Testkit-Lite User Guide

1. Introduction

This document gives an overview of Testkit-Lite and guides how to run test cases for Core and WebAPI components on Testkit-Lite.

2. Overview

Testkit-Lite is a test runner with command-line interface. It has the following functions:

- Accepts .xml test case descriptor files as input
- Drives automatic test execution
- Provides multiple options to meet various test requirements

3. Prerequisites

Before using Testkit-Lite, ensure that:

For Host (>=ubuntu11.10)

- Two packages have been installed: python-pip and requests.

You can run the following command to install the packages:

$ sudo apt-get install python-pip

$ sudo pip install requests

Note: python should be version 2.7 or later.

For Target device:

- **One packages have been installed: requests**

You can get requests rpm in Testkit-Lite package, and install it on target

$ rpm -ivh requests.rpm

- **An .xml test case descriptor file is ready**

Testkit-Lite recognizes and drives test cases through an.xml test case descriptor file, which describes test script, program, or HTML web test page.

4. Installing Testkit-Lite

To install Testkit-Lite from source code, perform the following steps:

- install on host:
  Navigate to the work directory testkit-lite:
  D**ouble click the xxx.deb to install testkit-lite**
- install on target (run in single mode):
  Navigate to the work directory testkit-lite:
  $ rpm - ivh xxx.rpm

5. Testkit-Lite Options

- Mandatory options

Option	Description
-f devices:"<test_descriptor_file>.xml"	Specify one or more test case descriptor files.
-e <Web Runtime Environment>	Specify Web Runtime Environment to run WebAPI testing, for example,WRTLauncher<test_widget_name>. Note: Only required for WebAPI testing.

- Optional options

Option	Description

-A	Testkit-Lite runs only auto test cases.
-M	Testkit-Lite runs only manual test cases.
-o <test_result_file>	Specify the name of result file. Testkit-Lite locates the result file under /opt/testkit/lite/latest/ by default.
-E	Specific test engine.
--fullscreen	Testkit-Lite runs WebAPI test cases in full screen mode.
--non-active	Disable the ability to set the result of core manual cases from the console
--enable-memorycollection	Enable the ability to release memory when the free memory is less than 100M
--version	Show version information
--deviceid	Specify sdb device serial information, get the first sdb devices serial information as default
--comm	Specify commodule type, set “tizenmobile” as default,'localhost' as single mode
--capability	Specify hardware capability file to filter test case

6. Running Testkit-Lite

- Show the help information:

$ testkit-lite –help

- Run non-WebAPI test cases:

$ testkit-lite -f device:”/PATH/<test_descriptor_file>.xml”

- Run WebAPI test cases:

$ testkit-lite -e "WRTLauncher<widget _name>" -f deivce:”/PATH/<test_descriptor_file>.xml”

- Run both non-WebAPI and WebAPI test cases:

$ testkit-lite -e "WRTLauncher<widget _name>" -f devices:
"/PATH/<test_descriptor_file_for_webapi>.xml /PATH/<test_descriptor_file_for_non_webapi>.xml"

- Run test in single mode:

$ testkit-lite -e "WRTLauncher <widget _name>" -f
devices:"/PATH/<test_descriptor_file_for_webapi>.xml " –comm localhost

- Run test cases with multiple test case descriptor files:

Testkit-Lite can accept as many test case descriptor files at one time as you want. You can type all test case descriptor files sequentially after the ‘–f’ option:

$ testkit-lite -f devices:”/PATH/TO/<test_descript_file>.xml … /PATH/TO/<test_descript_file_more>.xml”

- Run a subset of test cases in test case descriptor file with filters You can select and run a subset of test cases from the test case descriptor file by using filters.

Filter	Description
--type	Filter test cases by test case type: • functional_positive • functional_negative • security • performance • reliability • portability • maintainability • compliance • user_experience
--priority	Filter test cases by test case priority: • P0 • P1 • P2
--category	Filter test cases by test case category • IVI • Netbook • TV • Mobile
--status	Filter test case by test case status: • ready • approved • designed
--suite	Filter test case by test suite
--set	Filter test case by test set
--id	Filter test case by test case id
--component	Filter test case by test case component

- You can assign multiple values to each filter. For example, to select test cases of both P0 and P1 priority, run the following command:

$ testkit-lite --priority P0 P1 -f device:"<test_descriptor_file>.xml"

- You can use a group of filters at one time, because Testkit-Lite performs the AND logic when selecting test cases. For example, to select test cases by both priority and component filters, run the following command:

$ testkit-lite --priority P0 P1 --componentcomp1 comp2 -f device:"<test_descriptor_file>.xml"

- You can freely combine all those usages in one command to meet your test requirement. For example, to select and run test cases from two test case descriptor files by filters of priority, component and status run the following command:

$testkit-lite --priority P0 P1 --componentcomp1 comp2 --status ready -f device:"<test_descriptor_file_1>.xml <test_descriptor_file_2>.xml"

7. Checking Test Reports

After Testkit-Lite completes executing all test cases successfully, you can obtain an .xml test report from /opt/testkit/lite/latest

8. **Viewing Test Reports**

   Test report can be viewed in HTML format, so the data in the xml result file looks more human friendly.

   Please follow the following steps to view test report:

   - copy files: application.js back_top.png jquery.min.js testresult.xsl tests.css under directory /opt/testkit/lite/xsd/
   - put the files from step 1) under the same directory as the xml result file
   - open xml result file with a web browser(IE, Chrome or Firefox)