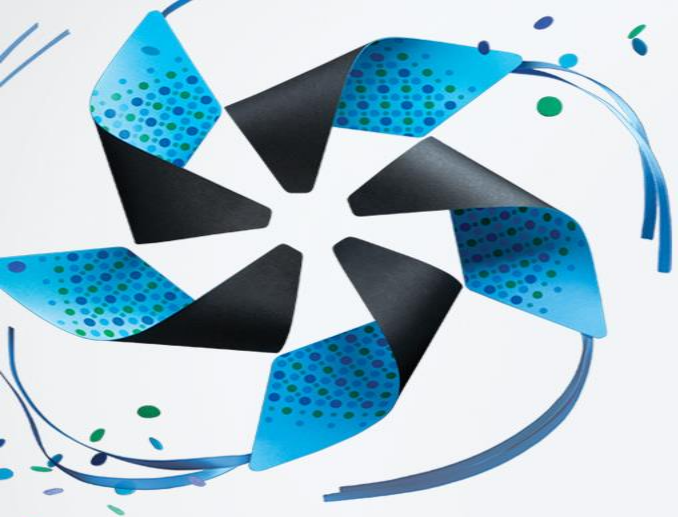
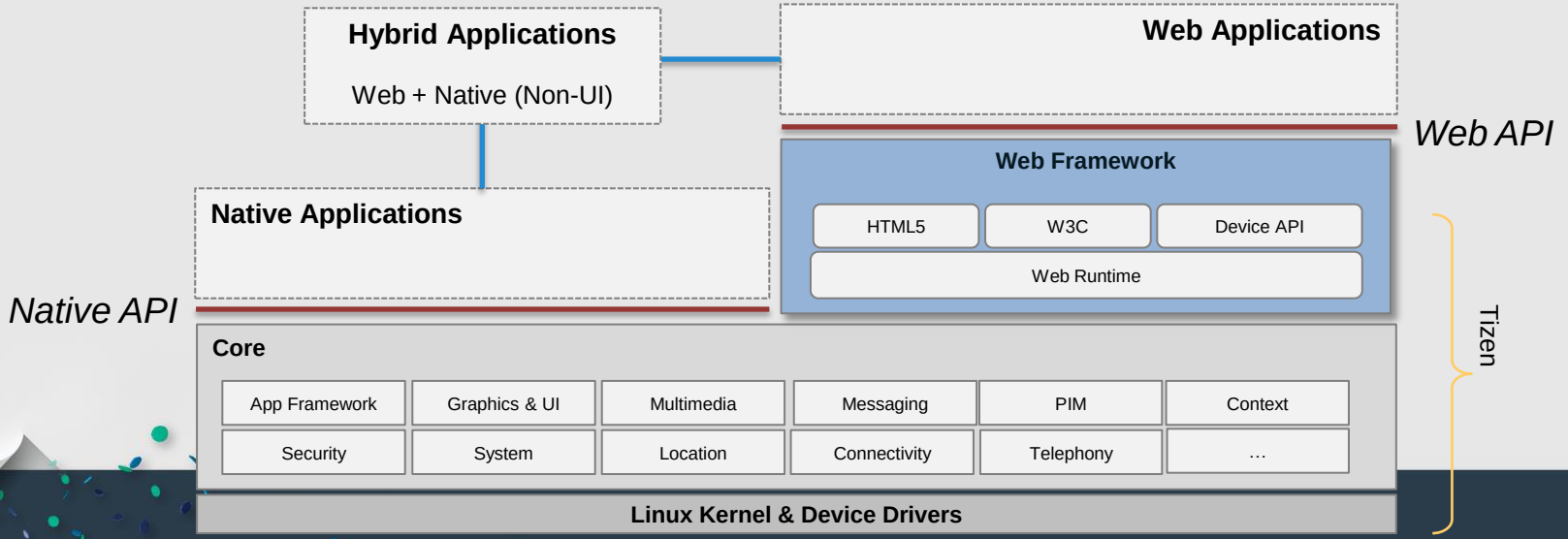


Tizen 2.3 API Overview

Hobum (Vincent) Kwon
Principal Engineer
at Samsung Electronics

Tizen Architecture & Public APIs & 3rd Party Applications

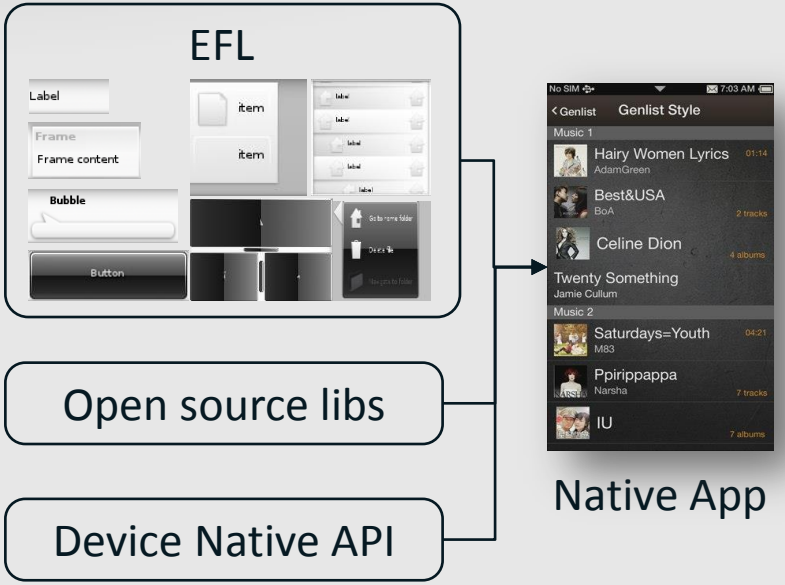
- Web API
  - Web is the primary application development environment for Tizen
- Native API
  - Fast, light-weight, scalable native applications can run smoothly on any device



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)**
  - 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

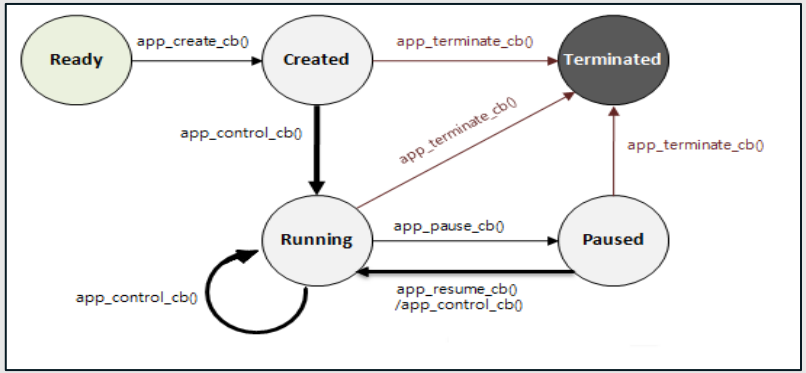
- For UI application (Applications with UI)
  - Developed using EFL for user interface and other APIs to utilize full mobile device features
  - Use template in the SDK: File → Tizen Native Project → Template → UI Application

- 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

Tizen Native Applications – UI Application Lifecycle (1/4)

- Application states

State	Description
READY	The application is launched
CREATED	The application starts the main loop
PAUSED	The application is running but invisible to users
RUNNING	The application is running and visible to users
TERMINATED	The application is terminated



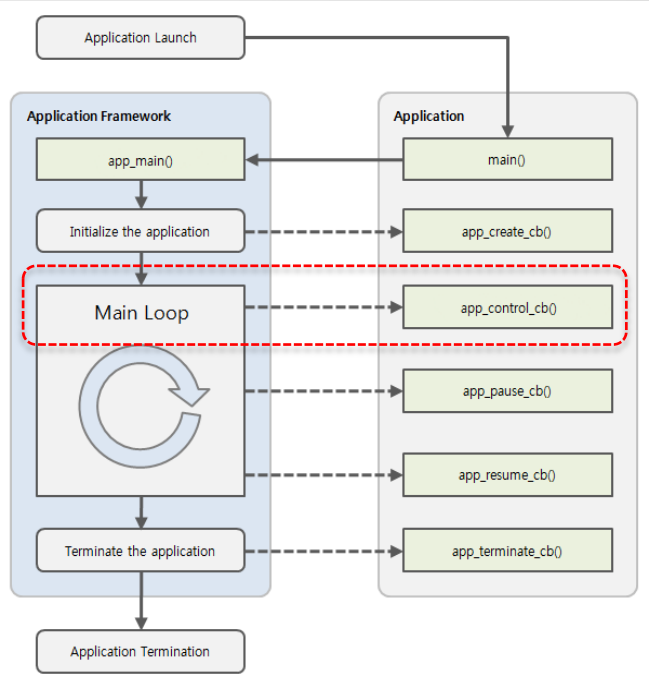
- State transition callbacks should be set up before starting main loop

Tizen Native Applications – UI Application Lifecycle (2/4)

- Application life-cycle callbacks

Callback	Description	Action (Example)
app_create_cb	Hook to take necessary actions before main event loop starts	UI generation code
app_pause_cb	Hook to take necessary actions when application becomes invisible	Releasing memory/resources
app_resume_cb	Hook to take necessary actions when application becomes visible	Re-allocating resources
app_terminate_cb	Hook to take necessary actions when application is terminating	Release all resources
app_control_cb	Hook to take necessary actions for responding to a launch request	Required action

