Tizen 2.3 API Overview

Hobum (Vincent) Kwon
Principal Engineer
at Samsung Electronics
Tizen Architecture & Public APIs & 3rd Party Applications

- **Web API**
  - Web is the primary application development environment for Tizen

- **Native API**
  - Fast, light-weight, scalable native applications can run smoothly on any device
Tizen 2.3 beta
Native API Update
Introducing new Native API in Tizen 2.3 Mobile Profile

- **Mature**
  - Technology was already in Tizen since 1.0
  - Now in SDK and Compliance for 3rd party developers

- **Powerful Graphics**
  - Powered by Enlightenment Foundation Libraries (EFL)
  - High performance, scalable, customizable styles

- **Lightweight**
  - Light-weight enough to fit in every Tizen Profile

- **More open source libraries**
  - EFL, Sqlite, openssl, Curl, json-glib, libexif, etc.
EFL Overview

- **EFL (Enlightenment Foundation Libraries)**
  - A collection of libraries supporting 2D and 2.5D UI rendering for Tizen platform
  - Built by the same team working on Enlightenment project
  - Built for the purpose of making E17 (Enlightenment 0.17)
  - Always focused on staying lean and still providing fanciness
  - Provides a set of libraries for adding common GUI widgets
  - Handling and routing input, managing data, communications and the main-loop

- **Enlightenment Open source Project** ([http://www.enlightenment.org](http://www.enlightenment.org))
  - A whole suite of libraries to help you create beautiful user interfaces with much less work
Create a Tizen Native Application Project – With UI or Not

▪ First download Tizen 2.3 beta SDK and select Mobile Profile
  • [https://developer.tizen.org/ko/downloads/tizen-sdk?langredirect=1#2.3](https://developer.tizen.org/ko/downloads/tizen-sdk?langredirect=1#2.3)

▪ For UI application (Applications with UI)
  • Developed using EFL for user interface and other APIs to utilize full mobile device features
  • Use template in the SDK: File → Tizen Native Project → Template → UI Application

▪ For Service application (Applications without UI)
  • Can use all Native API except UI related (EFL, OpenGL ES, ... )
  • Use template in the SDK: File → Tizen Native Project → Template → Service Application
### Application states

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>READY</td>
<td>The application is launched</td>
</tr>
<tr>
<td>CREATED</td>
<td>The application starts the main loop</td>
</tr>
<tr>
<td>PAUSED</td>
<td>The application is running but invisible to users</td>
</tr>
<tr>
<td>RUNNING</td>
<td>The application is running and visible to users</td>
</tr>
<tr>
<td>TERMINATED</td>
<td>The application is terminated</td>
</tr>
</tbody>
</table>

- **State transition callbacks should be set up before starting main loop**
## Tizen Native Applications – UI Application Lifecycle (2/4)

### Application life-cycle callbacks

<table>
<thead>
<tr>
<th>Callback</th>
<th>Description</th>
<th>Action (Example)</th>
</tr>
</thead>
<tbody>
<tr>
<td>app_create_cb</td>
<td>Hook to take necessary actions before main event loop starts</td>
<td>UI generation code</td>
</tr>
<tr>
<td>app_pause_cb</td>
<td>Hook to take necessary actions when application becomes invisible</td>
<td>Releasing memory/resources</td>
</tr>
<tr>
<td>app_resume_cb</td>
<td>Hook to take necessary actions when application becomes visible</td>
<td>Re-allocating resources</td>
</tr>
<tr>
<td>app_terminate_cb</td>
<td>Hook to take necessary actions when application is terminating</td>
<td>Release all resources</td>
</tr>
<tr>
<td>app_control_cb</td>
<td>Hook to take necessary actions for responding to a launch request</td>
<td>Required action</td>
</tr>
</tbody>
</table>
Who can launch applications?

