

# SailFin Tests

Sep 12, 2007

# SailFin Testing Process

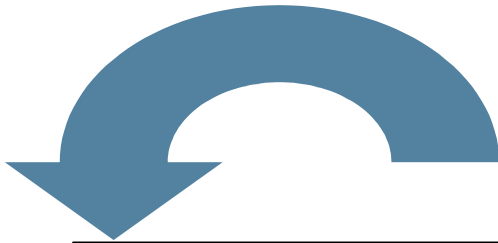
## Developer Tests

Basic module level unit/developer tests to test units separately



## QuickLook Tests

Integrated sanity tests into nightly builds that have highlevel functional tests.



## Smoke Tests

Higher bar than QuickLook to promote the builds.



## BAT Tests

Basic Acceptance Tests. Next level to accept the builds.



## System Tests

User level real world scenario based tests.

# SailFin Testing Framework

- Introduction
- Framework
- Workspace
- Sample Test Development
- Guidelines

# Introduction

- ◆ Unit Tests
  - ◆ Validate individual units of source code
  - ◆ Typically at Class (base/super, abstract/derived class)
- ◆ Who does
  - ◆ Developers and not end-users
- ◆ What can be expected
  - ◆ Make sure your unit of code works! Feel great;)
  - ◆ Focus on functionality at unit level
- ◆ What can't be expected
  - ◆ Catching all bugs (as in testing) with all possible paths
  - ◆ Integration errors, System wide errors, Performance problems

# Benefits

- ◆ Facilitates Change
  - ◆ Code refactoring easy – catch the method level bugs.
- ◆ Easy for Integration
  - ◆ Parts or units already work
- ◆ Separate interface with implementation
  - ◆ Use mockups
- ◆ Documentation
  - ◆ Sort of living doc
- ◆ Quality
  - ◆ An example on working code
  - ◆ Focus on end-user use-cases

# Framework

- ◆ Test Harness
  - ◆ ANT/Maven Based
- ◆ Test Framework
  - ◆ Java Based - Test NG/JUnit (Netbeans IDE helps)
  - ◆ SipUnit (under evaluation)
- ◆ Test Tools
  - ◆ SIPp (scenario based)
  - ◆ SIP client based on JAIN-SIP/NIST stack
- ◆ Test Report
  - ◆ Internal XML based
  - ◆ Junit

# Workspace

- ◆ Developer or community tests
  - ◆ sailfin/sailfin-tests/community
  - ◆ Use this workspace for your tests
- ◆ QL
  - ◆ sailfin/sailfin-tests/quicklook
  - ◆ This workspace is the reference and make changes in sync with Quality team

# Workspace

/sailfin/sailfin-  
tests

- |
- |-maven.xml
- |-build.xml
- |-build.properties
- |-config.properties
- |-cluster-  
  config.properties
- |-config
- |-<common-xml-  
  files>
- |-lib
- |-util
- |-build
- |-<module>
- |-<archive>
- |-<classes>
- |-reports

|--community [ Holding all development or unit tests ]

- |---maven.xml
- |---build.xml
- |---build.properties
- |---<module>
- |----<feature|testsuite>
- |-----<scenario|test>
- |-----build.xml
- |-----build.properties
- |-----config
- |-----sql
- |-----descriptor
- |-----src
- |-----client
- |-----ejb
- |-----web

- |---<module>
- |----<feature|testsuite>
- |-----<scenario|test>
- |-----build.xml
- |-----testng.xml
- |-----build.properties
- |-----nbproject
- |-----src
- |-----test
- |-----web

- |---<module>
- |----<feature|testsuite>
- |-----<scenario|test>
- |-----build.xml
- |-----testng.xml
- |-----build.properties
- |-----nbproject
- |-----src



# Sample Test Development

## ◆ Typical build.xml

- ◆ `<project name="mymodule-myfeature" default="usage" basedir=".">`
- ◆ `<import file="../../../../config/common.xml"/>`
- ◆ `<property file="{basedir}/build.properties"/>`
- ◆ `<target name="all" depends="clean, build, setup, deploy, run, undeploy, unsetup"/>`
- ◆ ...

