Web TCT Tool Packaging Guide

Version 1.0, for Tizen 2.3

Contents

1 Overview

This document is intended for developers who contribute Web TCT development.

You are supposed to have gained the following knowledge:

- Where to download TIZEN SDK 2.3
- Where to download source code of TCT-manager, TCT-shell, TCT-utilities
- Where to download source code of TCT test suites
- Where to download source code of testkit-lite, testkit-stub, tinyweb

2 Web TCT

Web TCT release is provided as an all-in-one tar ball in layout as below.

```
[TCT-<version>-<revision>]/
   ├─ [doc]/
        ├─<Tool Name>-User Guide-<version>.pdf
        ├─Release Notes
   ├─ [package]/
        ├─[common]/
        ├─[wearable]/
        ├─[mobile]/
        ├─[pkg_infos]/
   ├─ [resource]/
        ├─[testkit-stub]/
        ├─[tinyweb]/
        ├─getCap.wgt
        ├─tct-testconfig-<version>.zip
   ├─ [tools]/
        ├─ [manager]/
        ├─ [shell]/
        ├─ [testkitlite]/
        ├─ [sdb]/
        ├─ tct-mgr
        ├─ tct-shell
        ├─ testkit-lite
        ├─ full_pkg_generator
        ├─ tct-config-host.py
        ├─ healthcheck.ini
        └─ tct-config-device.py
```

- TCT-<version>-<revision>: Final name of a Web TCT release.
- Doc:
    - Web-TCT User Guide

  - TCT-manager User Guide
  - TCT-shell User Guide
  - TCT-behavior User Guide
  - TCT-tool-packaging User Guide
  - Release Notes
- Package:
  - [common]: TIZEN common test packages
  - [wearable]: TIZEN wearable specified test packages
  - [mobile]: TIZEN mobile specified test packages
  - [pkg_infos]: contains test packages list file for each TIZEN profile
    For tizen_2.3, we have two profiles: wearable and mobile
- Resource:
  - [testkit-stub]/: contains arm/IA executable binaries of testkit-stub
  - [tinyweb]/: contains arm/IA executable binaries of tinyweb
  - getCap.wgt: a widget used to get capabilities of TIZEN target
  - tct-testconfig: a common test configuration file for web test suites
- Tools:
  - [manager]/: contains the jar and resource of tct-manager
  - [shell]/: executable files and resource of tct-shell
  - [testkitlite]/: executable files and resource of of testkit-lite
  - [sdb]/: TIZEN sdb tool, 32bit/64bit
  - tct-mgr: launcher of tct-manager
  - tct-shell: launcher of tct-shell
  - full-pkg-generator: launcher of package list tool
  - testkit-lite: launcher of testkit-lite
  - tct-config-host.py: deployment script for tools in Host side
  - tct-config-device.py: deployment script for tools in Device side
  - healthcheck.ini: configure file for tct-config-device.py

3 Prerequisites

- One of the following Linux Distribution versions is installed

  - Windows 7 (32-bit)
  - Windows 7 (64-bit)
  - Ubuntu 12.04 (32-bit)
  - Ubuntu 12.04 (64-bit)
  - Ubuntu 12.10 (32-bit)
  - Ubuntu 12.10 (64-bit)

- Oracle JDK7 or Open-JDK7 is installed.
- TIZEN SDK 2.3 is installed (You need to select all components when install it).
- TIZEN Widget Signing Tool.
- Java build tool ant is installed. You can use the command below to install it.

```
$ sudo apt-get install ant
```

- Python 2.7
- Packages: zip

4 How to Package Web TCT

- **Download TCT utilities to prepare layout of the tar ball**
  - Use git clone to get tct-utilities from Tizendev.org

```
- $ git clone ssh://<uid>@review.tizendev.org:29418/test/tct/utilities.git tct-utilities
- $ cd tct-utilities && git checkout  tizen_2.3
```

Layout of a new initialized tct-utilities project as below

```
├─ [tct-utilities]/
    ├─ [doc]/
         ├─<***>-User Guide-<version>.pdf
         ├─Release Notes
    ├─ [package]/
         ├─[common]
         ├─[wearable]
         ├─[mobile]
         ├─[pkg_infos]
    ├─ [resource]/
        ├─[GetCap]/
    ├─ [tools]/
     ├─ [manager]/
        ├─ [shell]/
     ├─ tct-mgr
        ├─ tct-shell
        ├─ testkit-lite
        ├─ full_pkg_generator
        ├─ tct-config-host.py
        ├─ healthcheck.ini
        └─ tct-config-device.py
```

