

Tizen® 2.2.1 Compliance Specification for Mobile Profile

Version 1.0

Copyright© 2013 Samsung Electronics Co., Ltd.
Copyright© 2013 Intel Corporation

Linux is a registered trademark of Linus Torvalds.

Tizen® is a registered trademark of The Linux Foundation.

ARM is a registered trademark of ARM Holdings Plc.

Intel® is a registered trademark of Intel Corporation in the U.S. and/or other countries

* Other names and brands may be claimed as the property of others.

Revision History

Revision	Date	Author	Reason for Changes
Tizen 2.2.1 Compliance Specification for Mobile Profile, v1.0	11 Nov. 2013	Tizen TSG	Public release

Glossary

Term	Definition
ABI	Application Binary Interface, the runtime interface between a binary software program and the underlying operating system.
API	Application Programming Interface, the interface between software components, including methods, data structures, and processes.
Compliance	Certified for full conformance, which was verified by testing.
Conformance	How well the implementation follows a specification.
CSS	Cascading Style Sheets, a simple mechanism for adding style (for example fonts, colors, and spacing) to web documents.
DOM	Document Object Model, a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure, and style of documents.
DTV	Digital Television, a target of the TV Profile.
GPS	Global Positioning System.
HSPA	High Speed Packet Access, a mobile broadband technology.
IOMMU	Input/Output Memory Management Unit.
IPTV	Internet Protocol Television, a target of the TV Profile.
IVI	In-Vehicle-Infotainment, a target of the IVI Profile. System used for entertainment, such as music, video, and games, along with information, such as navigation and web. A platform target for Tizen.
jQuery	Portable client-side JavaScript library.
LTE	Long Term Evolution, a telephone and mobile broadband communication standard.
Mobile	Portable, connected devices, such as phones and tablets. A platform target for Tizen.
NFC	Near Field Communication, a form of contactless communication between devices containing an NFC tag, such as smartphones , tablets, smart signs, kiosks etc.
REST	Representational State Transfer, design model used by the World Wide Web based on a client/server architecture where the client requests information and the server processes the request and returns information.
SDB	Smart Development Bridge, a device management tool in the Tizen SDK.
STB	Television set-top box, a target of the TV Profile.
Side loading	Installing applications or components other than from a certified application installer package.

Term	Definition
Smack	Simplified Mandatory Access Control Kernel, an access control technology used by Tizen to protect data and prevent malicious programs from causing harm.
UI	User Interface, the widgets, theme, and layout of software components displayed on the device screen through which the user interacts with the device. Usually refers to the visual software elements but may also include hardware buttons or controls.
UX	User experience, the effect that the design of a system (both software and hardware) has on the user of the system.
Tizen Web API	Collection of Tizen web programming interfaces for applications. Includes approved specifications generically known as HTML5, as well as additional interfaces such as Tizen Web Device API and Tizen Web UI FW.
Tizen Native API	Collection of Tizen native(C++) interfaces, standard C and C++ libraries, and a selected set of open source libraries such as OpenAL, libxml2, etc.
WPS	Wi-Fi based Positioning System.

Table of Contents

1. Overview
 - 1.1. Why Compliance?
 - 1.2. Target Audience
 - 1.3. Tizen Compliance Model
 - 1.4. Revision Policy
 - 1.5. Tizen Source Code Modification Policy
 - 1.6. References
2. Mobile Profile Software Compliance
 - 2.1. General Principles
 - 2.2. Tizen Web API
 - 2.2.1. Namespace
 - 2.2.2. Tizen Web API Categories
 - 2.2.3. Preliminary Web APIs
 - 2.2.4. Behavior of Unsupported APIs
 - 2.3. Tizen Native API
 - 2.3.1. Namespace
 - 2.3.2. Tizen Native API Categories
 - 2.3.3. Behavior of Unsupported APIs
 - 2.3.4. Alias Application ID
 - 2.3.5. Native Application Model
 - 2.4. Application Binary Interface
 - 2.5. Application Control
 - 2.6. Platform Attributes
 - 2.7. Optional APIs
 - 2.7.1. Tizen Web API
 - 2.7.2. Tizen Native API
 - 2.8. Privilege
 - 2.8.1. Tizen Web API
 - 2.8.2. Tizen Native API
 - 2.9. Push Service
 - 2.10. Application Packaging Compatibility
 - 2.10.1. Web App Package Support
 - 2.10.2. Native App Package Support
 - 2.10.3. Hybrid Web/Native Package Support
 - 2.11. WebKit and Browser
 - 2.11.1. WebKit
 - 2.11.2. Browser
 - 2.12. Web Runtime
 - 2.13. User Interface
 - 2.13.1. Theme
 - 2.13.2. Notification Tray
 - 2.13.3. Keys
 - 2.14. Security
 - 2.15. Multimedia

- 2.16. Developer Tools
- 2.17. Software Update
- 2.18. Tizen Compliance Tests
- 3. Mobile Profile Hardware Compliance
 - 3.1. Mandatory Hardware Requirements
 - 3.1.1. Memory Storage
 - 3.1.2. Sound
 - 3.1.3. Connectivity / Networking
 - 3.1.4. Display
 - 3.1.5. USB
 - 3.1.6. Input Devices
 - 3.2. Optional Hardware Requirements
 - 3.2.1. Camera
 - 3.2.2. Graphics
 - 3.2.3. GPS
 - 3.2.4. Sensors
 - 3.2.5. Telephony
 - 3.2.6. Bluetooth
 - 3.2.7. Wi-Fi
 - 3.2.7.1. Wi-Fi Direct
 - 3.2.8. NFC
 - 3.2.9. Input Devices
 - 3.2.10. Secure Element
- 4. Mobile Profile Application Compliance
 - 4.1. API Use
 - 4.1.1. Limited use of C library functions
 - 4.2. Application Packaging
 - 4.3. Application Lifecycle
 - 4.4. Namespace
 - 4.5. Application Features and Privileges
- Appendix A. Additional Information
 - A.1. Tizen Application Control
 - A.2. Tizen Features
 - A.2.1. Tizen Web API
 - A.2.2. Tizen Native API
 - A.3. Tizen Privileges
 - A.3.1. Tizen Web API Privileges
 - A.3.2. Tizen Native API Privileges
 - A.4. Sensor Hardware Capabilities
 - A.5. Push C-language APIs

1. Overview

This specification defines the operating environment of the Tizen platform. It is intended to be used by both mobile device implementers and application developers to enable the development of portable application software.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" used in this document are to be interpreted as described in [ref. 5].

Tizen is a registered trademark of the Linux Foundation, which controls the usage of the brand and trademark. A requirement for permission to use this trademark in conjunction with products is compliance with the requirements of this specification.

1.1. Why Compliance?

Tizen Compliance is designed to ensure mobile device implementations and applications work together.

1.2. Target Audience

This specification is intended to be used by:

- **Application developers:** know how to create compatible applications that work across multiple devices, and how Tizen devices will behave.
- **Mobile device implementers:** know how to implement device hardware, security configurations, services, APIs, etc.
- **Operators:** know how to customize and enhance a device, while remaining within compliance guidelines.
- **End users:** know that applications will work on their device and are assured of a consistent user experience among compliant applications.

1.3. Tizen Compliance Model

To become Tizen compliant, a device **MUST** obtain Tizen Compliance certification from the Tizen Association for at least one Tizen Profile by satisfying the requirements of the Tizen Compliance Specification and passing all of the Tizen Compliance Tests.

A Tizen Profile describes the requirements for a category of Tizen devices that have a common application execution environment. Applications are created for a specific target profile and can run on devices compliant to that profile.

- **Device implementations:** if implemented to a profile, a device will provide applications with consistent behavior defined by that profile, as well as a consistent user experience.
- **Applications:** if built to a profile, applications will run on devices that are compliant to the profile.

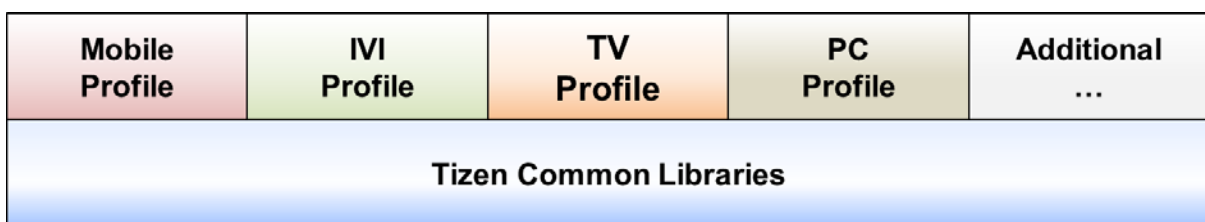
Each Tizen Profile is based on Tizen Common Libraries, which are the set of libraries common across all Tizen platforms. The set of libraries helps to unify multiple categories of devices by sharing common platform components.

The current available profile is:

- Mobile: handsets, tablets, etc.

Additional profiles are expected in the near future, possibly including:

- IVI: In-Vehicle Infotainment systems
- TV: DTV/STB/IPTV systems
- PC: laptops, convertibles, etc.



The Tizen Compliance Tests will measure conformance to the requirements.

Note: The initial release of this specification describes only the compliance requirements for the Tizen Mobile Profile.

1.4. Revision Policy

There will be a distinct release of the specification, as well as matching compliance tests, for each distinct release (version) of the Tizen platform. Updates may be issued between releases, if deemed necessary. All compliance requirements for the Mobile Profile specification must be approved by the Tizen Technical Steering Group (TSG) and may change from time to time, only by approval of the Tizen Technical Steering Group.

