IEC 62766-5-2

Edition 1.0   2017-07

INTERNATIONAL STANDARD

Consumer terminal function for access to IPTV and open internet multimedia services –
Part 5-2: Web standards TV profile

IEC 62766-5-2:2017-07(en)

About the IEC
The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications
The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue
The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub
The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished
Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

Electropedia - www.electropedia.org
The world's leading online dictionary of electronic and electrical terms containing 20 000 terms and definitions in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary
65 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc
If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

IEC 62766-5-2

INTERNATIONAL STANDARD

Consumer terminal function for access to IPTV and open internet multimedia services –
Part 5-2: Web standards TV profile

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

CONSUMER TERMINAL FUNCTION FOR ACCESS TO IPTV AND OPEN INTERNET MULTIMEDIA SERVICES –

Part 5-2: Web standards TV profile

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.

2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.

3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.

5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.

6) All users should ensure that they have the latest edition of this publication.

7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.

8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 62766 has been prepared by IEC technical committee 100: Audio, video and multimedia systems and equipment

The text of this International Standard is based on the following documents:

CDV	Report on voting
100/2549/CDV	100/2663/RVC

Full information on the voting for the approval of this International Standard can be found in the report on voting indicated in the above table.

This document has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts in the IEC 62766 series, published under the general title *Consumer terminal function for access to IPTV and open Internet multimedia services,* can be found on the IEC website.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under "http://webstore.iec.ch" in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

A bilingual version of this publication may be issued at a later date.

INTRODUCTION

The IEC 62766 series is based on a series of specifications that was originally developed by the OPEN IPTV FORUM (OIPF). They specify the user-to-network interface (UNI) for consumer terminals to access IPTV and open internet multimedia services over managed or non-managed networks as defined by OIPF.

IEC 62766-5-2 will be updated over time, as maturity of different standards (now works in progress) increase and/or new web standards are defined.

This document is organized as follows: the main body includes a list of references to specifications upon which the defined profile relies and that are considered necessary to enable an enhanced user experience. Annex A contains instead a detailed list of which features for each specification are considered stable enough and therefore can be safely supported by terminals and used by application developers. Such tables will be updated in future revisions of this document as maturity of the various specifications evolve. Support tables are omitted for those specifications that are required to be fully supported.

CONSUMER TERMINAL FUNCTION FOR ACCESS TO IPTV AND OPEN INTERNET MULTIMEDIA SERVICES –

Part 5-2: Web standards TV profile

1 Scope

This part of IEC 62766 specifies a profile of HTML5, CSS and other related web technologies for connected TVs. Its goal is to describe a common profile that can be relied on by content and service providers and implemented by manufacturers. It does not describe extensions or modification to any of the referenced technologies but only tries to define a subset of web standards that are suitable and useful for TV deployments and at the same time stable enough to provide a good degree of confidence that real interoperability can be achieved. It may add clarifications and/or additional constraints where these are needed due to the nature of the target deployment environment.

This part of IEC 62766 only describes a minimum subset of web technology that a terminal compliant with this profile is required to support. This does not preclude terminals to support more technologies than the ones described in this profile.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 62766-1, *Open IPTV Forum (OIPF) consumer terminal function and network interfaces for access to IPTV and open Internet multimedia services – Part 1: General*

IETF RFC 6265, *HTTP State Management Mechanism;* Adam Barth, April 2011

IETF RFC 6455, *The WebSocket Protocol;* I. Fette, A. Melnikov, December 2011 including verified errata

W3C Recommendation, *CSS 2.1 Cascading Style Sheets;* Bert Bos, Tantek Çelik, Ian Hickson, Håkon Wium Lie*,* 7 June 2011. Available from: http://www.w3.org/TR/2011/REC-CSS2-20110607/

W3C Recommendation, *CSS Color Module Level 3*; Tantek Çelik, Chris Lilley, L. David Baron, 7 June 2011. Available from http://www.w3.org/TR/2011/REC-css3-color-20110607/