- Users through the application launcher
- Another application which needs to perform a specific operation (AppControl)

```c
static void app_control(app_control_h app_control, void *user_data) {
    struct appdata *ad = (struct appdata *)user_data;
    char *operation;
    char *uri;
    char *mime_type;
    app_control_get_operation(app_control, operation);
    if (!strcmp(operation, APP_CONTROL_OPERATION_VIEW)) {
        app_control_get_uri(app_control, &uri);
        app_control_get_mime(app_control, &mime_type);
        if (uri && !strcmp(mime_type, "image/jpg"))
            display_image_file(ad, uri); // display a specific image
    }
}
```
### Additional callbacks for system events

<table>
<thead>
<tr>
<th>Callback</th>
<th>Description</th>
<th>Action (Example)</th>
</tr>
</thead>
<tbody>
<tr>
<td>app_low_memory_cb</td>
<td>Hook to take necessary actions in low memory situations</td>
<td>Save data into a persistent memory</td>
</tr>
<tr>
<td>app_low_battery_cb</td>
<td>Hook to take necessary actions in low battery situations</td>
<td>Stop heavy cpu/power consumption</td>
</tr>
<tr>
<td>app_device_orientation_cb</td>
<td>Hook to take necessary actions for handling a device orientation change</td>
<td>Change display orientation</td>
</tr>
<tr>
<td>app_language_changed_cb</td>
<td>Hook to take necessary actions for handling a language change event</td>
<td>Refresh display with a new language</td>
</tr>
<tr>
<td>app_region_format_changed_cb</td>
<td>Hook to take necessary actions for handling a region change event</td>
<td>Update time to show timezone change</td>
</tr>
</tbody>
</table>
### Tizen Native Applications – Service Application Lifecycle

**Application life-cycle callbacks**

<table>
<thead>
<tr>
<th>Callback</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>service_app_create_cb</td>
<td>Hook to take necessary actions before main event loop starts</td>
</tr>
<tr>
<td>service_app_terminate_cb</td>
<td>Hook to take necessary actions when application is terminating</td>
</tr>
<tr>
<td>service_app_control_cb</td>
<td>Hook to take necessary actions for responding to a launch request</td>
</tr>
<tr>
<td></td>
<td>(without UI, you still need to get app control to take a certain action)</td>
</tr>
</tbody>
</table>

**For system events, same as UI Application**

- Memory, battery, orientation, language, region format change callbacks
Tizen 2.3 beta
Native API Layout
Public Native API | Definition

- The Public Native API is defined in the Tizen Native API Reference in the SDK
  - You can find at SDK’s Help Contents → Tizen Mobile Native App Programming → API Reference
  - The API reference will be available at developer.tize.org at Tizen 2.3 official release
  - Must only use documented Public Native API for compatibility reason

- The APIs are originated from core subsystems

- Revised and improved over the past few months for API usability
  - Continuing effort to lower the learning curve for the new Public Native API
Native API Layout

Telephony:
- Telephony Info

Location:
- Location Manager

Web:
- WebView

Base:
- i18n
- glib2
- sqlite
- C++ StdLib
- minimizip
- libEXIF
- libxml2
- eglibc

Network:
- Bluetooth
- Application Manager
- Notification
- Wi-Fi
- Badge
- Message Port

Application Framework:
- Service Application
- Package Manager
- Shortcut
- Bundle
- Data Control

Messaging:
- Push
- Messages
- Email

Social:
- Contacts
- Calendar
- Account Manager

UI:
- EFL
- EFL Util
- TBM

Graphics:
- OpenGL-ES 1.1
- Fontconfig & Freetype
- cairo

Multimedia:
- Media Codec
- Camera
- Recorder
- Sound Manager
- Audio I/O
- Image Util
- WAV Player
- Metadata Extractor
- Video Util

Content:
- Media Content
- MIME Type
- Download
- Radio
- openAL

Security:
- openssl
- Key Manager

System:
- System Information
- System Settings
- Media-Key

Legend:
- Tizen Native Modules
- Open-source Modules
## Supported Open Source Libraries (1/2)

