Web TCT Tester User Guide

Version 1.0, for Tizen 3.0

Contents

1 Introduction 3
1.1 What is Web TCT? 3
1.2 How does Web TCT Work? 3
2 Test Environment Setup 4
2.1 Symbols and Abbreviations 4
2.2 Hardware Requirements 4
2.3 Software Requirements 4
2.4 The Installation Procedure 5
2.5 Installation Web TCT Test Tool 6
2.6 Installation Web TCT Behavior Test Tool 6
3 How to Execute TCT 7
3.1 Using Web TCT Manager 7
3.2 Using Web TCT Shell 15
3.3 Using the Web TCT Behavior Test Tool 17
A Appendix 19

1 Introduction

This document provides comprehensive information about Web TCT Test Set, including the following: Overview, Installation and Usage, Troubleshooting and Known Issue etc.

1.1 What is Web TCT?

TCT is short for the Tizen Compliance Tests, which validates platform compatibility for Tizen. Web TCT consists of Web TCT Manager (UI tool), Web TCT Shell (console tool), Testkit-lite (backend test runner), and Web TCT Behavior Test Tool (device behavior checker).

1.2 How does Web TCT Work?

Web TCT has four main components:

a. **Web TCT Manager** is a java GUI tool that runs on the host machine, allow users to create a test execution plan, trigger the test execution, and view the test report. By supporting both automated and manual web API testing, Web TCT Manager makes it much easier for users to conduct TCT tests and enter hardware capability information.

b. **Web TCT Behavior Test Tool** is a device behavior checker from a user perspective. It can be manually launched and operated by clicking the corresponding thumbnail icon on target devices.

c. **Web TCT Shell** is a lightweight console tool that runs on the host machine, allowing users to debug single failed case, or trigger TCT testing with an existing test plan by specifying a test suite list or test case ID. Test suites are executed on target devices under the management of Testkit-lite.

d. **Testkit-lite** is a back-end test runner that communicates with Testkit-stub through the Smart Development Bridge (SDB).

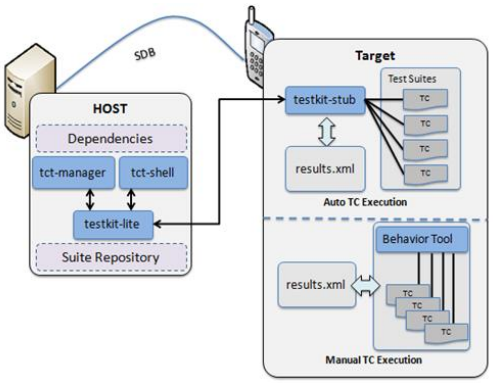

Figure 1-1: Web TCT Workflow

2 Test Environment Setup

2.1 Symbols and Abbreviations

TC - Test Case

TCT - Tizen Compliance Test

SDB - Smart Development Bridge

<name> - Mandatory argument

[name] - Optional argument

$ (in shell command) - Indicates the beginning of a command

(in shell command) – In long commands, the backslash character ensures that newline character is ignored (if you join consecutive lines, please remove unnecessary backslashes)

2.2 Hardware Requirements

Make sure these items in place are ready before starting:

a. PC or Laptop that will work as host on which TCT-Manager will be installed

b. Tizen device that will work as target on which TCs will be executed

c. USB Cable for connecting device to host

2.3 Software Requirements

a. Install 32 or 64 bit Ubuntu OS.

b. Install JDK 1.6 or newer version on Linux PC.

c. Install Tizen 3.0 SDK on Linux PC for SDB connection.

d. These packages should be installed before installing TCT-Manager

```
$ sudo apt-get install rpm2cpio
$ sudo apt-get install tree
$ sudo apt-get install python-pip
$ sudo apt-get install python-support
```

```
$ sudo apt-get install python-requests
$ sudo apt-get install python-setuptools
```

e. libudev1 or libudev-dev package should be installed for SDB.

First find the library 'libudev' installation location using command:~$ cd /lib/

```
$ find . –type f – name "libudev*"
```

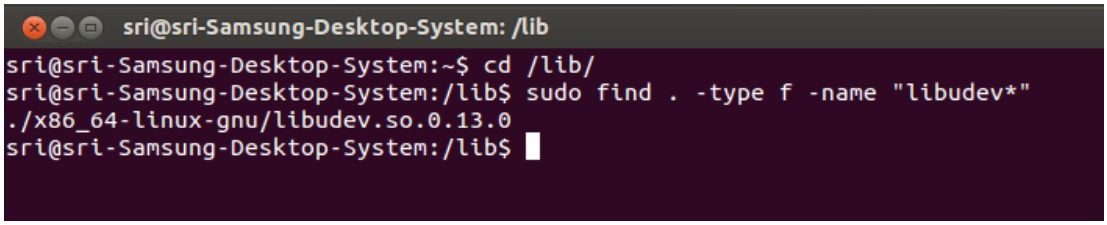

Figure 2-1: Getting Location of libudev

If the package is not properly linked, use the following command:

```
$ sudo ln –s /lib/<installation-folder>/libudev.so.<version> /lib/<installation-folder>/libudev.so.0
e.g. $ sudo ln –s /lib/i386-linux-gnu/libudev.so.0.13.0 /lib/i386-linux-gnu/libudev.so.0
```

2.4 The Installation Procedure

a. Download Web TCT release from http://download.tizen.org/tct/ to your host machine.

b. Reboot your device to make sure the device environment is clean.

c. To burn the new Tizen image to the target device (refer to https://source.tizen.org/documentation/reference/flash-device) and make sure the host machine is well connected to the target device through USB.

d. The device need enable 'USB debugging' in setting. If you already have SDB installed on your host machine, you can check the device connected with sdb command:

```
$ sdb devices
```

The key packages of Web TCT on Host: Web TCT Manager, Web TCT Shell, Testkit-lite.