W3C Recommendation, *Media Fragments URI 1.0 (basic)*; Raphaël Troncy et al, 25 September 2012. Available from http://www.w3.org/TR/2012/REC-media-frags-20120925/

W3C Recommendation, *Media Queries*; Håkon Wium Lie; Tantek Çelik; Daniel Glazman; Anne van Kesteren, 19 June 2012. Available from http://www.w3.org/TR/2012/REC-css3-mediaqueries-20120619/

W3C Recommendation, *Selectors API Level 3*; Daniel Glazman et al., 29 September 2011. Available from http://www.w3.org/TR/2011/REC-css3-selectors-20110929/

W3C Recommendation, *Web Storage*; Ian Hickson, 30 July 2013. Available from http://www.w3.org/TR/2013/REC-webstorage-20130730/

W3C Recommendation, *WOFF File Format 1.0*; Jonathan Kew, Tal Leming, Erik van Blokland, 13 December 2012. Available from http://www.w3.org/TR/2012/REC-WOFF-20121213/

W3C, *ECMAScript Language Specification, Edition 5.1*, June 2011. Available from http://www.ecma-international.org/publications/standards/Ecma-262.htm

W3C Candidate Recommendation, *CSS Backgrounds and Borders Module Level 3*; Bert Bos, Elika J. Etemad, Brad Kemper, 24 July 2012. Available from http://www.w3.org/TR/2012/CR-css3-background-20120724/

W3C Candidate Recommendation, *CSS Image Values and Replaced Content*; Elika J. Etemad, Tab Atkins Jr., 17 April 2012. Available from http://www.w3.org/TR/2012/CR-css3-images-20120417

W3C Candidate Recommendation, *CSS Flexible Box Layout Module*; Tab Atkins Jr., Elika J. Etemad, Alex Mogilevsky, 18 September 2012. Available from http://www.w3.org/TR/2012/CR-css3-flexbox-20120918/

W3C Candidate Recommendation, *CSS3 Module: Multi-column layout*; Håkon Wium Lie, 12 April 2011. Available from http://www.w3.org/TR/2011/CR-css3-multicol-20110412/

W3C Candidate Recommendation, *HTML Canvas 2D Context*; Rik Cabanier, Eliot Graff, Jay Munro, Tom Wiltzius, Ian Hickson, 17 December 2012. Available from http://www.w3.org/TR/2012/CR-2dcontext-20121217

W3C Candidate Recommendation, *HTML5*; Robin Berjon, Steve Faulkner, Travis Leithead, Erika Doyle Navara, Edward O'Connor, Silvia Pfeiffer, 6 August 2013. Available from http://www.w3.org/TR/2013/CR-html5-20130806/

W3C Candidate Recommendation, *HTML5 Web Messaging*; Ian Hickson, 01 May 2012. Available from http://www.w3.org/TR/2012/CR-webmessaging-20120501

W3C Candidate Recommendation, *Server-Sent Events*; Ian Hickson, 11 December 2012. Available from http://www.w3.org/TR/2012/CR-eventsource-20121211/

W3C Candidate Recommendation, *Web Workers*; Ian Hickson, 01 May 2012. Available from http://www.w3.org/TR/2012/CR-workers-20120501

W3C Candidate Recommendation, *The WebSocket API*; I. Hickson, 20 September 2012. Available from http://www.w3.org/TR/2012/CR-websockets-20120920/

W3C Working Draft, *CSS3 Basic User Interface Module*; Tantek Çelik, 17 January 2012. Available from http://www.w3.org/TR/2012/WD-css3-ui-20120117/

W3C Working Draft, *CSS Animations*; Dean Jackson, David Hyatt, Chris Marrin, Sylvain Galineau, L. David Baron, 19 February 2013. Available from http://www.w3.org/TR/2013/WD-css3-animations-20130219/

W3C Working Draft, *CSS Fonts Module Level 3*; John Daggett, 12 February 2013. Available from http://www.w3.org/TR/2013/WD-css3-fonts-20130212/

