# Testkit-lite Tutorial

## Contents

# Overview

Guide for test case developer. If you are looking for the answer of below question, this guide is for you.

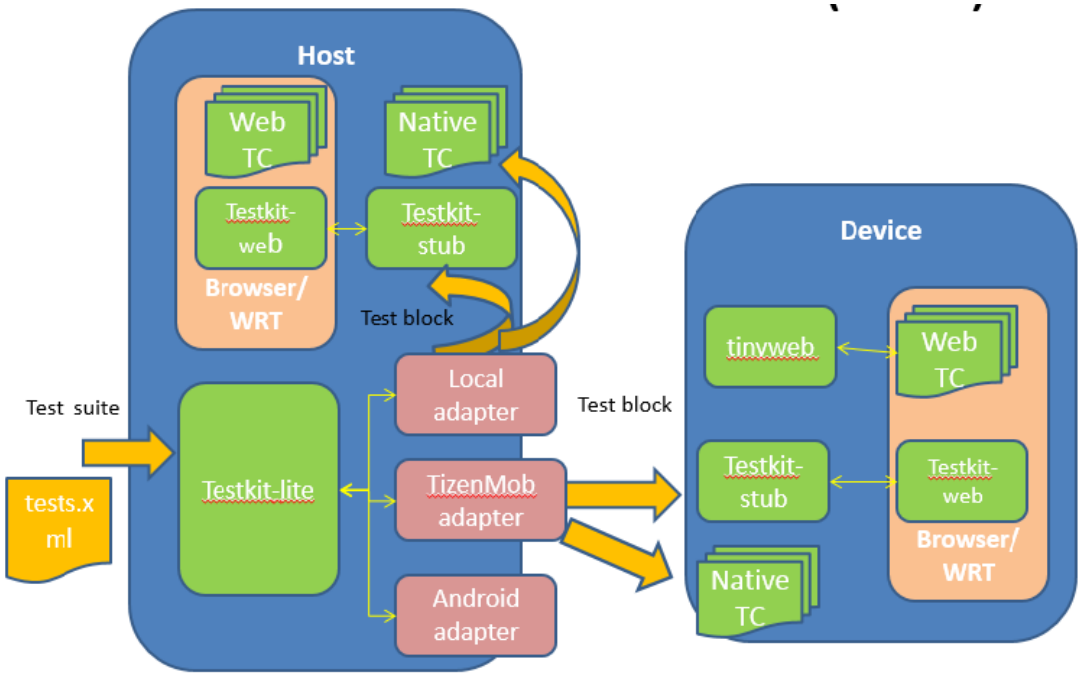
- What the testkit-lite can do
- How to use testkit-lite to run your test cases.
- What result you can get

Testkit-lite is a general test executor. It could be used for executing (manual/automatic) test-case on below target OS.

- Tizen (Mobile, IVI, SDK)
- Android
- Web browser (Chrome, Firefox, Opera, and so on).
- Almost all Linux distribution (Fedora, Ubuntu and so on)

Both ARM and x86 architecture are supported.

# Testkit-lite structure




## Deployment

Testkit-lite is consist of 3 parts

- Testkit-lite

  Testkit-lite is the command-line interface (CLI) of Testkit-lite, which provides comprehensive

options for TCs filtering and execution.

- Adapter

  Adapter is communication module of testkit-lite, which responsible for handling interaction with target device, for different test targets, such as TIZEN device, Android device or localhost target. It provides a set of same application interfaces for Testkit-lite.

- Testkit-stub

  Testkit-stub is a test stub app designed for web test only, runs in daemon mode.

  Generally testkit-stub works as an http server and serves its two users:testkit-lite and test app. It provides a group of web-APIs, and the API transfers data in JSON format via POST/GET http request.

2 kinds of deployment:

- Host-device

  Host-device mode called HD mode for short.

  In this mode, testkit-lite and adapter are deployed and runs on Host-side, while testkit-stub deployed and runs in device-side.

- Host-Host

  Host-Host mode also called Single mode.

  In this mode, testkit-lite, adapter and testkit-stub are all deployed and runs in a same machine, such as all installed in a TIZEN mobile device, or all installed in an Ubuntu Desktop.