The key packages of Web TCT on Device: Testkit-stub, Tinyweb.

2.5 Installation Web TCT Test Tool

Un-compress Web TCT tar ball to local path on Ubuntu Host.

You can get help information of the config script firstly.

```
$ cd /path/to/<TCT_pkg_folder>/<TCT_pkg_folder>/tools
$ python ./tct-config-host.py -h
$ python ./tct-config-device.py -h
```

a. Deployment on Host

```
$ cd /path/to/<TCT_pkg_folder>
$ sudo python ./tct-setup.py
```

b. Deployment on Device

```
$ cd /opt/tct/tizen_web_3.0/scripts
$ python ./tct-config-device.py
```

2.6 Installation Web TCT Behavior Test Tool

The Web TCT Behavior Test Tool widget is installed on the target device after run tct-config-device.py with option "--bhtest **<device_profile>**", for instance we can install behavior test tool in wearable device as below

a. On Windows Host

```
$ cd <path_to>\<TCT_pkg_folder>\<TCT_pkg_folder>\tools
$ c:\Python27\python.exe  tct-config-device.py --bhtest wearable
```

b. On Ubuntu Host

```
$ cd /path/to/<TCT_pkg_folder>/<TCT_pkg_folder>/tools
$  python ./tct-config-device.py --bhtest wearable
```

3 How to Execute TCT

3.1 Using Web TCT Manager

3.1.1 Launching Web TCT Manager on Host Machine

Launch the Web TCT Manager by shell command:

On Ubuntu Host:

```
$ tct-mgr
```

When the below screen is shown, Web TCT Manager is launched successfully as shown in Figure 3-1.

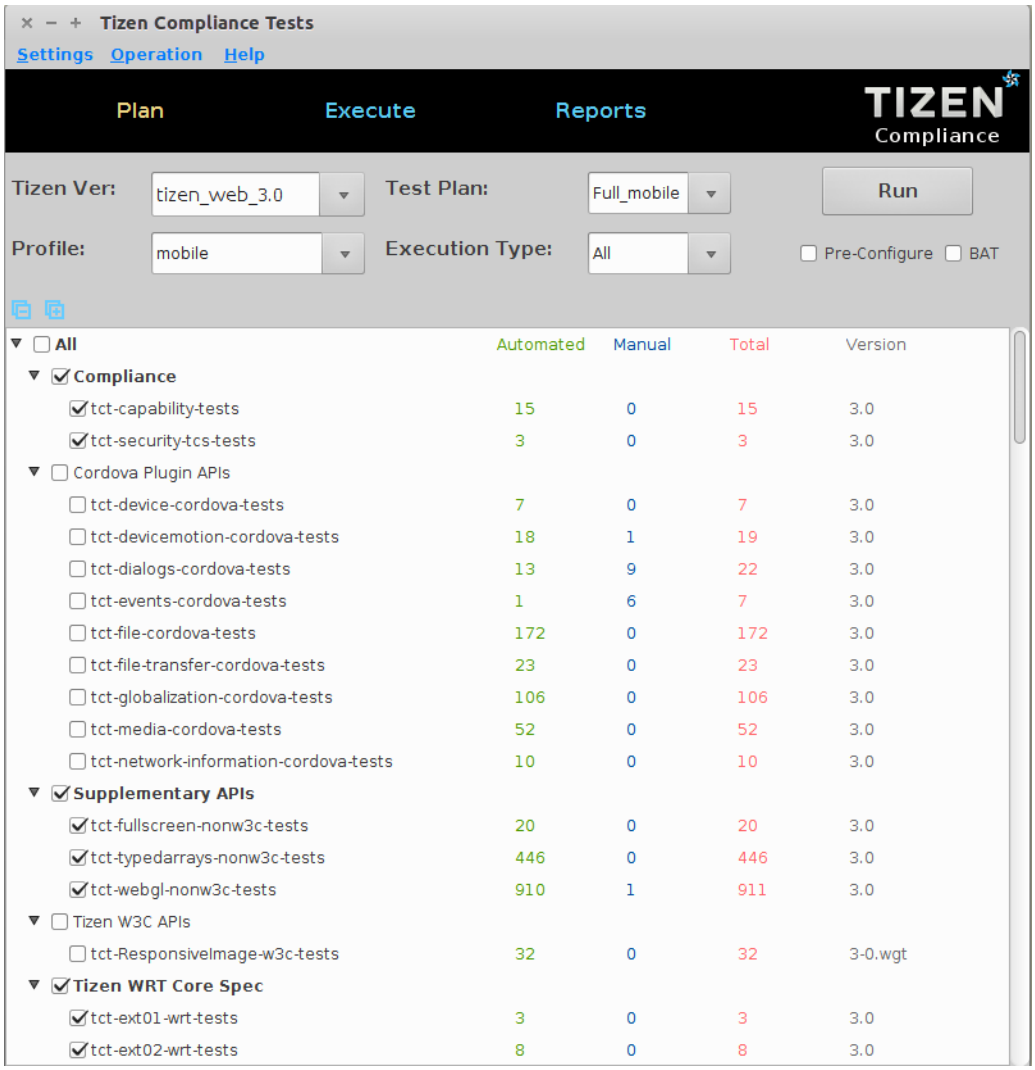

Figure 3-1: Web TCT Manager Plan Page

3.1.2 Creating or Selecting a Test Plan

On the Plan page, select a test profile to switch the test plan set supported for different profiles. For instance, the “Full_mobile” is a test plan includes all mobile test suites. Select some packages and click **Run** button.

