# Tizen Compliance Test (TCT)

### Hojun Jaygarl (Samsung Electronics), Cathy Shen (Intel)

TIZEN DEVELOPER CONFERENCE 2013 SAN FRANCISCO

### Contents

- Tizen Compliance Program
- Native TCT
- Web TCT



# **Tizen Compliance Program**

## **Tizen Compliance Program**

• Key components of the Tizen Compliance Program:



- **TCS** is a set of requirements for devices and applications to ensure that they work together smoothly .
- **TCT** is a part of the Tizen compliance process designed to verify mobile device implementation conformance to requirements in TCS.
- **Reference implementation** compensates TCS and TCT to guide the behavior of compliant implementation.



## **Tizen Compliance Program Flow**



\* This process is not fixed yet.



### **Tizen Compliance Specification Categories**

	General Principles (Web and Native API)		Memory Storage	
	Tizen Web API		Sound	
	Tizen Native API	Mandatory	Connectivity/Networking	
	Application Binary Interface	Hardware Requirements	Display	
	Application Control	Application Control		
	Platform Attributes		Input Devices (Touch, Key Input)	
Magalaka	Device Capability Report		Camera	
Software	Privilege		USB	
Requirements	Application Packaging Compatibility		Sensors	
	WebKit and Browser	Optional	Telephony	
	Web Runtime	Hardware Requirements	Bluetooth	
	User Interface		Wi-Fi	
	Security		NFC	
	Multimedia		Input Devices (Microphone)	
	Development Tools			
ĺ	Software Update			



## **TCT Objective**

- TCT verifies conformance of TCS requirements
  - All mandatory SW and HW features should pass a compliance test
  - If a device provides optional HW features, all relevant SW features should pass a compliance test

### Objective

- Uniform application development environment
- Consistent customer experience for applications
- Differentiated but compatible platform among multiple device manufacturers



### **Auto Test and Manual Test**





- To provide compatibility of APIs Signature tests, Platform API tests,
- Platform data model, Platform AppControls Platform privilege, Platform resources

- To test APIs and functions with manual input
- e.g. sensors, Wi-Fi, BT, NFC, audio quality
- To test UI results
- e.g. button, rotation



## **Compliance Test for Optional Features**

- Configuration must describe the supporting features of the device
  - The Tizen Store controls
     downloading only the properly
     functioning applications based on
     the device configuration file
- If optional HW is supported, relevant API should be provided and covered by the compliance test

#### Configuration.<sup>▶</sup> O♪ X Feature is restrict Compliance ed (Applications using rele OV Test vant features are filtere Coverage♪ Device d from the Tizen Store by configuration) Fail for Complian Х⊅ Non relevant ce Test

### Optional feature test criteria



# Native TCT

### **Native TCT Overview**

- Native TCT automatically tests native APIs and its behavior by running native test apps
- Native TCT consists of the following components
  - FtApp: native application that loads and executes test suites
  - Test Suite: FtApp loadable dynamic-link library, .so with the specific format of a configuration file, .conf. TCT native test suites use GTest as a test framework.
  - TCT Manager: a GUI tool to deploy test suites from host to target, manage crashes during test execution and generate reports of the test result.



### **Native TCT Workflow**





### **Native TCT Package**





### Installation

- Host PC @ 64bit or @32bit, Ubuntu 12.04 or 12.10, Target Device, USB connection (host PC -> target device)
- Download and decompress package file, Launch /installed\_path/eclipse
- Set (or switch) workspace path to /installed\_path/cts

😣 Workspace Launcher	
Select a workspace	
Eclipse stores your projects in a folder called a workspace Choose a workspace folder to use for this session.	·
Workspace: encts/eclipse_cts_linux32_20121220_v3.4.7_	demo/cts
<ul> <li>① Set work</li> <li>Output the Use this as the default and do not ask again</li> </ul>	xspace)
	Cancel OK 2 Click OK



### **Create Plan**





## **Select Test**

- 1. Edit the plan by selecting test project name(s).
- 2. Only selected items in the table are tested.
- 3. Save the plan when the editing is done.

