

# Web TCT Shell User Guide

Version 1.0, for Tizen 2.2.1

Copyright © 2013 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	Overview .....	3
3	Terminology .....	3
4	Prerequisites .....	4
5	Installing Web TCT Shell .....	4
6	Web TCT Shell Options .....	5
6.1	Mandatory Options .....	5
6.2	Optional Options.....	6
7	Typical Usage.....	7
7.1	Generating a Test Plan .....	7
7.2	Running Test.....	7
8	View Report.....	8

# 1 Introduction

---

This document provides comprehensive information about Web TCT Shell, including the following: Overview, Installation Instructions, Command Option Descriptions, and Typical Usage etc.

# 2 Overview

---

Web TCT Shell is a lightweight console tool and serves as a major component of the Tizen Compliance Tests (TCT) tool set. By providing an alternative way to execute the TCT tools with Testkit-lite, Web TCT Shell allows users to:

- Run test packages
- Run test plans
- Generate test reports
- Rerun failed tests
- Show test resources, such as plans, test results, and connected devices

# 3 Terminology

---

**Test Package:** .zip file that contains test cases.

**Test Plan:** .xml file that contains a series of test suites.

**Test Report:** .html file readable for users in the test result package.

**Test Result:** .xml file that contains the result information of a test execution.

## 4 Prerequisites

---

Make sure these items are in place before starting:

- One of these Linux distribution versions is installed:
  - Ubuntu 12.04 (32-bits)
  - Ubuntu 12.04 (64-bits)
  - Ubuntu 12.10 (32-bits)
  - Ubuntu 12.10 (64-bits)
- Web TCT Shell depends on Python and python-support ( 0.90.0 or later) is installed.
- Tizen capable devices with the latest the Tizen implementation (for example, M0 and Lunchbox) are available.
- The sdb software is correctly installed.
- There is a USB connection between the host and the target device.
- Enable “USB debugging” option in settings on target device
- Ensure the “unzip” command-line tool installed on target device.

## 5 Installing Web TCT Shell

---

To install Web TCT Shell, perform the following procedure:

**Note:** Make sure the installation environment is ready before starting. For detailed instructions, refer to the [“Installing the TCT Tools”](#) section in the *Web TCT User Guide*.

1. Switch to the TCT directory, where the TCT package is decompressed, by executing this command:

```
$cd <TCT_Directory>/tools
```

2. Install Web TCT Shell by using one of these methods, as appropriate:

- Install Web TCT Shell independently, using the .deb package:

```
$sudo dpkg -i tct-shell_<Version>.deb
```

- Install the entire TCT tool set:

```
$ sudo ./tct-config-host.sh  
$ ./tct-config-device.sh
```

## 6 Web TCT Shell Options

---

### 6.1 Mandatory Options

#### Mandatory options

Option	Description
--testplan	Specify testplan.xml as the test plan.
--test	Specify testing suites. If more than one suite is provided, list them all and separate them with whitespace.
--rerun-fail	Rerun all failed test cases, according to the specified XML.

## 6.2 Optional Options

Optional options	
Option	Description
<code>--output=<i>resultfile</i></code>	Specify the output file for the resulting XML output. If more than one test xml file is provided, results will be merged into this output file.
<code>--version</code>	Show version information.
<code>--skip-iu</code>	Do not install and uninstall suite packages during a test. Install and uninstall them manually.
<code>--all-suites</code>	Show all available test-suites in the local repository. Its location is defined in the configuration file named CONFIG, and its path is:/opt/tct/shell/CONFIG.
<code>--deviceid</code>	Set sdb device serial information.
<code>--plan-list</code>	List all existing plans in the plan folder. The plan folder is defined in the configuration file named CONFIG, and its path is:/opt/tct/shell/.
<code>--result-list</code>	List all history results in the result folder. The result folder is defined in the configuration file named CONFIG, and its path is:/opt/tct/shell/ CONFIG.
<code>--device-list</code>	List all connected devices.
<code>--all</code>	All test cases will be executed.
<code>--manual</code>	Only manual test cases will be executed.
<code>--id</code>	Specify the ID of a test case to run.
<code>-h, --help</code>	Show this help message and exit.

