

Testkit-Lite Dev Guide

Copyright © 2014 Intel Corporation. All rights reserved. No portions of this document may be reproduced without the written permission of Intel Corporation.

Intel is a trademark of Intel Corporation in the U.S. and/or other countries.

Linux is a registered trademark of Linus Torvalds.

Tizen® is a registered trademark of The Linux Foundation.

ARM is a registered trademark of ARM Holdings Plc.

*Other names and brands may be claimed as the property of others.

Any software source code reprinted in this document is furnished under a software license and may only be used or copied in accordance with the terms of that license.

Contents

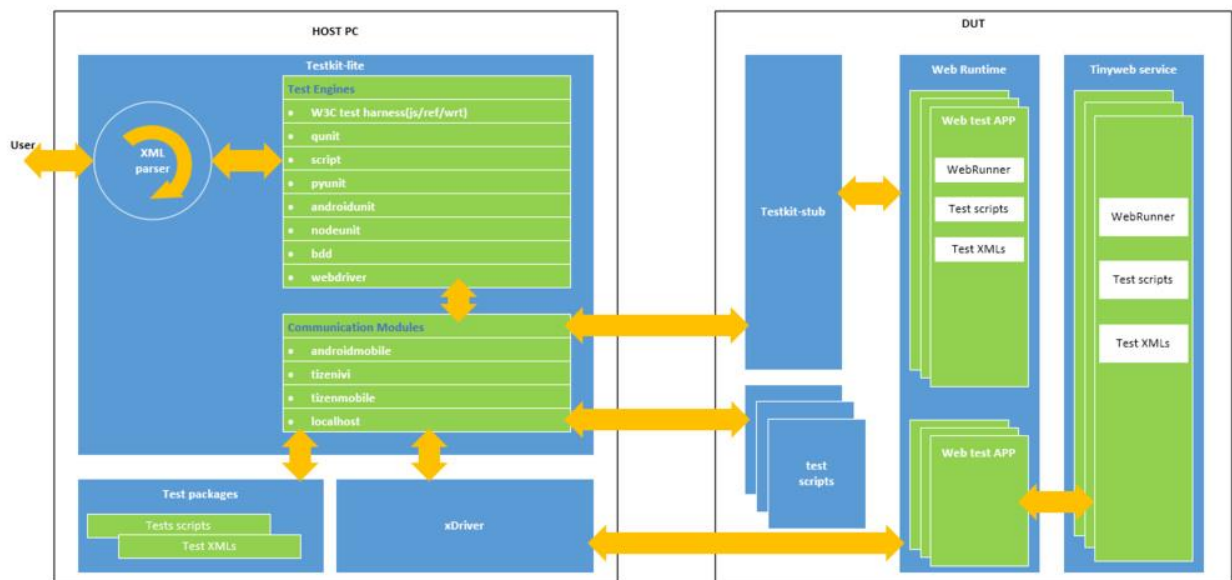
1	Introduction	3
2	Architecture	3
3	Pre-condition	4
4	Build testkit-lite.....	4
5	Build testkit-stub.....	5
6	Build tinyweb	6
7	Pack all-in-one testkit-lite package for release	8

1 Introduction

Testkit-lite is a light-weight testing execution framework, composed by 5 components:

- **testkit-lite** is the command line interface(CLI) of Testkit-lite. Which provides comprehensive options for web/core testing and supports cross platform.
In **TCT**, **CATS** usage, it is usually invoked as a background test runner.
- **test engines** is distribute engine responsible for handling various tests types.
- **com-module** is a common module responsible for handling interaction with target device, such as TIZEN device, Android device or localhost workstation.
- **testkit-stub** is a native process running on test target, which work as proxy between test suite and testkit-lite.
- **xDriver** is a special WebDriver implementation.

