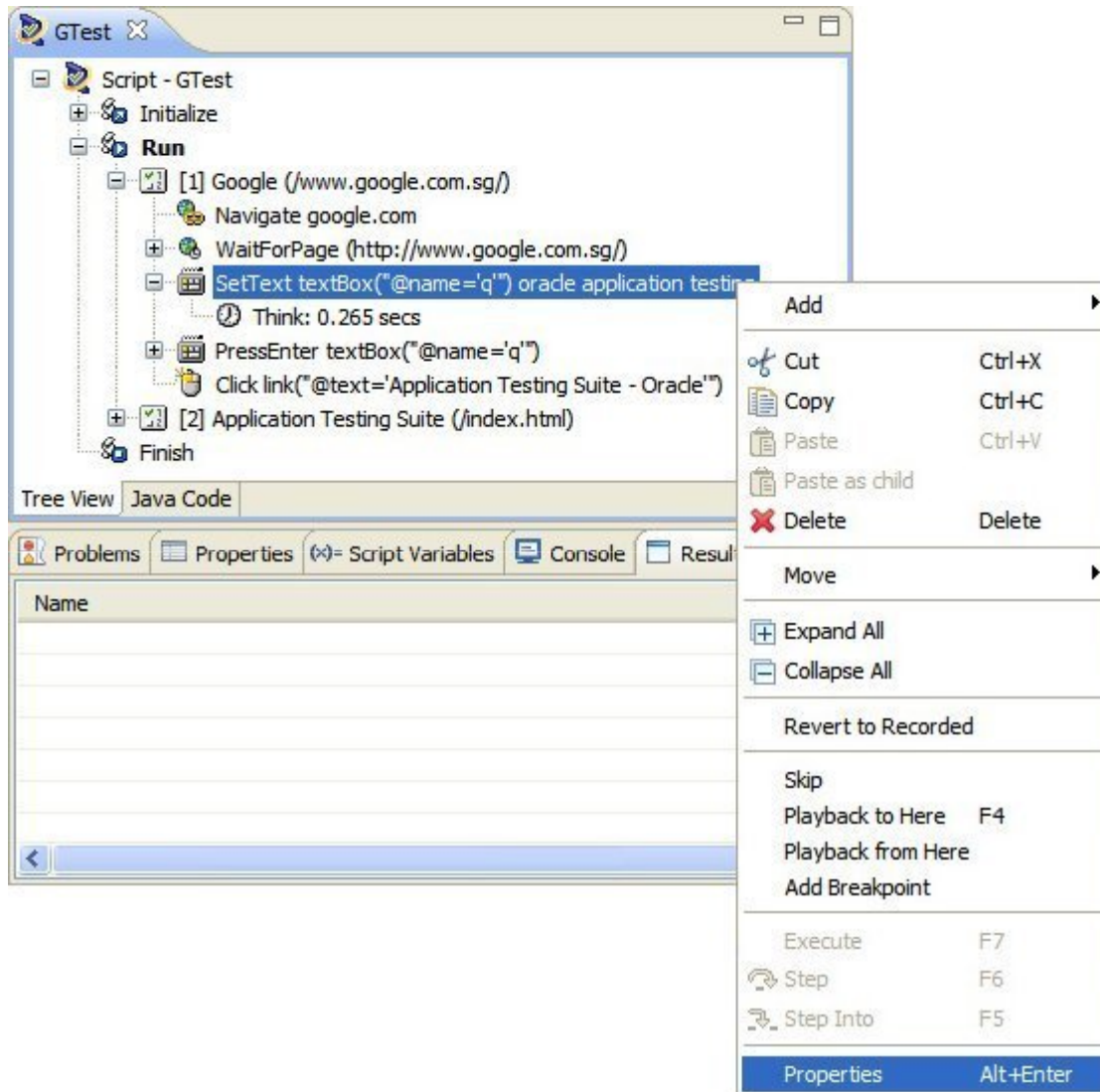
The results pane shows the page content, including the Google search interface and the text "oracle application testing" entered in the search box. The script execution time is noted as "Think: 0.265 secs".

The script code is shown in the following format:

```
}
web
    .textBox (
        8,
        "/web:window[@index='0' or @title='Google']/web
        .setText ("oracle application testing");
    {
        think(0.265);
    }
web
    .textBox (
```