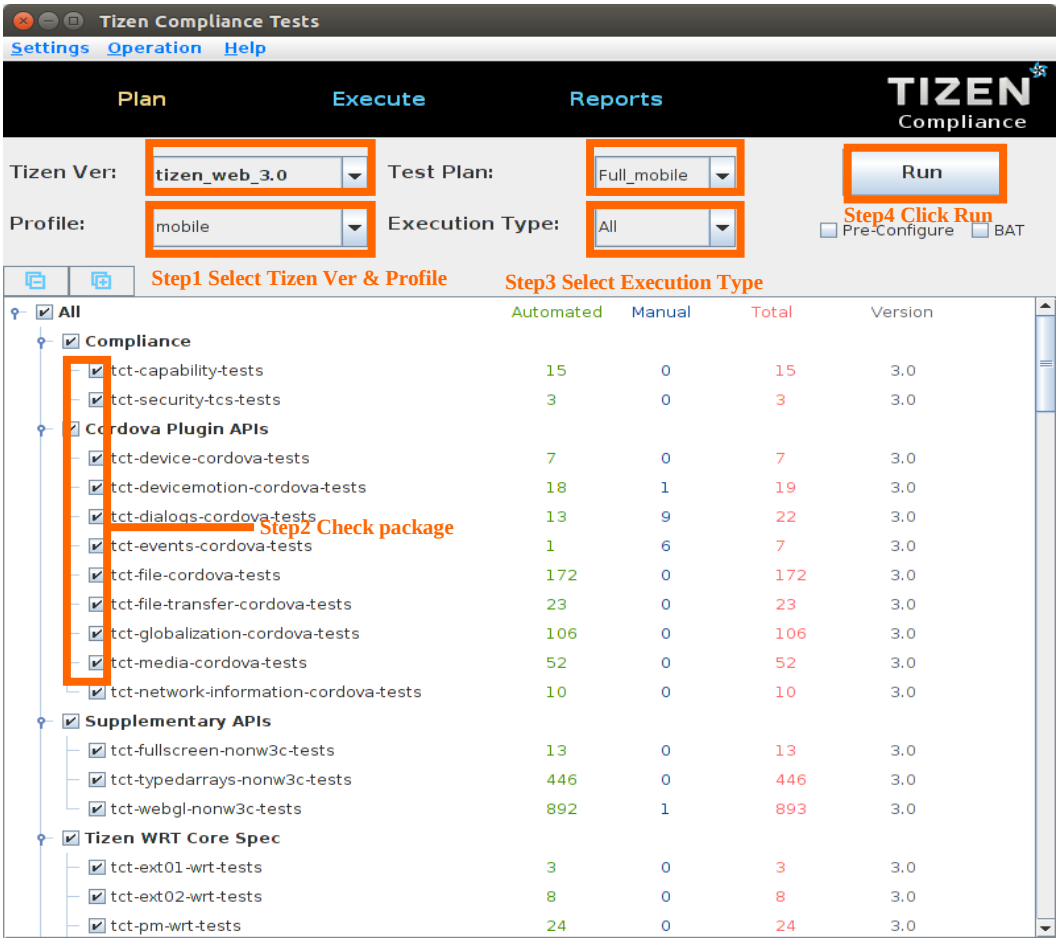

Figure 3-2: Change Test Plan

A dialog will pop-up to guide to save a new plan as shown in Figure 3-3, configure the device, and start to run the test.

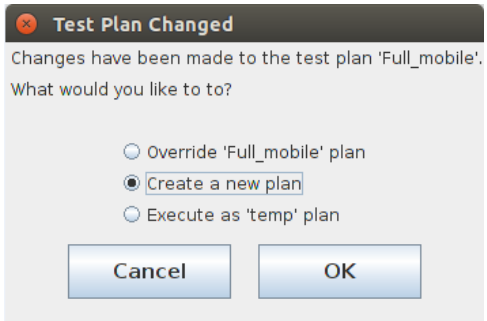

Figure 3-3: Prompt for Creating Test Plan

Select the item “Create a new plan” and click **OK** button to save a new test plan as shown in Figure 3-4

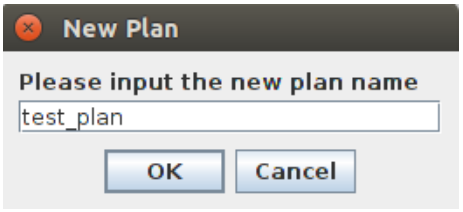

Figure 3-4: Create a New Test Plan

After entering the new plan name, click **OK** button. The Plan UI will switch to Execution UI, and run selected test plan as shown in Figure 3-5.