Tizen Native Applications – UI Application Lifecycle (3/4)



- Who can launch applications?
  - Users through the application launcher
  - Another application which needs to perform a specific operation (AppControl)

```
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
             ...
```

Tizen Native Applications – UI Application Lifecycle (4/4)

- Additional callbacks for system events

Callback	Description	Action (Example)
app_low_memory_cb	Hook to take necessary actions in low memory situations	Save data into a persistent memory
app_low_battery_cb	Hook to take necessary actions in low battery situations	Stop heavy cpu/power consumption
app_device_orientation_cb	Hook to take necessary actions for handling a device orientation change	Change display orientation
app_langage_changed_cb	Hook to take necessary actions for handling a language change event	Refresh display with a new language
app_region_format_changed_cb	Hook to take necessary actions for handling a region change event	Update time to show timezone change

Tizen Native Applications – Service Application Lifecycle

- Application life-cycle callbacks

Callback	Description
service_ app_create_cb	Hook to take necessary actions before main event loop starts
service_app_terminate_cb	Hook to take necessary actions when application is terminating
service_app_control_cb	Hook to take necessary actions for responding to a launch request (without UI, you still need to get app control to take a certain action

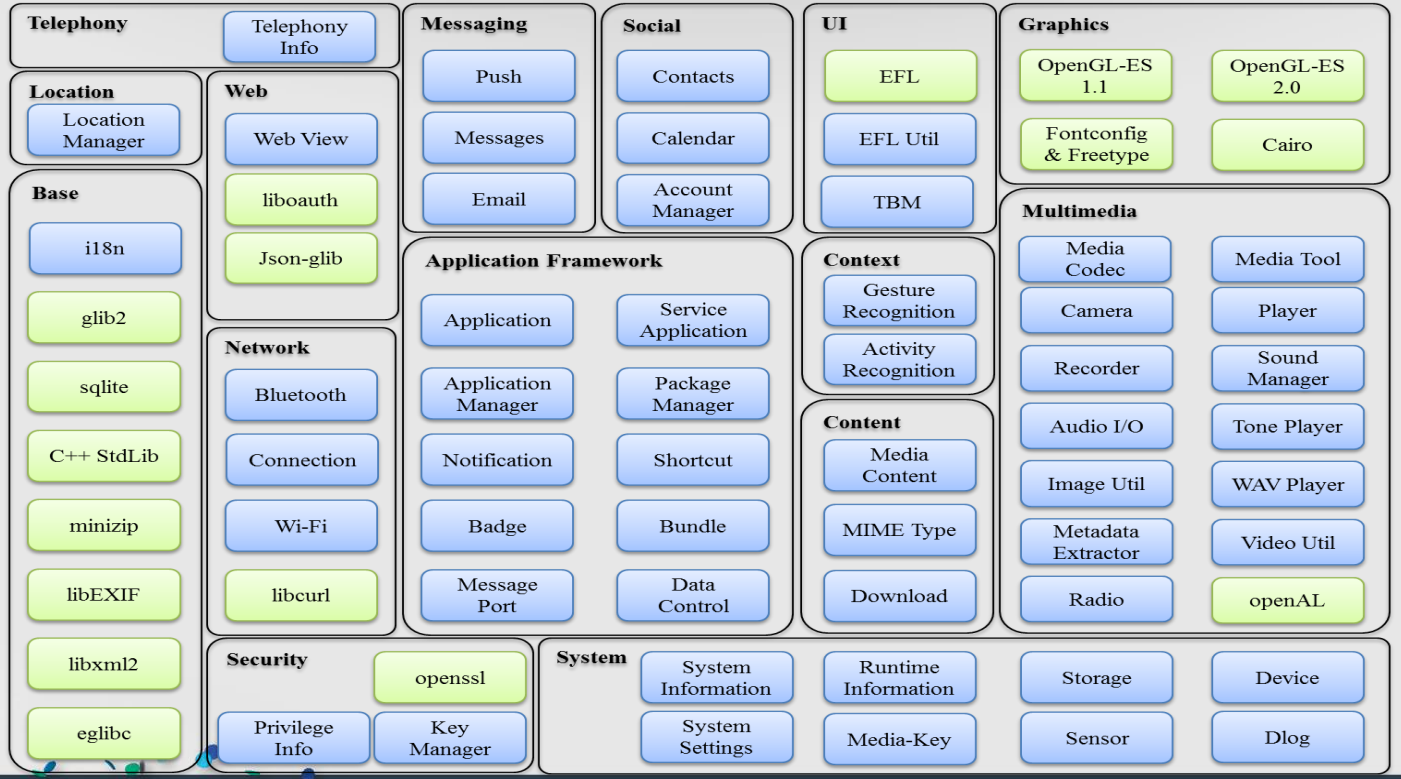
- For system events, same as UI Application
  - Memory, battery, orientation, language, region format change callbacks

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



Supported Open Source Libraries (1/2)

Lib	Version	Why we need to open this library?
EFL	1.7	EFL is the fundamental set of libraries underlying the Native API
libEXIF	0.6.21	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
Json-glib	0.10.4	Json-glib is a library for serializing and deserializing Javascript Object Notation (JSON) using Glib and Gobject data types.
Eglibc	2.13	Standard C library, needs to be available to programs written in ISO C language
Glib	2.32.3	Application building blocks which add data types and other programming facilities for C-language programs
Curl	7.28.1_24	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
libXML2	2.7.8	Library for parsing xml documents
Fontconfig	2.9.0	Font-handling library to let applications find a font or a closely matching font
Freetype	2.4.9	Text-rendering library
Minizip	1.2.5	Lightweight library building on top of zlib for processing files in the zip format
Sqlite	3.7.13	Implements a lightweight sql database within a library, widely used for embedded client-local storage.
Cairo	1.12.14	Library for 2-D vector graphics drawing
openssl	1.0.1g_1	Library implementation of secure sockets layer (ssl) and transport layer security (tls) to enable secure internet communications

Supported Open Source Libraries (2/2)

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

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)

Native API | Multimedia

- 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)

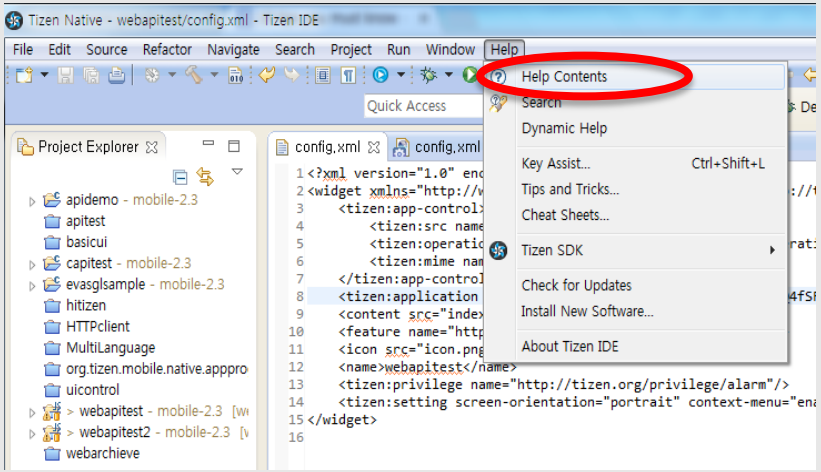
Native API | Web

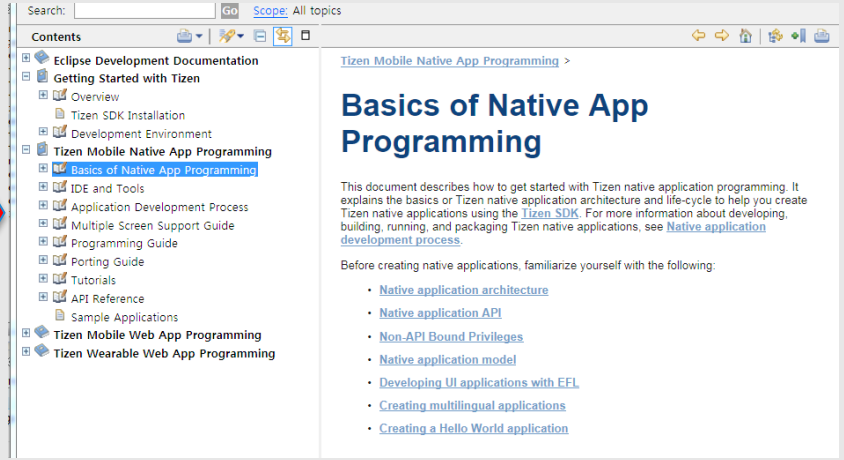
- Provides Web browser features
  - Displaying and controlling Web pages, such as browsing, tracking browsing history, and downloading Web content (WebView)

10 Must-know Tips
For Tizen Native App Development

Tip 1: What you are looking for is in SDK Help Contents (1/2)