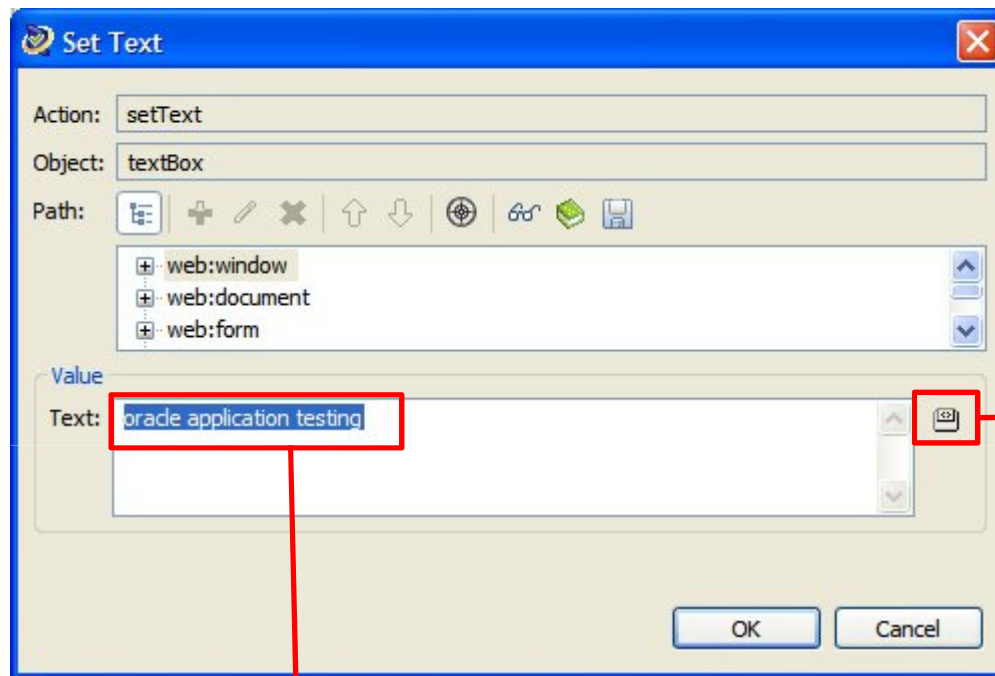
.. or view test script in its native Java format

Setting Parameters



- We can change the parameters, data sources, or values of any application element through the GUI
- Simply right-click on a parameter, then select its properties

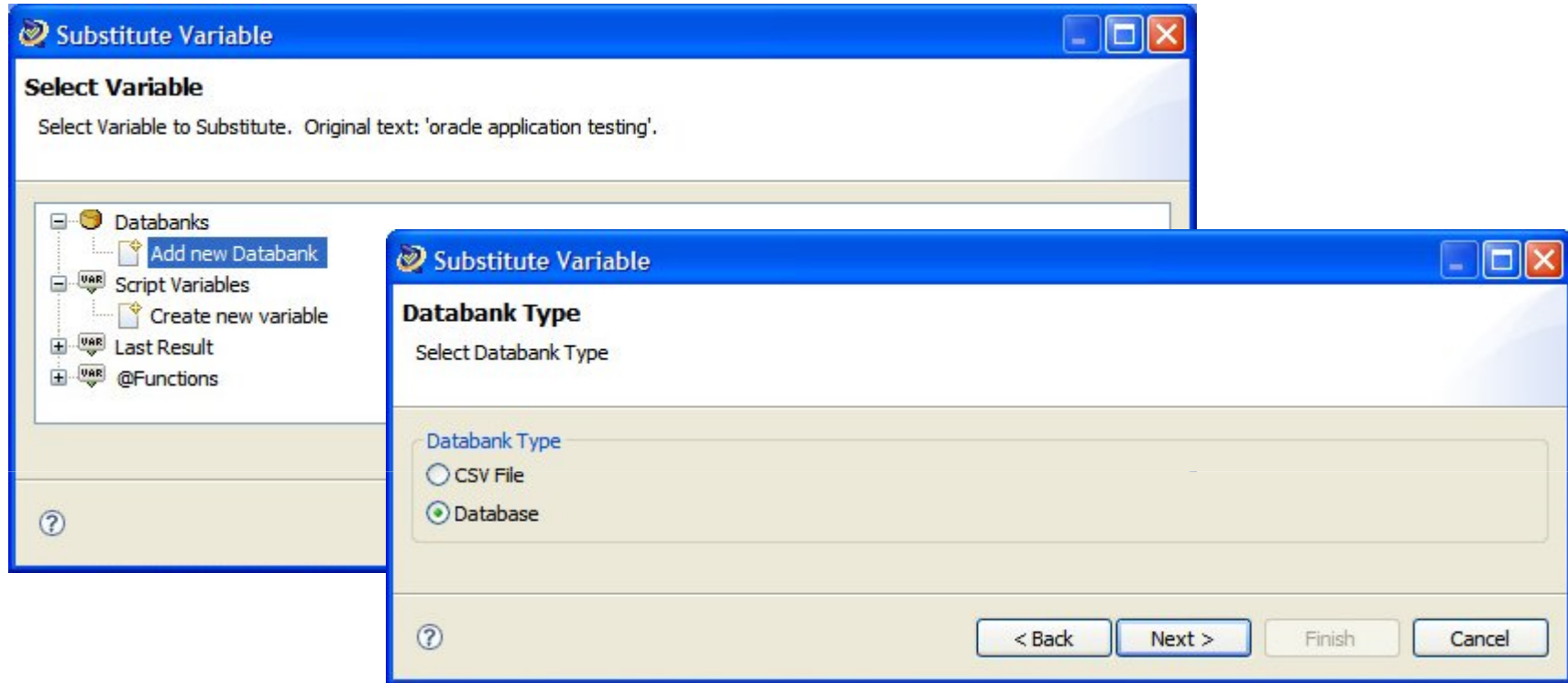
Setting Parameters continued..