W3C Working Draft, *CSS Text Module Level 3*; Elika J. Etemad, Koji Ishii, 13 November 2012. Available from http://www.w3.org/TR/2012/WD-css3-text-20121113/

W3C Working Draft, *CSS Transforms*; Simon Fraser, Dean Jackson, David Hyatt, Chris Marrin, Edward O'Connor, Dirk Schulze, Aryeh Gregor, 11 September 2012. Available from http://www.w3.org/TR/2012/WD-css3-transforms-20120911

W3C Working Draft, *CSS Transitions*; Dean Jackson, David Hyatt, Chris Marrin, L. David Baron, 12 February 2013. Available from http://www.w3.org/TR/2013/WD-css3-transitions-20130212/

W3C Working Draft, *CSSOM View Module*; Anne van Kesteren, 4 August 2011. Available from http://www.w3.org/TR/2011/WD-cssom-view-20110804

W3C Working Draft, *Document Object Model (DOM) Level 3 Events*; Travis Leithead, Jacob Rossi et al, 6 September 2012. Available from http://www.w3.org/TR/2012/WD-DOM-Level-3-Events-20120906/

W3C Working Draft, *XMLHttpRequest*; Julian Aubourg et al, 6 December 2012. Available from http://www.w3.org/TR/2012/WD-XMLHttpRequest-20121206/

3   Terms, definitions and abbreviated terms

3.1   Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62766-1 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

* IEC Electropedia: available at http://www.electropedia.org/
* ISO Online browsing platform: available at http://www.iso.org/obp

3.1.1
terminal
device running an interactive user-agent (browser) conformant to this document

Note 1 to entry: This is equivalent to the the OITF block as defined in Annex B of IEC 62766-1:2017.

3.1.2
application
interactive software authored using the set of languages defined by this profile and conformant with this profile

3.1.3
authoring tool
software program used to edit applications

3.2 Abbreviated terms

For the purposes of this document, the abbreviated terms given in IEC 62766-1, as well as the following apply.

API   Application Programming Interface

DOM   Document Object Model

CSS   Cascading Style Sheets

HTML   Hypertext Markup Language

URI   Uniform Resource Identifier

XHTML   Extensible HTML

WOFF   Web Open Font Format

4 Overview

4.1 General

This document is aimed at TV terminals that implement a browser-based application environment. Applications running in such environment are authored using a set of languages commonly referred to as "web technologies" or "web standards". This document lists the minimum set of languages that shall be supported by a terminal conforming to this specification. A terminal may support more languages than the ones listed in this document.

To avoid fragmentation and enhance interoperability with other web technologies-based devices and eco-systems, this document tries not to diverge from any of the referenced specifications it relies on. In some exceptional cases though, this document may decide to intentionally diverge from the referenced specifications. Such differences will be noted explicitly throughout this document.

This document references some specifications that are still under development. All features marked as "at risk" in such referenced specifications shall be considered as optional to support, unless this profile explicitly mandates support for them.

Normative clauses of the present document are marked as such. All authoring guidelines, diagrams and examples in the present document are informative.

4.2 Markup

The W3C Candidate Recommendation, HTML5 specification defines conformance requirements for user-agents and documents. Applications and authoring tools shall comply with conformance requirements for documents unless specified otherwise in Clause A.2. Terminals shall comply with conformance requirements for user agents unless specified otherwise in Clause A.2. In particular, a terminal shall support the HTML syntax and the XHTML syntax for HTML documents as defined in the W3C Candidate Recommendation, HTML5.

5 Style

5.1 General

Support for CSS as a whole is not required by HTML5, even though some features are defined in terms of specific CSS requirements. This clause defines requirements for CSS and other style and/or graphic related technologies.

5.2 Basic graphic

Terminals shall support W3C Working Draft, CSS3 Basic User Interface Module, as profiled in A.3.2.

Terminals shall support CSS 2.1 W3C Recommendation, Cascading Style Sheets, level 2 (CSS2) Specification. Although the CSS 2.1 specification includes a "tv" media type, this has not been widely used in practice. Terminals may ignore the "tv" media type and just use the "screen" media type.