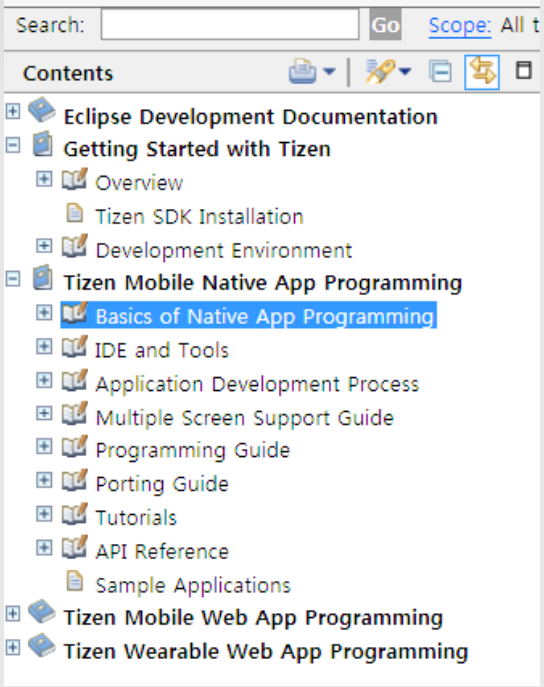
- All the Native API documents are in the [Help Contents] and will be added more





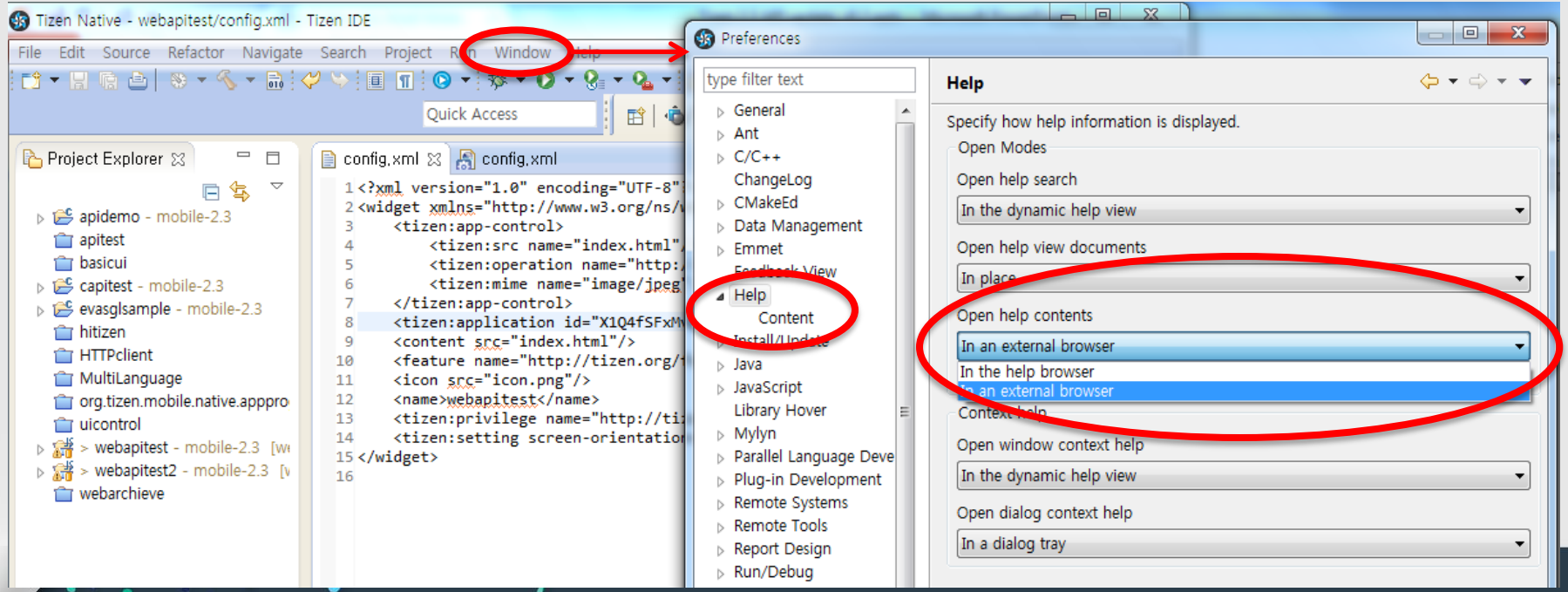
Tip 1: What you are looking for is in SDK Help Contents (2/2)

- 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



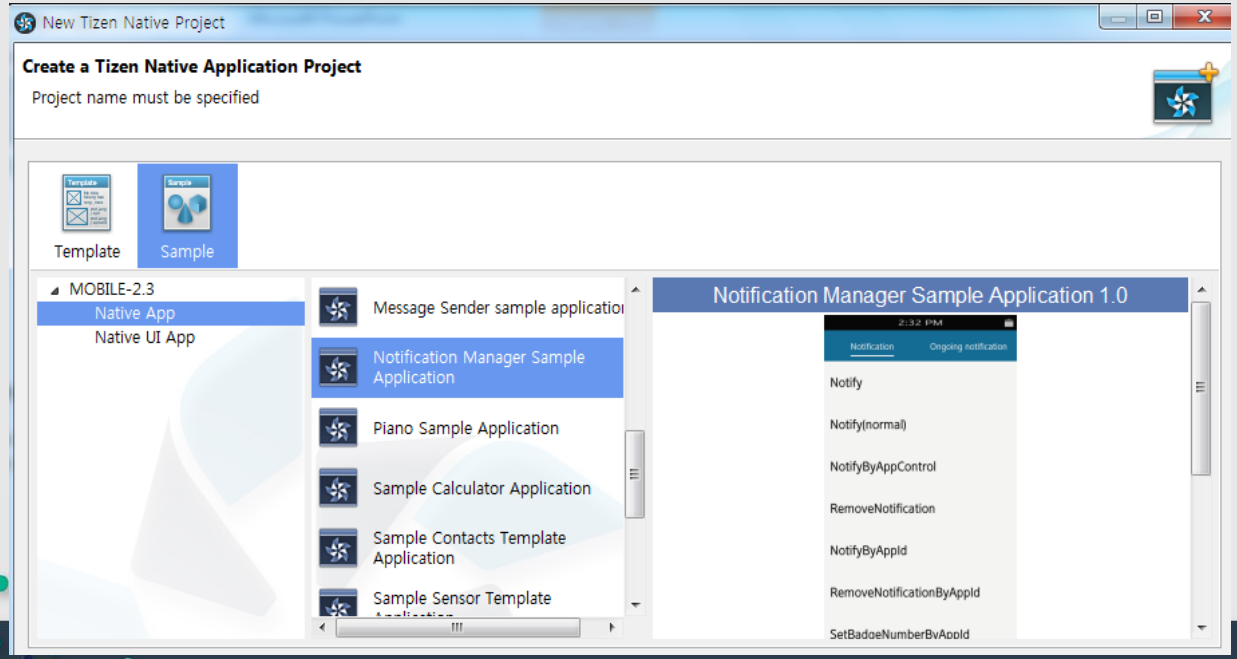
Tip 2: Open SDK help contents in a browser

- Go to [Window → Preference → Help → Open help contents]
- Select [In an external browser], then you can find info more easily



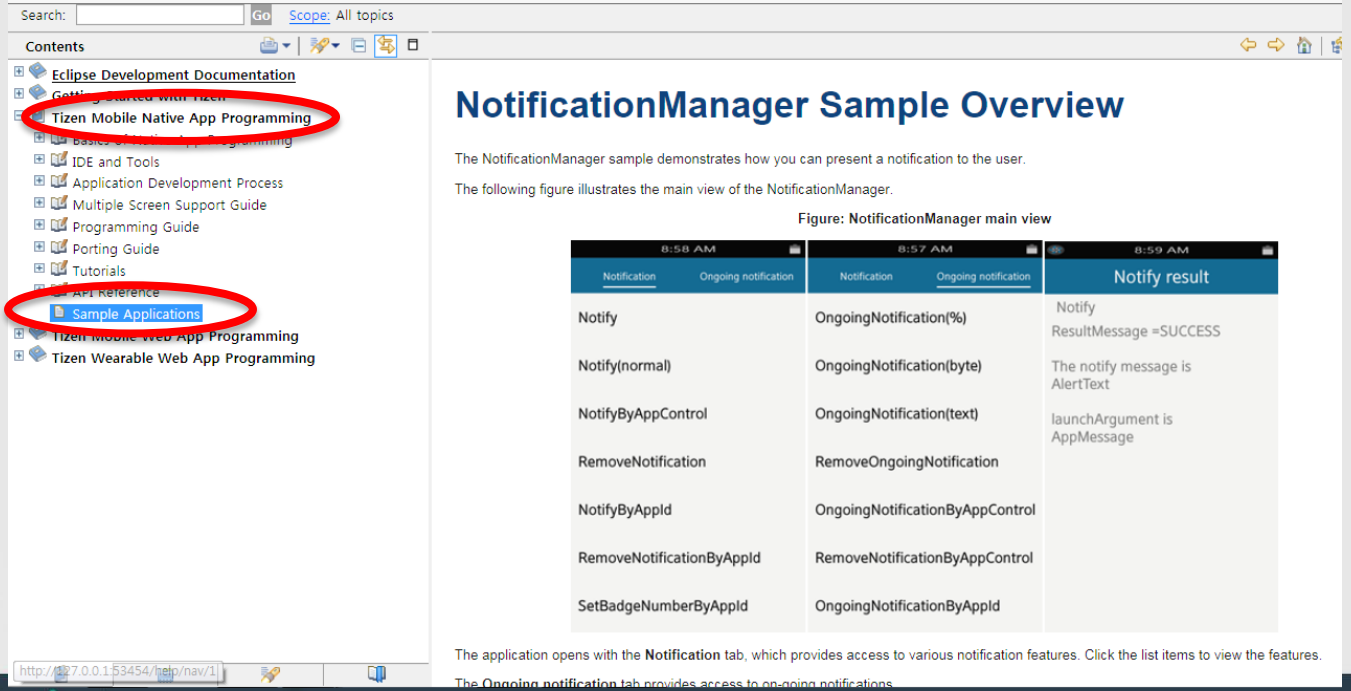
Tip 3: Refer to sample applications in the SDK (1/3)