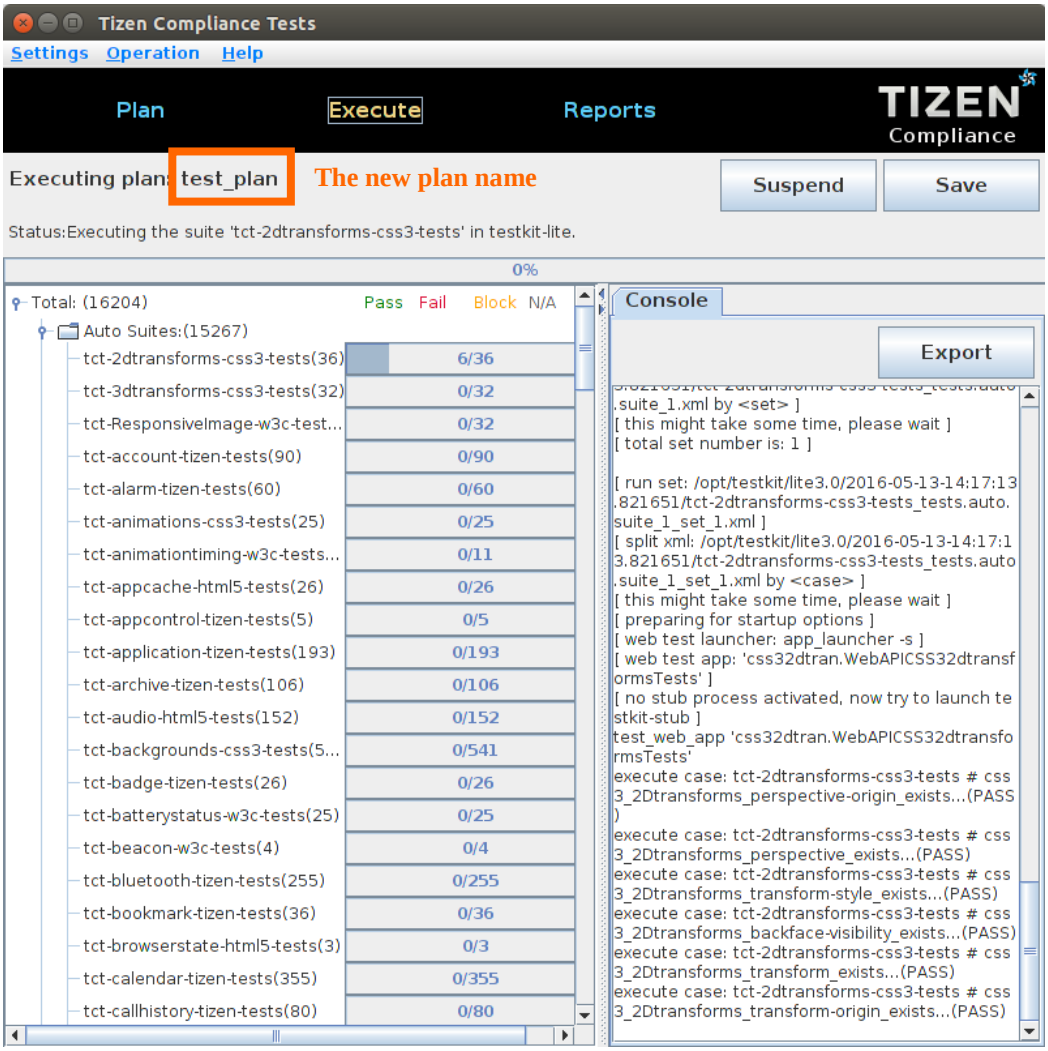

Figure 3-5: Execute a New Plan

Select a plan, the packages in the plan will be selected. As shown in Figure 3-6

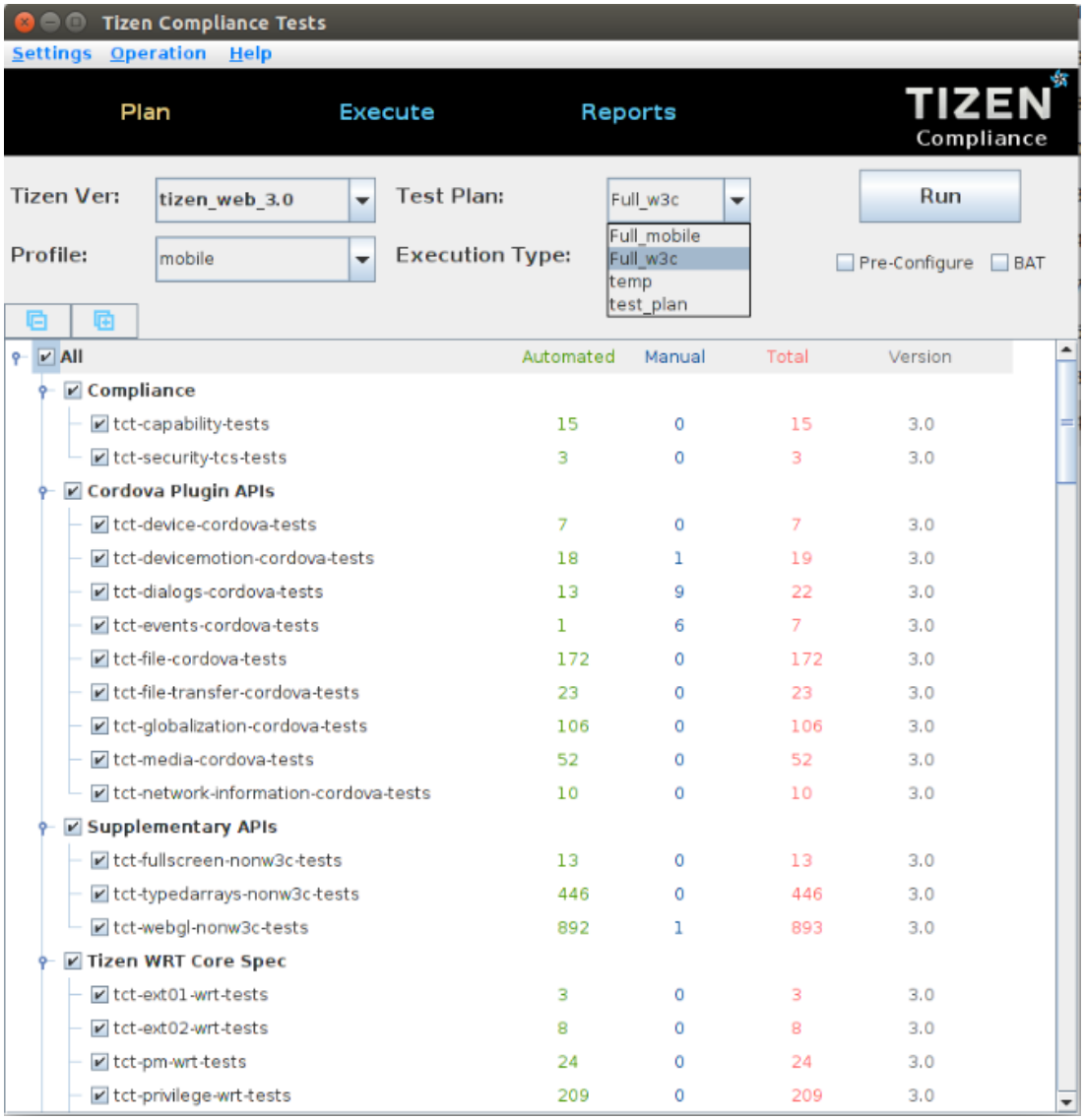

Figure 3-6: Select an Existing Plan

3.1.3 Configuring Test Environment

According to pre-configuration steps in the pop-up dialog, set the test environment before start testing.