Terminals shall support W3C Recommendation, CSS Color Module Level 3.

Terminals shall support W3C Candidate Recommendation, CSS Image Values and Replaced Content, as profiled in A.3.3.

Terminals shall support W3C Candidate Recommendation, CSS Backgrounds and Borders Module Level 3, as profiled in A.3.4.

5.3 Device adaptation, layout and processing

Terminals shall support W3C Recommendation, Selectors API Level 3.

Terminals shall support W3C Recommendation, Media Queries.

Terminals shall support W3C Candidate Recommendation, CSS3 module: Multi-column layout.

Terminals shall support W3C Candidate Recommendation, CSS Flexible Box Layout Module.

Terminals should support W3C Candidate Recommendation, CSS Conditional Rules Module Level 3.

5.4 Text and fonts

Terminals shall support W3C Working Draft, CSS Fonts Module Level 3, as profiled in A.3.5.

Terminals shall support the Web Open Font Format (WOFF) W3C Recommendation, WOFF File Format 1.0. Applications can link to WOFF fonts via a `@font-face` rule (W3C Working Draft, CSS Fonts Module Level 3).

NOTE WOFF packaged fonts may require a significant amount of space. See Clause 9 for some recommendations.

Terminals shall support W3C Working Draft, CSS Text Module Level 3, as profiled in A.3.6.

5.5 Advanced graphic

Terminals shall support W3C Working Draft, CSS Transforms, as profiled in A.3.7.

Terminals shall support W3C Working Draft, CSS Transitions, as profiled in A.3.8.

Terminals shall support CSS Animations, as specified in W3C Working Draft, CSS Animations.

Terminals shall support the W3C Candidate Recommendation HTML Canvas 2D Context as profiled in A.4.3.

6   Scripting

6.1   General

Scripts are small programs that can be embedded into applications. While defining features that rely on scripting, HTML5 does not mandate support for scripting for all user-agents. Furthermore scripting is defined using a syntax that in most cases is independent from the underlying scripting language. For these reasons, this document has additional requirements as defined in this clause.

6.2   ECMAScript

Terminals shall support scripting as defined in W3C Candidate Recommendation, HTML5.

Terminals shall support the W3C ECMAScript Language Specification, edition 5.1 scripting language.

6.3   Event model

In addition to the support for the DOM 3 events specification that is required by W3C Candidate Recommendation, HTML5, terminals shall support the focus and keyboard event types defined in 5.2.2 and 5.2.5 respectively of W3C Working Draft, Document Object Model (DOM) Level 3 Events.

NOTE 1   W3C Working Draft, XMLHttpRequest (referenced in this document) also requires support for a subset of event related functionality defined in W3C Working Draft, such as various exceptions and EventTarget.

In order to support legacy content, terminals shall support also the legacy attributes `keyCode` and `charCode` as defined in Appendix B of W3C Working Draft, Document Object Model (DOM) Level 3 Events.

Applications shall not rely on these attributes and shall use the new event model as defined in DOM3 Events. Values used for the `keyCode` property are implementation-specific, but shall be exposed to the application through the constants below, defined on the `KeyboardEvent` interface. For each constant, an equivalent key value is given (as defined for `KeyboardEvent.key` in DOM3, see 6.2.7.1 of W3C Working Draft, Document Object Model (DOM) Level 3 Events).

In order to support legacy content, these constants shall also be available via a `KeyEvent` interface. For example VK_OK can be accessed as `KeyEvent`.VK_OK.

Table 1 shows the values of virtual keycodes.

NOTE 2   There is no requirement for terminals to be able to generate all keycodes defined in Table 1.