<table>
<thead>
<tr>
<th>Lib</th>
<th>Version</th>
<th>Why we need to open this library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>EFL</td>
<td>1.7</td>
<td>EFL is the fundamental set of libraries underlying the Native API</td>
</tr>
<tr>
<td>libEXIF</td>
<td>0.6.21</td>
<td>Exif is an image file format used by camera and scanner devices (extends existing formats such as jpeg and tiff). Many Tizen devices have a camera and emit this format, libexif allows decoding</td>
</tr>
<tr>
<td>Json-glib</td>
<td>0.10.4</td>
<td>Json-glib is a library for serializing and deserializing Javascript Object Notation (JSON) using Glib and Gobject data types.</td>
</tr>
<tr>
<td>Eglibc</td>
<td>2.13</td>
<td>Standard C library, needs to be available to programs written in ISO C language</td>
</tr>
<tr>
<td>Glib</td>
<td>2.32.3</td>
<td>Application building blocks which add data types and other programming facilities for C-language programs</td>
</tr>
<tr>
<td>Curl</td>
<td>7.28.1_24</td>
<td>A client-side URL transfer library supporting http, https, ftp, file URIs and many more protocols. Allows applications to perform url-related activities without having to involve a web browser</td>
</tr>
<tr>
<td>libXML2</td>
<td>2.7.8</td>
<td>Library for parsing xml documents</td>
</tr>
<tr>
<td>Fontconfig</td>
<td>2.9.0</td>
<td>Font-handling library to let applications find a font or a closely matching font</td>
</tr>
<tr>
<td>Freetype</td>
<td>2.4.9</td>
<td>Text-rendering library</td>
</tr>
<tr>
<td>Minizip</td>
<td>1.2.5</td>
<td>Lightweight library building on top of zlib for processing files in the zip format</td>
</tr>
<tr>
<td>Sqlite</td>
<td>3.7.13</td>
<td>Implements a lightweight sql database within a library, widely used for embedded client-local storage.</td>
</tr>
<tr>
<td>Cairo</td>
<td>1.12.14</td>
<td>Library for 2-D vector graphics drawing</td>
</tr>
<tr>
<td>openssl</td>
<td>1.0.1g_1</td>
<td>Library implementation of secure sockets layer (ssl) and transport layer security (tls) to enable secure internet communications</td>
</tr>
</tbody>
</table>
Supported Open Source Libraries (2/2)

<table>
<thead>
<tr>
<th>Lib</th>
<th>Version</th>
<th>Why we need to open this library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenAL</td>
<td>1.13</td>
<td>Audio API designed for efficient rendering of 3-D positional audio.</td>
</tr>
<tr>
<td>OpenGL ES</td>
<td>1.1., 2.0</td>
<td>Library for rendering 3-D and 2-D graphics in embedded systems</td>
</tr>
<tr>
<td>C++ Standard Library</td>
<td>3.4.16 (GCC 4.6.4)</td>
<td>Standard C library, needs to be available programs written in ISO C++ language</td>
</tr>
<tr>
<td>libOAuth</td>
<td>0.9.4</td>
<td>Functions implementing the OAuth Core RFC 5849 protocol</td>
</tr>
</tbody>
</table>
Ready by Tizen 2.3 official release

- **More APIs:**
  - NFC, STT/TTS, media codec, window level adjustment, Tizen ID, etc.
  - Error name printer

- **More Documents:**
  - More programming guides and tutorials including EDJE
  - Kind explanation for sample apps

- **Tizen Compliance Support:**
  - Tizen Certification Specification for Tizen Native API
  - Tizen Certification Tests for Tizen Native API
Tizen 2.3 beta
Native API Modules
Native API | Application Framework / Base

- Provides
  - Managing the main event loop of an application or background application, managing application state changes, launching other applications using the application name, URI, or MIME type (Application, Service Application)
  - Storing and retrieving information related to packages installed on the device (Package manager)
  - Information about applications (Application manager)
  - Managing notifications (Notification)
  - Passing messages between applications (Message-port)
  - Simple string-based dictionary ADT (Bundle)
  - Exchanging specific data between applications (Data Control)
  - Flexible generation of number or date format patterns, formatting and parsing dates/number for any locale (i18n)
Native API | System / Security