## 7 Typical Usage

### 7.1 Generating a Test Plan

Test plans	
Suites to include	Command
Include all suites in the local repository	<code>\$ tct-plan-generator -o &lt;somewhere&gt;/testplan.xml</code>
Include all suites in the special repository	<code>\$ tct-plan-generator -o &lt;somewhere&gt;/testplan.xml -r &lt;somewhere&gt;/repository_folder</code>
Include the suites in the special repository where the name matches a specific regular expression	<code>\$ tct-plan-generator -o &lt;somewhere&gt;/testplan.xml -r &lt;somewhere&gt;/repository_folder --match '&lt;regex&gt;'</code>
Include the suites in the special repository where the name matches a specific regular expression, and exclude any file where the name matches another regular expression	<code>\$ tct-plan-generator -o &lt;somewhere&gt;/testplan.xml -r &lt;somewhere&gt;/repository_folder --match '&lt;regex&gt;' --unmatch '&lt;unmatch_regex&gt;'</code> For example, to exclude the Testkit-stub package: <code>\$ tct-plan-generator -o &lt;somewhere&gt;/testplan.xml -r &lt;somewhere&gt;/repository_folder --match '*.rpm' --unmatch 'stub'</code>

### 7.2 Running Test

Running tests	
Function	Command
Show the help information	<code>\$ tct-shell --help</code>
Show the plan list	<code>\$ tct-shell --plan-list</code> Test plans are placed in the '/opt/tct/shell/plan' folder, by default. Edit the '/opt/tct/shell/CONFIG' file to change it, if necessary: <code>TCT_PLAN_FOLDER = &lt;path-to-plans&gt;</code>
Show the package list	<code>\$ tct-shell --all-suites</code> The default package repository is in the /home/package folder. Edit '/opt/tct/shell/CONFIG' file to change it, if necessary: <code>TEST_SUITE_DIR = &lt;path-to-suites&gt;</code>

Running tests	
Function	Command
Show the test result list	<code>\$ tct-shell --result-list</code> Each test result is generated as a separate folder in the /opt/tct/shell/result folder.
Show the device list:	<code>\$ tct-shell --device-list</code>
Run a test plan	<code>\$ tct-shell --testplan '&lt;somewhere&gt;/testplan.xml'</code>
Run some test packages	<code>\$ tct-shell --test 'package1 package2 ... packageN'</code>
Run a test case by its ID:	<code>\$ tct-shell --test 'package' --id &lt;case-id&gt;</code>
Rerun all failed test cases	<code>\$ tct-shell --rerun-fail '&lt;somewhere&gt;/test-result.xml'</code>
Install and uninstall suite packages manually	Do not install and uninstall suite packages during a test (they need to be installed and uninstalled manually): <code>\$ tct-shell --testplan '&lt;somewhere&gt;/testplan.xml' --skip-iu</code>
Run a test package on a specific device:	<code>\$ tct-shell --testplan '&lt;somewhere&gt;/testplan.xml' --deviceid&lt;device-id&gt;</code>
Run all test cases, including automatic and manual cases	<code>\$ tct-shell --testplan '&lt;somewhere&gt;/testplan.xml' --all</code>
Run manual test cases	<code>\$ tct-shell --testplan '&lt;somewhere&gt;/testplan.xml' --manual</code>
Specify the output file for test results	<code>\$ tct-shell --testplan '&lt;somewhere&gt;/testplan.xml' -o &lt;somewhere&gt;/test-result.xml</code>

## 8 View Report

Test result is placed in /opt/tct/shell/result, in a new folder, with the test date-time string as the name. To view the result, navigate to the test result folder and open the summary.xml file with the web browser. This is a sample file:



### TCT Report

Test Summary		Device Information	
Test Plan Name	Full_test	Host Device	Linux-3.5.0-17-generic-i686-with-Ubuntu-12.10-quantal
Build ID	Tizen_Ref.Device-PQ_20131022.1656	Manufacturer	samsung
Test Total	994	Device Model	RD-PQ
Test Passed	981	Device ID	4d700e66263000
Test Failed	1	Screen Size	58mm x 103mm
Test Blocked	12	Resolution	720 x 1280
Test Not Executed	0		
Time	2013-10-25_10_18_31 ~ 2013-10-25_10_24_02		

### [Device Capability](#)

#### Test Summary by Suite

Suite	Total	Passed	Failed	Blocked	Not Executed	Ratio
<a href="#">tct-2dttransforms-css3-tests</a>	34	34	0	0	0	100.00%
<a href="#">tct-appcontrol-tizen-tests</a>	5	5	0	0	0	100.00%

Click on the suite name in the Test Summary by Suite table to see details.

This is an example of the test suite's details:

## Suite Test Results

[Show all](#)
[Show only failed](#)
[Show only blocked](#)
[Show only not executed](#)
[Summar](#)