Table 1 – Virtual keycode values

Virtual keycode constants	Equivelant key value (in DOM3)
VK_UNDEFINED	'Unidentified'
VK_CANCEL	'Cancel'
VK_BACK_SPACE	'Backspace'
VK_TAB	'Tab'
VK_CLEAR	'Clear'
VK_ENTER	'Enter'
VK_SHIFT	'Shift'
VK_CONTROL	'Control'
VK_ALT	'Alt'
VK_PAUSE	'Pause'
VK_CAPS_LOCK	'CapsLock'
VK_KANA	'KanaMode'
VK_FINAL	'FinalMode'
VK_KANJI	'KanjiMode'
VK_ESCAPE	'Esc'
VK_CONVERT	'Convert'
VK_NONCONVERT	'Nonconvert'
VK_ACCEPT	'Accept'
VK_MODECHANGE	'ModeChange'
VK_SPACE	space character, fix
VK_PAGE_UP	'PageUp'
VK_PAGE_DOWN	'PageDown'
VK_END	'End'
VK_HOME	'Home'
VK_LEFT	'Left'
VK_UP	'Up'
VK_RIGHT	'Right'
VK_DOWN	'Down'
VK_COMMA	','
VK_PERIOD	'.'
VK_SLASH	'/'
VK_0	'0'
VK_1	'1'
VK_2	'2'
VK_3	'3'
VK_4	'4'
VK_5	'5'
VK_6	'6'
VK_7	'7'
VK_8	'8'
VK_9	'9'
VK_SEMICOLON	';'
VK_EQUALS	'Equals'

Virtual keycode constants	Equivelant key value (in DOM3)
VK_A	'A'
VK_B	'B'
VK_C	'C'
VK_D	'D'
VK_E	'E'
VK_F	'F'
VK_G	'G'
VK_H	'H'
VK_I	'I'
VK_J	'J'
VK_K	'K'
VK_L	'L'
VK_M	'M'
VK_N	'N'
VK_O	'O'
VK_P	'P'
VK_Q	'Q'
VK_R	'R'
VK_S	'S'
VK_T	'T'
VK_U	'U'
VK_V	'V'
VK_W	'W'
VK_X	'X'
VK_Y	'Y'
VK_Z	'Z'
VK_OPEN_BRACKET	'['
VK_BACK_SLASH	'\'
VK_CLOSE_BRACKET	']'
VK_NUMPAD0	See note below
VK_NUMPAD1	See note below
VK_NUMPAD2	See note below
VK_NUMPAD3	See note below
VK_NUMPAD4	See note below
VK_NUMPAD5	See note below
VK_NUMPAD6	See note below
VK_NUMPAD7	See note below
VK_NUMPAD8	See note below
VK_NUMPAD9	See note below
VK_MULTIPLY	'Multiply'
VK_ADD	'Add'
VK_SEPARATER	'Separator'
VK_SUBTRACT	'Subtract'
VK_DECIMAL	'Decimal'

Virtual keycode constants	Equivelant key value (in DOM3)
VK_DIVIDE	'Divide'
VK_F1	'F1'
VK_F2	'F2'
VK_F3	'F3'
VK_F4	'F4'
VK_F5	'F5'
VK_F6	'F6'
VK_F7	'F7'
VK_F8	'F8'
VK_F9	'F9'
VK_F10	'F10'
VK_F11	'F11'
VK_F12	'F12'
VK_DELETE	'Del'
VK_NUM_LOCK	'NumLock'
VK_SCROLL_LOCK	'Scroll'
VK_PRINTSCREEN	'PrintScreen'
VK_INSERT	'Insert'
VK_HELP	'Help'
VK_META	'Meta'
VK_BACK_QUOTE	'`'
VK_QUOTE	'''
VK_RED	'Red'
VK_GREEN	'Green'
VK_YELLOW	'Yellow'
VK_BLUE	'Blue'
VK_GREY	'Grey'
VK_BROWN	'Brown'
VK_POWER	'Power'
VK_DIMMER	'Dimmer'
VK_WINK	'Wink'
VK_REWIND	'MediaRewind'
VK_STOP	'MediaStop'
VK_EJECT_TOGGLE	'Eject'
VK_PLAY	'MediaPlay'
VK_RECORD	'MediaRecord'
VK_FAST_FWD	'FastFwd'
VK_PLAY_SPEED_UP	'PlaySpeedUp'
VK_PLAY_SPEED_DOWN	'PlaySpeedDown'
VK_PLAY_SPEED_RESET	'PlaySpeedReset'
VK_RECORD_SPEED_NEXT	'RecordSpeedNext'
VK_GO_TO_START	'MediaTrackStart'
VK_GO_TO_END	'MediaTrackEnd'
VK_TRACK_PREV	'MediaPreviousTrack'

