

Tizen® 5.5 Compliance Specification for TV Profile

Version 0.95 beta

Copyright© 2014 - 2019 Samsung Electronics Co., Ltd.
Copyright© 2014, 2015 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
2.3 version 1.0	11 Dec 2014	Tizen TSG	Official release
2.4 version 1.0	18 Jan 2016	Tizen TSG	Official release
3.0 version 1.0	31 Jan 2017	Tizen TSG	Official release
4.0 version 0.9	30 Dec 2017	Tizen TSG	Release candidate
5.0 version 0.9	31 Dec 2018	Tizen TSG	Initial edits
5.5 version 0.9	31 Dec 2019	Tizen TSG	Initial edits

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.
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.
Mobile	Portable, connected devices, such as phones and tablets. A platform target for Tizen.
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.
TV	Connected smart televisions and set-top boxes. A platform target for

Term	Definition
	Tizen.
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 (if present) through which the user may interact 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.
Wearable	Miniature computing devices worn by the user on the body or clothing. A platform target for Tizen.
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. TV 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 C# API
 - 2.3.1. Namespace
 - 2.3.2. Tizen C# API Categories
 - 2.3.3. Behavior of Unsupported APIs
 - 2.4. Application Control
 - 2.5. Platform Attributes
 - 2.6. Optional APIs
 - 2.6.1. Tizen Web API
 - 2.6.2. Tizen C# API
 - 2.7. Privilege
 - 2.7.1. Tizen Web API
 - 2.7.2. Tizen C# API
 - 2.8. Application Packaging Compatibility
 - 2.8.1. Web App Package Support
 - 2.8.2. C# App Package Support
 - 2.9. Chromium and Browser
 - 2.9.1. Chromium
 - 2.9.2. Browser
 - 2.10. Web Runtime
 - 2.11. C# Runtime
 - 2.12. User Interface
 - 2.12.1. Keys
 - 2.13. Security
 - 2.14. Multimedia
 - 2.15. Developer Tools
 - 2.16. Software Update
 - 2.17. Tizen Compliance Tests
 - 2.17.1. Satisfying TCT preconditions
3. TV 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. Input Devices
 - 3.2. Optional Hardware Requirements
 - 3.2.1. Camera
 - 3.2.2. Tuner
 - 3.2.3. HDMI Input
 - 3.2.4. Wi-Fi
 - 3.2.5. Audio Input Devices
 - 3.2.6. USB
 - 3.2.7. Graphics
 - 3.2.8. Picture-in-picture (PIP)
 - 3.2.9. TV Information
 - 3.2.10. Bluetooth
- 4. TV Profile Application Compliance
 - 4.1. API Use
 - 4.2. Application Packaging
 - 4.3. Namespace
 - 4.4. Application Features and Privileges
 - 4.5. Profile Declaration
 - 4.6. Web UI Framework

1. Overview

This specification defines the operating environment of the Tizen platform. It is intended to be used by both application developers and TV device implementers 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. 2].

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 TV 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 TV devices, and know how Tizen devices will behave.
- **TV device implementers:** know how to implement device hardware, security configurations, services, APIs, etc.

1.3. Tizen Compliance Model

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

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:** built to a profile, applications will run on devices that are compliant to that profile.

The Tizen Compliance Tests for a profile will measure conformance to the Tizen Compliance Specification for that profile.

Note: *This specification describes only the compliance requirements for the Tizen TV Profile. Other supported profiles have their own related specifications.*

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. If deemed necessary, updates may be issued between releases. All compliance requirements for the Tizen TV 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 available at <https://source.tizen.org>. If modifications or replacements to the source code must be made, the implementer is 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. [N] Globalize 0.1.0a2: <https://github.com/jquery/globalize/tree/v0.1.0a2>
2. [N] IETF RFC 2119 "Key words for use in RFCs to Indicate Requirement Levels": <http://www.ietf.org/rfc/rfc2119.txt>
3. [N] jQuery 1.8.2: <http://blog.jquery.com/2012/09/20/jquery-1-8-2-released/>
4. [N] jQuery Mobile 1.2.0: <https://github.com/jquery/jquery-mobile/tree/1.2.0>
5. [I] Log View Reference: <https://developer.tizen.org/development/tizen-studio/native-tools/debugging-your-app/log-view>
6. [I] Smart Development Bridge: <https://developer.tizen.org/development/tizen-studio/native-tools/running-and-testing-your-app/sdb>
7. [N] Tizen Web Supplementary API References: https://www.tizen.org/tv/web_supplementary