.. or click on this button to substitute a variable or data source

change a property from its recorded value to something else..

Variable Substitution & Data Banking



- When substituting a value into a parameter, we can supply a Data Bank, a script variable, or functions
- A data bank's source may be a CSV file, or a JDBC data source

Data Banking

Substitute Variable
Add Databank
Configure the database.

Database Driver

- Oracle Thin (oracle.jdbc.OracleDriver)
 - Hostname:
 - Port:
 - SID
 - Service name
- ODBC (sun.jdbc.odbc.JdbcOdbcDriver)
 - Data source:

Url:

Username:

Password:

Query:

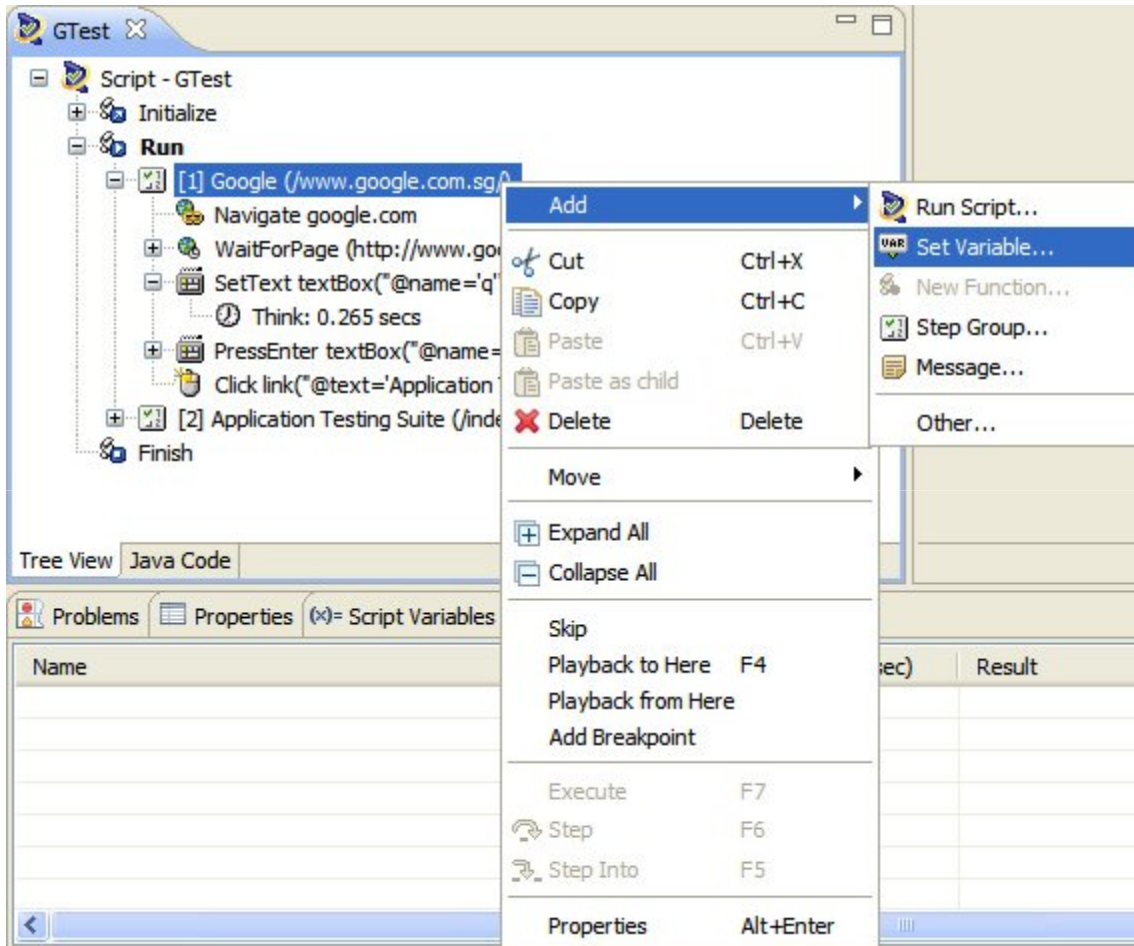
Alias:

Test

Url is required

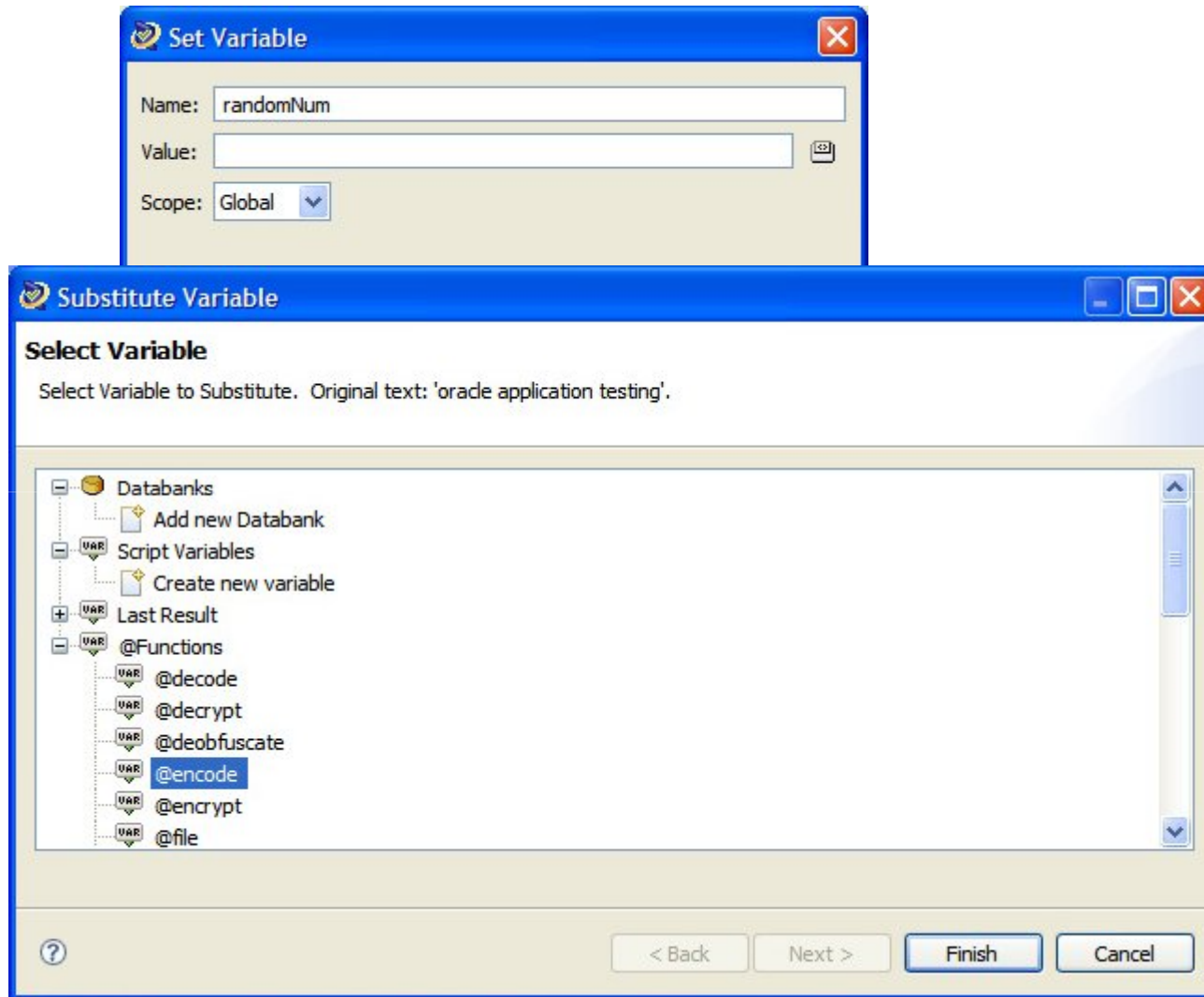
- Data banks can pull data from Oracle databases (via the JDBC type 4 thin driver)
- .. or any other database with the JDBC to ODBC shim

Script Variables



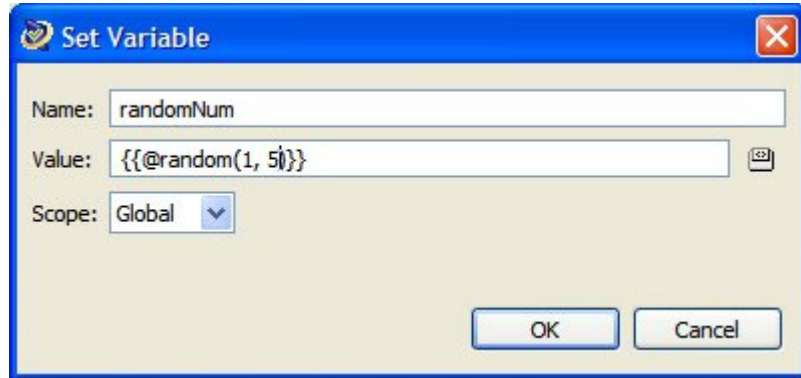
- Create script variables at any time by right clicking on the Tree View and selecting **Add** → **Set Variable**