Virtual keycode constants	Equivelant key value (in DOM3)
VK_TRACK_NEXT	'MediaNextTrack'
VK_RANDOM_TOGGLE	'RandomToggle'
VK_CHANNEL_UP	'ChannelUp'
VK_CHANNEL_DOWN	'ChannelDown'
VK_STORE_FAVORITE_0	'StoreFavorite0'
VK_STORE_FAVORITE_1	'StoreFavorite1'
VK_STORE_FAVORITE_2	'StoreFavorite2'
VK_STORE_FAVORITE_3	'StoreFavorite3'
VK_RECALL_FAVORITE_0	'RecallFavorite0'
VK_RECALL_FAVORITE_1	'RecallFavorite1'
VK_RECALL_FAVORITE_2	'RecallFavorite2'
VK_RECALL_FAVORITE_3	'RecallFavorite3'
VK_CLEAR_FAVORITE_0	'ClearFavorite0'
VK_CLEAR_FAVORITE_1	'ClearFavorite1'
VK_CLEAR_FAVORITE_2	'ClearFavorite2'
VK_CLEAR_FAVORITE_3	'ClearFavorite3'
VK_SCAN_CHANNELS_TOGGLE	'ScanChannelsToggle'
VK_PINP_TOGGLE	'PinPToggle'
VK_SPLIT_SCREEN_TOGGLE	'SplitScreenToggle'
VK_DISPLAY_SWAP	'DisplaySwap'
VK_SCREEN_MODE_NEXT	'ScreenModeNext'
VK_VIDEO_MODE_NEXT	'VideoModeNext'
VK_VOLUME_UP	'VolumeUp'
VK_VOLUME_DOWN	'VolumeDown'
VK_MUTE	'VolumeMute'
VK_SURROUND_MODE_NEXT	'AudioSurroundModeNext'
VK_BALANCE_RIGHT	'AudioBalanceRight'
VK_BALANCE_LEFT	'AudioBalanceLeft'
VK_FADER_FRONT	'AudioFaderFront'
VK_FADER_REAR	'AudioFaderRear'
VK_BASS_BOOST_UP	'AudioBassBoostUp'
VK_BASS_BOOST_DOWN	'AudioBassBoostDown'
VK_INFO	'Info'
VK_GUIDE	'Guide'
VK_TELETEXT	'Teletext'
VK_SUBTITLE	'Subtitle'
VK_BACK	'BrowserBack'
VK_MENU	'Menu'
VK_PLAY_PAUSE	'MediaPlayPause'

NOTE 3   Numpad values are not distinguished from other numerical key values in this set; a content author could use the KeyboardEvent.location attribute to discover if a key originated from the numeric keypad.

6.4   CSSOM view

Terminals shall support `innerWidth` and `innerHeight` properties of the `Window` interface as defined in W3C Working Draft, CSSOM View Module. Terminals should support other features defined in W3C Working Draft, CSSOM View Module.

NOTE Many of the features included in W3C Working Draft, CSSOM View Module, have been supported in browsers for a long time. For example, the `Screen` interface or extensions to the `Window` interface to obtain the width of the viewport through script. Future versions of this profile may mandate support for this standard, once it reaches a higher level of maturity.

7   Application APIs

A number of specifications from W3C or other organizations define APIs that provide additional functionalities to applications. This clause lists which specifications shall be supported by a terminal. Note that the APIs defined as part of the HTML5 specification are not included in this clause but can be found in A.2.3.