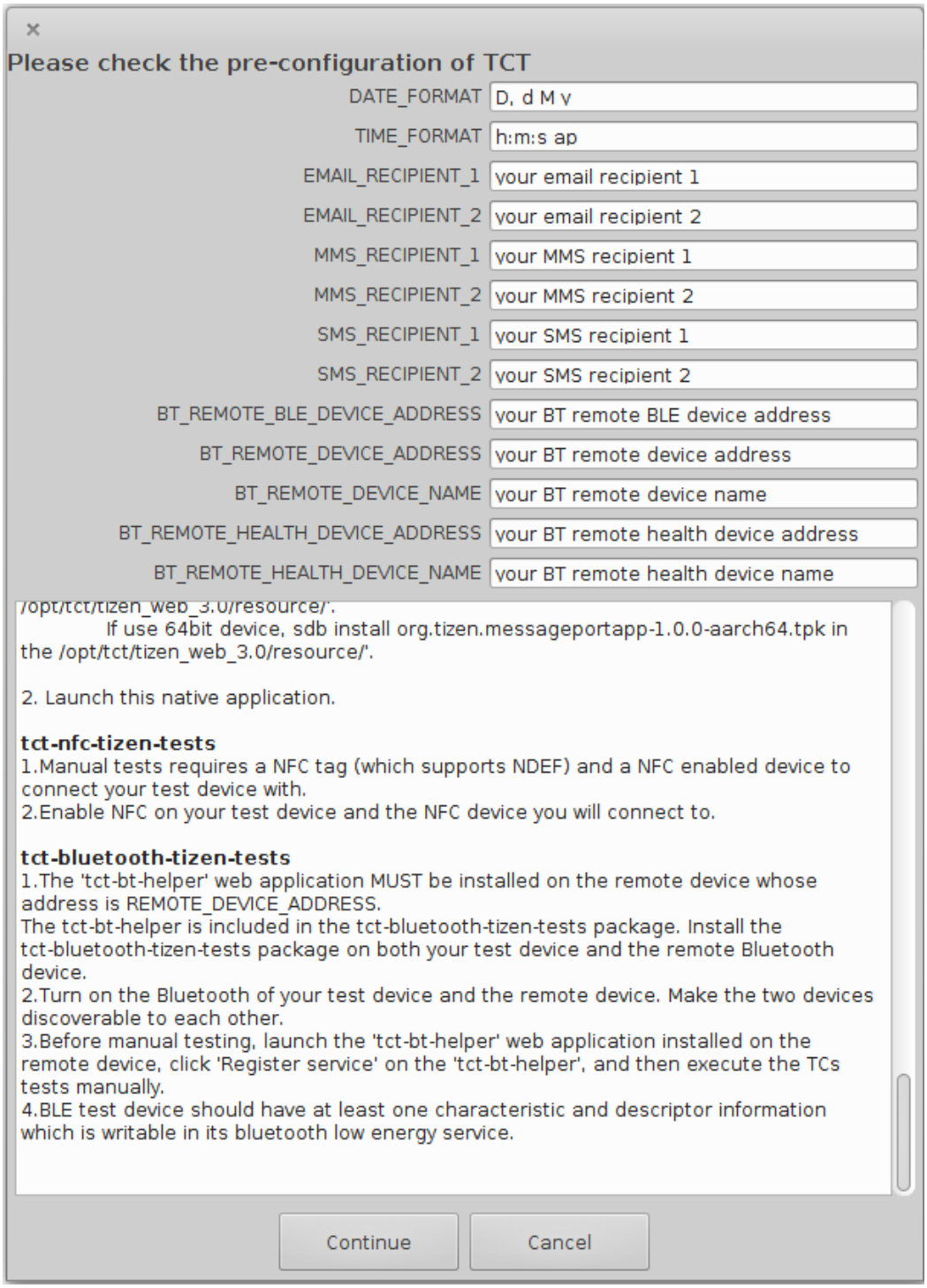

Figure 3-7: Web TCT Manager Pre-configuration Page

3.1.4 Monitoring Test Execution

After clicking **Continue** button in the configuration dialog, Web TCT Manager will go to Execute page. The test status and log information will appear there.

