Entering Tizen world for iOS & Android developers

Cheng Luo, DukSu Han
Samsung Platform Evangelist
Contents

1. Platform Overview
2. Frameworks
3. Native UI
4. Application Life Cycle
5. Event Handling
1. Platform Overview
Native

iOS

Objective C

Tizen

C++

Android

Java / C++
IDEs

iOS  Tizen  Android
Target Device

iOS  Tizen  Android
2. Frameworks
Frameworks of Tizen
Frameworks of Tizen

Native Applications

Tizen Native Framework

- App / Shell
- Base / Io
- Net
- System
- Graphics / Ui
- Uix
- Telephony
- Security
- Content
- Media
- Social
- Web / Xml
- Messaging
- Text / Locales

Tizen API
iOS to Tizen
Framework of iOS

Cocoa touch

Media

Core services

Core OS

iOS

• Accelerate Framework
• Core Bluetooth Framework
• External Accessory Framework
• Generic Security Service Framework
• Security Framework
• System
Framework of Tizen

- Tizen::Uix::Sensor
- Tizen::Net::Bluetooth
- Tizen::System::DeviceManager
- Tizen::Security
- Tizen::Base
Framework of iOS

Cocoa touch

Media

Core services

Core OS

iOS

- Accounts Framework
- Address Book Framework
- Ad Support Framework
- CFNetwork Framework
- Core Data Framework
- Core Foundation Framework
- Core Location Framework
- Core Media Framework
- Core Motion Framework
- Core Telephony Framework

- Event Kit Framework
- Foundation Framework
- Mobile Core Services Framework
- Newsstand Kit Framework
- Pass Kit Framework (iOS 6)
- Quick Look Framework
- Social Framework (iOS 6)
- Store Kit Framework
- System Configuration Framework
Framework of Tizen

Tizen

- Tizen::Social::Addressbook
- Tizen::Net
- Tizen::Base
- Tizen::IO
- Tizen::Location
- Tizen::Media
- Tizen::Telephony
- Tizen::System::SystemInfo
Framework of iOS

iOS

- Assets Library Framework
- AV Foundation Framework
- Core Audio
- Core Graphic Framework
- Core Image Framework
- Core MIDI Framework
- Core Text Framework
- Core I/O Framework
- GLKit Framework
- Media Player
- OpenAL, OpenGL ES
- Quartz Core Framework
Framework of Tizen

- Tizen::Media
- Tizen::Graphics
- Tizen::Io
- Tizen::Text
- OpenGL ES
- OpenAL
Framework of iOS

- Address Book UI Framework
- Event Kit UI Framework
- Game Kit Framework
- iAd Framework
- Map Kit Framework
- Message UI Framework
- Twitter Framework
- UIKit Framework
Framework of Tizen

Tizen

- Tizen::App::AppControl
- Tizen::UI
Android to Tizen
## Application Framework

<table>
<thead>
<tr>
<th>Android</th>
<th>Tizen</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Activity</strong></td>
<td><strong>UI application</strong></td>
</tr>
<tr>
<td><strong>Service</strong></td>
<td><strong>UI application</strong></td>
</tr>
<tr>
<td><strong>Content Provider</strong></td>
<td><strong>Service Application</strong></td>
</tr>
<tr>
<td><strong>App Widgets</strong></td>
<td><strong>Home Screen Widget</strong></td>
</tr>
<tr>
<td><strong>Preference</strong></td>
<td><strong>App</strong></td>
</tr>
<tr>
<td><strong>Intent(Call Action)</strong></td>
<td><strong>AppRegistry</strong></td>
</tr>
<tr>
<td><strong>Binder(IPC)</strong></td>
<td><strong>AppControl</strong></td>
</tr>
</tbody>
</table>