😣 🖻 🗉 🛛 Tizen CTS - CTS_workspace/NewPlanTest/NewPlanTest.pln - Eclipse								
File Edit Navigate Search Project CTS Run Window Help								
	○ ○ ○ □ □ □         □ <t< th=""></t<>							
ြဲ Project Explo 🕱 📃 🗖	► NewPlanTest 🖾 🗖 🗖							
□ 🔄 🕞 🍸	CTS Plan							
CTS_workspace	This page shows each Test Suite of Compatibility							
② Save	Save Plan * Plan has been changed. Please save.							
	UnitTest This section provides UnitTest lis							
	Name 2.0 2.1 TimeOut							
	osp-uts-app							
	osp-uts-base							
(	osp-uts-base-runtime 30							
	osp-uts-base-utility ① Select items to be tested							
	Select All/Deselect All							
📫 E E								
Files	Integration rest  This section provides IntegrationTest lists							
4d19f6b974a41f00 ( <unk< p=""></unk<>								
	<pre><plan> <configuration> <run> <report> <log></log></report></run></configuration></plan></pre>							
	E Console 🕱 🖻 Log							



## Configure

- 1. Select configuration.
- 2. Set the SDB path and model name.
- 3. Save the configuration.
- 4. Run the test.





## **Run Test**

- All test cases are automated (GTest is applied).
- TCT handles any problems occurring during the test.
- Test status is
   updated in real time.

😸 🗇 🗊 Tizen CTS - CTS_workspace/NewPlanTest/NewPlanTest.pln - Eclipse								
File Edit Navigate Search Project CTS Run Window Help								
<sup>1</sup> ▼ <sup>1</sup> ■ <sup></sup>								
ြာ Project Explo 🛚 🗖 🗖	oject Explo 🕱 🦳 🗖 🖻 NewPlanTest 🕱 👘 🗖							
⊑ 🔄 🖓 🌄	CTS Run							
CTS_workspace	This page shows CTS	running status						
	List of Compatibilit	y Test Suite						
	Category	Name	Elapsed time	Start time	Er			
	UnitTest	osp-uts-base-runtime-2.1	00:00:25	2012/12/27 15:09:18				
		osp-uts-base-utility-2.1						
	IntegrationTest	intApp-2.1						
				① Test statu	us table♪			
	(4)			)				
((()))))	< Plan > < Configurat	ion > < Run > < Report > < Log >						
Connection E 🛚 🖉 🗖	🗏 Console 🛿 🗟 L	.og		🔓 🚮 🛃	] <b>- ⊡</b> - □			
B B B	CTS Console							
Tiles	0][osp-uts-b 0][osp-uts-b	ase-runtime-2.1][TESTING][M] 🖺	[0;32m[ [0:32m[ RUN	OK ] [18][mUTsEventDriven 1 [19][mUTsEventDriven	Thread.UT(			
Files	101][osp-uts-b	ase-runtime-2.1][TESTING][M]	[0;32m[	OK ] DB [mUTsEventDriven	Thread.UT			
► 🖬 4019F6D974841F00 ( <unk< th=""><td>0][osp-uts-b</td><td>ase-runtime-2.1][TESTING][M]</td><td>[0;32m[ RUN</td><td>] [mUTsEventDriven]</td><td>Thread.UT</td></unk<>	0][osp-uts-b	ase-runtime-2.1][TESTING][M]	[0;32m[ RUN	] [mUTsEventDriven]	Thread.UT			
	0][osp-uts-b	ase-runtime-2.1][TESTING][M] 🖫	[0;32m[ RUN	] [][[mUTsEventDriven	Thread.UT			
	0][osp-uts-b	ase-runtime-2.1][TESTING][M]	[0;32m[	OK ] [][[mUTsEventDriven	Thread.UT			
1][osp-uts-base-runtime-2.1][TESTING][M] [B][0;32m[ RUN ] [B][mUTsEventDrivenThread.UTo								
2 Test log win	dow )][osp-uts-b	ase-runtime-2.1][TESTING][M] 🔢	[0;32m[ RUN	] [][[mUTsEventDriven	Thread.UT			
je root log mil	][osp-uts-b	ase-runtime-2.1][TESTING][M]	[0;32m[	OK ] $\frac{0.0}{1.8}$ [mUTsEventDriven	Thread.UT			
	1][osp-uts-b 439][osp-uts-b	ase-runtime-2.1][TESTING][M] <u>‼</u> ase-runtime-2.1][TESTING][I] [T	[[0;32m[ RUN EST STARTED]	J [밤[mUTsEventDriven [FtApp.exe] [23488] Test	process i			
			-					