- Update Release Notes according to updates

- **Download TCT test suite source codes and build test packages (zip)**
  - Use git clone to get source code of TCT test suites from Tizendev.org

```
- $ git clone ssh://<uid>@review.tizendev.org:29418/test/tct/webapi.git
- $ git clone ssh://<uid>@review.tizendev.org:29418/test/tct/webuifw.git
- $ git clone ssh://<uid>@review.tizendev.org:29418/test/tct/wrt.git
- $ git clone ssh://<uid>@review.tizendev.org:29418/test/tct/behavior.git
-     $ cd /path/to/webapi && git checkout  tizen_2.3
-     $ cd /path/to/webuifw && git checkout  tizen_2.3
-     $ cd /path/to/wrt && git checkout  tizen_2.3
-     $ cd /path/to/behavior && git checkout  tizen_2.3
```

- Use the packing script in the project to build all test packages (*.zip)

  For detailed steps of test suites packaging, please refer to the document *“**Web_TCT_2.3_TestSuite_Packing_Guide_v1.0.pdf**”*
- Copy test suite packages to sub folders of “package/” according to actual profile

  /path/to/tct-utilities/package/wearable

  /path/to/tct-utilities/package/mobile

● **Copy tct-testconfig-<version>.zip to /path/to/tct-utilities/resource/**

tct-testconfig is already packaged in the previous step, we just need to copy it.

● **Build wgt distribution for getCap from source code**

- Build and sign getCap wgt file

```
-     $ cd /path/to/tct-utilities/resource/GetCap/src
-     $ zip -rq /path/to/tct-utilities/resource/getCap.wgt *
-     $ /path/to/tizen-signing-tool/sign-widget.sh --dist platform /path/to/tct-utilities/resource/getCap.wgt
-     $ rm -rf /path/to/tct-utilities/resource/GetCap/
```

● **Download source code of TCT manager and build executable files**

- Use git clone to get source code of tct-manager

```
-     $ git clone ssh://<uid>@review.tizendev.org:29418/test/tct/manager.git tct-manager
-     $ cd /path/to/tct-manager && git checkout  tizen_2.3
```

- Make a directory src_jaxb and run ant to initialize it

```
-     $ mkdir src_jaxb
-     $ ant
```

- Copy the folder /path/to/tct-manager/dest/manager to /path/to/tct-utilities/tools/

Replace the existed folders or files with same name. The layout should be as below

```
├─ [tools]/
    ├─ [manager]/
        ├─ [doc]/
        ├─ [lib]/
        ├─ [style]/
        └─ [plan]/
```

- **Download source code of TCT-shell and copy it to tools folder**

  - Use git clone to get source code of tct-shell

```
-     $ git clone ssh://<uid>@review.tizendev.org:29418/test/tct/shell.git shell
-     $ cd /path/to/shell && git checkout  tizen_2.3
```

  - Copy the folder /path/to/shell to /path/to/tct-utilities/tools/

  Replace the existed folders or files with same name. The layout should be as below

```
├─ [tools]/
    ├─ [shell]/
        ├─ *.py
        ├─ [tctshell]/
        ├─ [style]/
        └─ [plan]/
```

- **Download source code of testkit-lite and copy it to tools folder**

  - Use git clone to get source code of testkit-lite from Github

```
-     $ git clone git@github.com:testkit/testkit-lite.git  testkitlite
-     $ cd /path/to/testkitlite && git checkout  tizen_2.3
```

  - Copy the folder /path/to/testkitlite to /path/to/tct-utilities/tools/

  The layout should be as below

```
├─ [tools]/
    ├─ [testkitlite]/
```

- **Download source code and build testkit-stub**

  - Download source code of testkit-stub from Github

```
-     $ git clone git@github.com:testkit/testkit-stub.git
-     $ cd /path/to/testkit-stub && git checkout  tizen_2.3
```

  - Build ARM/IA binaries of testkit-stub for Web TCT

```
-     $ cd /path/to/testkit-stub/CommandLineBuild
```

```
### build ARM
-     $ /path/to/tizen-sdk/tools/native-make clean
-     $ /path/to/tizen-sdk/tools/native-make -a armel -t  GCC-4.5
-     $ mkdir arm && cp  testkit-stub arm/
### build IA
-     $ /path/to/tizen-sdk/tools/native-make clean
-     $ /path/to/tizen-sdk/tools/native-make -a i386  -t  GCC-4.5
-     $ mkdir ia && cp testkit-stub ia/
### make testkit-stub folder
-     $ mkdir  /path/to/tct-utilities/resource/testkit-stub
```