1.5. Tizen Source Code Modification Policy

All Tizen implementations **MUST** provide the full behavior of the Tizen API and application execution environment as defined by the Tizen Profile for its device category. The best way to accomplish this is by using the source code for the Tizen reference implementation. If modifications or replacements to the source code must be made, the implementer is

Tizen 2.2.1 Compliance Specification
for Mobile Profile

responsible for making sure that there is no impact on compliant applications. The Tizen Compliance Tests may be used to measure the correctness of the implementation, but in case of ambiguities, errors, or incompleteness of this specification or of the Tizen Compliance Tests, the final arbiter of compatibility is the behavior of the Tizen reference implementation.

1.6. References

The following external specifications and other documents are referenced by this specification.

[N]: Normative Reference

[I]: Informative Reference

1. [I] Dynamic Analyzer Reference: https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.native.appprogramming/html/ide_sdk_tools/da_overview.htm
2. [N] Globalize 0.1.0a2: <https://github.com/jquery/globalize/tree/v0.1.0a2>
3. [I] EglIBC 2.13: <http://www.eglibc.org/home>
4. [I] GNU C/C++ compiler 4.5.3: <http://gcc.gnu.org>
5. [N] IETF RFC 2119 "Key words for use in RFCs to Indicate Requirement Levels": <http://www.ietf.org/rfc/rfc2119.txt>
6. [N] ISO 18092 "Near Field Communication – Interface and Protocol (NFCIP-1)": [http://standards.iso.org/ittf/PubliclyAvailableStandards/c038578_ISO_IEC_18092_2004\(E\).zip](http://standards.iso.org/ittf/PubliclyAvailableStandards/c038578_ISO_IEC_18092_2004(E).zip)
7. [I] libxml2 2.7.8: <http://www.xmlsoft.org/html/index.html>
8. [N] jQuery 1.8.2: <http://blog.jquery.com/2012/09/20/jquery-1-8-2-released/>
9. [N] jQuery Mobile 1.2.0: <https://github.com/jquery/jquery-mobile/tree/1.2.0>
10. [I] Log View Reference: https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.gettingstarted/html/dev_env/log_view.htm
11. [I] NFC Specifications: <http://www.nfc-forum.org/specs/>
12. [N] OpenAL 1.1 Specification and Reference: https://source.tizen.org/sites/default/files/page/openal_1.1_specification.pdf
13. [N] OpenGL ES 1.1: http://www.khronos.org/registry/gles/specs/1.1/es_full_spec_1.1.12.pdf
14. [N] OpenGL ES 2.0: http://www.khronos.org/registry/gles/specs/2.0/es_full_spec_2.0.25.pdf
15. [N] OpenMP Application Program Interface 3.0: <http://www.openmp.org/mp-documents/spec30.pdf>
16. [I] OProfile Reference: https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.native.appprogramming/html/ide_sdk_tools/oprofile.htm

17. [I] Smart Development Bridge: https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.gettingstarted/html/dev_env/smart_development_bridge.htm
18. [N] Tizen Alias Application ID: https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.gettingstarted/html/guides/alias_app_id.htm
19. [N] Tizen Native API Namespace List: <https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.native.apireference/namespaces.html>
20. [I] Tizen Native API Packaging: https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.native.appprogramming/html/app_dev_process/packaging_app.htm
21. [N] ISO/IEC 14882 2011, section "20.7.1 Class template unique_ptr": http://en.cppreference.com/w/cpp/memory/unique_ptr
22. [I] Tizen Native Application Development Process: https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.native.appprogramming/html/app_dev_process/overview_of_app_dev.htm
23. [I] Tizen Native Application Lifecycle: https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.native.appprogramming/html/basics_tizen_programming/tizen_app_model/application_lifecycle.htm
24. [N] Tizen Native Application Model: https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.native.appprogramming/html/basics_tizen_programming/tizen_app_model/application_model.htm
25. [I] Tizen Privilege Guide: https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.gettingstarted/html/tizen_overview/security.htm
26. [I] Tizen UX Guide: <https://developer.tizen.org/documentation/ux-guide>
27. [N] Tizen Web Supplementary API References: https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.web.supplementary.apireference/supplementary_api.html
28. [N] Tizen W3C/HTML5 API References: https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.web.w3c.apireference/w3c_api.html
29. [N] Tizen NFC reserved push Guide: https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.native.appprogramming/html/guide/net/nfc_reserved_push.htm
30. [N] Tizen Web Device API 2.2.1 Reference: <https://developer.tizen.org/documentation/dev->

- [guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.web.device.apireference/index.html](https://developer.tizen.org/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.web.device.apireference/index.html)
31. [N] Tizen Web Runtime Core Specification 2.2:
<https://source.tizen.org/sites/default/files/page/tizen-2.2-wrt-core-spec.pdf>
 32. [N] Tizen Web UI Framework Reference:
https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.web.uiwidget.apireference/html/web_ui_framework.htm
 33. [N] W3C Widget Access Request Policy (W3C Recommendation 7 February 2012 version): <http://www.w3.org/TR/widgets-access/>
 34. [N] Optional Tizen W3C/HTML5 API: <https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.web.apireference.relatedpage/html/feature/w3c.html>
 35. [N] Optional Tizen Web Device API: <https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.web.apireference.relatedpage/html/feature/device.html>
 36. [N] Optional Tizen Native API: <https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.native.apireference/feature.html>
 37. [I] Tizen Application Filtering: https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.gettingstarted/html/tizen_overview/application_filtering.htm
 38. [I] Application Controls for Tizen web applications:
https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.web.appprogramming/html/guide/app_guide/application_service.htm
 39. [I] Application Controls for Tizen native applications:
https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.native.appprogramming/html/guide/app/app_controls.htm
 40. [I] Tizen Web Application Security and Privacy:
https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.web.appprogramming/html/basics_tizen_programming/web_security_privacy.htm
 41. [I] Tizen Web Application Package Manager:
https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.web.appprogramming/html/basics_tizen_programming/application_package_manager.htm
 42. [I] Tizen Native Application Package Manager:
https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.native.appprogramming/html/basics_tizen_programming/tizen_app_model/application_package_manager.htm

43. [I] Tizen Hybrid Application Package Manager:
https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.web.appprogramming/html/app_dev_process/hybrid_app_package.htm
44. [N] Basics of Tizen Native App Programming - Limited use of C library functions:
https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.native.appprogramming/html/basics_tizen_programming/basics_tizen_programming/c_library_restrictions.htm
45. [N] Tizen Push Messaging: https://developer.tizen.org/documentation/dev-guide/2.2.1?redirect=https://developer.tizen.org/dev-guide/2.2.1/org.tizen.native.appprogramming/html/guide/messaging/push_messaging.htm

2. Mobile Profile Software Compliance

This chapter describes the software requirements that implementers **MUST** meet to create a compliant Tizen mobile device.

2.1. General Principles

Mobile device implementations **MUST** include support for both the Tizen Web API and the Tizen Native API.

- The mobile device implementation **MUST** accurately report the presence or absence of optional hardware and software features (see section 2.7) as platform attributes. If a mobile device implementation supports a particular optional hardware or software feature, it **MUST** implement the entire corresponding API, and it **MUST** report the presence of the hardware or software as platform attributes.
- If a mobile device implementation does not support a particular optional hardware or software feature, it **MUST** report the absence of the hardware or software as platform attributes.
- Whether a mobile device implementation supports or does not support a particular optional hardware or software feature, the compliance tests **MUST** be passed. If the feature is not supported, the corresponding API **MUST** return the unsupported return value, as described in sections 2.2.4 for Web API and 2.3.3 for Native API.

2.2. Tizen Web API

2.2.1. Namespace

Mobile device implementations MUST NOT modify the API namespace listed in the Tizen Web Device API Reference [ref. 30], including `tizen.*`.

2.2.2. Tizen Web API Categories

- **W3C/HTML5 APIs:** include the standard APIs defined by W3C, such as HTML5, CSS3, and Widget Specification. See [ref. 28].
- **Supplementary APIs:** non-W3C specifications, such as WebGL, Typed Array, FullScreen API, and viewport Meta Tag. See [ref. 27].
- **Web Device API:** defined by the Tizen project to facilitate the development of web applications by accessing various device features not fully covered by W3C APIs. The APIs enable interacting with device features, such as calendar, contact, Bluetooth, NFC, messaging, alarm, and system information. See [ref. 30].
- **Web UI Framework:** The Web UI Framework provides tools, such as widgets, themes, events, effects, and animations for web applications. The Web UI framework is based on jQuery version 1.8.2 [ref. 8], jQuery Mobile version 1.2.0 [ref. 9], and Globalize version 0.1.0a2 [ref. 2]. Use of the Web UI framework is an optional feature for Tizen application developers. Tizen application developers MAY just use W3C standard technologies, such as HTML/JavaScript/CSS. However, mobile device implementations MUST provide the Tizen Web UI framework for applications that use it. The versions listed above are strongly recommended. Mobile device implementations MAY use later versions if they pass the Tizen Compliance Tests. See the Tizen Web UI FW Reference [ref. 32].

2.2.3. Preliminary Web APIs

The Tizen Web API includes some preliminary Web API specifications which are in an early stage in the development cycle. Preliminary revisions are referred to as Editor's Draft (ED), Working Draft (WD), and Last Call Working Draft (LCWD). Application developers are cautioned that APIs in these specifications could be modified in a future version of Tizen to align with the developing progress of specifications. Preliminary APIs are indicated in the W3C/HTML5 APIs reference [ref. 28].

Mobile device implementations MUST support all Tizen Web APIs from the Tizen Web API specifications, including those indicated as preliminary.

2.2.4. Behavior of Unsupported APIs

Mobile device implementations MUST NOT omit any web API listed in the Tizen Web API specification, except those specified as optional in section 2.7.1 and not supported on the device. Optional APIs are dependent on particular hardware or software availability.

If an optional API is not supported on the device, it MUST return “undefined” when a whole module is not supported. For example, an attempt to access `tizen.nfc` MUST return “undefined” if the NFC module is not supported on the device. In case APIs in a module depend on a certain optional feature, those APIs MUST report `NotSupportedError` if the feature is not present. For example, if MMS is not supported on the device, an attempt to call `tizen.messaging.getMessageServices("messaging.mms", successCallback)` MUST report `NotSupportedError`.