## Host-device

testkit-lite and adapter are deployed on Host, currently they are packaged in one package

- For Host with Ubuntu OS, a debian install package provided, use the command-line below to install them:

```
>sudo dpkg -i /patch/to/testkit-lite_<version>.deb
```

- For Host with Fedora OS, a rpm install package provided, use the command-line below to install them:

```
>sudo rpm -i /patch/to/testkit-lite_<version>.rpm
```

Testkit-stub is deployed on device. Please select proper version meets the arch type of target device (armv7l, i686 or x86_64).Take TIZEN device as an instance, the command-line is as below

```
>sdb push /path/to/testkit-stub_<arch> /opt/home/developer/testkit-stub
>sdb shell chmod +x /opt/home/developer/testkit-stub
```

## Host-Host

For Host-Host mode, testkit-lite , Adapter and testkit-stub is deployed on a same machine.
On TIZEN device, we use the rpm install package to deploy testkit-lite and adapter, deploy executable binary file for testkit-stub

```
> sdb root on
> sdb push /path/to/testkit-lite_<version>.rpm /tmp/
> sdb shell rpm -i /tmp/testkit-lite_<version>.rpm
> sdb push /path/to/testkit-stub_<arch> /usr/bin/testkit-stub
> sdb shell chmod +x /usr/bin/testkit-stub
```

On Ubuntu desktop, we use the debian install package to deploy testkit-lite and adapter, deploy executable binary file for testkit-stub

```
> sudo dpkg -i /patch/to/testkit-lite_<version>.deb
> sudo cp /path/to/testkit-stub_<arch> /usr/bin/testkit-stub
> sudo chmod +x /usr/bin/testkit-stub
```

On Fedora desktop, we use the rpm install package to deploy testkit-lite and adapter, deploy executable binary file for testkit-stub

```
> sudo rpm -i /patch/to/testkit-lite_<version>.rpm
> sudo cp /path/to/testkit-stub_<arch> /usr/bin/testkit-stub
> sudo chmod +x /usr/bin/testkit-stub
```

# Quick Start

## A simple native case executed on device

## Test Case

Here is a test case “hello_testkit.sh”

```
#!/bin/bash
echo “Hello testkit”
exit 1
```

## Test Definition

For using testkit-lite to run this case, a test-definition XML is needed as below:

```
<?xml version="1.0" encoding="UTF-8"?>
<test_definition
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="test_definition.xsd">
<suite name="example suite">
    <set name="example set">
        <testcase purpose="hello testkit" type="functional"
status="ready" requirement_ref="A link or description of the
requirement" component="The component this case covered"
execution_type="auto" priority="P1" id="the unique ID of this test
case in test suites">
            <description>
                <pre_condition>Description for the condition which
should be ready before executing this test-case. This description will
be shown when execute manual test case on tct-manager, and will be shown
in CMD when executed as Native auto case </pre_condition>
                <post_condition>Post condition once the test case is
finished. This is not a checking point. It is just a description for
showing as  “pre_condition”</post_condition>
                <steps>
                    <step order="1">
                        <step_desc>Description for test step, the
description will be shown in test report</step_desc>
                        <expected>1</expected>
                    </step>
                </steps>
                <test_script_entry
test_script_expected_result="1">/opt/testkit_example/hello_testkit
.sh</test_script_entry>
            </description>
            <categories>
                <category>platform1</category>
                <category>platform2</category>
            </categories>
        </testcase>
    </set>
</suite>
</test_definition>
```

In this XML, a test case is announced belong to the test suite “example suite” and test set “example set”. Testkit-lite will execute this test case with the “test_script_entry” and compare the result of the script with “test_script_expected_result”.

## Deployment

The script and definition of test case should be put on device folder “/opt/testkit_example/” as announced in the definition.

## Command

Here is the command to execute this test case:

```
>testkit-lite -f device:/opt/testkit_example/tests.xml –o ./result.xml
```

## A simple native case executed on host

The script and definition of test case are same with device-example.
Only need to deploy the script and definition on host folder “/opt/testkit_example/”

## Command

Here is the command to execute this test case:

```
>testkit-lite -f /opt/testkit_example/tests.xml –o ./result.xml    --comm localhost
```

## A simple web case executed on device

Web case is designed to run under the W3C test harness. For more information of test harness, please refer to http://www.w3.org/2008/webapps/wiki/Harness

## Test Case

Here is a “Hello” test case based on test harness.

```
<!DOCTYPE html>
<html>
   <head>
      <title>CSS3 2D Transforms: 2dtransform_property_exist</title>
      <link rel="author" title="Intel" href="http://www.intel.com/" />
      <link rel="help" href="http://www.w3.org/TR/2012/WD-css3-transforms-20120911/" />
      <meta name="flags" content="" />
      <meta name="assert" content="Check if the 2dtransform property exists" />
      <script src="../resources/testharness.js"></script>
      <script src="../resources/testharnessreport.js"></script>
      <script>
      //pre-defined methods
      </script>
    </head>
<body>
            <div id="log"></div>
            <p>Hello Testkit-lite</p>
      <script>
      test(function () {
            assert_true(true);
      }, "True really is true");
      </script>
```