8. [N] Tizen W3C/HTML5 API References: https://www.tizen.org/tv/w3c_api
9. [N] Tizen Web Device API Reference: https://www.tizen.org/tv/web_device_api
10. [N] Tizen Web Runtime Core Specification 3.0:
<https://source.tizen.org/sites/default/files/page/tizen-3.0-wrt-core-spec.pdf>
11. [N] W3C Widget Access Request Policy (W3C Recommendation 7 February 2012 version): <http://www.w3.org/TR/widgets-access/>
12. [N] Optional Tizen W3C/HTML5 Features: https://www.tizen.org/tv/w3c_feature
13. [N] Optional Tizen Web Device Features:
https://www.tizen.org/tv/web_device_api_feature
14. [I] Tizen Application Filtering: <https://developer.tizen.org/development/getting-started/native-application/understanding-tizen-programming/application-filtering>
15. [I] Application Controls for Tizen Web applications:
https://developer.tizen.org/development/guides/web-application/tizen-features/application/application_controls
16. [I] Tizen Web Application Security and Privacy:
<https://developer.tizen.org/development/getting-started/web-application/understanding-tizen-programming/web-runtime>
17. [I] Tizen Web Application Package Manager:
<https://developer.tizen.org/development/getting-started/web-application/tizen-application-model>
18. [N] Tizen Web API Privileges: <https://www.tizen.org/tv/privilege>
19. [I] Document Object Model (DOM) Level 3 Events Specification:
<http://www.w3.org/TR/DOM-Level-3-Events-key/>
20. [I] Document Object Model (DOM) Level 3 Events Fixed Key Specification:
<http://www.w3.org/TR/2014/WD-DOM-Level-3-Events-20140925/#fixed-virtual-key-codes>

2. TV Profile Software Compliance

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

2.1. General Principles

TV device implementations **MUST** include support for the Tizen Web API and Tizen C# API.

- The TV device implementation **MUST** report the availability of optional hardware and software features (see section 2.6) as platform attributes.
- If a TV device implementation reports that it supports a particular optional hardware or software feature, it **MUST** implement the entire corresponding API.
- Whether a TV 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 section 2.2.4 for Web API.

2.2. Tizen Web API

2.2.1. Namespace

TV device implementations MUST NOT modify the API namespace listed in the Tizen Web Device API Reference [ref. 9], including `tizen.*`.

2.2.2. Tizen Web API Categories

- **W3C/HTML5 APIs:** includes the standard APIs defined by W3C, such as HTML5, CSS3, and Widget Specification. See [ref. 8].
- **Supplementary APIs:** non-W3C specifications, such as WebGL, Typed Array, FullScreen API, and viewport Meta Tag. See [ref. 7].
- **Web Device API:** defined by the Tizen project to facilitate the development of web applications by accessing various device features that are not fully covered by W3C APIs. The APIs enable interacting with device features, such as media content, alarm, or system information [ref. 9].
- **External Framework APIs:** includes cross-platform framework APIs supported by Tizen project, which function on more than one computer architecture or operating system. For example, the Apache Cordova plugin APIs such as device, file, media, events, and globalization are included. See [ref. 9].

2.2.3. Preliminary Web APIs

The Tizen Web API includes some preliminary Web API specifications which are in an early stage in the W3C development cycle. Preliminary revisions are referred to in the W3C maturity model 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. 8].

TV 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

TV device implementations MUST NOT omit any web API listed in the Tizen Web API specification, except those specified as optional in section 2.6.1 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.tvchannel` MUST

return “undefined” if the TVChannel module is not supported on the device. In case APIs in a module depend on a certain optional feature which is not present, those APIs MUST report `NotSupportedError`. For example, if a device is not cellular network enabled, an attempt to start a download over a cellular network by calling the `tizen.download.start` method MUST report `NotSupportedError`.