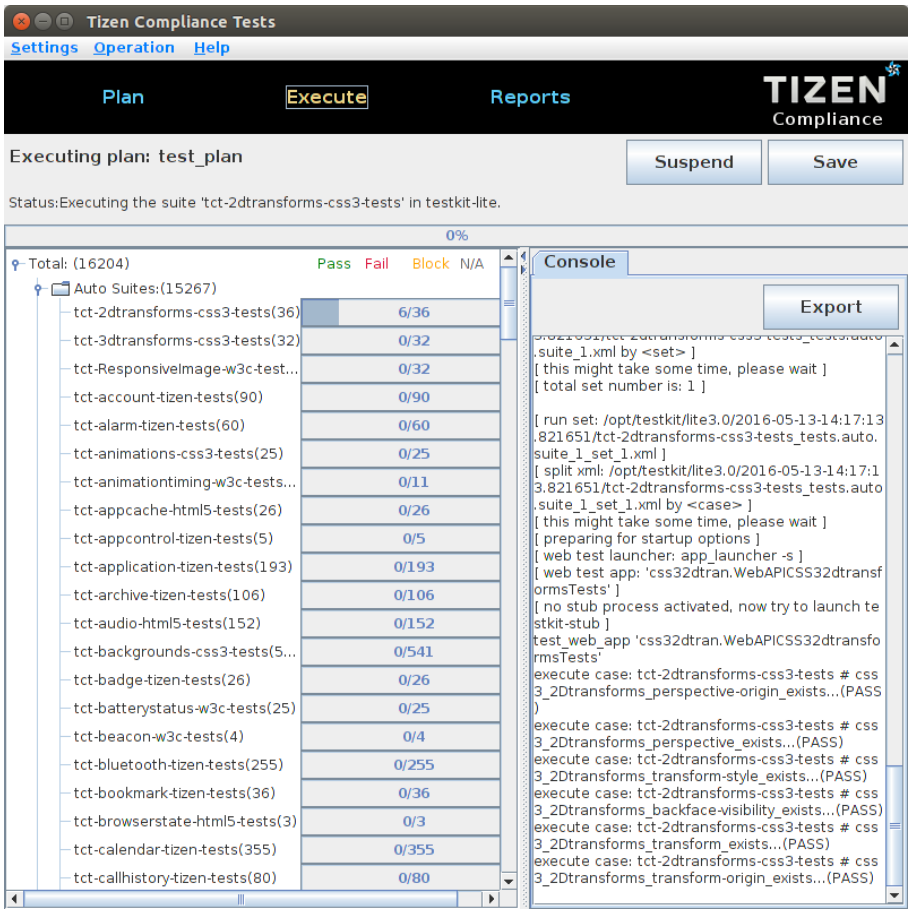

Figure 3-8: Web TCT Manager Execution UI Page

Click the **Suspend** button in *Execute UI* to stop executing the test plan. After stopping a test plan, click icon of *Reports UI* to resume test.

3.1.5 Retrieving Test Result

The test summary and details of the current test appear when the test is done. When select the Reports page, the history testing reports will show up.

To view all reports list in reports UI, click one item to view summary information. Click functional icons can easily export results or re-execute known failures for any test plan. As shown in Figure 3-9

a. Click the icon of  to download consolidated reports.

b. Click the icon of  will rerun non-passed test cases.

c. Click icon of  to resume the test plan if the status of test plan is stopped.

d. Select one or multiple test reports in reports UI, then click **Remove** button, the selected items will be removed after user confirming.

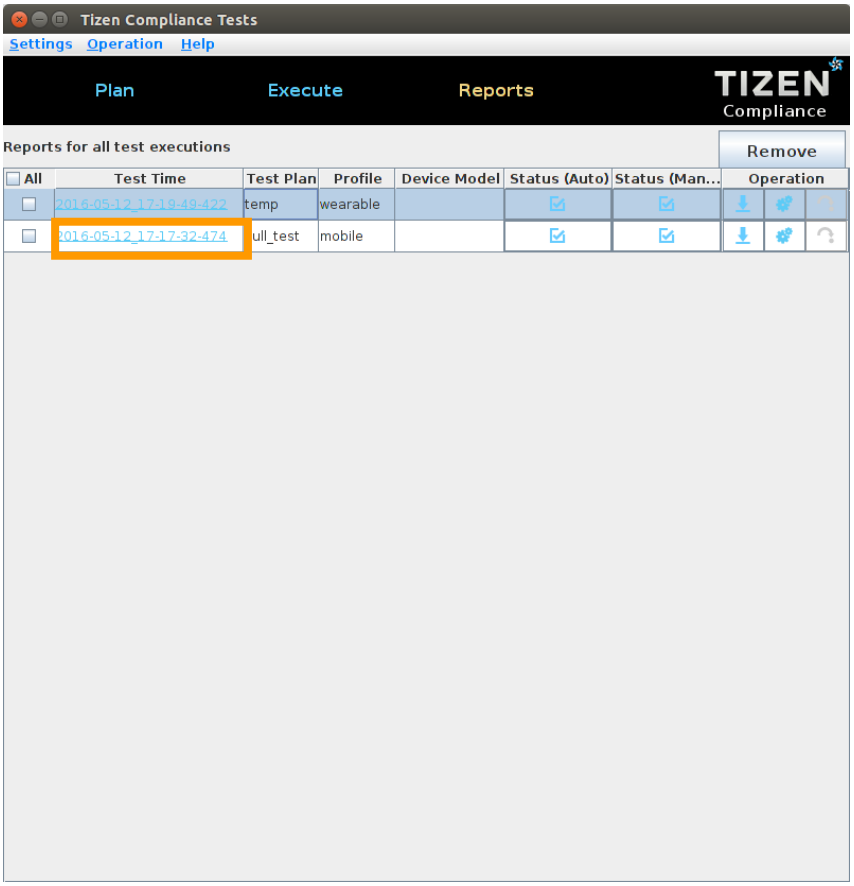

Figure 3-9: Report List

To view summary of the report, click on the **Test Time** entry, as shown in Figure 3-10