2.3. Tizen Native API

2.3.1. Namespace

The Tizen Native API namespace is `Tizen::*`, as defined by the Tizen Native API Namespace List [ref. 19]. Mobile device implementations MUST NOT modify this API namespace.

2.3.2. Tizen Native API Categories

The Tizen Native API is defined by a C++ library [ref. 4], which provides the namespace `Tizen::*` as defined in [ref. 19]. In addition, the following native APIs MUST be available to native applications:

- C library, as implemented by `eglibc 2.13` [ref. 3]
 - Mobile device implementations are not required to provide any specific commands for use by the `system()` and `popen()` interfaces and the `exec()` family of interfaces.
- C++ Standard Library: `ISO/IEC 14882-2003` and `ISO/IEC TR 19768:2007(C++ TR1)` compliant C++ standard library implementation within `g++ 4.5.3` [ref. 4]
- `std::unique_ptr` from `C++11` [ref. 21]
- `OpenAL 1.1` [ref. 12]
- `OpenMP 3.0` [ref. 15]
- `libxml2 2.7.8` [ref. 7]

A mobile device implementation MAY omit `OpenGL-ES 1.1` [ref. 13] and `2.0` [ref. 14] APIs. The mobile device implementation MUST accurately report the availability of OpenGL ES API versions 1.1 and 2.0 through the `Tizen::System::SystemInfo` API.

For all Native API libraries listed above, mobile device implementations MUST use the version shown or 100% compatible versions.

2.3.3. Behavior of Unsupported APIs

Mobile device implementations MUST NOT omit any native API listed in the Tizen Native API specification [ref. 19], except those specified as optional in section 2.7.2 and not supported on the device. Optional APIs are dependent on particular hardware or software availability.

If an optional API is not supported on the device, it MUST return an `E_UNSUPPORTED_OPERATION` error, when accessed.

2.3.4. Alias Application ID

The Tizen Native API MUST support alias application IDs [ref. 18]. The alias application ID is an extra nominal name for the application ID and can be used instead of the actual ID. The alias allows mobile device implementers to configure the device with different applications which satisfy certain functionality criteria, and application developers can use the functionality without requiring knowledge of the underlying implementation.

2.3.5. Native Application Model

The Tizen Native API MUST comply with the Native Application Model [ref. 24]. The Tizen native application model handles application life-cycle and system events in the native framework. The Tizen platform supports both UI applications (which have a graphical user interface) and service applications (which do not have a graphical user interface).

2.4. Application Binary Interface

The Application Binary Interface (ABI) describes the compatibility of executable object or binary programs. Use of the Tizen Native API will result in binary programs. Mobile device implementations MUST be compatible with the one of the following ABIs.

The ABI for ARM® Architecture CPUs is supported with these characteristics:

- ABI: aapcs-linux
- CPU architecture: armv7
- CPU instruction set: cortex-a5
- FPU option: vfpv3-d16
- Floating point ABI: softfp
- Endian-ness: little endian

The ABI for Intel® IA32 Architecture CPUs is supported with these characteristics:

- ABI: i386 psABI (gcc: -m32)
- CPU architecture: IA32

- CPU instruction set: SSSE3
- Floating point ABI: SSE math (gcc: -mfpmath=sse)
- Endian-ness: little endian

2.5. Application Control

The application control interface (AppControl) in both the Tizen Web API and the Tizen Native API enables launching an application directly using an app ID or invoking specific application functionality remotely through IPC.

Mobile device implementations MUST provide the mandatory platform AppControls, as defined in section A.1.

A Tizen application may register itself as an AppControl provider. The available AppControl values can be queried and invoked by a Tizen application.

Further details on Application Controls are provided in the developer documentation. (See [ref. 38 and ref. 39])

2.6. Platform Attributes

Mobile device implementations MUST provide accurate platform attributes through the System Information API for the Tizen Web API and the Tizen Native API.

Platform attributes include the following:

- Device capabilities (see section 2.7)
- Information about data storage devices
- Display information
- Information about the device orientation
- Locale information
- Network information

2.7. Optional APIs

The Tizen API may depend on available hardware capabilities and, in some cases, on software capabilities. Optional software features may be capabilities not part of the publicly available stack, or may require hardware capability that is beyond the minimum mobile device requirement (such as higher processing power/memory) (See section 3.1 for minimum hardware requirements).

Mobile device implementations MUST NOT omit any API listed in the API specification, except those specified as optional in this section. Optional APIs are dependent on particular hardware or software availability. Optional APIs are divided into two categories:

- Optional software: Software is optional, but if present, its corresponding API MUST be supported.
 - Web: Push
 - Native: Push, database encryption, recognition (voice, facial, motion), NFC reserved push
- Optional hardware: Hardware is optional, but if present its corresponding API MUST be supported.
 - Web: Bluetooth, telephony, MMS, NFC, camera, microphone, sensors, Location (GPS), Secure Element, Wi-Fi
 - Native: Bluetooth, telephony, MMS, NFC, camera, microphone, sensors, Wi-Fi, Wi-Fi Direct, location (WPS, GPS)

For more detail, see the optional Tizen Web APIs [ref. 34 and ref. 35] and the optional Tizen Native APIs [ref. 36].

Mobile device implementations MUST support the System Information APIs (web and native) to report device capabilities. Capability information can be used by application stores to check the capabilities of a given device and to select applications that will run with full capabilities on the device.

2.7.1. Tizen Web API

The Tizen Web APIs specified as optional in [ref. 34 and ref. 35] will not be implemented if a mobile device implementation does not include those features. These features can be used to filter out applications by using the SystemInfo API. The mobile device implementation MUST accurately report the availability of these features through the Tizen Web API `SystemInfo` API.

2.7.2. Tizen Native API

The Tizen Native APIs specified as optional in [ref. 36] will not be implemented if the mobile device implementation does not include the hardware or software feature. These features can be used to filter out applications by using the SystemInfo API. The mobile device implementation MUST accurately report the availability of these features through the Tizen Native API `Tizen::System::SystemInfo` API.

2.8. Privilege

Certain APIs have access to privacy-sensitive information (for example contacts, camera, geolocation) or have security or stability implications. If an application uses such APIs, then appropriate privileges MUST be declared in the configuration document for the application.

Privilege is affected by the privilege levels described below. In addition to declaring the privilege, the application MUST have access to the required privilege level:

- Public: for all Tizen developers
- Partner: for trusted application developers (for using security-sensitive API)
- Platform: for OEMs/operators (for development of preloaded applications)

See the Tizen Privilege Guide [ref. 25] for detailed information of the privilege level.

If an application declares a privilege that requires a level higher than public, and the application is not signed with a certificate granting it access to that level, then the implementation MUST block installation and execution of the application.

2.8.1. Tizen Web API

If a web application does not declare a required privilege in the *config.xml* file, access to the corresponding API MUST throw `SecurityError` as specified in the Tizen Web Device API Reference [ref. 30]. Mobile device implementations MUST support this mechanism.

Mobile device implementations MUST NOT change the semantics of permissions as documented in the Tizen Web Application Security and Privacy [ref. 40] for applications using the Tizen Web API. See section A.3.1 for a full list of privileges defined for the Tizen Web API.

2.8.2. Tizen Native API

If a native application does not declare a required privilege in the *manifest.xml* file, mobile device implementations MUST deny the access and throw `E_PRIVILEGE_DENIED` if the corresponding API is accessed.

In addition, certain privileged APIs can access private user data. When these API calls attempt access to private user data without successfully obtaining user consent, mobile device implementations MUST deny the access and throw `E_USER_NOT_CONSENTED` even if the application declares the relevant privileges.

Mobile device implementations MUST enforce a mechanism that limits an application using the Tizen Native API to use privileged APIs only if the privileges it requires are declared. Mobile device implementations MUST NOT change the semantics of permissions and support, as documented in the manifest specification for applications using the Tizen Native API. See section A.3.2 for the full list of privileges defined for the Tizen Native API.

2.9. Push Service

The Tizen Push API enables a Tizen mobile device to receive messages from the Tizen Push Service. The client-side Tizen Push API MAY be omitted as specified in section 2.7, but if implemented, device implementations MUST use the push messaging service provided by the Tizen Server. Tizen Web and Native Push APIs MUST use the C-language APIs (see section A.5) to communicate with the Tizen Push Server. For more detail, see [ref. 45]. The absence or presence of the Push API MUST be reported through the SystemInfo API.

The Tizen Push API is intended to guarantee availability of a common Tizen push service among all Tizen mobile implementations.

2.10. Application Packaging Compatibility

Tizen defines several mandatory application packaging formats. Mobile device implementations MUST correctly process packages in these formats. They MUST NOT extend these packaging formats in a way that would prevent packages generated on the implementation from running on other conforming mobile device implementations.

Nothing in this section precludes mobile device implementations from supporting additional packaging formats outside the requirements of this specification.

2.10.1. Web App Package Support

Mobile device implementations MUST be able to install, remove, list, and update Web application packages in the .wgt format as described in the Tizen Web Runtime Core Specification [ref. 31].

2.10.2. Native App Package Support

Mobile device implementations MUST be able to install, remove, list, and update Native application packages in .tpk format, as described in the Tizen Native Application Development Process [ref. 22].

2.10.3. Hybrid Web/Native Package Support

Mobile device implementations MUST be able to install, remove, list, and update hybrid Web/Native application packages in .wgt format. See [ref. 43].

2.11. WebKit and Browser

2.11.1. WebKit

The WebView and Web Runtime implementations on mobile device implementations MUST be based on the WebKit built from the Tizen reference implementation: WebKit 537.3. Any customizations made by device implementations MUST NOT alter the original web exposed behavior, based on this version of WebKit.