2 Architecture



3 Pre-condition

- Host: Ubuntu 12.04 LTS 64bit/32bit
- Pre-install the following tools/libs:
 - ✧ Python (2.7)
 - ✧ Java SDK (oracle)
 - ✧ Android SDK (android-15) & Eclipse
 - ✧ Android NDK (android-ndk-r10e)
 - ✧ Tizen SDK (tizen-sdk-ubuntu64-v2.2.71) & Tizen SDK Image
 - ✧ Git
 - ✧ setuptools, dpkg-dev, debhelper, python-support, rpm, libncurses5-dev
 - ✧ busybox, refer to <http://forum.geekspfone.com/index.php?topic=6135.0>

4 Build testkit-lite

- Downloads testkit-lite source codes from github:.
 - \$ git clone [git@github.com:testkit/testkit-lite.git](https://github.com/testkit/testkit-lite.git)
- Build deb/rpm package on Ubuntu 12.04 LTS Host:
 - \$ cd testkit-lite
 - \$ sudo rm -rf testkitmerge testkit-merge
 - ✧ Generate deb package: testkit-lite_<version>_all.deb
 - \$ sudo python setup.py install
 - \$ sudo dpkg-buildpackage
 - ###PS: Need install setuptools, dpkg-dev, debhelper, python-support firstly. After executed this command, it would build out ../testkit-lite_<version>_all.deb and ../testkit-lite_<version>.tar.gz etc.
 - \$ save generated testkit-lite_<version>_all.deb as testkit-lite-<version>/testkit-lite_<version>_all.deb
 - ✧ Generate rpm package: testkit-lite-<version>-1.noarch.rpm from above generated testkit-lite_<version>.tar.gz
 - \$ cp ../testkit-lite_<version>.tar.gz packaging

```
$ rpmbuild --define 'python_sitelib /usr/lib/python2.7/site-packages' -tb
packaging/testkit-lite-<version>.tar.gz --nodeps --target=noarch
###PS: Need install rpm firstly. After executed this command, it would build out
~/rpmbuild/RPMS/noarch/testkit-lite-<version>-1.noarch.rpm
$ save generated testkit-lite-<version>-1.noarch.rpm as
testkit-lite-<version>/testkit-lite-<version>-1.noarch.rpm
```

5 Build testkit-stub

- Download testkit-stub source codes from github:

```
$ git clone git@github.com:testkit/testkit-stub.git
```

- Make executable binary for Tizen/Ubuntu/Deepin

- ✧ Build executable binary for ARM

```
$ cd testkit-stub/CommandLineBuild
$ /path/to/tizen-sdk/tools/native-make clean
$ /path/to/tizen-sdk/tools/native-make -a armel -t GCC-4.5
#would generate executable testkit-stub under current directory
$ save generated testkit-stub as
testkit-lite-<version>/web-test-utilities/testkit-stub/tizen/arm/testkit-stub
```

- ✧ Build executable binary for IA32

```
$ cd testkit-stub/CommandLineBuild
$ /path/to/tizen-sdk/tools/native-make clean
$ /path/to/tizen-sdk/tools/native-make -a i386 -t GCC-4.5
#would generate executable testkit-stub under current directory
$ save generated testkit-stub as testkit-lite-<version>/web-test-utilities/testkit-
stub/debian(tizen)/ia32/testkit-stub
```

- ✧ Build executable binary for X86_64 on Ubuntu 12.04 LTS 64bit

```
$ cd testkit-stub
$ make
#would generate executable testkit-stub under current directory
```

```
$ save generated testkit-stub as testkit-lite-<version>/web-test-utilities/testkit-
stub/debian(tizen)/x64/testkit-stub
```

- Make Apk package for Android

- ✧ Generate executable binaries of all target APP_ABI in testkit-stub/android/libs folder

```
$ cd testkit-stub/android/jni
$ /path/to/android-ndk-r10e/ndk-build
```

- ✧ Make APK package by Eclipse tool:

```
$ Click File/Import... to import testkit-stub project
$ Click File/Export... to make Apk package
$ save generated testkit-stub apk as testkit-lite-<version>/web-test-utilities/testkit-
stub/android/testkit-stub_all.apk
```