## Test Result (1/2)

- Report (test result) is generated automatically when the test is done.
- The report contains total, passed, failed, and nonexecuted test cases in table and graph form.
- Test result, status, elapsed time, and log for each test project is available.
- Detailed messages are displayed for each failed test case.

### Test result

UnitTest		Result				Flapsed		
		Total	Passed	Failed	Not Executed	Time	Ratio	Log
$\checkmark$	osp-uts-app-2.0	171	170	1	0	00:00:37	99%	Log
$\checkmark$	osp-uts-app-3.0	226	225	1	0	00:00:19	99%	Log
$\checkmark$	osp-uts-base-2.0	909	909	0	0	00:01:58	100%	Log
$\checkmark$	osp-uts-base-3.0	1,160	1,160	0	0	00:00:52	100%	Log
N/E	osp-uts-base-collection-2.0	0	0	0	0	00:00:00		Log
N/E	osp-uts-base-collection-3.0	0	0	0	0	00:00:00		Log
$\checkmark$	osp-uts-base-runtime-2.0	63	63	0	0	00:01:37	100%	Log
$\checkmark$	osp-uts-base-runtime-3.0	92	92	0	0	00:01:53	100%	Log
$\checkmark$	osp-uts-base-utility-2.0	605	605	0	0	00:01:15	100%	Log
$\checkmark$	osp-uts-base-utility-3.0	605	605	0	0	00:00:26	100%	Log
$\checkmark$	osp-uts-content-2.0	281	131	150	0	00:02:34	46% 54%	Log
$\checkmark$	osp-uts-content-3.0	292	282	10	0	00:06:08	96%	Log
$\checkmark$	osp-uts-enriched-text-2.0	151	151	0	0	00:01:11	100%	Log



## Test Result (2/2)

#### Detailed Result for each test suite

#### Report Summary > UnitTest Result > osp-uts-system-2.1 Detailed Test Result This page provides detailed results of osp-uts-system-2.1 for all documented API exposed by the tizen SDK.





 $(\Box)$ 

## **FtApp Execution**

- FtApp can be executed without the Test Manager (the Eclipse IDE).
- It can be used as a test runner for your own test cases.