The user agent string reported by the WebKit MUST follow this format:

Mozilla/5.0 (Linux; Tizen *PLATFORM_VER*; *MODEL*) AppleWebKit/*APPLE_WEBKIT_VER*
(KHTML, like Gecko) *APP_NAME*/*APP_VER* Mobile Safari/*APPLE_WEBKIT_VER*

- The value of the *PLATFORM_VER* string MUST be "2.2".
- The value of the *MODEL* string SHOULD be the same as the name of the device. There is no specific format for this field.
- The value of the *APPLE_WEBKIT_VER* string MUST be "537.3".
- The value of the *APP_NAME* string SHOULD be the same as the name of the application.
- The value of the *APP_VER* string SHOULD be the same as the version of the application.
- Mobile device implementations MAY omit the word "Mobile" from the user agent string.

2.11.2. Browser

Mobile device implementations MUST include a browser.

The browser MUST meet the W3C/HTML5 and Supplementary API specifications [ref. 27 and ref. 28]. The default browser on mobile device implementations MUST be based on the WebKit built from the Tizen reference implementation. Any customizations made by device implementations MUST NOT alter the original web exposed behavior, based on the 537.3 version of WebKit.

2.12. Web Runtime

Mobile device implementations MUST support all mandatory requirements in the Tizen Web Runtime Core Specification. [ref. 31]

2.13. User Interface

The Tizen platform provides a full user interface (UI) implementation for both web and native applications. The UI implementation including both APIs and UI controls helps ensure a high quality mobile experience that is consistent across all Tizen mobile devices. The Tizen UX Guide [ref. 26] provides recommendations for styling, navigation and other UI elements.

2.13.1. Theme

Many UI resources are defined in a package called a theme. The Tizen platform provides the system themes for both the web and native UI frameworks. The Tizen reference implementation provides system themes for only 480x800 (WVGA) and 720x1280 (HD) resolutions. System themes for other resolutions are the responsibility of the mobile device implementer, who **MUST** ensure that the Tizen look and feel is compatible with the Tizen style defined by the Tizen UX Guide (see [ref. 26]).

While developers **MAY** create and use custom themes for their applications, the system themes **MUST** match the Tizen look and feel as defined in the Tizen UX Guide to ensure that applications using this theme run consistently across all Tizen mobile devices.

2.13.2. Notification Tray

The Tizen notification tray is a screen area used to display notifications, such as the most recent texts, calls, emails, and more. The notification tray is accessible in the user interface by pulling down on the status bar. Tizen provides APIs to display notifications to the user in the notification tray. Mobile device implementations **MUST** provide the notification tray.

2.13.3. Keys

Mobile device implementations **MUST** provide the following functions through dedicated physical keys.

- **Menu** - used to show menus from the application
- **Back** - used to navigate to previous view in the application

In addition, mobile device implementations **MUST** provide the following functions through either dedicated hardware or dedicated software-implemented keys. These keys **MUST** always be visible to the user and **MUST NOT** be part of the application's screen display.

- **Home** - used to navigate to the Home screen in an application. The key will always send the application in use to the background and bring the Home screen to the front.
- **Volume** - used to adjust volume of incoming call, notification, media, system, etc.
- **Power** - used to turn on/off the device or display.

2.14. Security

The following are security requirements for Tizen platforms.

- The device **MUST** follow the Linux standard security model, including:
 - Applications **MUST** run under a non-root user ID.

- An application **MUST** be allowed to read and write files in its home directory and shared media directory (/opt/usr/media).
- Smack-based access control and process isolation:
 - The device **SHOULD** have a Linux kernel including all Smack features from Linux kernel version 3.5 or later, and the Smack features **SHOULD** be enabled.
 - All applications **SHOULD** run with Smack labels different from the predefined Smack labels.
- Secure execution environment:
 - Native applications **SHALL** be launched by the application framework.
 - Web applications **SHALL** be launched by the web runtime.
 - There **SHOULD NOT** be any set-user-ID binaries in the device.
- Smack supported modules:
 - The device **SHOULD** contain coreutils, d-bus, udev, and Xorg with Smack capability enabled by Tizen.
 - The device **SHOULD** contain the Tizen rpm security plugin.
- Privileged information:
 - The device **MUST NOT** allow any other privilege than the permissions described in the application's manifest file.

2.15. Multimedia

The following media formats/codecs **MUST** be supported by mobile device implementations.

This following list of codecs is a minimum requirement on a Tizen Mobile Device. Please note that the Tizen Technical Steering Group makes no representation that these codecs are unencumbered by patents. Implementation of these codecs **MAY** require patent licenses from the relevant patent holders.

Format	Codec
Audio codec (Decoder)	AAC LC
	AAC+
	Enhanced AAC+
	AMR-NB
	AMR-WB
	MP3
	Vorbis
	PCM (raw PCM)
Audio codec (Encoder)	AAC LC
	AMR-NB
	Raw PCM
Video codec (Decoder)	H.263
	H.264 Baseline Profile
	MPEG-4 part 2
Video codec (Encoder)	H.263

	MPEG-4 part 2
Image codec (Decoder)	BMP
	GIF
	JPEG
	PNG
Image codec (Encoder)	JPEG
	PNG
	BMP

Type	File Type/Container Format
Audio	MPEG-4 (.mp4, .m4a)
	AMR (.amr)
	MP3 (.mp3)
	Ogg (.ogg)
	WAV (.wav)
Video	3GPP (.3gp)
	MPEG-4 (.mp4)
Image	BMP (.bmp)
	GIF (.gif)
	JPEG (.jpg)
	PNG (.png)

2.16. Developer Tools

Mobile device implementations MUST include services that enable communication with the Tizen SDK, enabling the following development tasks:

- Smart Development Bridge [ref. 17].
MUST support all SDB functions to interact with the Tizen SDK. The sdbd (SDB daemon) SHOULD support all commands documented in the SDB Commands section of the SDB reference. The implementation SHOULD allow sdbd to be activated by a device user.
- Log View [ref. 10].
MUST support the function to retrieve the dlog (Tizen platform log).
- OProfile [ref. 16].
MUST support all OProfile functions to interact with Tizen SDK. OProfile MUST be inactive by default, and there MUST be a user-accessible mechanism to turn on OProfile.
- Dynamic Analyzer [ref. 1].
MUST include the Dynamic Analysis framework and make it available for applications to use.

If a mobile device implementation is unrecognized by the SDB, mobile device implementers **MUST** provide suitable device drivers, allowing developers to connect the device using the SDB.

The implementation **MUST** permit side loading through SDB, for development purposes, of applications not signed with a distributor signature.

2.17. Software Update

Mobile implementations **SHOULD** provide a mechanism for updating system software. If provided, user data, application private data, and application shared data **SHOULD** be preserved.

2.18. Tizen Compliance Tests

The Tizen Compliance Tests (TCT) verify conformance to the requirements of this specification. Platforms **MUST** pass the TCT to be considered Tizen compliant.

3. Mobile Profile Hardware Compliance

This chapter describes mandatory and optional hardware components. If a mobile device includes an optional hardware component that has a corresponding API, the implementation **MUST** implement that API as described in this specification.

3.1. Mandatory Hardware Requirements

These minimum hardware features **MUST** be provided by a compliant Mobile device implementation.

3.1.1. Memory Storage

A Tizen mobile device **MUST** have at least 512 MB of RAM if it has IOMMU support. Without an IOMMU, additional RAM **MAY** be required.

Mobile device implementations **MUST** have at least 1 GB of internal storage.

Mobile device implementations **MUST** allow a host computer to access files in the folder `/opt/usr/media` on the device. The precise method is unspecified. Two optional methods are USB mass storage (UMS) and Media Transfer Protocol (MTP).

3.1.2. Sound

Mobile device implementations MUST support at least one audio output.

3.1.3. Connectivity / Networking

Mobile device implementations MUST support at least one form of data networking capable of accessing the Internet. Examples of acceptable data networking technologies include Wi-Fi, LTE, HSPA, Ethernet, etc. Implementations MAY omit any individual mechanism, as long as at least one method is supported.

Since the Tizen Mobile Profile is intended for mobile, internet-connected devices, if a physical networking standard such as Ethernet is provided, the implementation SHOULD also include for at least one common wireless data standard.

3.1.4. Display

Mobile device implementations MUST provide a minimum screen resolution of 320x480(HVGA). However, it is strongly recommended to use a display resolution of 480x800 (WVGA) or 720x1280 (HD) for a mobile device implementation. The Tizen reference implementation has only been validated with these two display resolutions.

The screen orientation MAY be fixed or dynamically rotatable at 90 degree angles.

Mobile device implementations SHOULD support a 32-bit frame buffer.

3.1.5. USB

Mobile device implementations MUST provide USB client functionality.

The implementation MUST support:

- USB 2.0 or later
- the Smart Development Bridge (SDB)

3.1.6. Input Devices

Mobile device implementations MUST provide applications a means of receiving keyboard input from users.

- Implementations MAY omit a full hardware keyboard.

- If no hardware keyboard is available, a soft keyboard **MUST** be provided.
- A soft keyboard or an input method setup **MUST** be able to augment keyboards not capable of a full QWERTY layout. For example, a 12 key number pad can allow a user to enter alphabetical letters through multiple presses of a numeric key.

Mobile device implementations **MUST** include a touchscreen capable of single touch. Multi-touch capability is recommended, if possible.

3.2. Optional Hardware Requirements

If a mobile device includes an optional hardware component that has a corresponding API, the implementation **MUST** implement that API, as described in this specification.

Mobile device implementations **MUST** accurately report the presence of optional hardware components in:

- the Tizen Web API `SystemInfo` API
- the Tizen Native API `Tizen::System::SystemInfo` API

3.2.1. Camera

A mobile device implementation **MAY** omit a camera device. If a camera device is present, at least a 1.5 megapixel rear-facing camera **MUST** be provided.

Mobile device implementations **MUST** accurately report the presence or absence of cameras.