- 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

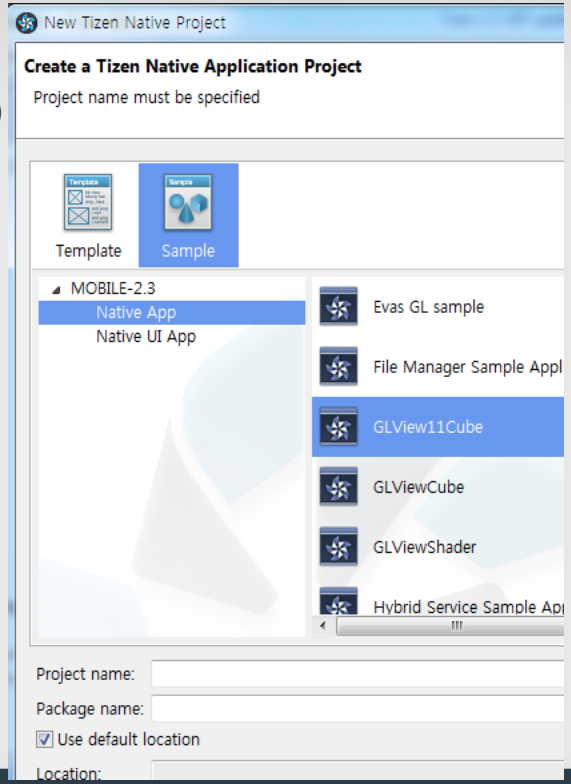


Tip 3: Refer to sample applications in the SDK (3/3)

Native UI Apps

: demonstrate UI only (not fully functioning)

Alignment UI	Message Bubble UI
Animation UI	Relative Position UI
ApplicationStore	SNS UI
Calculator	Settings UI
Clock	Theme extension UI
Media	Layout Samples UI
EDC Format	Layout sinal UI
Email UI	Language change UI
Gallery UI	More to come



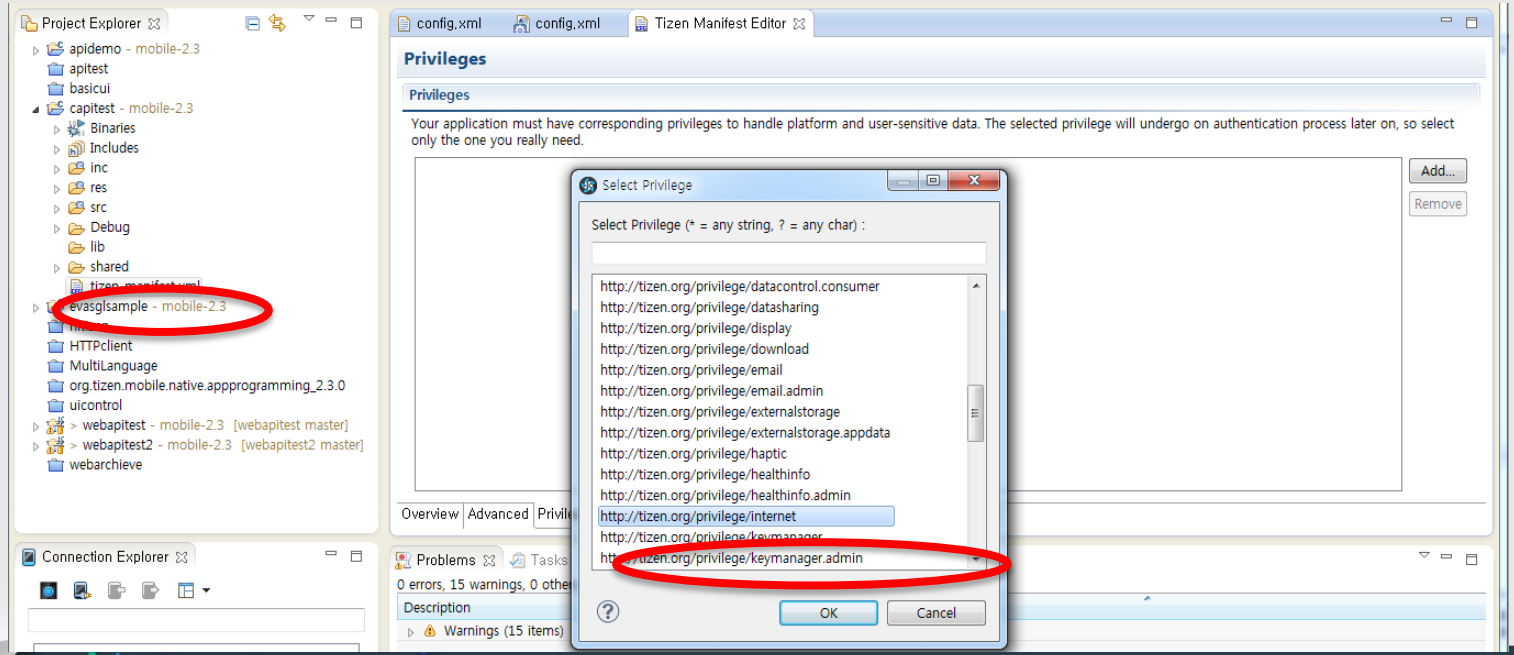
Native Apps

: fully functional sample apps

UI Controls	Notification
Evas GL	Piano
File	Calculator
GLView	Contacts
Hybrid application	Sensors
Media	Scheduler
More to come	

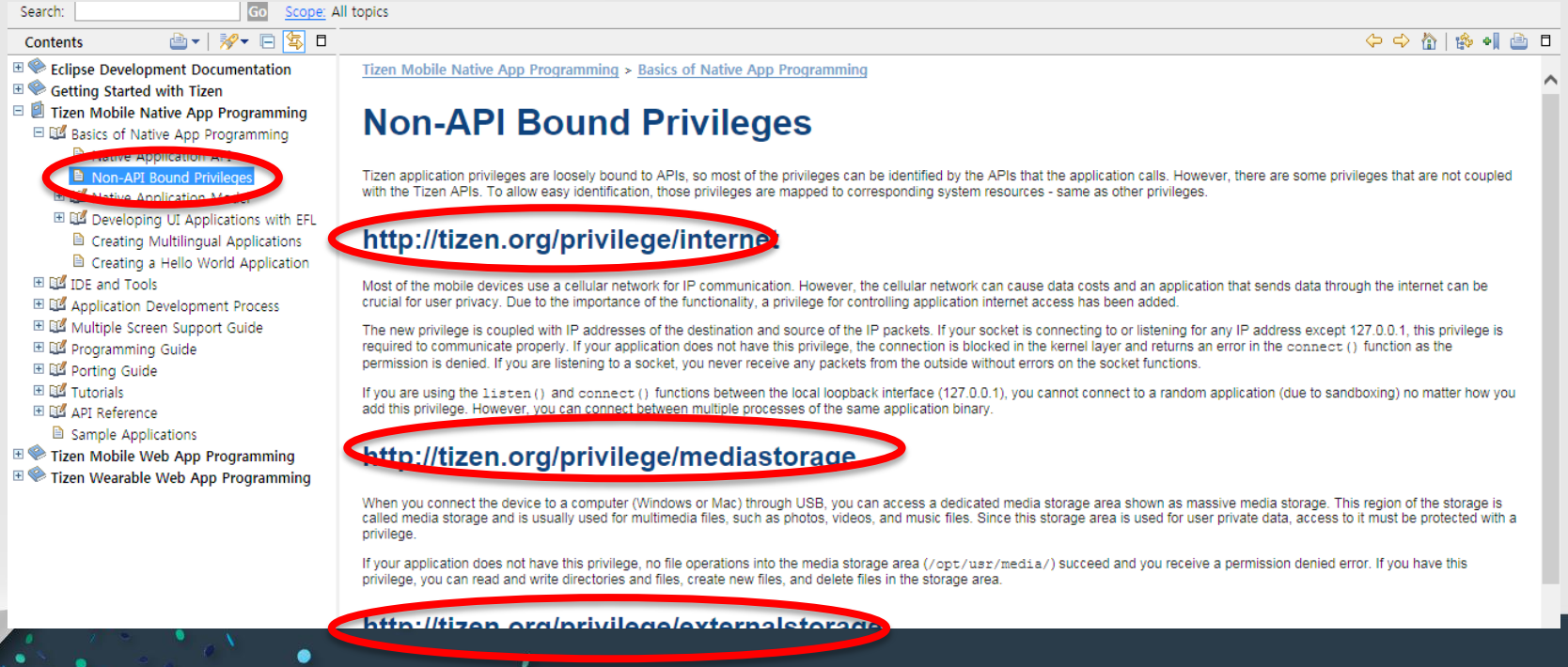
Tip 4: Don’t forget Non-API bound privileges (1/2)