2.3. Tizen C# API

2.3.1. Namespace

TV device implementations MUST provide all of the API namespaces and APIs listed in the Tizen FX API Reference [ref. 9], including `Tizen.*`.

2.3.2. Tizen C# API Categories

- **.NET Standard API:** implements the .NET Base Class library, allows you to use the well known C# language base class libraries and features, such as collections, threading, file I/O, and LINQ.c.
- **TizenFX API:** allows you to access platform-specific features not covered by the generic .NET and Xamarin.Forms features, such as system information and status, battery status, sensor data, and account and connectivity services. Some APIs may be indicated as “preview” which may change in their final official versions. They may be omitted from the scope of Tizen Compliance Tests.
- **Xamarin.Forms**, which allows you to efficiently build a user interface from standard components in C# or XAML. Xamarin.Forms APIs may exist in application packages as commonly used for all Xamarin.Forms supported platforms. In this case, the providers of Xamarin.Forms Nuget packages for the Tizen devices MUST be guarantee correct behavior of Xamarin.Forms APIs on compatible Tizen platform version and may be omitted from the scope of Tizen Compliance Tests.

2.3.3. Behavior of Unsupported APIs

TV device implementations MUST NOT omit any C# API listed in the TizenFX API. Specially, C# APIs are designed to be common to all Tizen profiles. If some APIs are not supported on the Tizen devices by software or hardware limitations, it MUST throw appropriate “`NotSupportedException`” exception as specified in the TizenFX specification.

2.4. Application Control

The application control interface in the Tizen Web API enables launching an application directly using an application ID or invoking specific application functionality remotely through inter-process communication (IPC).

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

There are no mandatory platform provided application controls in this profile, however TV device implementations **MUST** allow Tizen applications to register application controls for use by other Tizen applications.

Further details on Application Controls are provided in the developer documentation [ref. 15].

2.5. Platform Attributes

TV device implementations **MUST** provide accurate platform attributes via the appropriate interfaces in the Tizen Web API for System Information.

Platform attributes include but are not limited to the following:

- Device capabilities (see section 2.6)
- Information about data storage devices
- Display information
- Information about the device orientation
- Locale information
- Network information

2.6. 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 TV device requirement such as higher processing power or memory requirements (See section 3.1 for minimum hardware requirements).

TV device implementations **MUST** implement all APIs listed in the referenced API specifications, except those specified as optional in this section. Optional APIs are dependent on the availability of particular underlying hardware or software features.

2.6.1. Tizen Web API

The Tizen Web APIs specified as optional in [ref. 12 and ref. 13] will not be implemented if a TV device implementation does not include the corresponding hardware or software feature. The TV device implementation MUST accurately report the availability of these underlying features through the Tizen Web API System Information API.

2.6.2. Tizen C# API

See section 2.3.3. If a TV device implementation does not include the corresponding hardware or software feature, the APIs MUST throw “NotSupportedException” exception and the TV device implementation MUST accurately report the availability of these underlying features through the Tizen C# API System Information API.

2.7. Privilege

Certain APIs have access to privacy-sensitive information (for example the TVChannel API can be used to continuously update TV viewers about program information) or have security or stability implications. If an application uses such APIs, then appropriate privileges MUST be declared in the application's *config.xml* file.

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
- Platform: for OEMs/operators

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.

2.7.1. 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. 9]. TV device implementations MUST support this mechanism.

TV device implementations MUST NOT change the semantics of permissions as documented in the Tizen Web Application Security and Privacy [ref. 16] for applications using the Tizen

Web API. A full list of privileges defined for the Tizen Web API is available in the developer's guide [ref. 18].

2.7.2. Tizen C# API

If a C# application does not declare a required privilege in the `tizen-manifest.xml` file, access to the corresponding API MUST return errors.