If a mobile device implementation includes a camera hardware feature, it **MUST** support at least one of preview pixel formats for camera previews:

<i>RGB565</i>	The RGB565 pixel format
<i>ARGB8888</i>	The ARGB8888 pixel format
<i>R8G8B8A8</i>	The R8G8B8A8 pixel format The order of color component is guaranteed by the byte unit.
<i>YCbCr420_PLANAR</i>	The 8-bit Y-plane followed by 8-bit 2x2 sub sampled U-plane and V-plane
<i>JPEG</i>	The encoded formats
<i>NV12</i>	The NV12 pixel formats
<i>UYVY</i>	The UYVY pixel format

3.2.2. Graphics

A mobile device implementation **MAY** omit 3D Graphics hardware acceleration. However, it is strongly recommended that it include hardware acceleration, to provide the best possible user experience on the device.

3.2.3. GPS

A mobile device implementation MAY omit GPS hardware. If provided, it MUST support the GPS API.

Mobile device implementations MUST accurately report the presence or absence of GPS.

3.2.4. Sensors

A mobile device implementation MAY omit any and all sensors listed in this specification. If an implementation provides any sensor from this specification, it SHOULD meet the specific requirements for that sensor type. See section A.4 for details.

Mobile device implementations MUST accurately report the presence or absence of sensors.

3.2.5. Telephony

A mobile device implementation MAY omit telephony hardware features. If an implementation includes telephony hardware, it MUST support voice calls and the messaging API (SMS) using cellular technologies.

Mobile device implementations MUST accurately report the presence or absence of telephony.

3.2.6. Bluetooth

A mobile device implementation MAY omit Bluetooth capability. If an implementation includes Bluetooth hardware features, it MUST support the Bluetooth API.

Mobile device implementations SHOULD implement the Audio/Video Remote Control Profile (AVRCP) and the Object Exchange (OBEX) protocol.

Mobile device implementations MUST accurately report the presence or absence of Bluetooth.

3.2.7. Wi-Fi

A mobile device implementation MAY omit Wi-Fi capability. If an implementation includes Wi-Fi hardware features, it MUST support the Wi-Fi API.

Mobile device implementations MUST accurately report the presence or absence of Wi-Fi.

3.2.7.1. Wi-Fi Direct

A mobile device implementation MAY omit Wi-Fi Direct capability. If an implementation includes Wi-Fi Direct hardware features, it MUST support the Wi-Fi Direct API as well as the Wi-Fi API.

Mobile device implementations MUST accurately report the presence or absence of Wi-Fi Direct.

3.2.8. NFC

A mobile device implementation MAY omit NFC capability. If an implementation includes NFC hardware, it MUST support the NFC API.

Mobile device implementation MAY omit NFC reserved push even if NFC hardware is included in the implementation [ref. 29].

Mobile device implementations MUST read/write NFC Data Exchange Format (NDEF) messages in NFC standard formats, such as NFC Forum Tag Types 1, 2, 3, and 4.

A Tizen mobile device MUST support sending and receiving data using the following standards [ref. 11]:

- NFCIP-1 (ISO 18092)
- LLCP 1.0
- SNEP 1.0

Mobile device implementations MUST accurately report the presence or absence of NFC.

3.2.9. Input Devices

A mobile device implementation MAY omit a microphone.

Mobile device implementations MUST accurately report the presence or absence of a microphone.

3.2.10. Secure Element

A mobile device implementation MAY omit Secure Element capability. If an implementation includes the Secure Element feature, it MUST support the SecureElement API. Mobile device implementations MUST accurately report the presence or absence of Secure Element.

4. Mobile Profile Application Compliance

This chapter provides information for application developers to aid them in creating applications that will run on Tizen compliant devices.

4.1. API Use

Applications **MUST** use only the APIs defined in the Tizen Web API and the Tizen Native API specifications when making calls external to the application. Compliant web applications **MAY** also use any RESTful web APIs implemented using HTTP and the principles of REST (Representational State Transfer).

Web applications **MAY** also use RESTful APIs provided by other open services, as well as JavaScript libraries included in the resources of the application, subject to the condition that the web application's configuration specifies the REST API domain in the `<access>` tag, according to the W3C Widget Access Request Policy [ref. 33].

4.1.1. Limited use of C library functions

Note that mobile device implementations are not required to provide any specific commands for use by the C library `system()` and `popen()` interfaces and the `exec()` family of interfaces. Applications using those interfaces **MUST** use them only to execute user-provided commands. For more detail, see [ref. 44].

4.2. Application Packaging

Applications **MUST** follow the packaging guidelines, as defined for the platform. See [ref. 41 and ref. 42].

4.3. Application Lifecycle

Native applications **MUST** be implemented with the Tizen Native API application lifecycle. See [ref. 23].

4.4. Namespace

Applications **SHOULD** include a namespace, such as: `<company>.<application>`. Applications **MUST NOT** overwrite the Tizen API namespaces.

4.5. Application Features and Privileges

A Tizen application **MUST** declare the features (see section A.2) and privileges (see section A.3) that it uses, in the configuration document included in the application package. Further

details on how to implement this requirement are provided in the developer documentation. See [ref. 37 and ref. 25].

The application SHALL be granted privileges only for the listed APIs. In some circumstances, user consent MAY be required before a privilege is granted. User consent may be requested at install time or at access time.

The Tizen Web API configuration document (`config.xml`) uses syntax as shown in these examples:

```
<feature name="http://tizen.org/feature/network.nfc"/>
<tizen:privilege name="http://tizen.org/privilege/application.launch"/>
```

The Tizen Native API configuration document (`manifest.xml`) uses syntax as shown in these examples:

```
<Requirements>
  <Feature Name="http://tizen.org/feature/camera">true</Feature>
</Requirements>
<Privileges>
  <Privilege>http://tizen.org/privilege/notification</Privilege>
</Privileges>
```

Appendix A. Additional Information

This chapter contains tables of information providing further details for API aspects referenced elsewhere in this specification.

A.1. Tizen Application Control

Browse a webpage		
1	operation	http://tizen.org/appcontrol/operation/view
	uri	http://[PATH]
2	operation	http://tizen.org/appcontrol/operation/view
	uri	https://[PATH]
Display an image		
3	operation	http://tizen.org/appcontrol/operation/view
	uri	file://[PATH]
	mime	image/bmp
4	operation	http://tizen.org/appcontrol/operation/view
	uri	file://[PATH]
	mime	image/jpeg
5	operation	http://tizen.org/appcontrol/operation/view

	uri	file://[PATH]
	mime	image/gif
6	operation	http://tizen.org/appcontrol/operation/view
	uri	file://[PATH]
	mime	image/png
Play a sound		
7	operation	http://tizen.org/appcontrol/operation/view
	uri	file://[PATH]
	mime	audio/aac
8	operation	http://tizen.org/appcontrol/operation/view
	uri	file://[PATH]
	mime	audio/amr
9	operation	http://tizen.org/appcontrol/operation/view
	uri	file://[PATH]
	mime	audio/mp3
10	operation	http://tizen.org/appcontrol/operation/view
	uri	file://[PATH]
	mime	audio/wav
Play a video		
11	operation	http://tizen.org/appcontrol/operation/view
	uri	file://[PATH]
	mime	video/mp4
12	operation	http://tizen.org/appcontrol/operation/view
	uri	file://[PATH]
	mime	video/3gpp
Select a file		
13	operation	http://tizen.org/appcontrol/operation/pick
	mime	*/*
14	operation	http://tizen.org/appcontrol/operation/pick
	mime	image/*
15	operation	http://tizen.org/appcontrol/operation/pick
	mime	video/*
16	operation	http://tizen.org/appcontrol/operation/pick
	mime	audio/*

A.2. Tizen Features

Certain hardware or software features MUST be identified by the mobile device implementation and declared in application packages to allow matching of applications to

mobile device implementation so applications can run with full capabilities. The following tables list the official Tizen feature names.

A.2.1. Tizen Web API

Feature key	Description
http://tizen.org/feature/network.push	This key should be specified if an application requires the IP push service feature.
http://tizen.org/feature/network.secure_element	This key should be specified if an application requires the secure element feature.
http://tizen.org/feature/network.nfc	An application requires the Near Field Communication (NFC) feature.
http://tizen.org/feature/network.bluetooth	An application requires the Bluetooth feature.
http://tizen.org/feature/network.wifi	An application requires the Wi-Fi feature.
http://tizen.org/feature/network.telephony	An application requires the telephony feature.
http://tizen.org/feature/network.telephony.mms	An application requires the mms feature.
http://tizen.org/feature/camera	An application requires a camera.
http://tizen.org/feature/microphone	An application requires a microphone.
http://tizen.org/feature/location.gps	An application requires the Global Positioning System (GPS) feature.
http://tizen.org/feature/sensor.accelerometer	An application requires an accelerometer sensor.
http://tizen.org/feature/sensor.gyroscope	An application requires a gyroscope sensor.
http://tizen.org/feature/sensor.magnetometer	An application requires a magnetometer sensor.
http://tizen.org/feature/screen.size.normal	<p>This key should be specified if an application supports all possible current and future resolutions on normal screen size. This key is level 2 key.</p> <p>It should not specify multiple level screen.size.* key(s).</p> <p>But if it does that, only the most specific resolution key(s) are considered and the less specific resolution key(s) is(are) ignored.</p> <p>For example, if an application specifies both screen.size.normal (level 2) and screen.size.normal.320.480 (level 3), then only the screen.size.normal.320.480 is considered and screen.size.normal is ignored. It is strongly recommended to declare one of screen.size.* keys. If none of screen.size.* is declared, it is assumed that the application supports only screen.size.normal.720.1280.</p>