TCT Report

Test Summary	
TCT Version	TCT
Test Plan Name	temp
Test Profile	mobile
Build ID	N/A
Test Total	212
Test Passed	205
Test Failed	6
Test Blocked	1
Test Not Executed	0
Time	2016-05-13_21_59_13 ~ 2016-05-13_22_10_59

Device Information	
Host Device	Linux-3.13.0-24-generic-i686-with-Ubuntu-14.04-trusty
Manufacturer	N/A
Device Model	N/A
Device ID	0000d8e600006200
Screen Size	N/A
Resolution	N/A

Device Capability

Test Summary by Suite

Suite	Total	Passed	Failed	Blocked	Not Executed	Ratio
tct-geoallow-w3c-tests	212	205	6	1	0	96.70%

Figure 3-10: Test Summary by Suite

Click on the **suite** name in the Test Summary by Suite table to see details as shown in Figure 3-11

Suite Test Results

Show all Show only failed Show only blocked Show only not executed Summary

Test Suite: tct-messaging-email-tizen-tests (All)

Case_ID	Purpose	Result	Stdout
Test Set: Messaging-EMAIL-1st			dlog
MessageArraySuccessCallback_notexist	Check if interface MessageArraySuccessCallback exists, it should not.	PASS	[Message]
MessageArraySuccessCallback_onsuccess	Check if method onsuccess of MessageArraySuccessCallback works	FAIL	[Message] *MessageArraySuccessCallback_onsuccess: assert_unreac callback: name:UnknownError, msg:Error during getting account list Re
MessageAttachment_constructor	Check if MessageAttachment's constructor works	PASS	[Message]
MessageAttachment_constructor_minargs	Check if MessageAttachment's constructor works with minimum arguments	PASS	[Message]
MessageAttachment_exist	Check if MessageAttachment exists	PASS	[Message]
MessageAttachment_extend	Check if instance of interface MessageAttachment can be extended with new property	PASS	[Message]
MessageAttachment_filePath_attribute	Check if attribute filePath of MessageAttachment exists, has type DOMString and is readonly	FAIL	[Message] *MessageAttachment_filePath_attribute: assert_unreached: name:UnknownError, msg:Error during getting account list Reached unr
MessageAttachment_id_attribute	Check if attribute id of MessageAttachment exists, has type MessageAttachmentId and is readonly	FAIL	[Message] *MessageAttachment_id_attribute: assert_unreached: getMe name:UnknownError, msg:Error during getting account list Reached unr
MessageAttachment_messageId_attribute	Check if attribute messageId of MessageAttachment exists, has type MessageId and is readonly	FAIL	[Message] *MessageAttachment_messageId_attribute: assert_unreach callback: name:UnknownError, msg:Error during getting account list Re
MessageAttachment_mimeType_attribute	Check if attribute mimeType of MessageAttachment exists, has type DOMString and is readonly	PASS	[Message]
MessageAttachmentSuccessCallback_email_onsuccess	Check if callback is called and if its arguments have proper type - MessageAttachmentSuccessCallback, onsuccess, email	FAIL	[Message] *MessageAttachmentSuccessCallback_email_onsuccess: a error callback: name:UnknownError, msg:Error during getting account li

Figure 3-11: Show Detail Test Results

Choose one of the following to customize the report view, referring to Figure 6-5

e. **Show all**: show all the results

f. **Show only failed**: show cases that failed

g. **Show only blocked**: show cases that have blocked results

h. **Show only not executed**: show cases that have non-applicable(N/A) results

3.2 Using Web TCT Shell

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

3.2.1 Getting help

To use Web TCT Shell on the host machine to get help:

```
$ cd /opt/tct/tizen_web_3.0/
$ tct-shell --help
```

3.2.2 Generating a Test Plan

To use Web TCT Shell on the host machine:

Include all suites in the local repository

```
$ ./tct-plan-generator  -o  <somewhere>/testplan.xml
```

Include all suites in the special repository

```
$ ./tct-plan-generator  -o  <somewhere>/testplan.xml -r <somewhere>/repository_folder
```

Include the suites in the special repository where the name matches a specific regular expression

```
$ ./tct-plan-generator  -o  <somewhere>/testplan.xml -r <somewhere>/repository_folder --match '<regex>'
```

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

```
$ ./tct-plan-generator  -o  <somewhere>/testplan.xml -r <somewhere>/repository_folder --match ‘<regex>’ --unmatch ‘<unmatch_regex>’
```

3.2.3 Running Tests

To use Web TCT Shell on the host machine:

a. Run an existing test plan

```
$tct-shell --plan-list
$tct-shell --testplan '<somewhere/testplan.xml>' --tizen-version tizen_web_3.0
```

b. Run one or multiple test packages

```
$tct-shell --test '<package1, package2, ..., packageN>' --izen-version tizen_web_3.0
```

c. Run a test package on a specific device

```
$ tct-shell --device-list
$ tct-shell --testplan '<somewhere>/testplan.xml' --deviceid <device-id> --tizen-version tizen_web_3.0
```

d. Run one test case

```
$tct-shell --test 'package' --id <caseid> --tizen-version tizen_web_3.0
```

e. Run all test cases, including automatic and manual cases

```
$ tct-shell --testplan '<somewhere>/testplan.xml' --all --tizen-version tizen_web_3.0
```

f. Run manual test cases

```
$ tct-shell --testplan '<somewhere>/testplan.xml' --manual --tizen-version tizen_web_3.0
```

g. Specify the output file for test results

```
$ tct-shell --testplan '<somewhere>/testplan.xml' -o <somewhere>/test-result.xml --tizen-version tizen_web_3.0
```

h. Rerun cases that failed

```
$tct-shell --rerun-fail '<somewhere/test-result.xml>'
```

3.2.4 Checking Test Result

Check the result by viewing the test summary displayed on the web page.