2.8. Application Packaging Compatibility

Tizen defines a mandatory application packaging format. TV device implementations MUST correctly process packages in this format. They MUST NOT extend the packaging format in a way that would prevent packages generated on the implementation from running on other conforming TV device implementations.

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

2.8.1. Web App Package Support

TV 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. 10]. `.wgt` packages MUST NOT contain more than one UI application or the behavior is undefined.

2.8.2. C# App Package Support

TV device implementations MUST be able to install, remove, list, and update C# application packages in the `.tpk` format.

2.9. Chromium and Browser

2.9.1. Chromium

The WebView and Web Runtime implementations on TV device implementations SHOULD be based on Chromium version 47 or higher. This is strongly recommended for maintaining application compatibility across Tizen TV devices. Any customizations made by device implementations SHOULD NOT alter the original web exposed behavior from the Chromium version used.

If Chromium is used, the user agent string reported by the Chromium MUST follow this format:

Mozilla/5.0 (*DEVICE_TYPE*; Linux; Tizen *PLATFORM_VER*; *MODEL*) AppleWebKit/537.36 (KHTML, like Gecko) *APP_NAME*/*APP_VER* Chrome/47.0.2526.69 TV Safari/537.36

- The value of the *DEVICE_TYPE* string SHOULD be the same as the type of the device.
- The value of the *PLATFORM_VER* string MUST be "5.5".
- 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 *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.
- TV device implementations MAY omit the word "TV" from the user agent string.

2.9.2. Browser

TV device implementations SHOULD include a browser.

If a browser is included, the browser SHOULD meet the W3C/HTML5 and Supplementary API specifications [ref. 7 and ref. 8]. The default browser on TV device implementations SHOULD be based on Chromium 47 or higher. Any customizations made by device implementations SHOULD NOT alter the original web exposed behavior from the Chromium version used.

If a browser is included, TV device implementations MUST provide a pointer or pointer emulation which supports at least positioning, mouseover and left click events to enable access to UI components such as links, cascading menus and hover elements on web pages that are open in the browser.

2.10. Web Runtime

TV device implementations MUST support all mandatory requirements in the Tizen Web Runtime Core Specification [ref. 10].

2.11. C# Runtime

TV device implementations MUST support all mandatory requirements of .NET CoreCLR.

2.12. User Interface

2.12.1. Keys

TV device implementations **MUST** provide functionality to deliver the following mandatory events to a listening application in response to controller key presses:

- **Navigate** – "four direction" button or button array to navigate menus and guides. Sends DOM `'ArrowDown'`, `'ArrowLeft'`, `'ArrowRight'` and `'ArrowUp'` key events.
- **Enter** – used to choose menu or guide options. Sends a DOM `'Enter'` key event.
- **Back** – used to navigate to previous view in the applications. Sends a `'tizenhwkey'` event with `keyName == "back"`.

The following key codes **MUST** be used for mandatory events [ref. 20]:

Key event	Key code
<code>'ArrowLeft'</code>	37
<code>'ArrowUp'</code>	38
<code>'ArrowRight'</code>	39
<code>'ArrowDown'</code>	40
<code>'Enter'</code>	13

TV device implementations **MAY** provide functionality to deliver the following events to a listening application in response to controller key presses:

- **Channel control** – used to select channel. Sends DOM `'ChannelDown'` and `'ChannelUp'` key events.
- **Volume control** – used to set the volume. Sends DOM `'VolumeDown'` and `'VolumeUp'` key events.
- **Media Control** – used to control media content. Can be used to play, record, pause or seek media content. Sends a variety of DOM key events such as `'MediaPlay'` and `'MediaPause'`.
- **Color (red, green, yellow, blue etc.)** – used by application developers to carry out custom operations in their applications. Sends DOM `'ColorF0Red'`, `'ColorF1Green'`, `'ColorF2Yellow'` and `'ColorF3Blue'` key events.

To receive events from non-mandatory keys and redefine their operation within an application, application developers need to use the `registerKey()` method in the `TVInputDevice` API to receive input events and their corresponding DOM event key codes. TV device implementers **MUST** send key events which the device supports and **MUST** expose supported keys to applications via the `TVInputDevice` API [ref. 19].