## ◆ Typical Targets

- ◆ *start-domain, stop-domain, restart-appserver*
- ◆ *clean, compile, build, deploy, undeploy, package-sar*
- ◆ *cluster, delete-cluster*
- ◆ *as-admin-common*
- ◆ Note: Check common.xml for more targets

## ◆ Use Netbeans project

- ◆ Check invite-sample as an example

# Sample Test Development

- ◆ SIP client - SIPp runtime
  - ◆ Download the SIPp latest version and get the binary for the targeted platform.
    - ◆ Sources @URL  
[http://sourceforge.net/project/showfiles.php?group\\_id=10430](http://sourceforge.net/project/showfiles.php?group_id=10430)
    - ◆ Copy the binary (sipp) in \$WS\_HOME/lib directory (with file name sipp)
  - ◆ Create SIPp scenario xml (sipp.xml) under scenarios directory.
    - ◆ <test>/scenarios/sipp.xml
  - ◆ Run "ant sipp" --> This will parse the logs and include in the report.
    - ◆ Note: If you need fine control, set the appropriate parameters to sipp target.
  - ◆ Console o/p:
    - ◆ SIPp screen and final PASS/FAIL summary as below
    - ◆ All these results go to \$WS\_HOME/test\_resultsxxx.xml along with other test results.
  - ◆ Results
    - ◆ Run "ant report" and check test\_results.html

# Sample Test Development

## ◆ Internal Reporters

- ◆ SimpleReporterAdapter API (Generates test-results.xml and test-results-valid.xml)
  - ◆ static SimpleReporterAdapter reporter = new SimpleReporterAdapter();
  - ◆ reporter.addDescription(String testdescription)
  - ◆ reporter.addStatus(String testName, String testStatus)
  - ◆ reporter.printSummary()
- ◆ TestReporter API (new in SailFin) -Another abstraction on top of SimpleReporterAdapter API
  - ◆ TestReporter reporter = new TestReporter()
  - ◆ reporter.addMessageSequence(String... messages)
  - ◆ reporter.addSendStatus(String)
  - ◆ reporter.addReceiveStatus(String)
  - ◆ reporter.addSendMessage(String)
  - ◆ reporter.addReceiveMessage(String)
  - ◆ reporter.report(String testName, String testDescription)

# Sample Test Development

## ◆ Report generation

### ◆ ant report (for Internal xml report)

```
*****  
PASSED= 1  
----- =====  
FAILED= 0  
----- =====  
DID NOT RUN= 0  
----- =====  
Total Expected=1  
*****
```

### — ant report-junit (for Junit xml report)

```
Summary  
(2007-09-05 04:41 PDT)  
*****  
PASSED = 1  
----- =====  
FAILED = 0  
----- =====  
ERRORS = 0  
----- =====  
TOTAL = 1  
----- =====  
PASS RATE= 100%  
*****
```

# Guidelines

- ◆ Automate the tests
- ◆ Before checkins (normal code)
  - ◆ Run your module unit tests
  - ◆ Run dependent module unit tests (under sailfin-tests/community)
  - ◆ Run QuickLook tests (under sailfin-tests/quicklook)
- ◆ Run smoke tests for large code checkins
- ◆ Hookup your tests to Hudson/Tinderbox
- ◆ Watchout nightly build notifications – QL and SmokeTests may have failures!

# Current Status

- ◆ Sailfin-Tests workspace is ready for your checkins
- ◆ SIP clients – NIST stack, SIPp tool setup is ready
- ◆ Contact mailing lists
  - ◆ [quality@sailfin.dev.java.net](mailto:quality@sailfin.dev.java.net)