**Android api17 vs. Tizen2.1**
## Base

<table>
<thead>
<tr>
<th>Android(android.)</th>
<th>Tizen(Tizen::)</th>
</tr>
</thead>
<tbody>
<tr>
<td>java.lang collection</td>
<td>Base Collection</td>
</tr>
<tr>
<td>Base Collection</td>
<td>Text Base::Collection</td>
</tr>
<tr>
<td>Collection::stlConverter(STL)</td>
<td></td>
</tr>
<tr>
<td>util(regexp,zip) util(locale,timezone)</td>
<td>Utility</td>
</tr>
<tr>
<td>Utility</td>
<td>Base::Utility, Base::Locales</td>
</tr>
<tr>
<td>util.concurrent(lock, semaphore) os (system info.) hardware.input</td>
<td>Thread System</td>
</tr>
<tr>
<td>Thread System</td>
<td>Base::Runtime System(system info., external device)</td>
</tr>
<tr>
<td>IO Database</td>
<td>IO::Database</td>
</tr>
<tr>
<td>IO</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

22
## Graphics & Multimedia

<table>
<thead>
<tr>
<th>Android(android.)</th>
<th>Multimedia</th>
<th>Tizen(Tizen::)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphics (Canvas)</td>
<td>Graphics</td>
<td>Graphics (Canvas)</td>
</tr>
<tr>
<td>Opengl (1.1, 2.0)</td>
<td></td>
<td>Graphics::Opengl (1.1,2.0)</td>
</tr>
<tr>
<td>Media.mediaplayer</td>
<td>Multimedia</td>
<td>Media::Player</td>
</tr>
<tr>
<td>- Audio, Video</td>
<td></td>
<td>- Audio, Video</td>
</tr>
<tr>
<td>- Streaming</td>
<td></td>
<td>- Streaming: http,rtsp</td>
</tr>
<tr>
<td>hardware.camera</td>
<td></td>
<td>Media::Camera</td>
</tr>
<tr>
<td>media:Facedetector</td>
<td></td>
<td>Uix::Vision(Facedetector, Image Recognizer, QR code)</td>
</tr>
</tbody>
</table>
# Communication

<table>
<thead>
<tr>
<th>Android(android.)</th>
<th>Tizen(Tizen::)</th>
</tr>
</thead>
<tbody>
<tr>
<td>net</td>
<td>Net</td>
</tr>
<tr>
<td>bluetooth</td>
<td>Net::Bluetooth</td>
</tr>
<tr>
<td>net.http</td>
<td>Net::Http</td>
</tr>
<tr>
<td>Java.socket</td>
<td>Net::Sockets</td>
</tr>
<tr>
<td>net.wifi</td>
<td>Net::Wifi</td>
</tr>
<tr>
<td>nfc</td>
<td>Net::Nfc</td>
</tr>
<tr>
<td>telephony</td>
<td>Telephony</td>
</tr>
<tr>
<td>Notification</td>
<td>Messaging(sms,mms,email,push receive)</td>
</tr>
</tbody>
</table>
## Web & Contents

<table>
<thead>
<tr>
<th>Android(android.)</th>
<th>Web</th>
<th>Tizen(Tizen::)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Webkit</td>
<td>Web</td>
<td>Web::Controls</td>
</tr>
<tr>
<td>Webkit.webview</td>
<td></td>
<td>Web::Json</td>
</tr>
<tr>
<td>org.json</td>
<td></td>
<td>Libxml2</td>
</tr>
<tr>
<td>org.xml, javax.xml</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# User Interface

<table>
<thead>
<tr>
<th>Android(android.)</th>
<th>Tizen(Tizen::)</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>UI</td>
</tr>
<tr>
<td>view.animation</td>
<td>Ui::Animations</td>
</tr>
<tr>
<td>View.widget</td>
<td>Ui::Controls</td>
</tr>
<tr>
<td>view.inputmethodservice</td>
<td>Ui::Ime</td>
</tr>
<tr>
<td>Activity, Intent</td>
<td>Ui::Scenes</td>
</tr>
<tr>
<td>gesture</td>
<td>UI::GestureDetector</td>
</tr>
<tr>
<td></td>
<td>Ui::Effects</td>
</tr>
<tr>
<td>hardware.sensor</td>
<td>UX</td>
</tr>
<tr>
<td>speech</td>
<td>Uix::Sensor</td>
</tr>
<tr>
<td>os.vibrator</td>
<td>Uix::Speech</td>
</tr>
<tr>
<td></td>
<td>Uix::Vibrator</td>
</tr>
</tbody>
</table>
Native UI
UI Structure – iOS