### Test Suite: tct-capability-tests (All)

Case_ID	Purpose	Result	Stdout
<b>Test Set: capability</b>			
<a href="#">caps_accelerometer</a>	Check if accelerometer is supported on the device	FAIL	[Message]###Test Start###Capability Test: caps_accelerometer###Test End#####Error2 Start###assert_true: accelerometer capability is mismatched expected true got false(stack: assert@file:///opt/usr/apps/api0capab0/res/wgt/opt/tct-capability-tests/resources/testharness.js:2022 assert_true@file:///opt/usr/apps/api0capab0/res/wgt/opt/tct-capability-tests/resources/testharness.js:615 check_capability@file:///opt/usr/apps/api0capab0/res/wgt/opt/tct-capability-tests/capability/support/caps.js:329 file:///opt/usr/apps/api0capab0/res/wgt/opt/tct-capability-tests/capability/caps_accelerometer.html:49 step@file:///opt/usr/apps/api0capab0/res/wgt/opt/tct-capability-tests/resources/testharness.js:1121 test@file:///opt/usr/apps/api0capab0/res/wgt/opt/tct-capability-tests/resources/testharness.js:414 global code@file:///opt/usr/apps/api0capab0/res/wgt/opt/tct-capability-tests/capability/caps_accelerometer.html:50)###Error2 End###
<a href="#">caps_barometer</a>	Check if barometer is supported on the device	PASS	[Message]
<a href="#">caps_bluetooth</a>	Check if bluetooth is supported on the device	FAIL	[Message]###Test Start###Capability Test: caps_bluetooth###Test End#####Error2 Start###assert_true: bluetooth capability is mismatched expected true got false(stack: assert@file:///opt/usr/apps/api0capab0/res/wgt/opt/tct-capability-tests/resources/testharness.js:2022 assert_true@file:///opt/usr/apps/api0capab0/res/wgt/opt/tct-capability-tests/resources/testharness.js:615 check_capability@file:///opt/usr/apps/api0capab0/res/wgt/opt/tct-capability-tests/capability/support/caps.js:329 file:///opt/usr/apps/api0capab0/res/wgt/opt/tct-capability-tests/capability/caps_bluetooth.html:52 step@file:///opt/usr/apps/api0capab0/res/wgt/opt/tct-capability-tests/resources/testharness.js:1121 test@file:///opt/usr/apps/api0capab0/res/wgt/opt/tct-capability-tests/resources/testharness.js:414 global code@file:///opt/usr/apps/api0capab0/res/wgt/opt/tct-capability-tests/capability/caps_bluetooth.html:53)###Error2 End###

Click on the corresponding link, at the top of the page, to filter the failed, blocked, or not executed test cases.

This is an example of the filtered information:

## Suite Test Results

[Show all](#)
[Show only failed](#)
[Show only blocked](#)
[Show only not executed](#)
[Summary](#)

### Test Suite: tct-capability-tests (Failed only)

Case_ID	Purpose	Result	Stdout
Test Set: capability			
<a href="#">caps_accelerometer</a>	Check if accelerometer is supported on the device	FAIL	<pre>[Message]###Test Start###Capability Test: caps_accelerometer###Test End#####Error2 Start###assert_true: accelerometer capability is mismatched expected true got false(stack: assert@file:///opt/usr/apps /api0capab0/res/wgt/opt/tct-capability-tests/resources /testharness.js:2022 assert_true@file:///opt/usr/apps /api0capab0/res/wgt/opt/tct-capability-tests/resources /testharness.js:615 check_capability@file:///opt/usr/apps /api0capab0/res/wgt/opt/tct-capability-tests/capability/support /caps.js:329 file:///opt/usr/apps/api0capab0/res/wgt/opt/tct- capability-tests/capability/caps_accelerometer.html:49 step@file:///opt/usr/apps/api0capab0/res/wgt/opt/tct-capability- tests/resources/testharness.js:1121 test@file:///opt/usr/apps /api0capab0/res/wgt/opt/tct-capability-tests/resources /testharness.js:414 global code@file:///opt/usr/apps /api0capab0/res/wgt/opt/tct-capability-tests/capability /caps_accelerometer.html:50)###Error2 End###</pre>
<a href="#">caps_bluetooth</a>	Check if bluetooth is supported on the device	FAIL	<pre>[Message]###Test Start###Capability Test: caps_bluetooth###Test End#####Error2 Start###assert_true: bluetooth capability is mismatched expected true got false(stack: assert@file:///opt/usr/apps/api0capab0/res/wgt /opt/tct-capability-tests/resources/testharness.js:2022 assert_true@file:///opt/usr/apps/api0capab0/res/wgt/opt/tct- capability-tests/resources/testharness.js:615 check_capability@file:///opt/usr/apps/api0capab0/res/wgt /opt/tct-capability-tests/capability/support/caps.js:329 file:///opt/usr/apps/api0capab0/res/wgt/opt/tct-capability- tests/capability/caps_bluetooth.html:52 step@file:///opt /usr/apps/api0capab0/res/wgt/opt/tct-capability-tests/resources /testharness.js:1121 test@file:///opt/usr/apps/api0capab0 /res/wgt/opt/tct-capability-tests/resources/testharness.js:414 global code@file:///opt/usr/apps/api0capab0/res/wgt/opt/tct- capability-tests/capability/caps_bluetooth.html:53)###Error2 End###</pre>