[FTAPP] FtSoExecutor - DllLoadLibrary: /opt/apps/org.tizen.2s4jm6firv/data/libTestTc.so Facial Engine - Called constructor [FTAPP] dlopen is success Signature:201208141222 [FTAPP] Creating Tests [FTAPP] success to crete a log file - /opt/apps/org.tizen.2s4jm6firv/data/ftapputs.log [FTAPP] FtSoExecutor - CreateTSMain done [FTAPP] TSMain SetLogFilePath:/opt/apps/org.tizen.2s4jm6firv/data/ [FTAPP] TSMain SetLogFileName:ftapputs.log [FTAPP] FtSoExecutor - LoadSo done Google Test filter = UTsBooleanP.UTcToBoolP01:UTsBooleanP.UTcOperatorNonEquivalenceP01:UTsBooleanP.UTcToString UTcParseP02:UTsBooleanP.UTcGetFalseP01:UTsBooleanP.UTcBooleanP01:UTsBooleanP.UTcBooleanP02:UTsBooleanP.UTcGet Running 15 tests from 1 test case. Global test environment set-up. 15 tests from UTsBooleanP UTsBooleanP.UTcBooleanP01 UTsBooleanP.UTcBooleanP01 (0 ms) UTsBooleanP.UTcBooleanP02 UTsBooleanP.UTcBooleanP02 (0 ms) UTsBooleanP.UTcBooleanP03 OKSILUTsBooleanP.UTcBooleanP03 (0 ms) UTsBooleanP.UTcOperatorEquivalenceP01 UTsBooleanP.UTcOperatorEquivalenceP01 (0 ms) UTsBooleanP.UTcOperatorNonEquivalenceP01 OK ] UTsBooleanP.UTcOperatorNonEquivalenceP01 (0 ms) UTsBooleanP.UTcEqualsP01 OK ] UTsBooleanP.UTcEqualsP01 (0 ms) UTsBooleanP.UTcEqualsP02 OK ] UTsBooleanP.UTcEqualsP02 (0 ms) UTsBooleanP.UTcToBoolP01 UTsBooleanP.UTcToBoolP01 (0 ms) UTsBooleanP.UTcParseP01 UTsBooleanP.UTcParseP01 (1 ms) UTsBooleanP.UTcParseP02 OK ] UTsBooleanP.UTcParseP02 (0 ms) UTsBooleanP.UTcParseP03 OK ] UTsBooleanP.UTcParseP03 (0 ms) UTsBooleanP.UTcToStringP01 OK ] UTsBooleanP.UTcToStringP01 (0 ms) UTsBooleanP.UTcToStringP02 UTsBooleanP.UTcToStringP02 (0 ms)



## **Native TCT Test Cases**

### Areas covered

- Signature: all Tizen native APIs are available
- Native API: all Tizen native public APIs work correctly
- App Control: data is delivered correctly between apps
- Privileges: privileged APIs are correctly allowed and denied for apps
- Device capabilities: optional features supported by the platform and device work correctly



## **Native Verifier App**

#### **Tizen Application for Checking Tizen device behavior Manually** •

1:50PM
Tizen CTS Verifier
Multimedia Codecs
Local Test
Streaming Test
ImageView
AppControl
operation/pick Test
operation/view Test
Sensors
Accelerometer Test
Gyroscope Test
Proximity Test
Light Test
Magnetometer Test
Delete Viewer Save

1. Select TestCase

<u>1:52pm</u>
Tizen CTS Verifier Multimedia Codecs
Local Test
Streaming Test
ImageView
AppControl
operation/pick Test
operation/view Test
Sensors
Accelerometer Test
Gyroscope Test
Proximity Test
Light Test
Magnetometer Test
Delete Viewer Save 5
3. Verifvina)

#### 2. Check Behavior



### **Native Verifier Test Cases**

### Test Categories

- Multimedia Codec Test
- App Control Test
- HW Requirement Test
- Sensor Test
- Platform Resource Test
- Others



### **Statistics**

### • Types

- Unit TC: Verify behavior of all public APIs supported by the Tizen native framework.
- Integration TC: Verify behavior of basic combinations of APIs based on use case scenarios.
- Compliance TC: Verify compliance specifications in TCS.
- Manual TC

### Statistics

• Not fixed.