## Test Definition

For executing this example web case in testkit-lite, below test definition should be provided.

```
<?xml version="1.0" encoding="UTF-8"?>
      <?xml-stylesheet type="text/xsl" href="./testcase.xsl"?>
<test_definition>
   <suite launcher="WRTLauncher" name="example-tests" category="Example">
      <set name="exampleSet">
         <testcase component="Hello Test Harness" execution_type="auto" id="example1"
purpose="An example for web case">
            <description>

<test_script_entry>/opt/harness_sample/harness_hello.html</test_script_entry>
            </description>
            </testcase>
      </set>
      </suit>
</test_definition>
```

2 different point with the definition of native case.

- The element “test_script_entry” is the path of the test case html file.
- The attribute “launcher” of “suit” element is "WRTLauncher"

## Deployment

Besides the html and definition file of test case, the test harness file and web-runner (<testkit_lite_HOME>/web/index.html) should also be deployed.
All of this file should be packed as a wgt/xpk file. and install it on device.

```
>testkit-lite -f device:/opt/xpk_folder/tests.xml –o ./result.xml
```

## Command

## A simple web case executed on host

The script and definition of test case are same with device-example.
Only need to deploy the script and definition on host folder “/opt/testkit_example/”

## Command

Here is the command to execute this test case in Chrome:

```
>testkit-lite -f /opt/testkit_example/tests.xml -e "google-chrome --allow-file-access-from-files --disable-web-security --start-maximized --user-data-dir=/home/test/result/data /opt/testkit_example/index.html" --comm localhost
```

# Test-definition XML

More details of the XML definition, please refer to the guide “Test Definition Schema.docx”

# Test-result Collecting

## Supported result

4 kinds of result are supported in the generated XML.
PASS
FAIL
BLOCK
N/A

## How testkit-lite check expecting result for web case

The result of web case is assigned by harness

## How testkit-lite check expecting result for native case

PASS: Once the test case is executed and return code is same with the attribute “test_script_expected_result”. The result is assigned with “PASS”.
FAIL: Once the test case is executed and return code is different with the attribute “test_script_expected_result”. The result is assigned with “FAIL”.
BLOCK: When the test case is launched, and no result is available (time out, crash, and so on). Testkit-lite will assign the result of this test case as “BLOCK”.
N/A: When the test case is failed to be launched. The result will be marked as “N/A”.

# Testkit-lite option list

- Mandatory option

| Option | Description |
|---|---|
| -f device:"<test_descriptor_file>.xml" | Specify one or more test definition xml files as test input.<br>Note: test definition XML schema, refer to the chapters above |

- Optional– Basic options

| Option | Description |
|---|---|
| -e <Web Runtime Environment> | Specify external test app launching command-line.<br>Such as, "WRTLauncher" is for TIZEN webapi test;<br>and "xwalk" is for TIZEN/android crosswalk webapi test<br>Note: Only required for webapi test. |
| -o <test_result_file> | Specify the name of result file.<br>Testkit-Lite locates the result file under "**/opt/testkit/lite/latest/**" by default. |
| --testprefix <location_prefix> | set prefix for test case location.<br>Such as, "**/opt/usr/media/tct**" is for TIZEN webapi test cases. |
| --version | Show version information |
| --deviceid | Specify device serial number to assign a device for execution,<br>Get the first device by default when "--deviceid" omitted. |
| --comm | Specify Adapter communication type,<br>Use "tizenmobile" by default when "--comm" omitted.<br>Note: 'localhost' is for Host-Host mode. |
| --capability | Specify hardware capability file to filter test case |
| --non-active | Disable the ability to set the result of core manual cases from the console |

- Optional – Filter options

| Filter | Description |
|---|---|
| -A | A shortcut of filter auto test cases.<br>execution_type:auto |
| -M | A shortcut of filter manual test cases.<br>execution_type:manual |
| --type | Filter test cases by test case type:<br>• functional_positive<br>• functional_negative<br>• security<br>• performance<br>• reliability<br>• portability<br>• maintainability<br>• compliance<br>• user_experience |
| --priority | Filter test cases by test case priority:<br>• P0<br>• P1<br>• P2 |
| --status | Filter test case by test case status:<br>• ready<br>• approved<br>• designed |
| --suite | Filter test case by test suite name |
| --set | Filter test case by test set name |
| --id | Filter test case by test case id |
| --component | Filter test case by test case component |