

Wind River Framework for Automated Software Testing

Device testing represents the largest portion of the development schedule. Reductions in the testing cycle correlate to reductions in time-to-market. Wind River Framework for Automated Software Testing can cut development costs, improve quality, and reduce testing cycle time. The framework acts as a meta-test execution tool for commercial mobile Linux platforms. It can easily be integrated with proprietary test suites, test suites from third parties, or open source test suites.

Wind River Framework for Automated Software Testing can help determine compliance, measure performance, and prove stability of mobile devices running either Android or Moblin. This test tool is designed to accept the results of a build, execute a selection of hundreds of Wind River–authored tests and tens of thousands of existing scripts, archive the relevant artifacts, and provide the results in a single report, all in a completely automated way.

Bring Software and Devices to Market on Time

Because Linux is embedded in devices that can vary widely from one another, testing each device can take significant time and money. Framework for Automated Software Testing provides a solution to complex, comprehensive testing,

allowing tested devices to go to market on time and within budget. Engineering development and integration efforts using Framework for Automated Software Testing can decrease by as much as 25%.

Framework for Automated Software Testing can test a target device on multiple levels including the following:

- Application
- Email
- Browser
- Display
- GPS
- Power management
- Hardware and driver

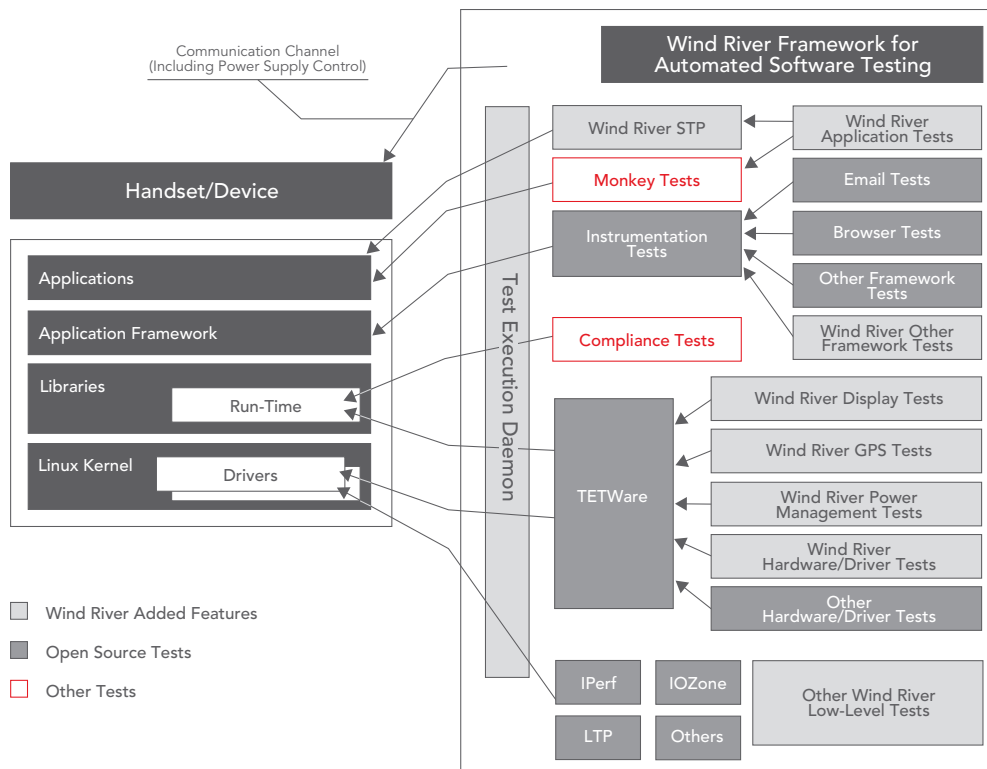


Figure 1: Wind River Framework for Automated Software Testing overview

A Comprehensive Test Management Tool

Framework for Automated Software Testing is a collection of custom test scripts as well as “wrapper” test scripts that use other tools/frameworks/test suites/etc. to perform device testing. The wrapper scripts allow easy integration of any third-party or open source test suite, enabling Framework for Automated Software Testing to run any type of test from any source. Framework for Automated Software Testing combines multiple test systems into one easy-to-use system.

Framework for Automated Software Testing employs a collection of scripts and test tools to test all levels of software on the device. Depending on the chosen operating system, Framework for Automated Software Testing integrates the following test suites:

- Android Compatibility Test Suite (CTS)
- Android Monkey
- Moblin Compliance
- Java instrumentation
- Hardware Abstraction Layer (HAL)
- Linux Test Project (LTP)
- IOzone
- Iperf

Framework for Automated Software Testing makes it easy to add test tools, such as customer-specific frameworks, and integrate them into the test plan hierarchy and test reports.

Framework for Automated Software Testing allows for full automation of testing, including device control. Target devices can be flashed, powered on and off, booted, and tested. Once a target device is properly configured and tests set up, there are no manual steps required. This allows engineers more time to spend on development and integration because testing can happen quickly without manual interaction.

Many Tests, One Button

Framework for Automated Software Testing defines a hierarchy that can automatically test a device for a broad range of functionality at different levels. This hierarchy of tests can be used to define a test plan for the device. To execute a test run of the test plan, you can make a simple selection from a

graphical user interface or use a simple command-line interface. Either way, one action results in a test run that executes all of the tests in the test plan, collects the results of all the tests, and reports them in the ways you have chosen. Testing of devices, especially repetitive testing, is simplified to a single user action.

Many Tests, One Report

Framework for Automated Software Testing consolidates the results of all the tests in a test run into one comprehensive test report. The tests may be executed by different underlying test frameworks (e.g., LTP, Iperf, IOzone) and may explore several levels of testing on the device, but all of the results are combined into one report that provides complete information about the test results.