- Copy arm and ia binaries to /path/to/tct-utilities/resource/testkit-stub according to layout

```
 ├─ [resource]/
     ├─ [testkit-stub]/
          ├─ [arm]
              └─testkit-stub
          ├─ [ia]
               └─testkit-stub
```

- **Download source code and build tinyweb**
  - Download source code of tinyweb from Github

```
-     $ git clone git@github.com:testkit/tinyweb.git
-     $ cd /path/to/tinyweb && git checkout  tizen_2.3
```

- Build ARM/IA binaries of testkit-stub for Web TCT

```
-     $ cd /path/to/tinyweb/CommandLineBuild
### build ARM
-     $ /path/to/tizen-sdk/tools/native-make clean
-     $ /path/to/tizen-sdk/tools/native-make -a armel -t  GCC-4.5
-     $ mkdir arm
-     $ cp tinyweb  arm/
-     $ cp cgi-getcookie arm/
-     $ cp cgi-getfield arm/
-     $ cp server.pem arm/
-     $ cp *.so arm/
### build IA
-     $ /path/to/tizen-sdk/tools/native-make clean
-     $ /path/to/tizen-sdk/tools/native-make -a i386  -t  GCC-4.5
-     $ mkdir ia
-     $ cp tinyweb  ia/
-     $ cp cgi-getcookie ia/
-     $ cp cgi-getfield ia/
-     $ cp server.pem ia/
-     $ cp *.so ia/
### make tinyweb folder
-     $ mkdir  /path/to/tct-utilities/resource/tinyweb
```

- Copy arm and ia binaries to /path/to/tct-utilities/resource/tinyweb according to layout below

```
├─ [resource]/
    ├─ [tinyweb]/
         ├─ [arm]
              ├─tinyweb
              ├─cgi-getcookie
              ├─cgi-getfield
              ├─libmongoose.so
              ├─echo.so
              └─server.pem
         ├─ [ia]
              ├─tinyweb
              ├─cgi-getcookie
              ├─cgi-getfield
              ├─libmongoose.so
              ├─echo.so
              └─server.pem
```

- **Download TIZEN sdb tool**
  - Download sdb 32bit version and sdb 64bit version from tizendev.org release
  - Unzip the package and copy sdb to /path/to/tct-utilities/tools/ according to layout

```
├─ [tools]/
     ├─ [sdb]/
          ├─sdb-32
          └─sdb-64
```

- **Packaging the tar ball of Web TCT**
  - Rename the /path/to/tct-utilities/ to /path/to/TCT-<version>-<revision>
  - Compress the TCT-<version>-<revision> to tar ball for release

Appendix-1 Package List File Sample

The package list xml file contains all the test suite package information

```
<?xml version="1.0" encoding="UTF-8"?>
<ns3:testplan xmlns:ns3="http://www.example.org/plan/" xmlns="">
  <suite name="tct-webdatabase-w3c-tests" launcher="WRTLauncher" category="W3C/HTML5 APIs">
    <auto_tcn>50</auto_tcn>
    <manual_tcn>0</manual_tcn>
    <total_tcn>50</total_tcn>
    <pkg_name>tct-webdatabase-w3c-tests-2.3-1.zip</pkg_name>
  </suite>
  <suite name="tct-jsenhance-html5-tests" launcher="WRTLauncher" category="W3C/HTML5 APIs">
    <auto_tcn>6</auto_tcn>
    <manual_tcn>0</manual_tcn>
    <total_tcn>6</total_tcn>
    <pkg_name>tct-jsenhance-html5-tests-2.3-1.zip</pkg_name>
  </suite>
  <suite name="tct-multicolumn-css3-tests" launcher="WRTLauncher" category="W3C/HTML5 APIs">
    <auto_tcn>141</auto_tcn>
    <manual_tcn>18</manual_tcn>
    <total_tcn>159</total_tcn>
    <pkg_name>tct-multicolumn-css3-tests-2.3-1.zip</pkg_name>
  </suite>
  <suite name="tct-messaging-mms-tizen-tests" launcher="WRTLauncher" category="Tizen Web Device APIs">
    <auto_tcn>122</auto_tcn>
```

```
            <manual_tcn>0</manual_tcn>

            <total_tcn>122</total_tcn>

            <pkg_name>tct-messaging-mms-tizen-tests-2.3-1.zip</pkg_name>

          </suite>

          <suite name="tct-workers-w3c-tests" launcher="WRTLauncher"
         category="W3C/HTML5 APIs">

            <auto_tcn>85</auto_tcn>

            <manual_tcn>1</manual_tcn>

            <total_tcn>86</total_tcn>

            <pkg_name>tct-workers-w3c-tests-2.3-1.zip</pkg_name>

          </suite>

          <suite name="tct-namespace-tizen-tests" launcher="WRTLauncher"
         category="Tizen Web Device APIs">

            <auto_tcn>25</auto_tcn>

            <manual_tcn>0</manual_tcn>

            <total_tcn>25</total_tcn>

            <pkg_name>tct-namespace-tizen-tests-2.3-1.zip</pkg_name>

 </suite>




</ns3:testplan>
```