UIWindow

Navigation Bar

Tool Bar

Table View

Indicator Bar

Header

Controls

ListView

Form

Footer
UI Structure – Android to Tizen

- Indicator
- Action bar
- Widgets

- ListView
  - Home
  - Message
  - Alarm
  - Call

- Controls
  - Custom

- Footer

- Indicator bar
- Header

- Frame
UI Structure – Android to Tizen
## Controls – iOS

<table>
<thead>
<tr>
<th>Label</th>
<th>Round Rect Button</th>
<th>Segmented Control</th>
<th>Text</th>
<th>Slider</th>
<th>Switch</th>
<th>Active indicator view</th>
<th>Progress view</th>
<th>Page control</th>
<th>Stepper</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table view</td>
<td>Table view cell</td>
<td>Collection view</td>
<td>Collection view cell</td>
<td>Collection reusable view</td>
<td>Image view</td>
<td>Text view</td>
<td>Web view</td>
<td>Map view</td>
<td>Scroll view</td>
</tr>
<tr>
<td>Date picker</td>
<td>Picker view</td>
<td>Ad Banner view</td>
<td>GLKit view</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>View</td>
<td>Container view</td>
<td>Navigation bar</td>
<td>Navigation item</td>
<td>Search bar</td>
<td>Search bar and search display controller</td>
<td>Toolbar</td>
<td>Bar button item</td>
<td>Fixed space bar button item</td>
<td>Flexible space bar button item</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Controls – iOS vs. Tizen (native)

- Label
- Round Rect Button
- Segmented Control
- Text field
- Slider
- Switch
- Active indicator view
- Progress view
- Page control
- Stepper
- Table view
- Table view cell
- Collection view
- Collection view cell
- Collection reusable view
- Image view
- Text view
- Web view
- Map view
- Scroll view
- Date picker
- Picker view
- Ad Banner view
- GLKit view
- View
- Container view
- Navigation bar
- Navigation item
- Search bar
- Search bar and search display controller
- Toolbar
- Bar button item
- Fixed space bar button item
- Flexible space bar button item
Widgets – Android

- TextView
  - Large
  - Medium
  - Small
- Button
  - Small
  - OFF
- CheckBox
- Radio Button
- Checked TextView
- Spinner
- Sub Item
- Image View
- ImageButton
  - Gallery
- Media Controller
- Video View
- Time Picker
- DatePicker
- ListView
  - Sub Item
  - Expandable List
  - Sub Item
  - Grid View
  - Vertical Scroll View
- Horizontal Scroll View
- Search View
- Sliding Drawer
- Chronometer
  - 8:28:32 PM
# Android to Tizen - Layouts

<table>
<thead>
<tr>
<th>Android</th>
<th>Tizen</th>
</tr>
</thead>
<tbody>
<tr>
<td>GridLayout</td>
<td>Grid layout</td>
</tr>
<tr>
<td>Card layout</td>
<td>Card layout</td>
</tr>
<tr>
<td>RelativeLayout</td>
<td>Relative layout</td>
</tr>
<tr>
<td>LinearLayout</td>
<td>Linear layout</td>
</tr>
<tr>
<td></td>
<td>HorizontalBox</td>
</tr>
<tr>
<td></td>
<td>VerticalBox</td>
</tr>
<tr>
<td>FragmentLayout</td>
<td>Fragment layout</td>
</tr>
<tr>
<td>TableLayout</td>
<td>Table layout</td>
</tr>
<tr>
<td></td>
<td>TableView</td>
</tr>
</tbody>
</table>
## Android to Tizen - Form widgets