Data Sources for Script Variables

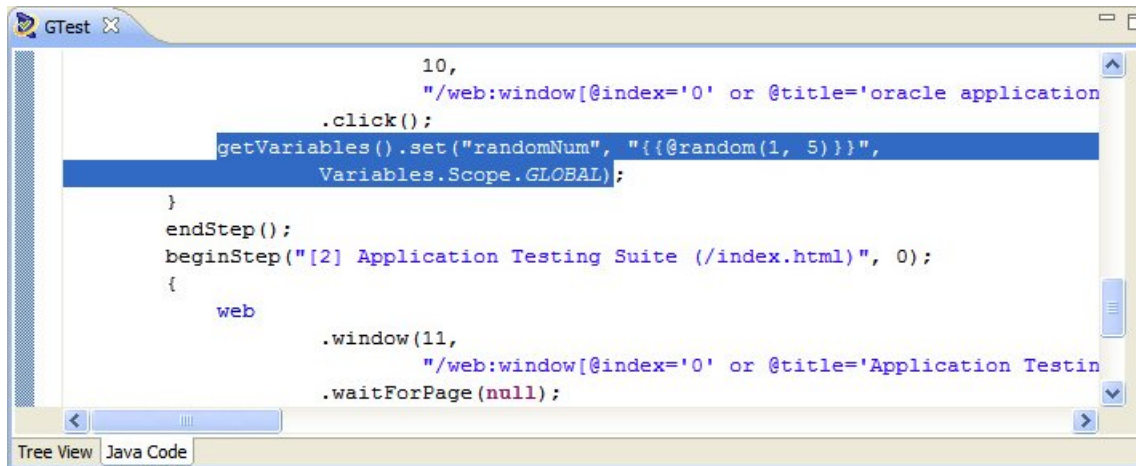


- script variables can be set to literals
- or take their values from data banks, built-in functions, or other variables

Built-in Script Variable Functions

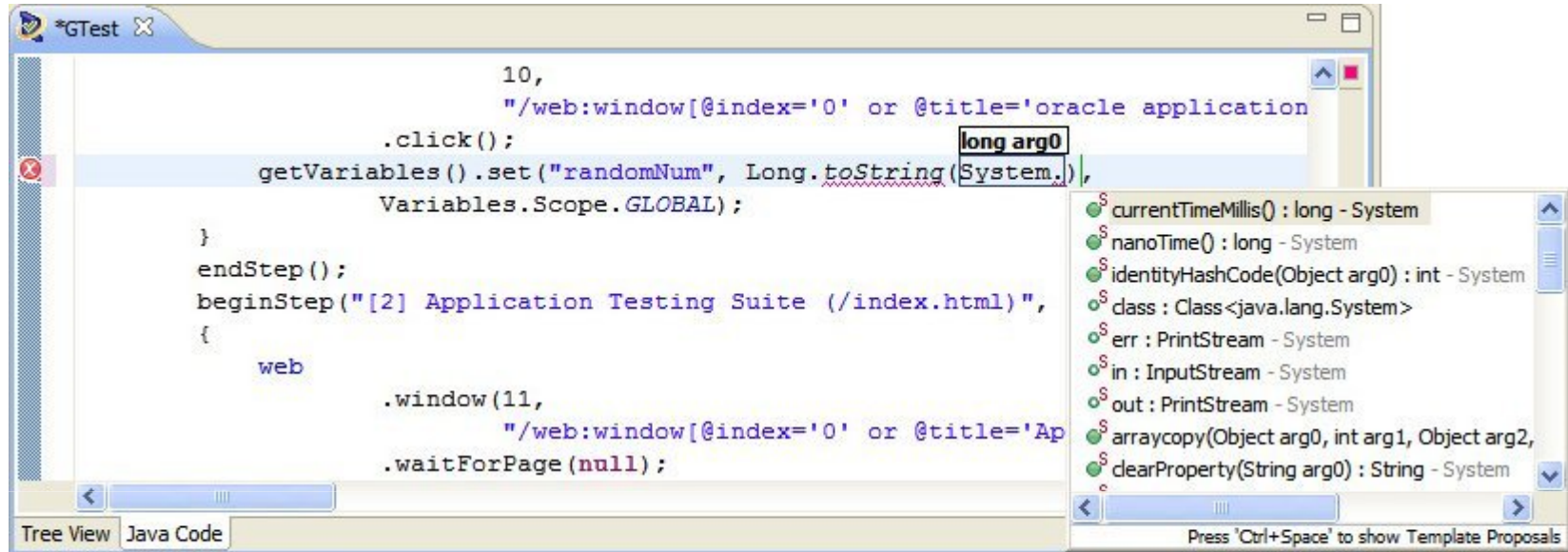


- Here is an example of a script variable taking its value from a built-in function, **@random**
- We can do the variable assignment from the Eclipse GUI..



- or from the Java code window for maximum flexibility

Setting Variables from Java code



```
10,
    "/web:window[@index='0' or @title='oracle application
    .click();
    getVariables().set("randomNum", Long.toString(System.
    Variables.Scope.GLOBAL);
}
endStep();
beginStep("[2] Application Testing Suite (/index.html)",
{
    web
        .window(11,
            "/web:window[@index='0' or @title='Ap
        .waitForPage(null);
```