▪ Provides system and device management features
  • Interfaces for accessing devices such as sensors, USB, MMC, battery, CPU, and display (Sensor, Device)
  • Getting information about the device (System Information, Runtime Information)
  • Getting system settings containing miscellaneous system preference (System Settings)
  • Sending log output for debug activities (dlog)
  • Getting information about storage (Storage)
  • Providing a secure repository protected by user’s passwords for keys, certificates, and sensitive data (Key-manager)
  • Retrieving and displaying privilege information (Privilege-Info)
Native API | Location

- Provides location-based services (LBS)
  - Position information, satellite, GPS status, geofencing (Location Manager)
Native API | Network

- Provides network and connectivity related functionalities
  - Managing modem data connections (Connection)
  - Managing Bluetooth devices (Bluetooth)
  - Managing Wi-Fi and monitoring the state of Wi-Fi (Wi-Fi)
Native API | Telephony

- Provides cellular functionalities communicating with a modem
  - Managing call-related information and services, obtaining information from a SIM card, accessing the cellular network status information (Telephony-information)
Native API | Messaging

- Provides messaging services
  - Creating, setting properties (recipients, body), and sending SMS, MMS messages (*Messages*)
  - Managing E-mails (*Email*)
  - Push service (*Push*)
Provides

• Encoding, decoding, and transforming images (Image Util)
• Transcoding a media file (Video Util)
• Recording from the audio device and playing raw audio data (Audio I/O)
• Playing multimedia contents from a file, network, and memory (Player)
• Playing the tone and Waveform audio files (Tone Player, Wav Player)
• Controlling a camera device (Camera)
• Recording audio and video (Recorder)
• Accessing Radio (Radio)
• Extracting meta data from an input media file (Metadata-extractor)
• Directly accessing media codes on the device (Media-codec)
• Handling AV packet buffer for interworking between multimedia framework modules (Media tool)
Native API | Content / Context

Provides
- Managing information about media files (Media Content)
- Downloading the contents of a URL to the storage asynchronously (Download)
- Mapping MIME types to file extensions and vice versa (Mime-type)
- Controlling information of the user and device including motions, activities (Activity Recognition, Gesture Recognition)
Native API | Social

▪ Provides PIM-related services
  • Managing account information on the device (Account Manager)
  • Managing calendar events and accessing calendar database (Calendar)
  • Managing contacts and contact groups and accessing contact database (Contacts)
Native API | UIX

- **Provides UI features & interaction services**
  - Providing surface for Tizen (**TBM Surface**)
  - Getting and setting the priority order of the notification window (**EFL-util**)
Native API | Web

- Provides Web browser features
  - Displaying and controlling Web pages, such as browsing, tracking browsing history, and downloading Web content (WebView)
10 Must-know Tips
For Tizen Native App Development
Tip 1: What you are looking for is in SDK Help Contents (1/2)

- All the Native API documents are in the [Help Contents] and will be added more
Tip 1: What you are looking for is in SDK Help Contents (2/2)

- **Basics of Native App Programming**
  - Where you first visit before write an app
    - Application Model
    - Multi-lingual guide
    - Brief UI programming with EFL guide
    - Hello world example

- **Programming Guide**
  - Module overview and major feature introduction

- **Tutorials**
  - Step by step API usage example
  - Tutorial helper app will be added soon to run tutorial codes by copy & paste to the SDK
Tip 2: Open SDK help contents in a browser

- Go to [Window ➔ Preference ➔ Help ➔ Open help contents]
- Select [In an external browser], then you can find info more easily
Tip 3: Refer to sample applications in the SDK (1/3)

- Go to [File → New → Project → Tizen → Tizen Native Project]
- Select Sample what you want to see
Tip 3: Refer to sample applications in the SDK (2/3)

- Check out samples in the SDK

![NotificationManager Sample Overview](image-url)

**NotificationManager Sample Overview**

The NotificationManager sample demonstrates how you can present a notification to the user. The following figure illustrates the main view of the NotificationManager.