Terminals shall support the XMLHttpRequest API, W3C Working Draft, XMLHttpRequest, as profiled in A.4.1.

Terminals shall support the Web Messaging API W3C Candidate Recommendation, HTML5 Web Messaging.

Terminals shall support W3C Candidate Recommendation, The WebSocket API with the protocol defined in IETF RFC 6455.

Terminals shall support the Web Workers API, W3C Candidate Recommendation, Web Workers, as profiled in A.4.2.

Terminals shall support the Server-Sent Events API, W3C Candidate Recommendation, Server-Sent Events.

Terminals shall support the Web Storage API, W3C Recommendation, Web Storage.

8   Media formats and protocols
Media fragment URI

Whenever an application points to a media resource through a URI (e.g. via the `src` attribute of the `<video>` element), it may include a Fragment Identifier (see IETF RFC 3986). The semantics of such fragments for resources of type audio/*, image/* and video/* are defined in W3C Recommendation, Media Fragments URI 1.0 (basic).

Terminals shall support temporal clipping based on Normal Play Time as defined in 4.2.1 of W3C Recommendation, Media Fragments URI 1.0 (basic). There is no requirement to support other means of specifying temporal clipping mentioned in W3C Recommendation, Media Fragments URI 1.0 (basic), such as SMPTE timecodes or real-world clock time, or to support the URI query mechanism.

9   Memory usage

NOTE This clause is for information only.

This clause provides some guidelines about the recommended minimum memory requirements of some of the specifications referenced by this document. Table 2 lists the minimum memory requirements.

Table 2 – Minimum memory requirements

Feature	Memory requirement	Comment
Downloadable fonts, W3C Recommendation, WOFF File Format 1.0	1 MB for Western Fonts. 10 MB for Asian fonts.	It is recommended to cache WOFF packages for for a better user experience.
W3C Recommendation, Web Storage	1 MB per origin. 10 MB global.	Terminals may provide methods to free-up memory as described in W3C Recommendation, Web Storage If the global limit is exceeded, a terminal may not be able to guarantee the per-origin storage limit.
Cookies (IETF RFC 6265)	At least 4 096 B per cookie (as measured by the sum of the length of the cookie's name, value, and attributes). At least 20 cookies per domain. At least 100 cookies total. At least 5 120 B for the "Set-Cookie" header.	

Annex A
(normative)

Support tables

A.1   General

This annex includes support tables for various specifications referenced by this profile. Each feature is given a short description and hyperlinked to its place within the related specification.

Terminals shall support all features marked as "Yes" in the "Mandatory" column; if restrictions apply, the feature support is indicated as "Partial" and the "Comment" column provides a description of the normative requirements. Support for all features marked as "No" is optional.

Specifications that are required to be fully supported do not have a related support table in this annex.

A.2   HTML5 profile

A.2.1   Elements

Table A.1 lists HTML elements that are either new HTML5 elements or elements already defined in HTML4 and now re-defined in HTML5. Some of the HTML4 elements have slightly modified meanings in HTML5 to better reflect how they are used on the Web or to make them more useful.