Framework for Automated Software Testing also provides support for multiple targets and multiple projects, allowing automated testing of multiple devices at one time.

Enhanced Test Engineer Interface

Framework for Automated Software Testing offers test engineers two interfaces for controlling automated test execution: a web user interface (web UI) and a command-line interface (CLI). Working in the interface in which they are most comfortable, test engineers can determine which tests to run and which results to view.

```
root@dhcp-147-212:~# /opt/fast/bin/fast start-testrun test_run.ini
Testrun with id = 9 started successfully

root@dhcp-147-212:~# /opt/fast/bin/fast get-testrun-status 9
[TG 9] MainTestGroup
[TC 155][2009-06-25 11:30:43 - None][RUNNING] Monkey

TESTRUN:          9
Percent completed: 0%
Total tests:      1
Tests completed:  0
Total results:    0
Results passed:   0
Results failed:   0

root@dhcp-147-212:~# /opt/fast/bin/fast get-log 155
+-----+-----+-----+
| id | result name | status |
+-----+-----+-----+
| 554 | Monkey | failed |
+-----+-----+-----+

root@dhcp-147-212:~# /opt/fast/bin/fast get-result 554
+-----+-----+-----+-----+
| id | artifact name | type | size |
+-----+-----+-----+-----+
| 183 | Monkey.log | text/plain | 721kb |
+-----+-----+-----+-----+

+-----+-----+-----+
| id | keyval key | keyval value |
+-----+-----+-----+
```

Figure 2: Automated test tool CLI

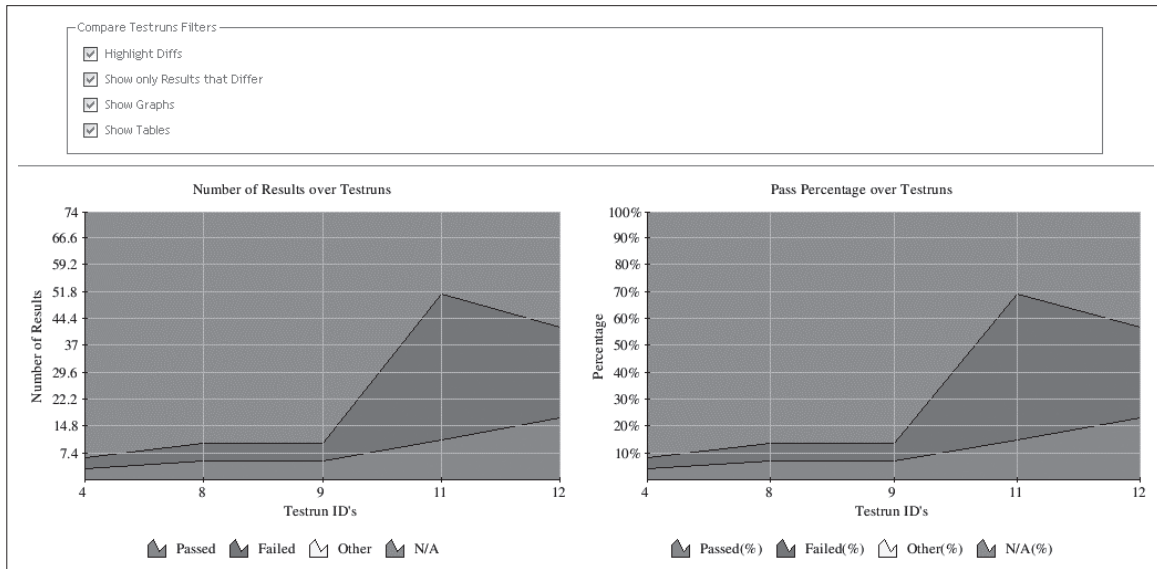


Figure 3: Regression tool graphs

Simple Test Configuration

Tests are easy to configure and run. In addition to setting up testing through the web UI, test engineers can create an INI file through the CLI that specifies all aspects of the tests, including the following:

- Test targets
- User running the tests
- Which tests to run
- Test run priority

Automated Test Reporting

Framework for Automated Software Testing includes options for reporting, making test results convenient to generate and easy to read:

- **Regression tool:** Available through the web UI, this tool allows test engineers to compare test results against a baseline test run.
- **Nightly test execution reporting:** This report is produced at the end of each test run and emailed to recipients designated in the INI file or via the web UI.

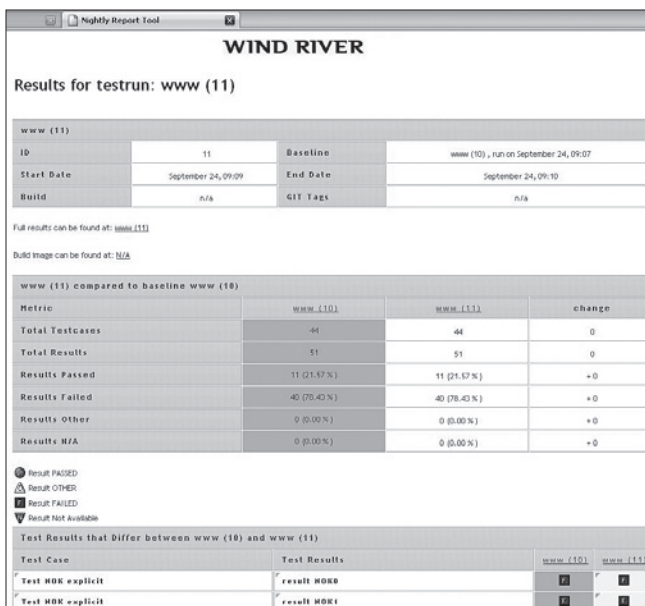


Figure 4: Nightly reporting tool web UI