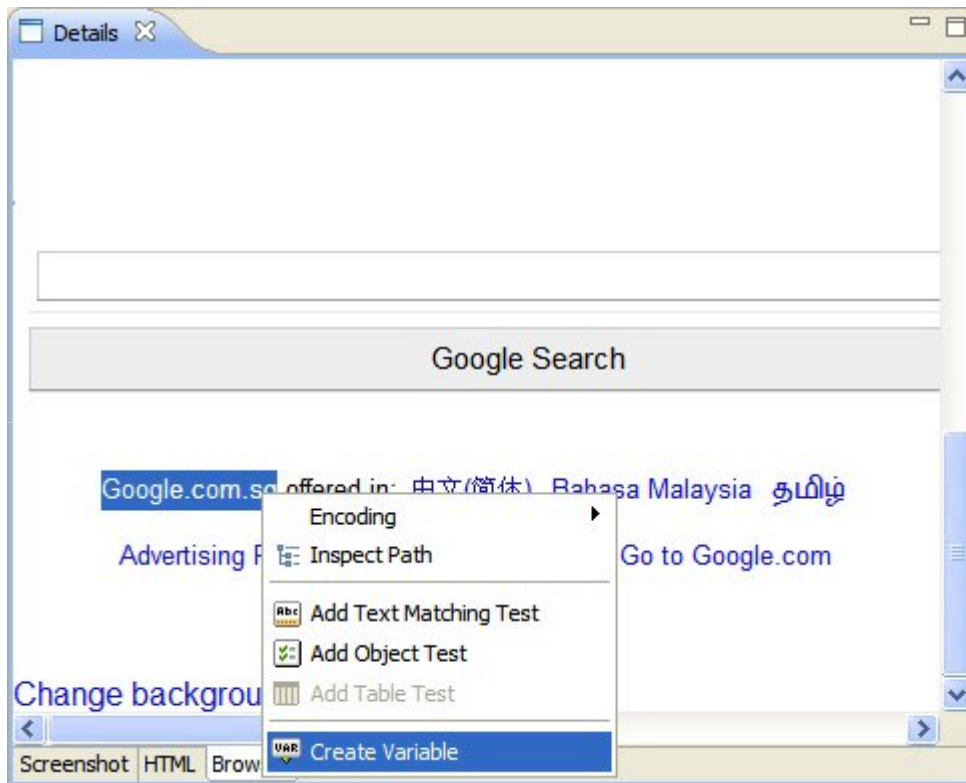
long arg0

- currentTimeMillis(): long - System
- nanoTime(): long - System
- identityHashCode(Object arg0): int - System
- class: Class<java.lang.System>
- err: PrintStream - System
- in: InputStream - System
- out: PrintStream - System
- arraycopy(Object arg0, int arg1, Object arg2,
- clearProperty(String arg0): String - System

Tree View Java Code Press 'Ctrl+Space' to show Template Proposals

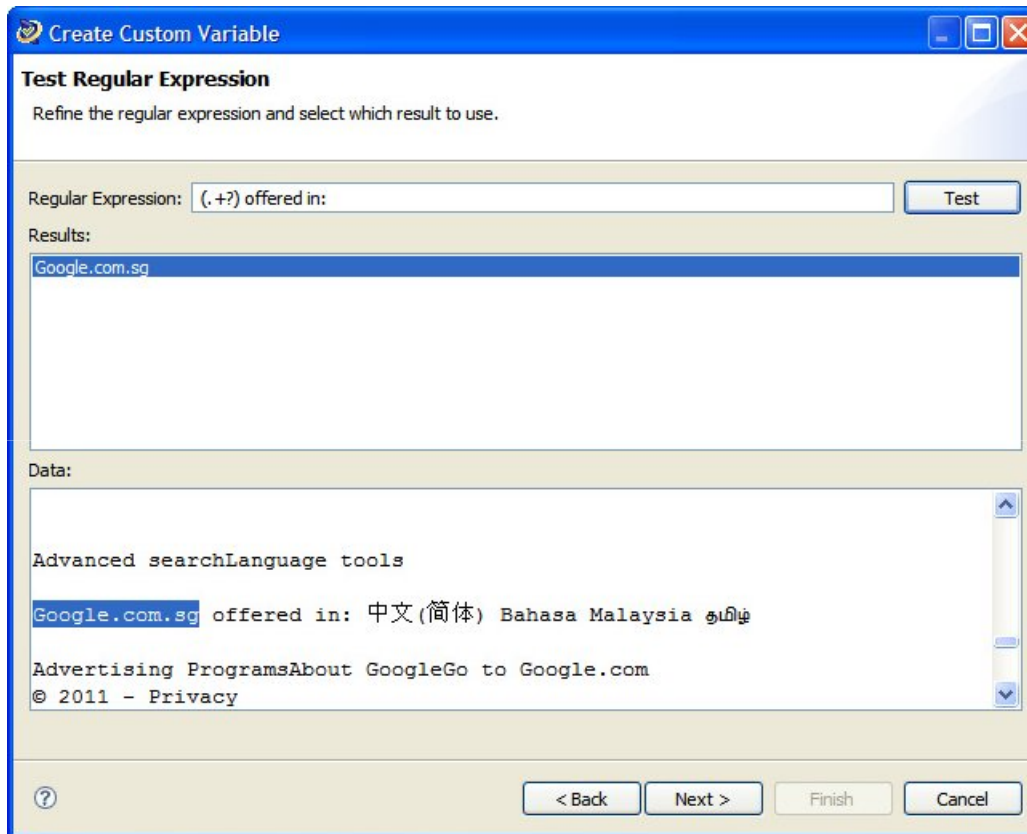
- script variables can also be set to arbitrary Java code, in this case a trivial example (*System.currentTimeMillis()*) but any Java code is possible, hence OpenScript tests are infinitely extensible
- note that full Eclipse functionality including syntax completion and code refactoring are available in OpenScript!