<p>http://tizen.org/feature/screen.size.normal.320.480</p>	<p>This key should be specified if an application supports 320x480 resolution on the normal screen size. This key is the level 3 key. It may specify multiple screen resolution keys if it can support multiple screen resolutions for normal screen size. It should not specify multiple level screen.size.* key(s). But if it does that, only the most specific resolution key(s) are considered and the less specific resolution key(s) is(are) ignored. For example, if an application specifies both screen.size.normal (level 2) and screen.size.normal.720.1280 (level 3), then only the screen.size.normal.720.1280 is considered and screen.size.normal is ignored. The platform will return true if it supports 320x480 resolution on the normal screen size. It is strongly recommended to declare one of screen.size.* keys. If none of screen.size.* is declared, it is assumed that the application supports only screen.size.normal.720.1280.</p>
<p>http://tizen.org/feature/screen.size.normal.480.800</p>	<p>This key should be specified if an application supports 480x800 resolution on the normal screen size. This key is the level 3 key. It may specify multiple screen resolution keys if it can support multiple screen resolutions for normal screen size. It should not specify multiple level screen.size.* key(s). But if it does that, only the most specific resolution key(s) are considered and the less specific resolution key(s) is(are) ignored. For example, if an application specifies both screen.size.normal and screen.size.normal.720.1280, then only the screen.size.normal.720.1280 is considered and screen.size.all is ignored. It is strongly recommended to declare one of screen.size.* keys. If none of screen.size.* is declared, it is assumed that the application supports only screen.size.normal.720.1280.</p>
<p>http://tizen.org/feature/screen.size.normal.540.960</p>	<p>This key should be specified if an application supports 540x960 resolution on the normal screen size. This key is the level 3 key. It may specify multiple screen resolution keys if it can support multiple screen resolutions for normal screen size. It should not specify multiple level</p>

	<p>screen.size.* key(s). But if it does that, only the more specific resolution key(s) are considered and the less specific resolution key(s) is(are) ignored. For example, if an application specifies both screen.size.normal and screen.size.normal.720.1280, then only the screen.size.normal.720.1280 is considered and screen.size.all is ignored. The platform will return true if it supports 540x960 resolution on the normal screen size. It is strongly recommended to declare one of screen.size.* keys. If none of screen.size.* is declared, it is assumed that the application supports only screen.size.normal.720.1280.</p>
<p>http://tizen.org/feature/screen.size.normal.600.1024</p>	<p>This key should be specified if an application supports 600x1024 resolution on the normal screen size. This key is the level 3 key. It may specify multiple screen resolution keys if it can support multiple screen resolutions for normal screen size. It should not specify multiple level screen.size.* key(s). But if it does that, only the most specific resolution key(s) are considered and the less specific resolution key(s) is(are) ignored. For example, if an application specifies both screen.size.normal and screen.size.normal.720.1280, then only the screen.size.normal.720.1280 is considered and screen.size.all is ignored. The platform will return true if it supports 600x1024 resolution on the normal screen size. It is strongly recommended to declare one of screen.size.* keys. If none of screen.size.* is declared, it is assumed that the application supports only screen.size.normal.720.1280.</p>
<p>http://tizen.org/feature/screen.size.normal.720.1280</p>	<p>This key should be specified if an application supports 720x1280 resolution on the normal screen size. This key is the level 3 key. It may specify multiple screen resolution keys if it can support multiple screen resolutions for normal screen size. It should not specify multiple level screen.size.* key(s). But if it does that, only the most specific resolution key(s) are considered and the less specific resolution key(s) is(are) ignored.</p>

	<p>For example, if an application specifies both <code>screen.size.normal</code> and <code>screen.size.normal.720.1280</code>, then only the <code>screen.size.normal.720.1280</code> is considered and <code>screen.size.all</code> is ignored.</p> <p>The platform will return true if it supports 720x1280 resolution on the normal screen size. It is strongly recommended to declare one of <code>screen.size.*</code> keys. If none of <code>screen.size.*</code> is declared, it is assumed that the application supports only <code>screen.size.normal.720.1280</code>.</p>
<p>http://tizen.org/feature/screen.size.normal.1080.1920</p>	<p>This key should be specified if an application supports 1080x1920 resolution on the normal screen size. This key is the level 3 key.</p> <p>It may specify multiple screen resolution keys if it can support multiple screen resolutions for normal screen size. It should not specify multiple level <code>screen.size.*</code> key(s).</p> <p>But if it does that, only the most specific resolution key(s) are considered and the less specific resolution key(s) is(are) ignored. For example, if an application specifies both <code>screen.size.normal</code> and <code>screen.size.normal.720.1280</code>, then only the <code>screen.size.normal.720.1280</code> is considered and <code>screen.size.all</code> is ignored.</p> <p>The platform will return true if it supports 1080x1920 resolution on the normal screen size. It is strongly recommended to declare one of <code>screen.size.*</code> keys. If none of <code>screen.size.*</code> is declared, it is assumed that the application supports only <code>screen.size.normal.720.1280</code>.</p>
<p>http://tizen.org/feature/screen.size.all</p>	<p>This key may be specified if an application supports all possible current and future screen sizes and all possible current and future resolutions per a screen size. This key is the level 1 key.</p> <p>It should not specify multiple level <code>screen.size.*</code> key(s).</p> <p>But if it does that, only the most specific resolution key(s) are considered and the less specific resolution key(s) is(are) ignored. For example, if an application specifies both <code>screen.size.all</code>(level 1) and <code>screen.size.normal</code>(level 2), then only the <code>screen.size.normal</code> is considered and</p>

	screen.size.all is ignored. It is strongly recommended to declare one of screen.size.* keys. If none of screen.size.* is declared, it is assumed that the application supports only screen.size.normal.720.1280.
--	--

A.2.2. Tizen Native API

Feature key	Description
http://tizen.org/feature/camera	This key should be specified if the application requires any camera.
http://tizen.org/feature/camera.back	This key should be specified if an application requires a back-facing camera.
http://tizen.org/feature/camera.back.flash	This key should be specified if an application requires a back-facing camera with a flash.
http://tizen.org/feature/camera.front	This key should be specified if an application requires a front-facing camera.
http://tizen.org/feature/camera.front.flash	This key should be specified if an application requires a front-facing camera with a flash.
http://tizen.org/feature/database.encryption	This key should be specified if an application requires the database encryption feature.
http://tizen.org/feature/graphics.acceleration	This key should be specified if an application requires hardware acceleration for 2D/3D graphics.
http://tizen.org/feature/input.keyboard	This key should be specified if an application requires a built-in physical keyboard.
http://tizen.org/feature/input.keyboard.layout	This key with a specific keyboard layout (String type) value should be specified if an application requires a built-in physical keyboard supporting the specified keyboard layout.
http://tizen.org/feature/location	This key should be specified if an application requires any location positioning feature.
http://tizen.org/feature/location.gps	This key should be specified if an application requires the Global Positioning System (GPS) feature.
http://tizen.org/feature/location.wps	This key should be specified if an application requires the Wi-Fi-based Positioning System (WPS) feature.
http://tizen.org/feature/microphone	This key should be specified if an application requires a microphone.

http://tizen.org/feature/multi_point_touch.pinch_zoom	This key should be specified if an application requires a pinch-zoom gesture feature.
http://tizen.org/feature/multi_point_touch.point_count	This key with a specific number of multi touch points (int type) should be specified if an application requires the specified number of multi touch points at minimum.
http://tizen.org/feature/network.bluetooth	This key should be specified if an application requires the Bluetooth feature.
http://tizen.org/feature/network.nfc	This key should be specified if an application needs to use ANY API which is documented as this feature is required.
http://tizen.org/feature/network.nfc.reserved_push	This key should be specified if an application requires the NFC reserved push feature.
http://tizen.org/feature/network.push	This key should be specified if an application requires the IP push service feature.
http://tizen.org/feature/network.secure_element	This key should be specified if an application requires the secure element feature.
http://tizen.org/feature/network.telephony	This key should be specified if an application needs to use ANY API which is documented as this feature is required.
http://tizen.org/feature/network.telephony.mms	This key should be specified if an application requires the MMS feature.
http://tizen.org/feature/network.telephony.sms.cbs	This key should be specified if an application requires the CBS feature.
http://tizen.org/feature/network.wifi	This key should be specified if an application needs to use ANY API which is documented as this feature is required.
http://tizen.org/feature/network.wifi.direct	This key should be specified if an application requires the Wi-Fi direct feature.
http://tizen.org/feature/opengles.texture_format.3dc	This key should be specified if an application requires the 3DC texture format for OpenGL ES.
http://tizen.org/feature/opengles.texture_format.atc	This key should be specified if the application requires the ATC texture format for OpenGL ES.
http://tizen.org/feature/opengles.texture_format.etc	This key should be specified if the application requires the ETC texture format for OpenGL ES.
http://tizen.org/feature/opengles.texture_format.ptc	This key should be specified if the

	application requires the PTC texture format for OpenGL ES.
http://tizen.org/feature/opengles.texture_format.pvrtc	This key should be specified if the application requires the PVRTC texture format for OpenGL ES.
http://tizen.org/feature/opengles.texture_format.utc	This key should be specified if an application requires the UTC texture format for OpenGL ES.
http://tizen.org/feature/opengles.version.1_1	This key should be specified if an application requires the OpenGL ES version 1.1 at minimum. It should specify at most one OpenGL ES version key. If it specifies more than one, the highest version key is considered and any other keys are ignored.
http://tizen.org/feature/opengles.version.2_0	This key should be specified if an application requires the OpenGL ES version 2.0 at minimum. It should specify at most one OpenGL ES version key. If it specifies more than one, the highest version key is considered and any other keys are ignored.
http://tizen.org/feature/platform.core.cpu.arch.armv7	This key should be specified if an application requires the ARMv7 CPU architecture. This key is automatically added by the Tizen IDE during application compilation.
http://tizen.org/feature/platform.core.cpu.arch.x86	This key should be specified if an application requires the x86 CPU architecture. This key is automatically added by the Tizen IDE during application compilation.
http://tizen.org/feature/platform.core.fpu.arch.sse2	This key should be specified if an application requires the SSE2 FPU architecture. This key is automatically added by the Tizen IDE during application compilation.
http://tizen.org/feature/platform.core.fpu.arch.sse3	This key should be specified if the application requires the SSE3 FPU architecture. This key is automatically added by the Tizen IDE during application compilation.
http://tizen.org/feature/platform.core.fpu.arch.ssse3	This key should be specified if the application requires the SSSE3 FPU architecture. This key is automatically added by the Tizen IDE during application compilation.
http://tizen.org/feature/platform.core.fpu.arch.vfpv3	This key should be specified if the application requires the VFPv3 FPU architecture. This key is automatically