2.13. 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.
- Smack-based access control and process isolation:
 - The device **MUST** have all Smack features from Linux kernel version 3.12 or later, and the Smack features **MUST** be enabled.
 - All applications **MUST** run with Smack labels different from the predefined Smack labels.
 - The device **MUST** use file systems which supports extended attributes (XATTR) and traditional discretionary access control (DAC) attributes such as owner, group, and permissions except for the case of external storage such as USB mass storage.
- Secure execution environment:
 - 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 or equivalent, d-bus, and udev with Smack capability enabled by Tizen.
 - The device **SHOULD** contain the Tizen rpm security plugin.
- Privileged information:
 - The device **MUST** only allow an application to carry out an operation if it has the privilege and permission to do so. Privileges will be declared in the application's manifest file.

2.14. Multimedia

The following tables list media formats/codecs for TV device implementations and whether they are **REQUIRED** or **OPTIONAL**. 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	Requirement
Audio Codec (Decoder)	AAC LC	REQUIRED
	HE-AAC	OPTIONAL
	AC3	REQUIRED
	EAC3	OPTIONAL
	MPEG	REQUIRED
	DTS (Core)	OPTIONAL
	WMA v7/v8/v9 (Standard v1/v2/v3)	OPTIONAL

	Vorbis	REQUIRED
	PCM (raw PCM)	REQUIRED
Video Codec (Decoder)	MPEG1	OPTIONAL
	MPEG2 (SP/MP)	REQUIRED
	MPEG4 (SP/ASP)	REQUIRED
	H.264	REQUIRED
	Divx 3.11/4/5/6	OPTIONAL
	VP6/8	OPTIONAL
	Theora	OPTIONAL
Image Decoder (Decoder)	JPEG	REQUIRED
	BMP	OPTIONAL
	PNG	OPTIONAL

Type	File Type / Container format	Requirement
Audio	MPEG (*.mp3)	REQUIRED
	MPEG4 (*.m4a, *.mpa)	REQUIRED
	OGG (*.ogg)	REQUIRED
	WMA (*.wma)	OPTIONAL
	WAV (.wav)	OPTIONAL
Video	AVI (*.avi, *.divx)	REQUIRED
	MKV (*.mkv)	REQUIRED
	MP4 (*.mp4)	REQUIRED
	PS (*.mpg, *.mpeg)	OPTIONAL
	TS (*.ts, *.tp, *.trp, *.m2ts, *.mts)	REQUIRED
Image	JPEG (*.jpeg, *.jpg)	REQUIRED
	PNG (*.png)	OPTIONAL
	BMP (*.bmp)	OPTIONAL

2.15. Developer Tools

TV device implementations MAY include services that enable communication with the Smart Development Bridge (SDB) in the Tizen SDK. If the implementation includes such support, the following development tasks MUST be available:

- MUST support all SDB functions [ref. 6] to interact with the Tizen SDK. The SDB daemon (sdbd) 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.
- MUST support the Log View [ref. 5] function to retrieve the Tizen platform log (dlog).

In addition, if the implementation supports SDB it MUST support either USB 2.0 or later or another data networking technology listed in section 3.1.3.

While SDB support is OPTIONAL in production devices, device implementation MUST have a device driver available enabling connection to SDB in order to execute the Tizen Compliance Tests (TCT). This driver need not be available in production devices.

2.16. Software Update

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

2.17. Tizen Compliance Tests

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

2.17.1. Satisfying TCT preconditions

TV device implementations MUST satisfy preconditions to pass TCT.

The list of TCT preconditions that MUST be satisfied are available in the Tizen 5.5.0 TCT for TV Profile.

3. TV Profile Hardware Compliance

This chapter describes mandatory and optional hardware components.

3.1. Mandatory Hardware Requirements

These minimum hardware features MUST be provided by a compliant TV device implementation and any corresponding API must be fully implemented.

3.1.1. Memory Storage

A Tizen TV device MUST have at least 512MB of RAM to run Web applications smoothly.