Types,⋟	# of TCs⊅
Unit TCs♪	11740♪
Integration TCs.	1025⊅
Compliance TCs	589♪
Manual TCs)	171♪
Total TCs♪	13525♪







### Web TCT Overview

- Web TCT is a set of tools and test cases to test web requirements defined in Tizen Compliance Specification
  - **TCT-manager** is GUI tool to run Web test cases
  - **TCT-shell** is lightweight console tool to automatically run Web test cases and debug test case failure
  - **TCT-behavior** tool is used to test behavior of hardware and software features in interactive mode
  - Web Test Suite covers Tizen Web API, Tizen Web Runtime Core, Web UI Framework and Device Capability Features
  - User Guide, Installation Scripts









### Web TCT Architecture





### Installation

### Platform Prerequisites

- Host PC: @ 64bit or @32bit, Ubuntu 12.04 or 12.10, Browser (Googlechrome or Firefox are preferred), network capability
- Target Device (Lunchbox, M0 or other Tizen Devices)
- USB connection (host PC -> target device)

### Installation is just easy by install.sh

- On host side, set up SDB, test suite, dependencies and TCT test tools
- On target device, set up embedded web server (http directory, websocket, CGI), test media, test runner, behavior test tool



### **TCT-shell**

### Console tool for Web TCT test

cathy@cathy-desktop:~\$ tct-shell Usage: tct-shell [options] --testplan <somewhere/testplan.plan> examples: tct-shell --test package1 package2 ... packageN tct-shell --rerun-fail '<somewhere>/test-result.xml' tct-shell --testplan <somewhere>/testplan.xml --capability <somewhere>/capability list run a test plan: tct-shell --testplan <somewhere>/testplan.xml --capability <somewhere>/capability list -o /tmp/wekit-tests-result.xml ... run some test packages: tct-shell --test package1 package2 ... packageN -o /tmp/wekit-tests-result.xml ... rerun all unpassed test: tct-shell --rerun-fail '<somewhere>/test-result.xml' ... show all existed testplan which is in the folder (configured in /opt/testkit/shell/CONFIG): tct-shell --plan-list show all history result which is in the folder (configured in /opt/testkit/shell/CONFIG): tct-shell --result-list show all connected devices: tct-shell --device-list



## TCT-manager | Test Plan

	43					(?) HELP	(i) ABOUT
		T2.1	PLAN	EXECUTE	REPORT		
	Test Plan	Full_test		•	Run		
	Execution Type	Automated		•			
1	Package Name				Case Number	Version	
1	tct-3dtransforms-tests				3998	2.2.2-143	
1	tct-alarm-tests				54	2.2.2-1	
1	tct-animations-tests				101	2.2.2-143	
1	tct-animationtiming-test	s			15	2.2.2-143	
1	tct-appcache-tests				51	2.2.2-143	
1	tct-appcontrol-tests				7	2.2.2-1	
1	tct-application-tests				154	2.2.2-1	
1	tct-audio-tests				329	2.2.2-143	
1	tct-backgrounds-tests				664	2.2.2-143	i



### TCT-manager | Execution

	PLAN E	(?) HELP (i) ABOUT
Execute Tests		Stop Test
	Test cases H Please wait.	nave been running 41 seconds
Total	19/243	<19:05> start time: 2000-01-03 17:16:15
Auto Test		<19:05> [case] execute case: CSS3Colors_color_0ff <19:05> last_test_result: PASS <19:05>
webapi-css3-colors-tests	19/46	<19:05> start time: 2000-01-03 17:16:15 <19:05>
webapi-css3-fonts-tests	(81)	<19:05> [case] execute case: CSS3Colors_color_0FF <19:05> last_test_result: PASS <19:05>
webapi-css3-text-tests	(116)	<19:05> start time: 2000-01-03 17:16:15 <19:05> <19:05> [case] execute case: CSS3Colors color rgb 0 255 255
Manual Test		<pre>&lt;19:05&gt; last_test_result: PASS &lt;19:05&gt; &lt;19:05&gt; start time: 2000-01-03 17:16:15 &lt;19:05&gt; (ase] execute case: CSS3Colors_color_rgb_0pct_100pct_100pc &lt;19:05&gt; last_test_result: PASS &lt;19:05&gt; &lt;19:05&gt; start time: 2000-01-03 17:16:16 &lt;19:05&gt; &lt;19:05&gt; [case] execute case: CSS3Colors_color_CCFF00 &lt;19:05&gt; [ast_test_result: PASS &lt;10:05&gt; [ast_test_r</pre>