	added by the Tizen IDE during application compilation.
http://tizen.org/feature/platform.native.osp_compatible	This key should be specified if an application requires the bada compatibility mode.
http://tizen.org/feature/screen.auto_rotation	This key should be specified if an application requires the auto rotation feature.
http://tizen.org/feature/screen.size.normal	This key should be specified if an application supports all possible current and future resolutions on normal screen size. This key is level 2 key. It should not specify multiple level screen.size.* key(s). But if it does that, only the most specific resolution key(s) are considered and the less specific resolution key(s) is(are) ignored. For example, if an application specifies both screen.size.normal(level 2) and screen.size.normal.320.480(level 3), then only the screen.size.normal.320.480 is considered and screen.size.normal is ignored. It is strongly recommended to declare one of screen.size.* keys. If none of screen.size.* is declared, it is assumed that the application supports only screen.size.normal.720.1280.
http://tizen.org/feature/screen.size.normal.320.480	This key should be specified if an application supports 320x480 resolution on the normal screen size. This key is the level 3 key. It may specify multiple screen resolution keys if it can support multiple screen resolutions for normal screen size. It should not specify multiple level screen.size.* key(s). But if it does that, only the most specific resolution key(s) are considered and the less specific resolution key(s) is(are) ignored. For example, if an application specifies both screen.size.normal(level 2) and screen.size.normal.720.1280(level 3), then only the screen.size.normal.720.1280 is considered and screen.size.normal is ignored. The platform will return true if it supports 320x480 resolution on the normal screen size. It is strongly recommended to declare one of screen.size.* keys. If none of

	screen.size.* is declared, it is assumed that the application supports only screen.size.normal.720.1280.
http://tizen.org/feature/screen.size.normal.480.800	This key should be specified if an application supports 480x800 resolution on the normal screen size. This key is the level 3 key. It may specify multiple screen resolution keys if it can support multiple screen resolutions for normal screen size. It should not specify multiple level screen.size.* key(s). But if it does that, only the most specific resolution key(s) are considered and the less specific resolution key(s) is(are) ignored. For example, if an application specifies both screen.size.normal and screen.size.normal.720.1280, then only the screen.size.normal.720.1280 is considered and screen.size.all is ignored. It is strongly recommended to declare one of screen.size.* keys. If none of screen.size.* is declared, it is assumed that the application supports only screen.size.normal.720.1280.
http://tizen.org/feature/screen.size.normal.540.960	This key should be specified if an application supports 540x960 resolution on the normal screen size. This key is the level 3 key. It may specify multiple screen resolution keys if it can support multiple screen resolutions for normal screen size. It should not specify multiple level screen.size.* key(s). But if it does that, only the more specific resolution key(s) are considered and the less specific resolution key(s) is(are) ignored. For example, if an application specifies both screen.size.normal and screen.size.normal.720.1280, then only the screen.size.normal.720.1280 is considered and screen.size.all is ignored. The platform will return true if it supports 540x960 resolution on the normal screen size. It is strongly recommended to declare one of screen.size.* keys. If none of screen.size.* is declared, it is assumed that the application supports only screen.size.normal.720.1280.
http://tizen.org/feature/screen.size.normal.600.1024	This key should be specified if an application supports 600x1024

	<p>resolution on the normal screen size. This key is the level 3 key. It may specify multiple screen resolution keys if it can support multiple screen resolutions for normal screen size. It should not specify multiple level screen.size.* key(s). But if it does that, only the most specific resolution key(s) are considered and the less specific resolution key(s) is(are) ignored. For example, if an application specifies both screen.size.normal and screen.size.normal.720.1280, then only the screen.size.normal.720.1280 is considered and screen.size.all is ignored. The platform will return true if it supports 600x1024 resolution on the normal screen size. It is strongly recommended to declare one of screen.size.* keys. If none of screen.size.* is declared, it is assumed that the application supports only screen.size.normal.720.1280.</p>
<p>http://tizen.org/feature/screen.size.normal.720.1280</p>	<p>This key should be specified if an application supports 720x1280 resolution on the normal screen size. This key is the level 3 key. It may specify multiple screen resolution keys if it can support multiple screen resolutions for normal screen size. It should not specify multiple level screen.size.* key(s). But if it does that, only the most specific resolution key(s) are considered and the less specific resolution key(s) is(are) ignored. For example, if an application specifies both screen.size.normal and screen.size.normal.720.1280, then only the screen.size.normal.720.1280 is considered and screen.size.all is ignored. The platform will return true if it supports 720x1280 resolution on the normal screen size. It is strongly recommended to declare one of screen.size.* keys. If none of screen.size.* is declared, it is assumed that the application supports only screen.size.normal.720.1280.</p>
<p>http://tizen.org/feature/screen.size.normal.1080.1920</p>	<p>This key should be specified if an application supports 1080x1920 resolution on the normal screen size. This key is the level 3 key. It may specify</p>

	<p>multiple screen resolution keys if it can support multiple screen resolutions for normal screen size. It should not specify multiple level screen.size.* key(s). But if it does that, only the most specific resolution key(s) are considered and the less specific resolution key(s) is(are) ignored. For example, if an application specifies both screen.size.normal and screen.size.normal.720.1280, then only the screen.size.normal.720.1280 is considered and screen.size.all is ignored. The platform will return true if it supports 1080x1920 resolution on the normal screen size. It is strongly recommended to declare one of screen.size.* keys. If none of screen.size.* is declared, it is assumed that the application supports only screen.size.normal.720.1280.</p>
<p>http://tizen.org/feature/screen.size.all</p>	<p>This key may be specified if an application supports all possible current and future screen sizes and all possible current and future resolutions per a screen size. This key is the level 1 key. It should not specify multiple level screen.size.* key(s). But if it does that, only the most specific resolution key(s) are considered and the less specific resolution key(s) is(are) ignored. For example, if an application specifies both screen.size.all(level 1) and screen.size.normal(level 2), then only the screen.size.normal is considered and screen.size.all is ignored. It is strongly recommended to declare one of screen.size.* keys. If none of screen.size.* is declared, it is assumed that the application supports only screen.size.normal.720.1280.</p>
<p>http://tizen.org/feature/sensor.accelerometer</p>	<p>This key should be specified if an application requires an accelerometer sensor.</p>
<p>http://tizen.org/feature/sensor.accelerometer.wakeup</p>	<p>This key should be specified if an application requires the wake-up feature by accelerometer sensor.</p>
<p>http://tizen.org/feature/sensor.gyroscope</p>	<p>This key should be specified if an application requires a gyroscope sensor.</p>
<p>http://tizen.org/feature/sensor.gyroscope.wakeup</p>	<p>This key should be specified if an application requires the wake-up feature</p>

	by gyroscope sensor.
http://tizen.org/feature/sensor.magnetometer	This key should be specified if the application requires a magnetometer sensor.
http://tizen.org/feature/sensor.magnetometer.wakeup	This key should be specified if the application requires the wake-up feature by magnetometer sensor.
http://tizen.org/feature/sensor.proximity	This key should be specified if the application requires a proximity sensor.
http://tizen.org/feature/sensor.proximity.wakeup	This key should be specified if the application requires the wake-up feature by proximity sensor.
http://tizen.org/feature/sensor.tiltmeter	This key should be specified if the application requires a tiltmeter sensor.
http://tizen.org/feature/sensor.tiltmeter.wakeup	This key should be specified if the application requires the wake-up feature by tiltmeter sensor.
http://tizen.org/feature/shell.appwidget	This key should be specified if the application requires the AppWidget (DynamicBox) feature.
http://tizen.org/feature/sip.voip	This key should be specified if an application requires the Voice over Internet Protocol (VoIP) feature.
http://tizen.org/feature/speech.recognition	This key should be specified if an application requires the speech recognition (STT) feature.
http://tizen.org/feature/speech.synthesis	This key should be specified if an application requires the speech synthesis (TTS) feature.
http://tizen.org/feature/usb.accessory	This key should be specified if an application requires the USB client (or accessory) feature.
http://tizen.org/feature/usb.host	This key should be specified if an application requires the USB host feature.
http://tizen.org/feature/vision.face_recognition	This key should be specified if an application requires the face recognition feature.
http://tizen.org/feature/vision.image_recognition	This key should be specified if an application requires the image recognition feature.
http://tizen.org/feature/vision.qrcode_generation	This key should be specified if an application requires the QR code generation feature.
http://tizen.org/feature/vision.qrcode_recognition	This key should be specified if an application requires the QR code recognition feature.

A.3. Tizen Privileges

Certain APIs have restricted usage models, either requiring an elevated privilege level or some confirmation mechanism (such as user consent) prior to usage. Applications declare their intent to use privileged APIs by including a privilege declaration in the application configuration document. The following tables list the official Tizen privilege names.