- If your app access the Internet, add the below privilege
- Non-API bound privilege need media storage access and external storage access also



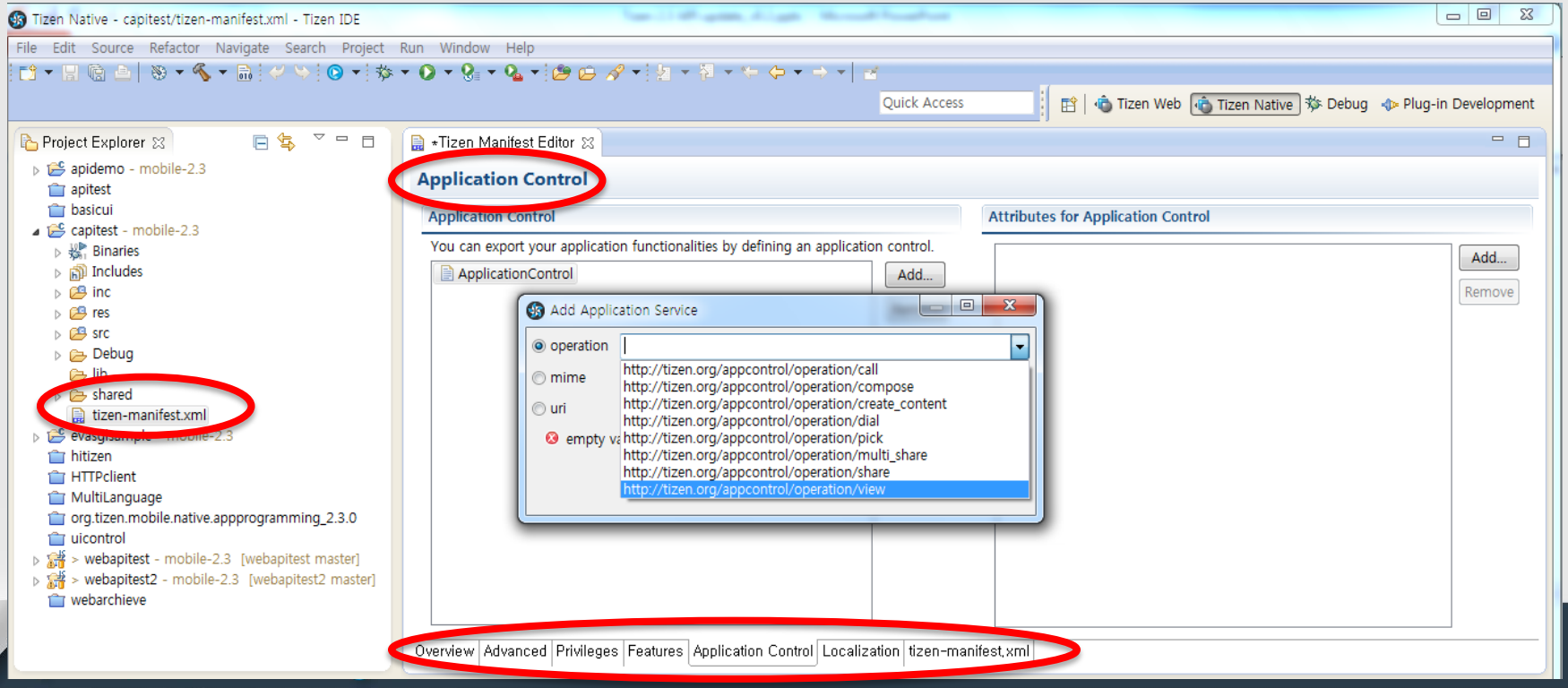
Tip 4: Don’t forget Non-API bound privileges (2/2)

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



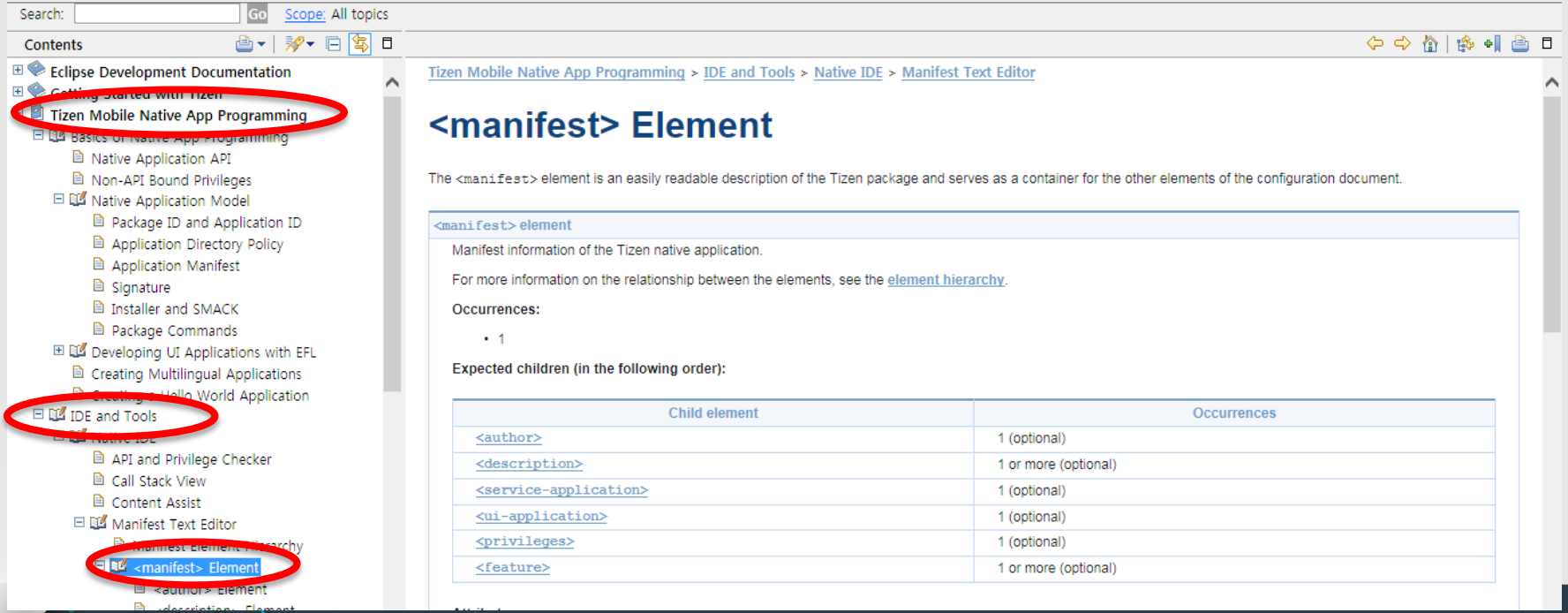
Tip 5: Use Tizen Manifest Editor for tizen-manifest.xml (1/2)

- Easy to configure App Control (below), Privileges, Features, etc.



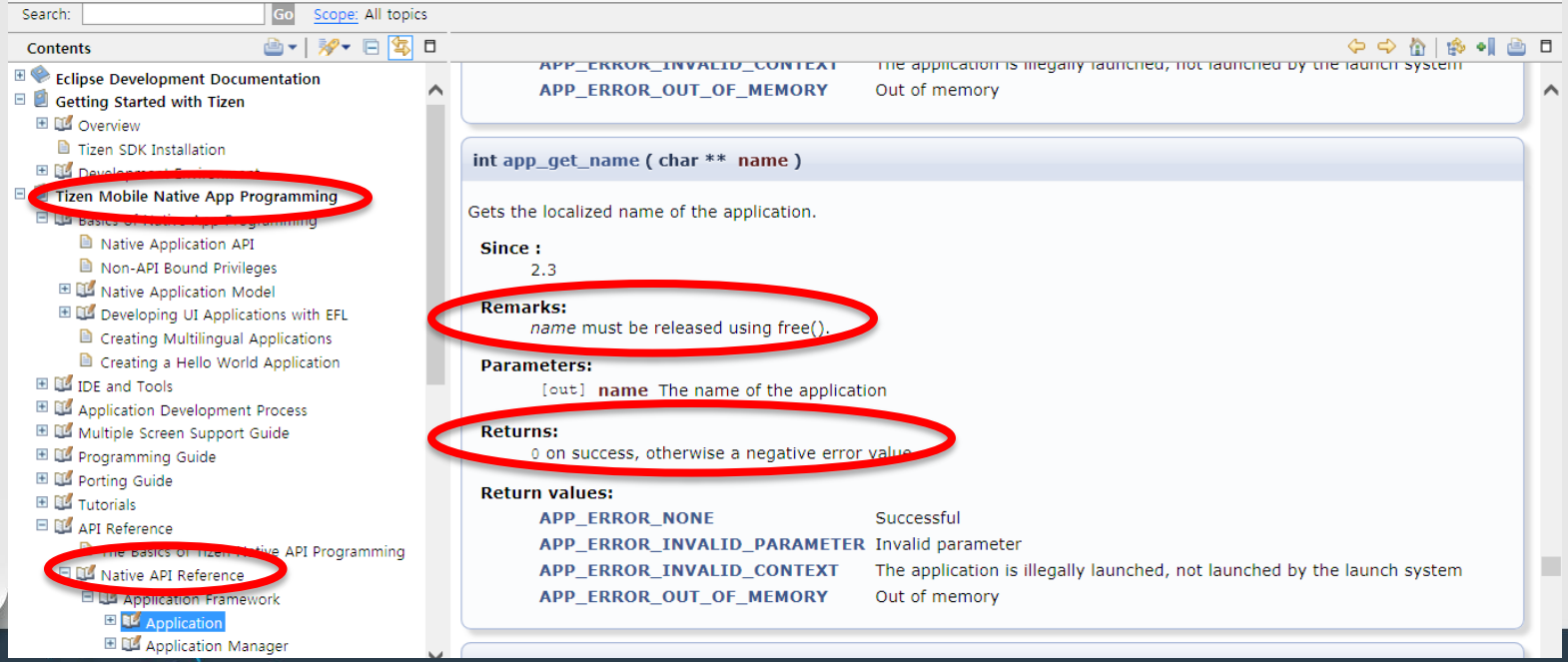
Tip 5: Use Tizen Manifest Editor for tizen-manifest.xml (2/2)

