Testing Convention

Here I try to summarize what we have discussed about Harmony testing configurations (TestNG) in harmony-dev, and list the problems which we shall solve and make common agreements in community.

Please feel free to edit this page. - Richard Liang

How to handle bootclasspath test

We have agreed that bootclasspath tests and classpath tests should be put into different directories

Reference: How to handle bootclasspath tests

Grouping Harmony Test

We may define Harmony test groups according the following dimensions:

Operating System: os.<os id>

- ***os.<os id>*** - group of tests which are designed for one specific operating system. A test may be in more than one of the groups. e.g., @Test (groups={"os.win", "os.linux"})
- ***@Test*** - default group of tests which pass on any operating system.

Architecture: arch.<arch id>

- ***arch.<arch id>*** - group of tests which are designed for one specific architecture. A test may be in more than one of the groups. e.g., @Test (groups={"arch.IA32", "arch.IA64"})
- ***@Test*** - default group of tests which pass on any architecture.

Test Type: type.impl

- ***type.impl*** - groups of tests which are designed for Harmony-specific implementation
- ***@Test*** - group of tests which are tests for APIs in the Java Specification. We may call it type.api group, but to make the annotations simple we omit "type.api"

Test Level: level.regression, level.integration, level.stress, etc. (***Levels of Test*** refer to the increase in complexity as moving through test cycle.)

- *level.regression - group of tests which are regression for some bugs.* (We have no clear requirement to exclude/include regression test, so this group may not be defined.)
- ***level.integration*** - group of tests which aim to verify the interaction between modules
- ***level.stress*** - group of tests which aim to verify performance
- ***@Test*** - group of unit test

Test State: state.broken, state.broken.[os id].[arch id], state.broken.<vm>.[vm id]

- ***state.broken*** - group of tests which fail on every platform, because of bugs of tests or implementation. We need to fix the bugs of tests or implementation to make them pass.

- ***state.broken.[os id].[arch id]*** - group of tests which fail on one specific platform. "os id" and "arch id" may be omitted. For example, *state.broken.win* means the test fails on windows with any architecture

- ***state.broken.<vm>.[vm id]*** - group of tests which fail on one specific vm

Environment: env.display

- ***env.display*** - group of tests which need a display to run. For example, ATW and Swing tests.
- ***@Test*** - group of tests which have no particular requirement on testing environment.

VM: vm.<vm id>

- ***vm.<vm id>*** - group of tests which are designed for on specific vm
- ***@Test*** - group of tests which pass on all vm.