Setting Variables from Application Results



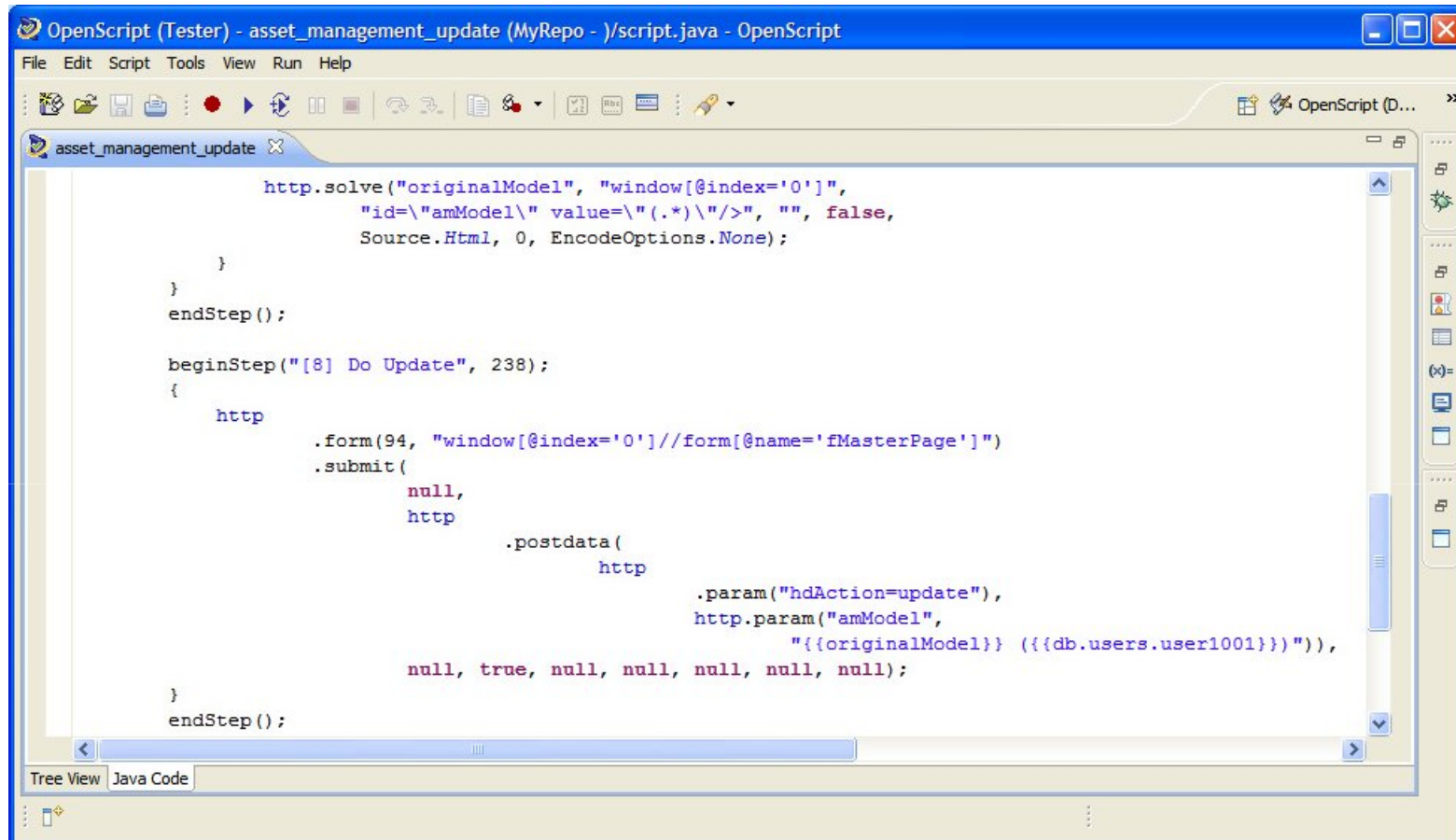
- It is also trivially simple to extract application output and assign to a script variable
- Simply highlight the text desired in the **Results** window, right-click, and select **Create Variable**

Regular Expression Builder



- OpenScript automatically creates a regular expression to extract the desired content
- Testers can also modify the regular expressions and test these changes immediately in the built-in regular expression builder

The possibilities are endless!



```
OpenScript (Tester) - asset_management_update (MyRepo - )/script.java - OpenScript
File Edit Script Tools View Run Help
asset_management_update
http.solve("originalModel", "window[@index='0']",
           "id=\"amModel\" value=\"(.*)\"/>", "", false,
           Source.Html, 0, EncodeOptions.None);
}
endStep();
beginStep("[8] Do Update", 238);
{
  http
  .form(94, "window[@index='0']//form[@name='fMasterPage']")
  .submit(
    null,
    http
    .postdata(
      http
      .param("hdAction=update"),
      http.param("amModel",
                 "{{originalModel}} ({{db.users.user1001}})"),
    null, true, null, null, null, null);
}
endStep();
Tree View Java Code
```

Thanks to the power of 100% Java test scripting.