- To learn elements in the manifest xml file, visit the below page in the SDK Help Contents



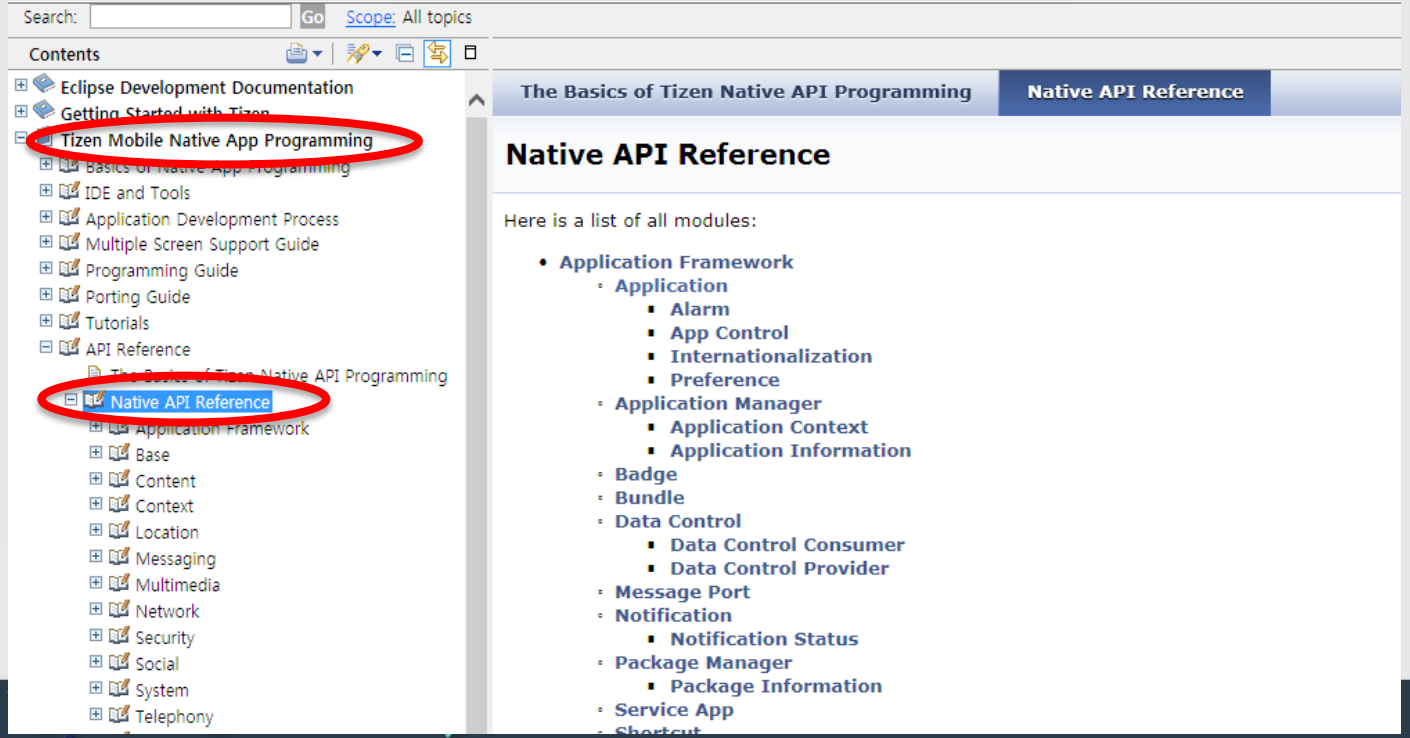
Tip 6: Things to remember when see the API Reference (1/2)

- In most case, [Returns] indicates 0 on success, negative on error
- Carefully read [Remarks]. e.g., string must be freed



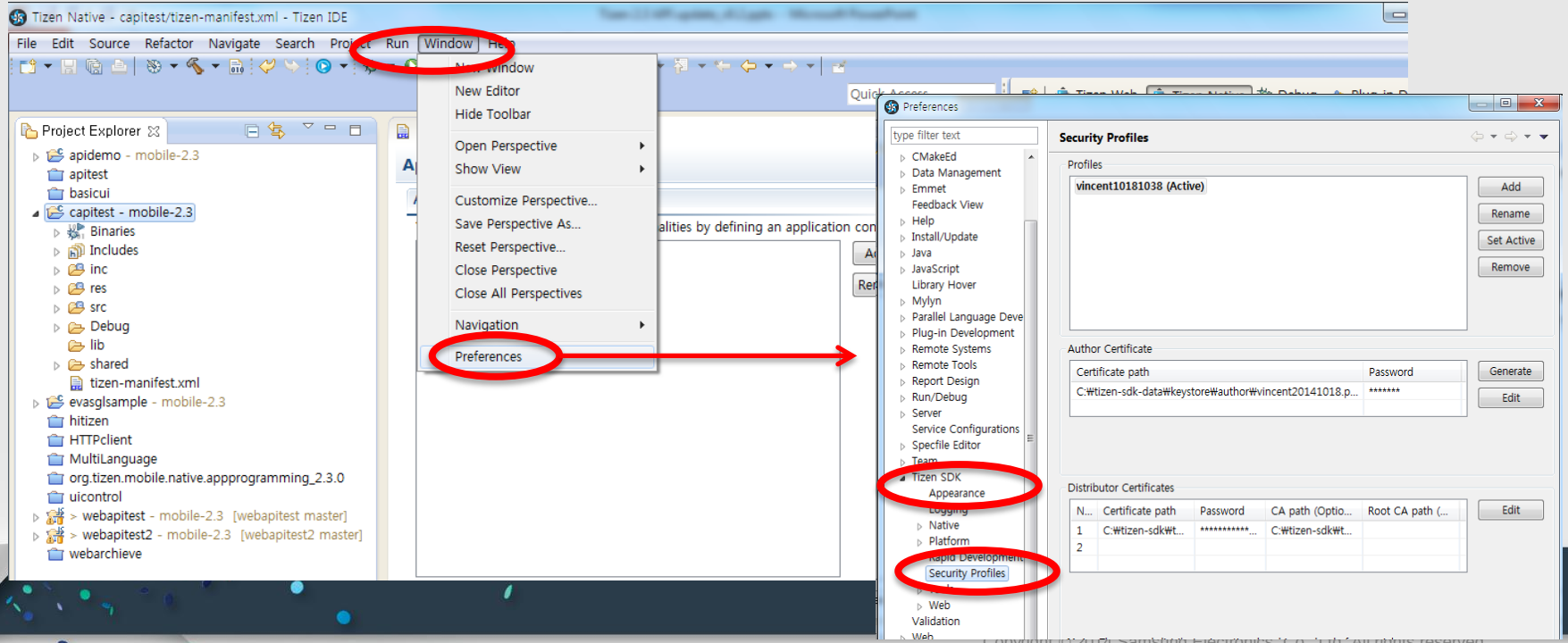
Tip 6: Things to remember when see the API Reference (2/2)

- Understand hierarchy of API modules (logical categorized)



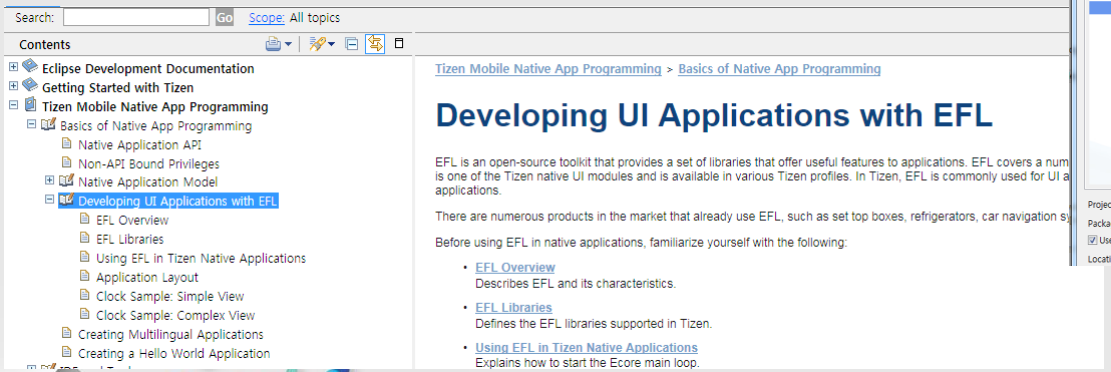
Tip 7: Security Profile

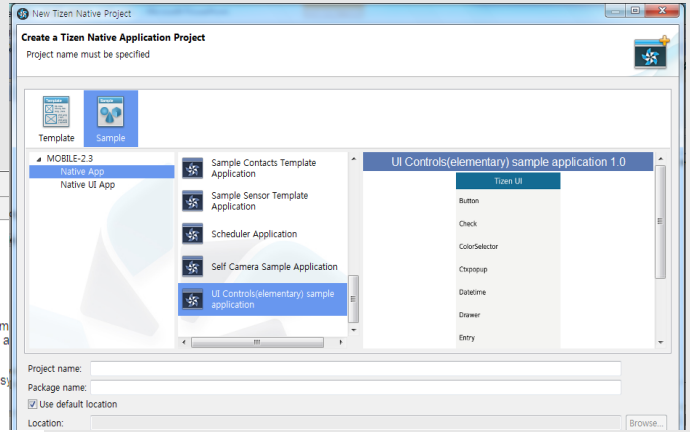
- Need to add Security Profile when you install SDK
- Suspect missing or wrong security profile if launching app failed with security error shown