- **Notify**
- **Notify(normal)**
- **NotifyByAppControl**
- **RemoveNotification**
- **NotifyByAppId**
- **RemoveNotificationByAppId**
- **SetBadgeNumberByAppId**

The application opens with the Notification tab, which provides access to various notification features. Click the list items to view the features. The Ongoing notification tab provides access to ongoing notifications.
### Tip 3: Refer to sample applications in the SDK (3/3)

#### Native UI Apps
- Alignment UI
- Animation UI
- ApplicationStore
- Calculator
- Clock
- Media
- EDC Format
- Email UI
- Gallery UI

#### Native Apps
- UI Controls
- Notification
- Evas GL
- Piano
- File
- Calculator
- GLView
- Contacts
- Hybrid application
- Sensors
- Media
- Scheduler
- More to come
Tip 4: Don’t forget Non-API bound privileges (1/2)

- If your app access the Internet, add the below privilege
- Non-API bound privilege need media storage access and external storage access also
Tip 4: Don’t forget Non-API bound privileges (2/2)

- [ Help Contents → Mobile Native App Programming → Non API Bound Privileges ]

Non-API Bound Privileges

Tizen application privileges are loosely bound to APIs, so most of the privileges can be identified by the APIs that the application calls. However, there are some privileges that are not coupled with the Tizen APIs. To allow easy identification, those privileges are mapped to corresponding system resources - same as other privileges.

Most of the mobile devices use a cellular network for IP communication. However, the cellular network can cause data costs and an application that sends data through the internet can be crucial for user privacy. Due to the importance of the functionality, a privilege for controlling application internet access has been added.

The new privilege is coupled with IP addresses of the destination and source of the IP packets. If your socket is connecting to or listening for any IP address except 127.0.0.1, this privilege is required to communicate properly. If your application does not have this privilege, the connection is blocked in the kernel layer and returns an error in the accept() function as the permission is denied. If you are listening to a socket, you never receive any packets from the outside without errors on the socket functions.

If you are using the listen() and connect() functions between the local socket interface (127.0.0.1), you cannot connect to a random application (due to sandboxing) no matter how you add this privilege. However, you can connect between multiple processes of the same application binary.

- [http://tizen.org/privilege/internet]
- [http://tizen.org/privilege/mediastorage]
- [http://tizen.org/privilege/externalstorage]
Tip 5: Use Tizen Manifest Editor for tizen-manifest.xml (1/2)

- Easy to configure App Control (below), Privileges, Features, etc.
Tip 5: Use Tizen Manifest Editor for tizen-manifest.xml (2/2)

- To learn elements in the manifest xml file, visit the below page in the SDK Help Contents
Tip 6: Things to remember when see the API Reference (1/2)

- In most case, [ Returns ] indicates 0 on success, negative on error
- Carefully read [ Remarks ]. e.g., string must be freed
Tip 6: Things to remember when see the API Reference (2/2)

- Understand hierarchy of API modules (logical categorized)
Tip 7: Security Profile

- Need to add Security Profile when you install SDK
- Suspect missing or wrong security profile if launching app failed with security error shown
Tip 8: If you are new to EFL, here comes docs for you (1/2)

- Visit official EFL documents page (samples):

- If you are interested in UI layout (EDJE), following site will help you alot:

- Be friend with UI control sample (on the right)
- In the SDK help contents, check out below link
Tip 8: If you are new to EFL, check out EFL documents (2/2)

- More in-depth guide in Programming Guide in Help Contents
Tip 9: Use SDB to push or pull files to your app project (1/2)

- Pull or push files (e.g., png files) from or to your project
- Your project is located /opt/usr/apps/APP_ID for SDK emulator
- You need to understand app directory structure (next page)
Tip 9: Use SDB to push or pull files to your app project (2/2)

- **Application directory policy in Help Contents**
  - For all pre-existing (not created at runtime), read-only file, if only used by own app, place it in [ res ] folder
  - For exclusive runtime data, use data folder
  - If need to share with other apps, such as icons, place it in [ shared/res ] for read-only data
Tip 10: Understand main loop for Native Application

- Ecore main loop is used for UI Application (as of 2.3 beta)
- Service applications will use ecore main loop (at the 2.3 official release)
- Can be used with GMainLoop libraries (handled internally)
Tizen 2.3 beta
Web API Update
Brief Overview: Tizen Web Application and Web Runtime