A.3.1. Tizen Web API Privileges

W3C/HTML5 API related Privileges

Privilege	Privilege Level	User Consent
http://tizen.org/privilege/location	Public	Needed
http://tizen.org/privilege/notification	Public	
http://tizen.org/privilege/mediacapture	Public	
http://tizen.org/privilege/unlimitedstorage	Public	

Supplementary API related Privileges

Privilege	Privilege Level	User Consent
http://tizen.org/privilege/fullscreen	Public	

Tizen Web Device API related Privileges

Privilege	Privilege Level	User Consent
http://tizen.org/privilege/alarm	Public	
http://tizen.org/privilege/application.info	Public	
http://tizen.org/privilege/application.launch	Public	
http://tizen.org/privilege/appmanager.certificate	Partner	
http://tizen.org/privilege/appmanager.kill	Partner	
http://tizen.org/privilege/bluetooth.admin	Public	Needed
http://tizen.org/privilege/bluetooth.gap	Public	Needed
http://tizen.org/privilege/bluetooth.health	Public	Needed
http://tizen.org/privilege/bluetooth.spp	Public	Needed
http://tizen.org/privilege/bluetoothmanager	Platform	Needed
http://tizen.org/privilege/bookmark.read	Platform	
http://tizen.org/privilege/bookmark.write	Platform	
http://tizen.org/privilege/calendar.read	Public	Needed
http://tizen.org/privilege/calendar.write	Public	Needed
http://tizen.org/privilege/callhistory.read	Public	Needed
http://tizen.org/privilege/callhistory.write	Public	Needed

http://tizen.org/privilege/contact.read	Public	Needed
http://tizen.org/privilege/contact.write	Public	Needed
http://tizen.org/privilege/content.read	Public	
http://tizen.org/privilege/content.write	Public	
http://tizen.org/privilege/datacontrol.consumer	Public	
http://tizen.org/privilege/datasync	Public	
http://tizen.org/privilege/download	Public	
http://tizen.org/privilege/filesystem.read	Public	
http://tizen.org/privilege/filesystem.write	Public	
http://tizen.org/privilege/messaging.read	Public	Needed
http://tizen.org/privilege/messaging.write	Public	Needed
http://tizen.org/privilege/networkbearerselection	Partner	
http://tizen.org/privilege/nfc.admin	Public	Needed
http://tizen.org/privilege/nfc.common	Public	Needed
http://tizen.org/privilege/nfc.p2p	Public	Needed
http://tizen.org/privilege/nfc.tag	Public	Needed
http://tizen.org/privilege/notification	Public	
http://tizen.org/privilege/package.info	Public	
http://tizen.org/privilege/packagemanager.install	Platform	
http://tizen.org/privilege/power	Public	
http://tizen.org/privilege/push	Public	
http://tizen.org/privilege/secureelement	Public	
http://tizen.org/privilege/setting	Public	
http://tizen.org/privilege/system	Public	
http://tizen.org/privilege/systemmanager	Partner	
http://tizen.org/privilege/websetting	Public	

A.3.2. Tizen Native API Privileges

Privilege	Privilege Level	User Consent
http://tizen.org/privilege/account.read	Public	Needed
http://tizen.org/privilege/account.write	Public	Needed
http://tizen.org/privilege/alarm	Public	
http://tizen.org/privilege/application.launch	Public	
http://tizen.org/privilege/appmanager.kill	Partner	
http://tizen.org/privilege/appmanager.launch	Partner	
http://tizen.org/privilege/appmanager.setting	Platform	
http://tizen.org/privilege/appsetting	Platform	
http://tizen.org/privilege/appusage	Partner	
http://tizen.org/privilege/appwidgetprovider.install	Public	

http://tizen.org/privilege/audiomanager.route	Partner	
http://tizen.org/privilege/audiorecorder	Public	
http://tizen.org/privilege/bluetooth.admin	Public	Needed
http://tizen.org/privilege/bluetooth.gap	Public	Needed
http://tizen.org/privilege/bluetooth.health	Public	Needed
http://tizen.org/privilege/bluetooth.opp	Public	Needed
http://tizen.org/privilege/bluetooth.spp	Public	Needed
http://tizen.org/privilege/bluetoothmanager	Platform	Needed
http://tizen.org/privilege/bookmark.read	Platform	
http://tizen.org/privilege/bookmark.write	Platform	
http://tizen.org/privilege/calendar.read	Public	Needed
http://tizen.org/privilege/calendar.write	Public	Needed
http://tizen.org/privilege/callforward	Platform	
http://tizen.org/privilege/camera	Public	
http://tizen.org/privilege/cellbroadcast	Platform	
http://tizen.org/privilege/certificate.read	Partner	
http://tizen.org/privilege/certificate.write	Partner	
http://tizen.org/privilege/contact.read	Public	Needed
http://tizen.org/privilege/contact.write	Public	Needed
http://tizen.org/privilege/content.read	Public	
http://tizen.org/privilege/content.write	Public	
http://tizen.org/privilege/customnetaccount	Partner	
http://tizen.org/privilege/datacontrol.consumer	Public	
http://tizen.org/privilege/dns	Public	
http://tizen.org/privilege/download	Public	
http://tizen.org/privilege/geolocationpermission.read	Platform	
http://tizen.org/privilege/geolocationpermission.write	Platform	
http://tizen.org/privilege/http	Public	
http://tizen.org/privilege/ime	Partner	
http://tizen.org/privilege/imemanager	Platform	
http://tizen.org/privilege/inputmanager	Platform	
http://tizen.org/privilege/location	Public	Needed
http://tizen.org/privilege/lockmanager	Platform	
http://tizen.org/privilege/messaging.read	Public	Needed
http://tizen.org/privilege/messaging.write	Public	Needed
http://tizen.org/privilege/netstatisticsmanager	Platform	
http://tizen.org/privilege/network.account	Public	
http://tizen.org/privilege/network.connection	Public	
http://tizen.org/privilege/network.statistics	Public	
http://tizen.org/privilege/networkmanager	Platform	
http://tizen.org/privilege/nfc.admin	Public	Needed
http://tizen.org/privilege/nfc.common	Public	Needed
http://tizen.org/privilege/nfc.p2p	Public	Needed
http://tizen.org/privilege/nfc.tag	Public	Needed

http://tizen.org/privilege/nfcmanager	Platform	
http://tizen.org/privilege/notification	Public	
http://tizen.org/privilege/package.info	Public	
http://tizen.org/privilege/packagemanager.info	Platform	
http://tizen.org/privilege/packagemanager.install	Platform	
http://tizen.org/privilege/packagemanager.setting	Platform	
http://tizen.org/privilege/power	Public	
http://tizen.org/privilege/privacymanager.read	Platform	
http://tizen.org/privilege/privacymanager.write	Platform	
http://tizen.org/privilege/privilegemanager.read	Partner	
http://tizen.org/privilege/push	Public	
http://tizen.org/privilege/secureelement	Public	
http://tizen.org/privilege/setting	Public	
http://tizen.org/privilege/settingmanager.read	Platform	
http://tizen.org/privilege/settingmanager.write	Platform	
http://tizen.org/privilege/shortcut.install	Public	
http://tizen.org/privilege/smstrigger	Partner	
http://tizen.org/privilege/socket	Public	
http://tizen.org/privilege/system	Public	
http://tizen.org/privilege/systemmanager	Partner	
http://tizen.org/privilege/telephony	Public	
http://tizen.org/privilege/telephonymanager	Platform	
http://tizen.org/privilege/uimanager	Platform	
http://tizen.org/privilege/userprofile.read	Public	Needed
http://tizen.org/privilege/userprofile.write	Public	Needed
http://tizen.org/privilege/vibrator	Public	
http://tizen.org/privilege/videorecorder	Public	
http://tizen.org/privilege/wappush	Platform	
http://tizen.org/privilege/web.privacy	Public	
http://tizen.org/privilege/web.service	Public	
http://tizen.org/privilege/wifi.admin	Public	
http://tizen.org/privilege/wifi.read	Public	
http://tizen.org/privilege/wifi.wifidirect.admin	Public	Needed
http://tizen.org/privilege/wifi.wifidirect.read	Public	Needed
http://tizen.org/privilege/wifimanager	Platform	

A.4. Sensor Hardware Capabilities

The following table details strongly recommended capabilities of sensors which have corresponding programming interfaces in Tizen.

Sensor Type	Required Capabilities
Accelerometer	Axis: 3 (x, y, z)

	Data range: -2G ~ 2G
	Minimum data rate: 50Hz
	Minimum resolution 0.1m/s ²
	Unit: G, 9.8m/s ² = 1G
Gyroscope	Axis: 3 (x, y, z)
	Data range: -8.73 rad/s ~ 8.73 rad/s
	Minimum data rate: 50Hz
	Minimum resolution 0.01 rad / s
	Unit: rad/s, radians per second
Magnetometer	Axis: 3 (x, y, z) with azimuth, pitch, roll
	Data range: -1200 μT ~ 1200 μT
	Minimum data rate: 50Hz
	Minimum resolution 1μT
	Unit: μT, micro tesla
Proximity	Provide the lux value, can turn on/off
	Data range: 0 ~ 5 cm
	Minimum data rate: 10Hz
	Minimum resolution 1 cm
	Unit: cm

A.5. Push C-language APIs

The following Push C-language APIs MUST be used to provide the Tizen Push API.

- API list
 - int **push_connect**(const char *app_id, push_state_cb state_callback, push_notify_cb notify_callback, void *user_data, push_connection_h *connection);
 - void **push_disconnect**(push_connection_h connection);
 - int **push_register**(push_connection_h connection, service_h service, push_result_cb result_callback, void *user_data);
 - int **push_deregister**(push_connection_h connection, push_result_cb callback, void *user_data);
 - int **push_get_notification_data**(push_notification_h notification, char **data);
 - int **push_get_notification_message**(push_notification_h notification, char **msg);
 - int **push_get_notification_time**(push_notification_h notification, long long int *received_time);
 - int **push_get_unread_notification**(push_connection_h connection, push_notification_h *noti);
 - int **push_get_registration_id**(push_connection_h connection, char **reg_id);

- void **push_free_notification**(push_notification_h noti);
- Callback list
 - typedef void (***push_state_cb**)(push_state_e state, const char *err, void *user_data);
 - typedef void (***push_notify_cb**)(push_notification_h noti, void *user_data);
 - typedef void (***push_result_cb**)(push_result_e result, const char *msg, void *user_data);