Tip 8: If you are new to EFL, here comes docs for you (1/2)

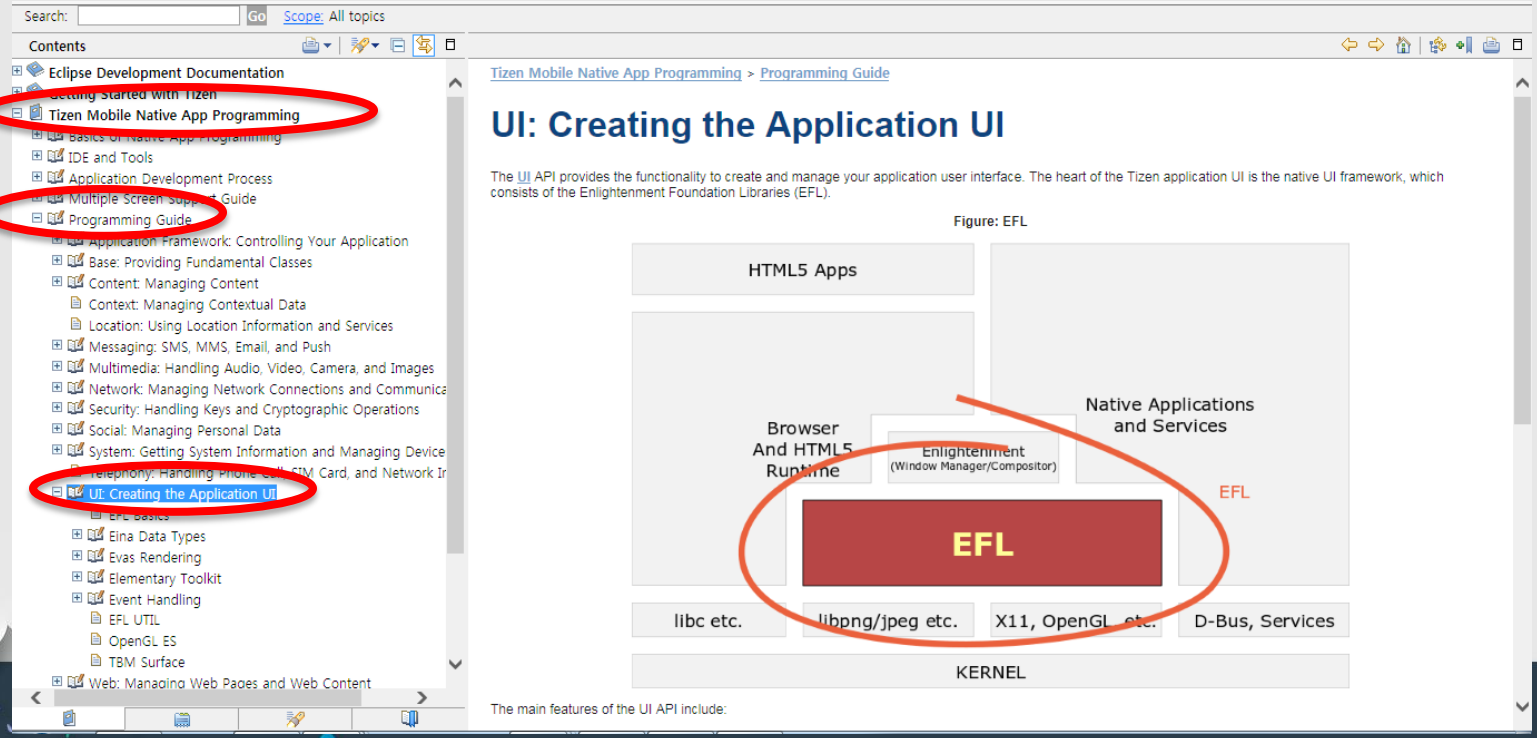
- Visit official EFL documents page (samples):
  - https://enlightenment.org/p.php?p=docs&l=en
- If you are interested in UI layout (EDJE), following site will help you alot:
  - http://docs.enlightenment.org/api/edje/doc/html/edcref.html
- Be friend with UI control sample (on the right)
- In the SDK help contents, check out below link





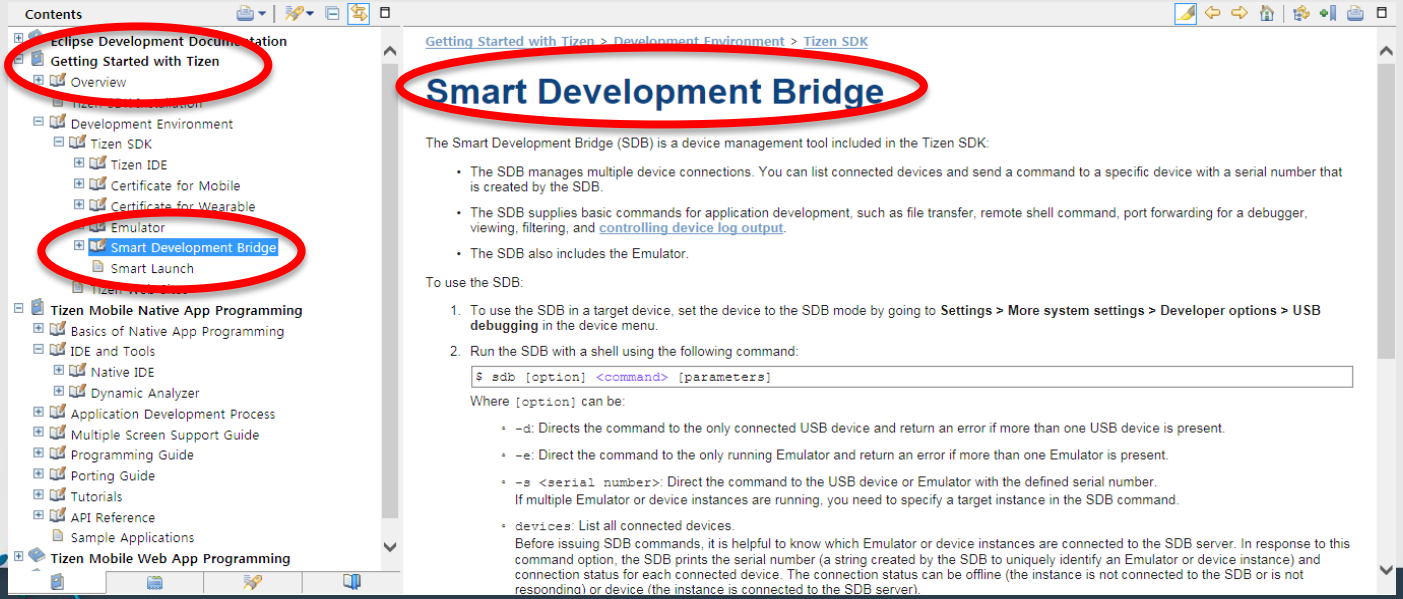
Tip 8: If you are new to EFL, check out EFL documents (2/2)

- More in-depth guide in Programming Guide in Help Contents



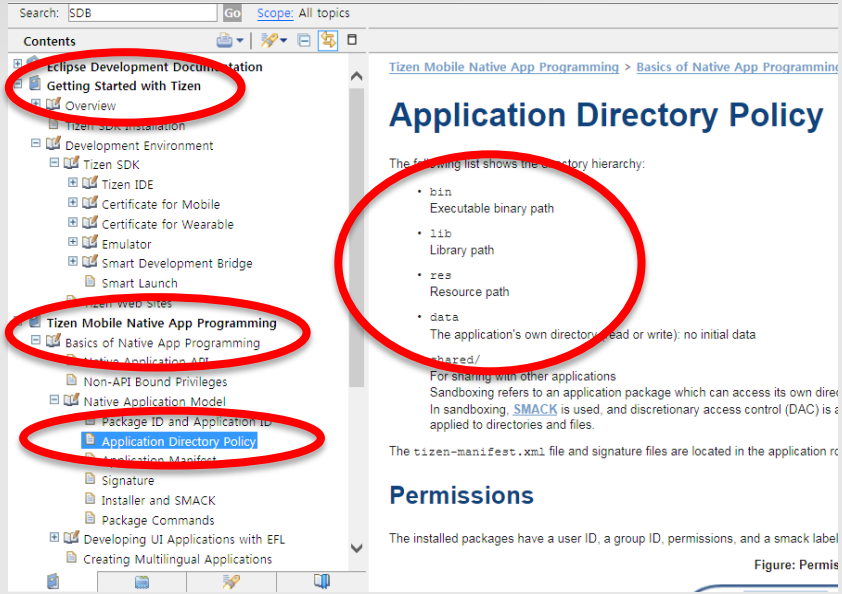
Tip 9: Use SDB to push or pull files to your app project (1/2)

