Where is it?

- VisualWorks 7.7.1 (and ObjectStudio and WebVelocity)
- VASmalltalk 8.0.2
- Pharo 1.1
- GemStone
- Dolphin 6.1
- Smalltalk/X
- VSE port in progress

Thanks to James Foster, Tim MacKinnon, Jan Vrany, Leandro and Valeria, Yuri Mironenko , ...

Anywhere else it should be?

(and if so, will you port it?)

Cross-dialect Compatibility

Pharo

- Pure SUnit 4.0 in www.squeaksource.com/SUnit
- SUnit 4.0 + Pharo in www.squeaksource.com/PharoInbox
  - (#cleanUpInstanceVariables not called by default in the basic merge)

Smalltalk/X:

- pure SUnit 4.0 version
- 4.0 + unintrusive tweak to capture last result of test run

Other dialects: pure SUnit 4.0   (VSE in progress)

Please merge 4.0 diff: 3.1 into whatever you use

What’s new: TestResource

Resource-processing is ordered

- — a test’s resources setUp in order and tearDown in reverse order
- — a resource sets up before and tears down after everything that uses it
- — tearDown ensured if setUp entered (like TestCase)

Resources understand #assert:… protocol

- — easy to refactor between MyTest>>setUp and MyResource>>setUp

Resources are made available just-in-time:

- — first using test sets it up; later tests see it has (or failed to) set up
- — MyResource>>setUp called by framework, not by MyResource>>new

Using TestResources trades test isolation for performance

- — TestCase>>setUp, testSomething, tearDown) similar functions,
- — TestResource>>setUp, isAvailable, tearDown) different epochs

Niall
Ross

What’s new: Pluggable Exceptions

runCase: dispatches on the exception

- TestFailure adds failure, Error adds error
- create subclasses to plugin specialised behaviour
- (see #sunitAnnounce:toResult:)

Trivial improvements

- better-integrated calling of #shouldInheritSelectors, #isAbstract
  - new method #lookUpHierarchyRoot for rare inheritance cases
- pluggability tweaks, e.g. self addPass: … instead of passed add: …,
  - (more can be done to improve pluggability)
- failing resource logs the calling test
  - (more logging improvements needed !)
- TestAsserter: superclass for any delegate classes a test needs

Future: SUnit 4.1 can exploit ANSI

Time to retire a lot of compatibility methods?

- #sunitSelectors
- #sunitOn:do:
- #sunitEnsure:
- #sunitName
- #sunitAsSymbol

and a class?

- who uses SUnitDelay?

and maybe some more?

- #sunitMatch:
- #sunitAddDependent:, #sunitRemoveDependent:

SUnit: core and friends

SUnit core: the Camp Smalltalk project

- robust cross-dialect test running
- compatibility base for SUnit variants in each dialect

SUnit variants

- aim: merge each release into dialect's main variant
- pluggability improvements will let tweaks be moved out of core
  - but new ideas may put new tweaks back in
    - move out later … or move into project core
    - (first make it run, then make it right)

SUnit welcomes ideas

SUnit will remain cross-dialect, backward-compatible, small

Niall Ross