<table>
<thead>
<tr>
<th>Android</th>
<th>Tizen</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text View</td>
<td>Text</td>
</tr>
<tr>
<td>Label</td>
<td>TextBox Label</td>
</tr>
<tr>
<td>Button</td>
<td>Button</td>
</tr>
<tr>
<td>On/Off button Switch</td>
<td>On/Off Button</td>
</tr>
<tr>
<td>Checkbox</td>
<td>Checkbox Button (radio Style)</td>
</tr>
<tr>
<td>Spinner</td>
<td>Expanding Items</td>
</tr>
<tr>
<td>Rating, Contact badge</td>
<td>Pre-Made widgets</td>
</tr>
</tbody>
</table>

![Diagram showing Android and Tizen widgets comparison](image-url)
# Android to Tizen - Layouts

<table>
<thead>
<tr>
<th>Android</th>
<th>Tizen</th>
</tr>
</thead>
<tbody>
<tr>
<td>ListView</td>
<td>List View</td>
</tr>
<tr>
<td>ExpandableList</td>
<td>ExpandableList</td>
</tr>
<tr>
<td>GridView</td>
<td>Grid View</td>
</tr>
<tr>
<td>ScrollView</td>
<td>Scrollable page</td>
</tr>
<tr>
<td>SearchView</td>
<td>Search</td>
</tr>
<tr>
<td>Tab</td>
<td>Tab</td>
</tr>
<tr>
<td>WebView</td>
<td>Web view</td>
</tr>
</tbody>
</table>
## Android to Tizen – Image, Medias, Data and Time

<table>
<thead>
<tr>
<th>Android</th>
<th>Tizen</th>
<th>Tizen</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImageView</td>
<td>Image View</td>
<td>Panel</td>
</tr>
<tr>
<td>ImageButton</td>
<td>Image Button</td>
<td>Button(with image)</td>
</tr>
<tr>
<td>Gallery</td>
<td>Gallery</td>
<td>Gallery</td>
</tr>
<tr>
<td>VideoView</td>
<td>Video View</td>
<td>OverlayRegion</td>
</tr>
<tr>
<td>TimePicker</td>
<td>Time Picker</td>
<td>EditTime</td>
</tr>
<tr>
<td>DatePicker</td>
<td>Date Picker</td>
<td>EditDate</td>
</tr>
<tr>
<td>CalendarView</td>
<td>Date &amp; Time Picker</td>
<td>DateTimePicker CustomControl</td>
</tr>
<tr>
<td>Chronometer, Digital Clock</td>
<td>Pre-Made widgets</td>
<td>CustomControl</td>
</tr>
</tbody>
</table>
UI builder & Workflow – Tizen
Interface Builder & Storyboard – iOS
Using Interface Builder – Android
Application Life Cycle
Tizen

Create → Initializing

- Call the OnAppInitializing() method.
- Call the OnAppInitialized() method.
- Call the OnAppWillTerminate() method.
- Exit event loop.
- Call the OnAppTerminating() method.
- Destroy allocated resources.

OnAppInitializing() return true

OnAppInitialized() return false

Running → Terminating

OnAppWillTerminate() return false

OnAppWillTerminate() return true

Terminated

Terminating

OnAppInitializing() return false
Frame states
iOS to Tizen
iOS App Lifecycle - Status

Foreground

Active

Inactive

Not Running

Background

Running

Suspended
Launching an app

- Foreground:
  - Active
  - Inactive
  - Not Running
  - Running
  - Suspended

- Background
Launching an app

iOS

- **Foreground**
  - Active
  - Inactive
  - Not Running

- **Background**
  - Running
  - Suspended

**DidFinishLaunchingWithOptions**: application: didFinishLaunchingWithOptions:

**OnAppInitializing**: OnAppInitializing()
Launching an app

iOS

applicationDidBecomeActive:

foreground:
  - Active
  - Inactive
  - Not Running
  - Running
  - Suspended

background:

Tizen

OnAppInitialized()
Switching from an app

iOS

- Foreground:
  - Active
  - Inactive
  - Not Running
- Background:
  - Running
  - Suspended

Tizen
Switching from an app

iOS

applicationWillResignActive:

Foreground
- Active
- Inactive

Not Running

Running

Suspended

Background

Tizen

OnWindowDeactivated()
Entering background

iOS

Tizen

Background

Foreground

Active

Inactive

Not Running

Running

Suspended

applicationDidEnterBackground:

OnBackground()

OnWindowStateChanged()
Entering background

iOS

Background

Forefront

Active

Inactive

Not Running

Running

Suspended

Tizen

OnAppCheckPointing()
Terminating

iOS

Tizen

Foreground

Active

Inactive

Not Running

Running

Suspended

Background
Terminating

iOS

applicationWillTerminate:

- Active
- Inactive
- Not Running
- Running
- Suspended

Tizen

- OnAppTerminating()
- OnTerminating()
Terminating

iOS

Tizen

Foreground
- Active
- Inactive

Background
- Running
- Suspended

Not Running
Android to Tizen
Life Cycle - Android vs. Tizen

OnCreate() → Created → OnAppInitializing() → Frame::OnInitializing()
Life Cycle - Android vs. Tizen

- **OnStart()**
- **Started**
- **Resumed**
- **Paused**

Frame::OnWindowStateChanged()
Life Cycle - Android vs. Tizen

Created

Started

Resumed

Paused

Window::OnWindowActivated()
Life Cycle - Android vs. Tizen

Control::FocusLost()

Created

Started

Resumed

Paused

OnPause()
Android App Lifecycle - Status

UIApp::OnBackground()
Window::OnWindowDeactivated()
Android App Lifecycle - Status

OnAppWillTerminate()
Frame::OnTerminating()
OnAppTerminating()

Foreground
- Started
- Resumed
- Paused

Background
- Stopped
- Destroyed

- OnDestory()
Android App Lifecycle - Status

UiApp::OnForeground()
Frame::OnWindowActivated()
iOS to Tizen
Delegate vs. Listener

**iOS**
- Handle the events from UI controls/ outlets
- Protocol methods

**Tizen**
- Handle events from the registered controls
- Pre-defined virtual methods
Delegate vs. Listener

iOS

- Handle the events from UI controls/outlets
- Protocol methods
- Connection between control, action and delegate has to be made from Interface builder

Tizen

- Handle events from the registered controls
- Pre-defined virtual methods
- Connection between control and listener normally made by calling `AddXXXListener()` function
#import <CoreMotion/CoreMotion.h>

@interface RootViewController : UIViewController {
    CMMotionManager *motionManager;
}
@property (nonatomic, retain) CMMotionManager *motionManager;

#import "RootViewController.h"
@implementation RootViewController
@synthesize motionManager;

(Void (^accelerometerHandler)(CMAccelerometerData *, NSError *)) = ^(CMAccelerometerData *accelerometerData, NSError *error) {
    NSLog(@"X= %.04f, Y=%.04f, Z=%.04F", accelerometerData.acceleration.x, accelerometerData.acceleration.y, accelerometerData.acceleration.z);
};

[self.motionManager startAccelerometerUpdatesToQueue: [NSOperationQueue mainQueue] withHandler: accelerometerHandler];
Android to Tizen
Tizen Listener vs. Android Listener

Tizen

class myClickListener : public Tizen::Ui::IActionEventListener {

  Button *pButtonOK
  //...
  pButtonOk->addActionEventListener(*myClickListener);

Android

class myClickListener implements OnClickListener {

  Button button;
  //...
  Button.setOnClickListener(myClickListener)

Thank you TIZEN™ DEVELOPER CONFERENCE 2013 SAN FRANCISCO