- 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)



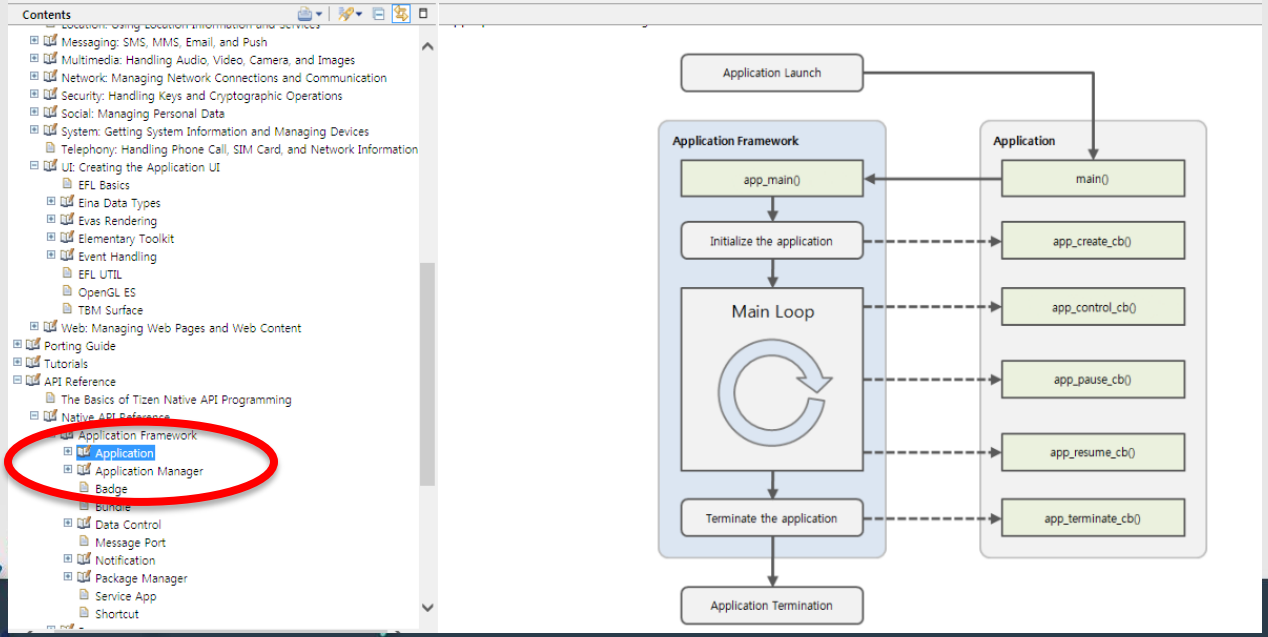
Tip 9: Use SDB to push or pull files to your app project (2/2)

- 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)



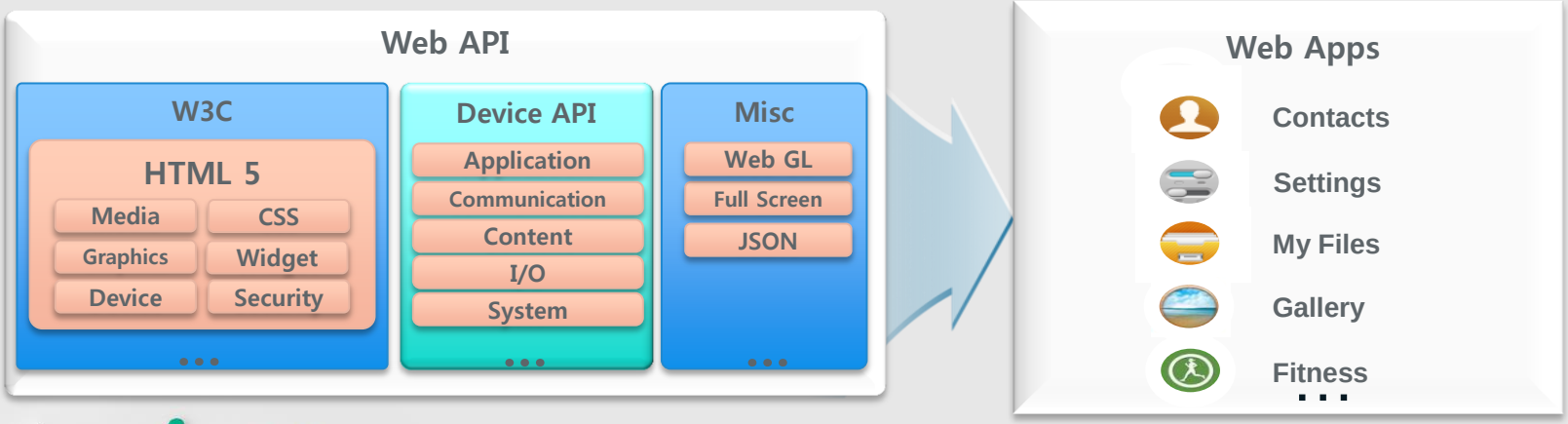
Tizen 2.3 beta
Web API Update

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

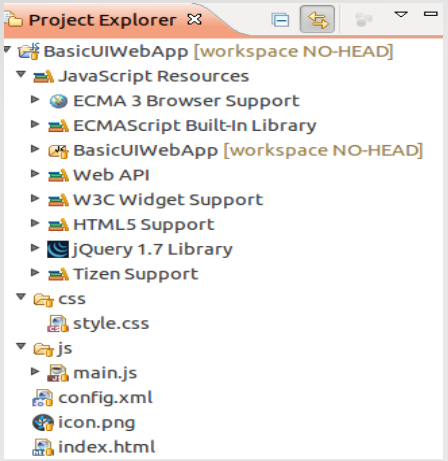
Brief Overview: Tizen Web Device APIs

- 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




```
<?xml version="1.0" encoding="UTF-8"?>
<widget xmlns="http://www.w3.org/ns/widgets" xmlns:tizen="http://tizen.org/ns/widgets" id="http://yourdomain/BasicUIWebAPp" versior
    <tizen:application id="RM2Gaf4ztV.BasicUIWebAPp" package="RM2Gaf4ztV" required_version="2.2"/>
    <content src="index.html"/>
    <feature name="http://tizen.org/feature/screen.size.all"/>
    <feature name="http://tizen.org/feature/network.bluetooth"/>
    <icon src="icon.png"/>
    <name>BasicUIWebAPp</name>
    <tizen:privilege name="http://tizen.org/privilege/application.info"/>
</widget>
```

Tizen 2.3 Web Device API Modules (as of 2.3 beta)

Tizen Common	
Tizen	Manages Bluetooth device and supports RFCOMM and HDP
Application	
Alarm	Schedules an application to be launched at a specific time
Application	Provides information about applications and controls applications
Data Control	Provides information about packages and install/uninstall packages
Package	Provides information install/uninstall package and get information about installed packages.
Communication	
Bluetooth	Manages Bluetooth device and supports RFCOMM and HDP
Messaging	Sends and receives SMS, MMS and Email message
Network Bearer Selection	Provides interfaces and methods for users to set network bearer for a specific IP address.
NFC	Manages NFC device and detects NFC tag and peer
Push	Receives push notifications from push server
Secure Element	provides interfaces and methods for access to Secure Elements.
Content	
Content	Discovers multimedia content (such as images, videos or music)
Download	Downloads remote objects by HTTP request
Exif	Provides interfaces and methods for manipulating EXIF data from JPEG file.
Input / Output	
Archive	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).
Filesystem	Provides access to the file system of a device. This API might be obsolete in the future when W3C File APIs are extended to access system-sensitive files by Web applications.
Message Port	Provides the functionality for communication with other applications.

Multimedia	
Sound	Provides a way to control sound volume level
Communication	
Account	Provides interfaces and methods for managing accounts(e.g. create an account, change the account information)
Bookmark	Provides interfaces and methods for access to Bookmarks
Calendar	Enables the management of calendar information
Call History	Allows accessing call history for phone calls
Contact	Enables the management of contact information
Data Synchronization	Provides methods to synchronize device data to the server using the OMA DS 1.2 protocol
System	
FM Radio	Provides interfaces and methods for listening to FM radio
Human Activity Monitor	Provides interfaces and methods for retrieving human activity data from the various sensors on the device
Power	Provides interfaces and methods for controlling power resources
Sensor	Provides interfaces and methods for getting sensor data from the various sensors on the device
System Information	Provides information about the device's display, network, storage and other capabilities
System Setting	Provides system setting functionality
Time	Exposes information about date, time and time zones
Web Setting	Manages the setting states of the web view in web applications
User Interface	
Badge	Provides a way to display the count of notifications(badge) on home screen
Notification	Provides a way to notify the user of events that happen in the application

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)

TIZEN™ DEVELOPER SUMMIT 2014

SHANGHAI

TIZEN开发者峰会（上海）