TV device implementations MUST have at least 2 GB of internal storage.

TV device implementations SHOULD allow an external device to access files in the shared media folder on the device. The precise method is unspecified.

3.1.2. Sound

TV device implementations MUST support at least one audio output.

3.1.3. Connectivity / Networking

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

3.1.4. Display

A TV device implementation, whether it is an integrated device with display or a controller-type device without its own display, MUST support driving a display with a minimum screen resolution of 1280x720 (HD). However, it is strongly recommended to use a display resolution of 1920x1080 (FHD) or 3840x2160 (UHD) for a TV device implementation.

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

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

3.1.5. Input Devices

TV 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.

3.2. Optional Hardware Requirements

If a TV device implementation reports that it includes an optional hardware component that has a corresponding optional API, the implementation MUST fully implement that API, as described in this specification. TV device implementers MAY report a hardware component as absent if they choose not to support the full API. Partial API implementations are not permitted.

3.2.1. Camera

A TV device implementation MAY omit camera devices.

If a TV device implementation reports that it includes a camera hardware feature, it MUST support at least one of the 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
<i>H.264</i>	MPEG-4 AVC video compression format

3.2.2. Tuner

A TV device implementation MAY omit tuner hardware. If an implementation reports that it includes tuner hardware, it MUST meet the broadcast standards (ex. ATSC/DVB/ISDB) of the target locations where it will be released. Such an implementation SHOULD meet the specific requirements for that tuner type.

3.2.3. HDMI Input

A TV device implementation MAY omit HDMI input hardware.

3.2.4. Wi-Fi

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

3.2.5. Audio Input Devices

A TV device implementation MAY omit a microphone. TV device implementations MUST accurately report the presence or absence of a microphone.

3.2.6. USB

TV device implementations SHOULD provide USB client functionality.

TV device implementations MUST accurately report the presence or absence of USB client functionality.

3.2.7. Graphics

A TV device implementation SHOULD provide 3D Graphics hardware acceleration. While it MAY be omitted, doing so will provide a degraded user experience.

TV device implementations MUST accurately report the presence or absence of acceleration hardware.

3.2.8. Picture-in-picture (PIP)

A TV device implementation MAY omit picture-in-picture (PIP) capability. If an implementation reports that it includes PIP capability, it MUST support the TV Window API.

3.2.9. TV Information

A TV device implementation MAY omit TV Information capability such as closed captioning. If an implementation reports that it includes TV Information capability, it MUST support the TV Information API.

3.2.10. Bluetooth

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

4. TV 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 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. 11].

4.2. Application Packaging

Applications MUST follow the packaging guidelines, as defined for the platform [ref. 17].

Web .wgt packages MUST NOT contain more than one UI application or the behavior is undefined.

4.3. Namespace

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

4.4. Application Features and Privileges

A Tizen application MUST declare the features and privileges that it uses in the configuration document included in the application package [ref. 17]. Further details on how to implement this requirement are provided in the developer documentation [ref. 14 and ref. 18].

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 this example:

```
<feature name="http://tizen.org/feature/tv.tuner"/>
<tizen:privilege name="http://tizen.org/privilege/tv.channel"/>
```

4.5. Profile Declaration

A Tizen application MUST declare the Tizen profile it is capable of running on. If this declaration is omitted, application stores MAY not correctly select the application for installation. For the Tizen TV profile, the following declaration style is used.

In the Tizen Web API configuration document (*config.xml*):

```
<widget xmlns="http://www.w3.org/ns/widgets"
  xmlns:tizen="http://tizen.org/ns/widgets" ...>
  <tizen:profile name="TV"/>
```

4.6. Web UI Framework

The Tizen 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. 3], jQuery Mobile version 1.2.0 [ref. 4], and Globalize version 0.1.0a2 [ref. 1]. The Web

UI framework is not a part of the required platform implementation. Tizen applications MAY use the Web UI framework, by including it with the application. This operation is supported by the Tizen SDK. Tizen applications MAY omit the Web UI framework and construct a UI using just W3C standard technologies, such as HTML/JavaScript/CSS.