- What: Introduction to Oracle Application Testing Suite
- How: a Worked Example
- **Testing EBS & Web Services Applications**
- Conclusion

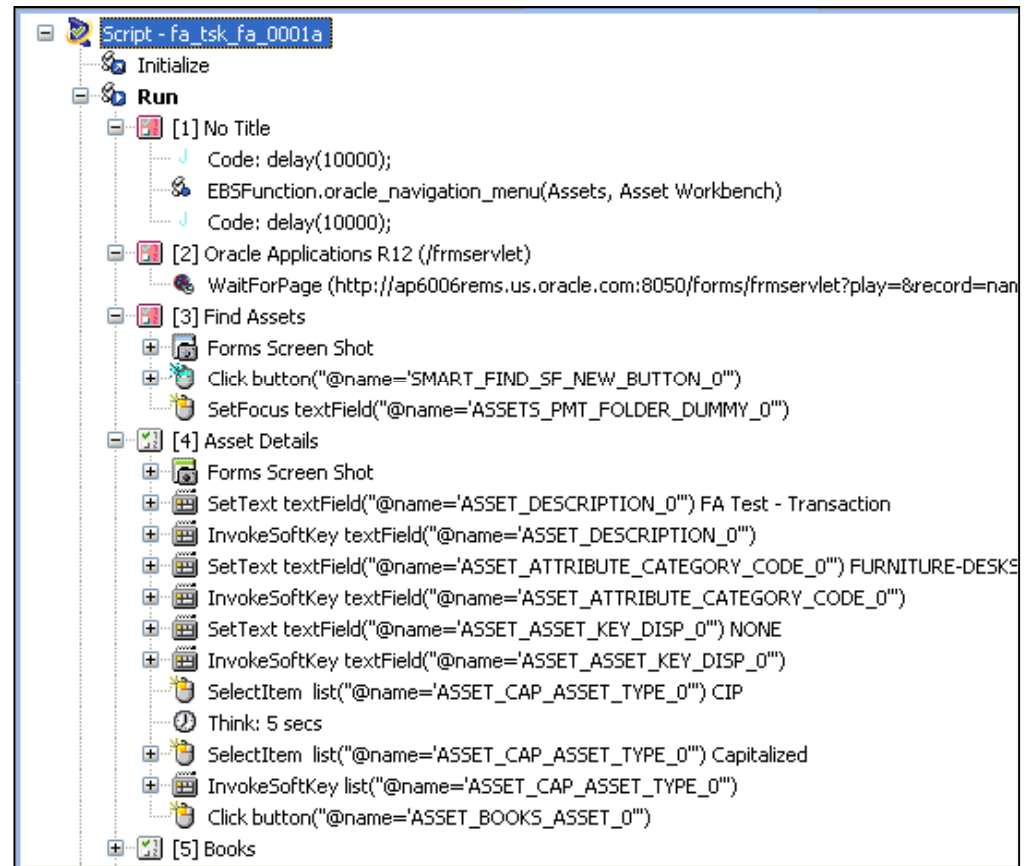
Application Testing Suite – Key capabilities

- Test Starter Kits for EBS 11i & R12
- Web services Accelerator
- Database testing accelerator
 - Import RAT DB captures and create synthetic scripts
- Automatic script generation
 - Zero-effort load testing script development
 - Scripts generated using RUEI
- Smart integration with EM GC for application performance diagnosis
 - JVM Diagnostics for mid-tier
 - DB Profiles for Oracle Database



Oracle E-Business Suite 11i & R12 Test Starter Kits & Sample Scripts

- New and updated test starter kits for Oracle e-Business Suite R12 and 11i
 - EBS 12.1.1, EBS 12.1.2
 - EBS 11.5.10.2
- Include ~75 unique test scripts, covering 17 different EBS modules
- Also includes load test scripts based on EBS Performance Benchmark Kits
- Scripts were created against a Vision demo database





EBS Apps Covered by Test Starter Kit

- **Financial Products:**
 - Oracle Payables
 - Oracle Receivables
 - Oracle General Ledger
 - Oracle Assets
 - Oracle Internet Expenses
- **Manufacturing Products:**
 - Oracle Purchasing
 - Oracle Inventory
 - Oracle Order Management
 - Oracle Cost Management
 - Oracle Materials Requirement Planning
 - Oracle Process Manufacturing Products
- **CRM Products:**
 - Oracle Contracts
 - Oracle Service
 - Oracle Marketing
- **Human Resources Product:**
 - Oracle Human Resources
- **Projects Product:**
 - Oracle Projects
- **Technology Product:**
 - Oracle Application Object Library



Web Services Testing Accelerators Overview

- Web Services Testing Accelerators enable automated functional and load testing for SOA Web Services
 - Supports SOAP/HTTP-based Web Services
 - Built on the OpenScript Web/HTTP module framework for creating and executing scripts
 - Integrated with Oracle Load Testing to execute scripts for load testing across thousands of concurrent VUs
 - Includes an integrated WSDL parser and manager
 - Lets you create SOAP requests directly from WSDL files
 - XML Editor to edit & parameterize SOAP requests
 - Integrate Web Services scripts/requests with other script types like Web/HTTP

ORACLE®