Eclipse Testing Tools

[Draft]

This page lists testing tools integrated with Eclipse IDE by topic. How this list is related to Harmony project? Since integrated development environments are collections of mature development tools, it worth to check this list before starting tool implementation from scratch.

Eclipse Extensibility Model

Eclipse is freely extensible by means of plugins. The list of plugins can be found at Eclipse Plugin Central. There is testing plugins subcategory which includes Test and Performance Tools Platform (TPTP) plugin. This plugin is an Eclipse top-level project which tries to establish a unified platform for testing tools.

Test Formats

- JUnit is a recommended tool for developing tests. Is de-facto standard and it is integrated into Eclipse by default.
- TestNG is a new generation of JUnit tests which is integrated into Eclipse by means of the following plugin. It can run JUnit tests without any changes, and has an automatic converter for JUnit tests to testNG format. Javadoc taglets or java 5.0 annotations can be used to specify test types and dependencies.

Test Execution

Test Finders

Test finders allow to select and run a subset of available test base.

- Eclipse has a primitive test finder which allows to run all tests in the selected project, package or source folder.
- A test can be excluded from test run by excluding the test from build process (<Right Mouse Button> -> Build Path -> Exclude).
- There is a feature request for advanced test finders.

Release Engineering

Release Engineering is a process of creating binary builds from source code. It includes the following steps:

- Acquire a source code from a source contol system
- Build the source code
- Bundle the binaries
- Run promotion checks
- Maintain integrity of the source repository

If promotion checks pass, the build is promoted for the test cycle.

Release Engineering plugin used by Eclipse team is reported to be hardly configurable. Though there are many aternatives.

Source Control

CVS is a default source control system for Eclipse. Many other source control servers such as Subversion are CVS-compatible. Specialized plugins can be found here.

Build Tools

Build tools are script engines which handle local and remote dependencies.

- Local dependency management allows incremental builds which rebuild changed files only and files indirectly affected by the change.
- Remote dependencies are taken into account for automatic download and updates of binary or source library distributions from the Internet.

Eclipse has the following support for building tools:

- XML-based build scripts are highlighted and folded in the editor.
- By default the internal Eclipse builder, Apache Ant and a command line tool are supported. Eclipse is extensible with custom builders.
- Eclipse is integrated with Maven by means of the followingplugin.
- Eclipse requires C/C++ development tools to be installed to edit GNU makefiles.

Continuous Integration

Continuous Integration is an agile software development practice for a teamwork with the following features:

- Daily synchronization with a common repository
- Continuous repository verification (automated build and testing)

Eclipse supports the following systems for continuous integration:

- Any supported builder can be used in "Build Automatically" mode.
- Here is a tutorial on integration of the popular coninuous integration systems CruiseControl, Luntbuild, and Anthill.
- Continuous Testing technology runs unit tests in parallel with development process.
- TPTP has an extensible SOA-based design.
- CruiseController Eclipse Plugin allows remote management of CruiseControl build queues.

UI Testing Tools

Abbot and other JUnit based testing tools are used to create automated tests for GUI applications. Test and Performance Tools Platform embeds these tools and enriching them with Eclipse-specific event hooks.

Distributed Testing Tools

- TPTP describes remote agents architecture.
- JUnit has a set of distributed test runners
- TestNG intself supports distribution of operations on slave machines

Tracing and Profiling Tools

- Tracing and Profiling Tools Project

Source Code Coverage and Other Metrics

Here is a list of Eclipse plugins which provide source code metrics.

Stress/Performance Test Generators

The followng tools allows combining existent functional JUnit tests into stress/performance tests.

- JUnitPerf combines tests programmatically.
- JUNITPP allows combining tests from the command line.

Automatic Test Generators

Most test generators just help test developers to obtain a trivial JUnit template. The folloiwng generators are more complex and worth additional investigation.

- Javver generates random bytecode
- Cricket Cage creates test cases recording an exeecution of a valid program

Test Reports

By default Eclipse reports JUnit results in a widget. TestNG have its own HTML reporting. JUnit has an extension to geneerate PDF reports.

Additional References

1. Mikhail Voronin at Intel
2. Testing Tools Inside Eclipse