## **Test Report (1/2)**

#### **TCT Report**

Test Summary						
TCT Version	TCT2.1					
Test Plan Name	Plan: /opt/testkit/shell/plan/example_plan.xml					
Build	Tizen-2.1_20130516.2					
Test Total	186					
Test Passed	163					
Test Failed	13					
Test Blocked	5					
Test Not Executed	5					
Time	2013-05-10_15_33_38 ~ 2013-05-10_15_36_30					

Device Information				
Host Device	ASUS Fedora			
Device Name	TRATS			
Device Model	armv7l			
Device ID	X0000000000X			
Firmware Version	VXXXXXXXX			
Screen Size	58mm x 103mm			
Resolution	720 x 1280			

#### Test Summary by Suite

Suite	Total	Passed	Failed	Blocked	Not Executed	Ratio
webapi-tizen-alarm-tests	54	39	5 5	5	72% 9% 9% 9%	
webapi-tizen-application-tests	132	124	8	0	0	94% 6%
						Back to Tor





### **Suite Test Results**

Show all Show only failed Show only blocked

#### Test Suite: webapi-tizen-application-tests (All)

Case_ID		Purpose		Stdout				
Test Set: Application								
Application_appInfo_attribute		check Application.appInfo attribute	PASS	[Message]				
Application_ContextId_attribute		check Application.ContextId attribute	PASS	[Message]				
Application_exit_exist		check if method Application.exit exist and can be overriden	PASS	[Message]				
Application_extend		check if Application possible extend	PASS	[Message]				
Step 1:           Application_inde_exist           Application_hide_exist           Check if method Application.getRequestedAppControl exist           Expected:method getRequestedAppControl exist		check if method Application.getRequestedAppControl exist and	PASS	[Message]				
		Control exist and can be overriden an sist and can be overriden	PASS	[Message]				
Application_notexist	Entry:		PASS	[Message]				
Voptwebapi-tizen-application-tests/application/		Application_getRequestedAppControl_exist.html		[Message]###Test Start###ApplicationCertificate_extend###Test End####Error2 Start###(stack: getAppCerts@[native code] file:///opt/usr/apps/api1appli0/res/wgt /opt/webapi-tizen-application-tests/application/ApplicationCertificate_extend.html:49 step@file:///opt/usr/apps/api1appli0/res/wgt/opt/webapi-tizen-application-tests/resources /testharness.js:1121 test@file:///opt/usr/apps/api1appli0/res/wgt/opt/webapi-tizen- application-tests/resources/testharness.js:414 global code@file:///opt/usr/apps/api1appli0				



Show only not executed Summary

## Debugging

### To debug failed or non-executed cases:

- 1. Get the caseID of the failed case from the report page.
- 2. Execute \$tct-shell --test package --id caseID.



## **TCT-behavior**

• Test the behavior of hardware and software features in interactive mode

Home page







### Web TCT Test Cases

- Web API Tests
  - W3C/HTML5, Supplementary, Device API
- Web Runtime Tests
  - Package Management, Runtime & UI, Security & Privacy, Widget
- Web UI Framework Tests
  - Web Widgets, Events, Effects, Animations
- Device Capability Tests
  - Mandatory and Optional Hardware Capability, Unsupported Features
- TCT Behavior Tests
  - Hardware Feature, Media Streaming, Networking, Location, USB, Sensors etc.



### Web TCT Test Case Statistics

	Total TCs	Auto TCs	Manual TCs
Web API Tests	9300	8600	700
Web Runtime Tests	500	0	500
Web UI Framework Tests	849	849	0
Device Capability Tests	28	28	0
TCT Behavior Tests	24	0	24

• TCs are not fixed, and Manual TCs are under optimization



# TIZEN DEVELOPER CONFERENCE 2013 SAN FRANCISCO