6 Build tinyweb

- Download tinyweb source codes from github:

```
$ git clone git@github.com:testkit/tinyweb.git
```

- Make executable binary for Tizen/Ubuntu/Deepin

- ✧ Build executable binaries for ARM

```
$ config gcc as arm-linux-androideabi-gcc
$ cd tinyweb
$ make
#would generate executable cgi-getcookie,cgi-getfield,tinyweb under current
directory
```

```
$ save generated cgi-getcookie,cgi-getfield,tinyweb as
testkit-lite-<version>/web-test-utilities/tinyweb/tizen/arm/cgi-getcookie,
cgi-getfield,tinyweb
$ cp server.pem as
testkit-lite-<version>/web-test-utilities/tinyweb/tizen/arm/server.pem
```

- ✧ Build executable binaries for IA32 on Ubuntu 12.04 LTS 32 bit

```
$ cd tinyweb
```

```
$ make
```

```
#would generate executable cgi-getcookie,cgi-getfield,tinyweb under current
directory
```

```
$ save generated cgi-getcookie,cgi-getfield,tinyweb as
```

```
testkit-lite-<version>/web-test-utilities/tinyweb/debian(tizen)/ia32/
```

```
cgi-getcookie,cgi-getfield,tinyweb
```

```
$ cp server.pem as
```

```
testkit-lite-<version>/web-test-utilities/tinyweb/tizen/ debian(tizen)/server.pem
```

- ✧ Build executable binaries for X86_64 on Ubuntu 12.04 LTS 64bit

```
$ cd tinyweb
```

```
$ make
```

```
#would generate executable cgi-getcookie,cgi-getfield,tinyweb under current
directory
```

```
$ save generated cgi-getcookie,cgi-getfield,tinyweb as
```

```
testkit-lite-<version>/web-test-utilities/debian(tizen)/tinyweb/x64/
```

```
cgi-getcookie,cgi-getfield,tinyweb
```

```
$ cp server.pem as
```

```
testkit-lite-<version>/web-test-utilities/tinyweb/tizen/x64/server.pem
```

- Make APK package for Android

- ✧ Generate executable binaries of all target APP_ABI in

```
tinyweb/android/native /libs folder
```

```
$ cd tinyweb/android/native/jni
```

```
$ /path/to/android-ndk-r10e/ndk-build
```

```
$ cp -r ../libs ../../assets/system
```

- ✧ Build android busybox according to APP_ABI

```
$ export PATH=/path/to/android-ndk-
```

```
<version>/toolschans/path/to/$CROSS_COMPILER-VERSION/prebuild/[linux-
x86|linux-x86_64]/bin:$PATH
```

```
$ cd tinyweb/third-party/busybox-1.22.1
```

```
$ configure CONFIG_CROSS_COMPILER_PREFIX && CONFIG_SYSROOT &&
CONFIG_EXTRA_CFLAGS in configs/android_ndk_defconfig
```

```
$ make
```

```
$ save generated busybox as
```

```
tinyweb/android/assets/system/libs/<APP_ABI>/busybox
```

✧ Make APK package by Eclipse tool:

```
$ Click File/Import... to import tinyweb project
```

```
$ Click File/Export... to make Apk package
```

```
$ save generated tinyweb apk as testkit-lite-<version>/web-test-
utilities/tinyweb/android/tinyweb_all.apk
```

7 Pack all-in-one testkit-lite package for release

- Copy the following document files in testkit-lite/doc into testkit-lite-<version>/docs folder
 - test_definition_schema.pdf
 - testkit-lite_tutorial.pdf
 - testkit-lite_user_guide.pdf
- Download webruuner repo and copy webrunner folder (exclude webrunner/.git) into testkit-lite-<version> folder


```
$ git clone git@github.com:testkit/tinyweb.git
```
- Execute above 4-6 **Build testkit-lite/testkit-stub/tinyweb**
- Zip testkit-lite-<version> folder as testkit-lite-<version>.tar.gz package


```
$ tar czvf testkit-lite-<version>.tar.gz testkit-lite-<version>
```