- **Web Application**
  - Using Web based technologies
  - Accessing local device / platform resources

- **Web APIs**
  - W3C/HTML5 and de-facto Supplementary APIs
  - Tizen Web Device APIs
  - Tizen Web UI FW

- **Web Runtime**
  - Lifecycle Management of web applications
  - Execution of web application
  - Access to device resources via JS API
  - Device and Platform integration
Brief Overview: Tizen Web Device APIs

- Apps need access to device features outside of the HTML
- Can access various device features via Web Device API
- Apps to work like native apps with minimum porting effort
Web Application Packaging

- **Packaging**
  - W3C Widget Packaging and Configuration
  - An example of an app structure (right)

- **Config.xml**
  - Web App Configurations
    - Application basic info (ID, icon, name, ...)
    - Localization
    - Preferences
    - Feature
    - Privilege
### Tizen 2.3 Web Device API Modules (as of 2.3 beta)

<table>
<thead>
<tr>
<th><strong>Tizen Common</strong></th>
<th><strong>Multimedia</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tizen</td>
<td>Sound</td>
</tr>
<tr>
<td></td>
<td>Provides a way to control sound volume level</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Application</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Alarm</td>
</tr>
<tr>
<td>Schedules an application to be launched at a specific time</td>
</tr>
<tr>
<td>Application</td>
</tr>
<tr>
<td>Provides information about applications and controls applications</td>
</tr>
<tr>
<td>Data Control</td>
</tr>
<tr>
<td>Provides information about packages and install/uninstall packages</td>
</tr>
<tr>
<td>Package</td>
</tr>
<tr>
<td>Provides information install/uninstall package and get information about installed packages</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Communication</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bluetooth</td>
</tr>
<tr>
<td>Manages Bluetooth device and supports RFCOMM and HDP</td>
</tr>
<tr>
<td>Messaging</td>
</tr>
<tr>
<td>Sends and receives SMS, MMS and Email message</td>
</tr>
<tr>
<td>Network Bearer Selection</td>
</tr>
<tr>
<td>Provides interfaces and methods for users to set network bearer for a specific IP address</td>
</tr>
<tr>
<td>NFC</td>
</tr>
<tr>
<td>Manages NFC device and detects NFC tag and peer</td>
</tr>
<tr>
<td>Push</td>
</tr>
<tr>
<td>Receives push notifications from push server</td>
</tr>
<tr>
<td>Secure Element</td>
</tr>
<tr>
<td>Provides interfaces and methods for access to Secure Elements.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Content</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Content</td>
</tr>
<tr>
<td>Discovers multimedia content (such as images, videos or music)</td>
</tr>
<tr>
<td>Download</td>
</tr>
<tr>
<td>Downloads remote objects by HTTP request</td>
</tr>
<tr>
<td>Exif</td>
</tr>
<tr>
<td>Provides interfaces and methods for manipulating EXIF data from JPEG file</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Input / Output</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Archive</td>
</tr>
<tr>
<td>Provides interfaces and methods to create an archive file as well as various other kinds of manipulation (e.g. extract files, add a file to an archive file)</td>
</tr>
<tr>
<td>Filesystem</td>
</tr>
<tr>
<td>Provides access to the file system of a device. This API might be obsolete in the future when Web console APIs are extended to access system-sensitive files by Web applications.</td>
</tr>
<tr>
<td>Message Port</td>
</tr>
<tr>
<td>Provides the functionality for communication with other applications.</td>
</tr>
</tbody>
</table>
New 2.3 Web Device API Features

- **FM Radio API**
  - Turn on/off, tune on/off

- **Sensors**
  - proximity, light, pressure, magnetic, ultraviolet API

- **Human activity monitor**
  - pedometer, wrist up

- **Archive**
  - create / extract zip archive

- **Content Playlist**
  - manage your playlist (audio, video)

- **Account**
  - Account provider and accounts interfaces

- **Sound**
  - Adjust various type of sounds on the device

- **Badge**
  - Update badge (number on icon in menu screen)