Note: Upon test completion, Web TCT Shell automatically launches Firefox to display the test summary.

3.2.5 Debugging Test Case

After getting the case ID from the test summary, show the log and perform debugging by executing this command:

```
$tct-shell --test ‘package’ --id <caseid> --tizen-version tizen_web_3.0
```

To get dlog information when debugging one case, following steps should be run beforehand:

a. Enable dlog logging on target device

```
$ dlogctrl set platformlog 1
```

b. Reboot target device

c. Run web testing by Web TCT Shell

d. Check dlog information in report

3.3 Using the Web TCT Behavior Test Tool

Web TCT Behavior Test Tool is used to check how correct the device behavior is. All of the cases that run in Web TCT Behavior Test Tool are manual or semi-manual.

Web TCT Behavior Test Tool is a device behavior checker that uses jQuery. It is a major component of the Tizen Compliance Tests (TCT) tool set.

Web TCT Behavior Test Tool has these features:

a. A list of behaviors you can test, with information about test cases

b. Test cases for checking and evaluating the behavior of a device

c. Automatically generated test reports you can view and save for further analysis

3.3.1 Launching the Web TCT Behavior Test Tool

The Web TCT Behavior Test Tool can be manually launched and operated by clicking the corresponding thumbnail icon on target device.

Figure 3-12: Web TCT Behavior Test Tool Home Screen

The footer bar has buttons for working with the test results:

a. **Export:** Save the test result XML file to the Tizen file system. This is default full path that includes both location and file name: /opt/home/owner/content/Document/tct-behavior-result/tct-behavior-tests_<timestamp>.result.xml.

b. **Reset:** Reset the test results.

c. **Exit:** Quit.

3.3.2 Checking the Behavior Cases

Click on every item, and go to the detailed check screen for current device behavior. Take MediaPlayback as an example. Below screen will show:

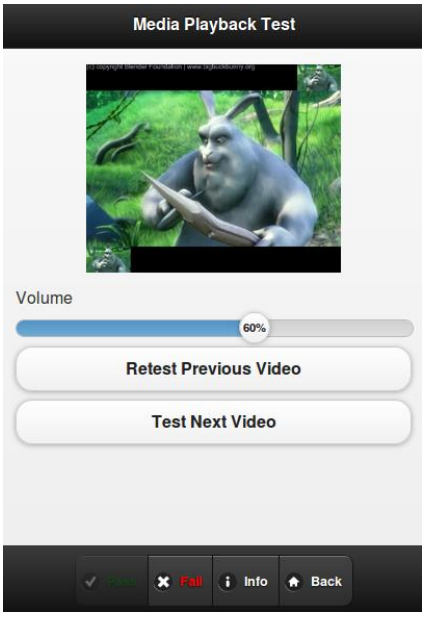

Figure 3-13: Web TCT Behavior Test Tool MediaPlayBack Screen

Click the **Info** button to view the test case description, and perform the test according to it.

Check the playback quality of the video clip, and then click **Pass** or **Fail** button based on the output quality.

3.3.3 Getting the Result File

After all the items are checked, click **Export** on the home screen and export the XML test report to local file system.

The report will be saved in location "**/opt/home/owner/content/Document/tct-behavior-result/tct-behavior-tests_<timestamp>.result.xml**"

You can view the summary information, including the case title and result by open the xml with browser.

A Appendix

a. Known Issues

Symptom: All automated testing fails and an error message says: “fail to connect with test service.”

Solution: After changing to use another target device, need to rerun tct-config-device.sh on host side to set up the test environment of the new target device.

Symptom: When rerunning failed cases, the UIFW package will not be tested even there is failed cases in UIFW.

Solution: Rerunning UIFW failed case is not supported yet.

b. Troubleshooting

Q: On target device, power consumption is faster than power charging through a USB cable. What should I do to make sure the full TCT test can be executed on my device?

A: Use target device with a power supply

Q: Some web test packages failed to be installed on the target device. What should I do?

A: It might be because the certification for these packages did not pass. Set the target device’s time and date to the current date to avoid this issue.

c. Web TCT Shell Options

Mandatory options	
Option	Description
--scenario/-s	Specify scenario.xml as the test plan.
--testplan/-p	Specify testplan.xml as the test plan.
--test/-t	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.
--tizen-version	Specify the name of tizen-version you want to run test. The tizen-version is defined in the local repository. And its path is ‘/opt/tct/’

Optional options	
Option	Description
--output=*resultfile*	Specify the output file for the result XML output. If more than one test xml file is provided, results will be merged into this output file.
--version	Show version information.
--all-suites	Show all available test-suites in the local repository ‘opt/tct/’
--deviceid	Set sdb device serial information.
--plan-list	List all existed plans in the plan folder. The plan folder is defined in the configuration file named CONFIG, and its path is:/opt/tct/shell/CONFIG.
--result-list	List all history results in the result folder. and its path is:/opt/tct/tizen_web_3.0/shell/ result/
--device-list	List all connected devices. Just same with ‘sdb devices’
--all	Both manual and auto test cases will be executed, without this option, only auto testcase will be executed.
--manual	Only manual test cases will be executed.
--id	Specify to run a test case by id.
--distribute	Specify the testplan.xml for distributing a plan to multiple devices
--dbutedevid	Set sdb device serial information for distribute mode.
--stub-port	Specify a port number. Network port used by the testkit-stub
--log/-l	Set log level. Logs that are less severe than the level will be ignored. Log levels (in descending severity order): [CRITICAL, ERROR, WARNING, NOTEST, INFO, DEBUG]
-h, --help	Show this help message and exit.