HUT

Back to Test Tracking

Summary

- Mission and Status
- How to configure and run Harmony Unit Tests
- Results

Mission and Status

Harmony Unit Tests (aka HUT) are one of Harmony's core test suites. HUTs consist of unit tests for classes from all classlib modules and include API tests, implementation-specific tests and regression tests. API tests normally reside in *<module root>/src/test/api* folder, implementation-specific in *<module root>/src/test/impl*. Regression tests are usually marked with comments that contain numbers of corresponding JIRA issues.

HUTs can be effectively used for regression testing. All Harmony committers and contributors are strongly encouraged to run these tests before creating patches or committing changes to Harmony repository.

Back to Summary

How to configure and run Harmony Unit Tests

Make sure that PATH environment variable contains JDK 1.5 (use RI) and Apache Ant (v1.6.5 or later), and that JAVA_HOME and ANT_HOME are properly set up. If you use proxy server, specify proxy settings by setting ANT_OPTS environment variable:

Windows:

```
set ANT_OPTS="-Dhttp.proxyHost=<proxy host> -Dhttp.proxyPort=<proxy port>"
```

Linux:

```
export ANT_OPTS="-Dhttp.proxyHost=<proxy host> -Dhttp.proxyPort=<proxy port>"
```

Before running the tests you may need to download required dependencies and build classlib itself. This can be achieved by doing the following:

```
cd <classlib trunk>
ant fetch-depends
ant
```

Let's assume you have compiled JRE you want to run the tests on at <JRE home>. The following command will run all non-excluded unit tests:

```
ant -Dtest.jre.home=<JRE home> test
```

NOTE: exclude lists for Harmony Unit Tests are plain text files that reside in *<module root>/make* directory and contain names of java files with excluded tests. Excluded tests won't be included into the test run under normal conditions.

To run tests from the selected classlib module only (i.e. beans, luni and etc.) you should type:

```
ant -Dtest.jre.home=<JRE home> -Dbuild.module=<module> test
```

You may also specify a pattern for the test class name to run. Please see the examples below.

```
    # run IntrospectorExceptionTest
    ant -Dtest.jre.home=<JRE home> -Dbuild.module=beans -Dtest.case=org.apache.harmony.beans.tests.java.beans.
IntrospectionExceptionTest test
    # another form to run IntrospectorExceptionTest
    ant -Dtest.jre.home=<JRE home> -Dbuild.module=beans -Dtest.case=org/apache/harmony/beans/tests/java/beans
/IntrospectionExceptionTest.java test
    # run all tests those names start with Introspect
    ant -Dtest.jre.home=<JRE home> -Dbuild.module=beans -Dtest.case=org/apache/harmony/beans/tests/java/beans
/Introspect*.java test
```

Pay attention to fact that if you specify a pattern then processing of exclude lists is not performed. Thus the following command will run all BEANS tests regardless the exclude lists (please be careful since this may crash the VM 😉):

```
    ant -Dtest.jre.home=<JRE home> -Dbuild.module=beans -Dtest.case=**/*.java test
```

You may also pass some parameters like JIT modes and heap size to JRE under test by utilizing *hy.test.vmargs* system property. For example:

```
    # this defines the value for prop1 system property and runs DRLVM in interpreter mode
    ant -Dtest.jre.home=<JRE home> -Dhy.test.vmargs="-Dprop1=data1 -Xint" test
```

Back to Summary

Results

Test reports can be found at <*classlib trunk*>*/build/test_report* .

Back to Summary

Back to Test Tracking