Appendix-2 Test Plan File Samples

Full_test_mgr.xml is a test plan which contains all suites, both auto and manual tests.

```
<?xml version="1.0" encoding="UTF-8"?>
<ns3:testplan xmlns:ns3="http://www.example.org/plan/" xmlns=""
name="Full_test">
  <execute_type>All</execute_type>
  <suite name="tct-webdatabase-w3c-tests" launcher="WRTLauncher"
category="W3C/HTML5 APIs">
    <auto_tcn>50</auto_tcn>
    <manual_tcn>0</manual_tcn>
    <total_tcn>50</total_tcn>
    <pkg_name>tct-webdatabase-w3c-tests-2.3-1.zip</pkg_name>
  </suite>
  <suite name="tct-jsenhance-html5-tests" launcher="WRTLauncher"
category="W3C/HTML5 APIs">
    <auto_tcn>6</auto_tcn>
    <manual_tcn>0</manual_tcn>
    <total_tcn>6</total_tcn>
    <pkg_name>tct-jsenhance-html5-tests-2.3-1.zip</pkg_name>
  </suite>
  <suite name="tct-messaging-mms-tizen-tests" launcher="WRTLauncher"
category="Tizen Web Device APIs">
    <auto_tcn>122</auto_tcn>
    <manual_tcn>0</manual_tcn>
    <total_tcn>122</total_tcn>
    <pkg_name>tct-messaging-mms-tizen-tests-2.3-1.zip</pkg_name>
  </suite>
  <suite name="tct-multicolumn-css3-tests" launcher="WRTLauncher"
category="W3C/HTML5 APIs">
```

```
    <auto_tcn>141</auto_tcn>
    <manual_tcn>18</manual_tcn>
    <total_tcn>159</total_tcn>
    <pkg_name>tct-multicolumn-css3-tests-2.3-1.zip</pkg_name>
  </suite>
  <suite name="tct-workers-w3c-tests" launcher="WRTLauncher"
  category="W3C/HTML5 APIs">
    <auto_tcn>85</auto_tcn>
    <manual_tcn>1</manual_tcn>
    <total_tcn>86</total_tcn>
    <pkg_name>tct-workers-w3c-tests-2.3-1.zip</pkg_name>
  </suite>
</ns3:testplan>
```

Full_auto_shell.xml is a test plan which only contains the auto suites.

```
<ns3:testplan name="Full_auto_shell" xmlns=""
xmlns:ns3="http://www.example.org/plan/">
  <execute_type>Auto</execute_type>
  <suite launcher="WRTLauncher" name="tct-animationtiming-w3c-tests">
    <auto_tcn>11</auto_tcn>
    <manual_tcn>7</manual_tcn>
    <total_tcn>18</total_tcn>
    <pkg_name>tct-animationtiming-w3c-tests-2.3-1.zip</pkg_name>
  </suite>
  <suite launcher="WRTLauncher" name="tct-batterystatus-w3c-tests">
    <auto_tcn>23</auto_tcn>
    <manual_tcn>0</manual_tcn>
    <total_tcn>23</total_tcn>
    <pkg_name>tct-batterystatus-w3c-tests-2.3-1.zip</pkg_name>
  </suite>
```

```
            <suite launcher="WRTLauncher" name="tct-audio-html5-tests">
             <auto_tcn>171</auto_tcn>
             <manual_tcn>1</manual_tcn>
             <total_tcn>172</total_tcn>
             <pkg_name>tct-audio-html5-tests-2.3-1.zip</pkg_name>
            </suite>
            <suite launcher="WRTLauncher" name="tct-ext02-wrt-tests">
             <auto_tcn>7</auto_tcn>
             <manual_tcn>0</manual_tcn>
             <total_tcn>7</total_tcn>
             <pkg_name>tct-ext02-wrt-tests-2.3-1.zip</pkg_name>
            </suite>
</ns3:testplan>
```