Table A.1 – HTML elements

Element name	Description	Mandatory	Comment
<body>	The body element of a document is the first child of the html element that is either a body element or a frameset element. If there is no such element, it is null.	Yes	
<base>	The base element allows authors to specify the document base URL for the purposes of resolving relative URLs, and the name of the default browsing context for the purposes of the following hyperlinks. The element does not represent any content beyond this information.	Yes	
<meta>	The meta element represents various kinds of metadata that cannot be expressed using the title, base, link, style, and script elements.	Yes	
<style>	The style element allows authors to embed style information in their documents.	Partial	Support for the scoped attribute is optional.
**
**	The br element represents a line break.	Yes	
<div>	The div element has no special meaning at all. It represents its children. It can be used with the class, lang, and title attributes to mark up semantics common to a group of consecutive elements.	Yes	
<h1> <h2> <h3> <h4> <h5> <h6>	These elements represent headings for their sections. They have a rank given by the number in their name. The h1 element is said to have the highest rank, the h6 element has the lowest rank, and two elements with the same name have equal rank.	Yes	
<html>	The html element represents the root of an HTML document.	Yes	
<iframe>	The iframe element represents a nested browsing context.	Partial	Support for the sandbox, srcdoc and seamless attributes is optional.
<img>	An img element represents an image.	Yes	
<li>	The li element represents a list item.	Yes	
<link>	The link element allows authors to link their document to other resources.	Partial	Support for the sizes attribute is optional. Support for the IDL attribute relList is optional
<ol>	The ol element represents a list of items, where the items have been intentionally ordered, such that changing the order would change the meaning of the document.	Yes	
<option>	The option element represents an option in a select element or as part of a list of suggestions in a datalist element.	Yes	
<p>	The p element represents a paragraph.	Yes	
<pre>	The pre element represents a block of preformatted text, in which structure is represented by typographic conventions rather than by elements.	Yes	
<dl>	The dl element represents an association list consisting of zero or more name-value groups (a description list).	Yes	

Element name	Description	Mandatory	Comment
\<dt>	The dt element represents the term, or name, part of a term-description group in a description list (dl element).	Yes	
\<dd>	The dd element represents the description, definition, or value, part of a term-description group in a description list (dl element).	Yes	
\<em>	The em element represents stress emphasis of its contents.	Yes	
\<s>	The s element represents contents that are no longer accurate or no longer relevant.	Yes	
\<q>	The q element represents some phrasing content quoted from another source.	Yes	
\<dfn>	The dfn element represents the defining instance of a term.	Yes	
\<abbr>	The abbr element represents an abbreviation or acronym, optionally with its expansion.	Yes	
\<code>	The code element represents a fragment of computer code.	Yes	
\<samp>	The samp element represents (sample) output from a program or computing system.	Yes	
\<var>	The var element represents a variable.	Yes	
\<kbd>	The kbd element represents user input (typically keyboard input, although it may also be used to represent other input, such as voice commands).	Yes	
\<sup>,\<sub>	The sup element represents a superscript and the sub element represents a subscript.	Yes	
\<u>	The u element represents a span of text with an unarticulated, though explicitly rendered, non-textual annotation, such as labeling the text as being a proper name in Chinese text (a Chinese proper name mark), or labeling the text as being misspelt.	Yes	
\<bdi>	The bdi element represents a span of text that is to be isolated from its surroundings for the purposes of bidirectional text formatting.	No	
\<bdo>	The bdo element represents explicit text directionality formatting control for its children. It allows authors to override the Unicode bidirectional algorithm by explicitly specifying a direction override.	Yes	
\<ins>	The ins element represents an addition to the document.	Yes	
\<del>	The del element represents a removal from the document.	Yes	
\<object>	The object element can represent an external resource, which, depending on the type of the resource, will either be treated as an image, as a nested browsing context, or as an external resource to be processed by a plugin.	Yes	
\<param>	The param element defines parameters for plugins invoked by object elements. It does not represent anything on its own.	Yes	

Element name	Description	Mandatory	Comment
`<map>`	The `map` element, in conjunction with any area element descendants, defines an image map. The element represents its children.	Yes	
`<area>`	The `area` element represents either a hyperlink with some text and a corresponding area on an image map, or a dead area on an image map.	Partial	Support for the IDL attribute `relList` is optional.
`<caption>`	The `caption` element represents the title of the `table` that is its parent, if it has a parent and that is a `table` element.	Yes	
`<table>`	The `table` element represents data with more than one dimension, in the form of a table.	Partial	Support for the `summary` attribute is optional.
`<colgroup>`	The `colgroup` element represents a group of one or more columns in the table that is its parent, if it has a parent and that is a table element.	Yes	
`<col>`	If a `col` element has a parent and that is a `colgroup` element that itself has a parent that is a `table` element, then the `col` element represents one or more columns in the column group represented by that `colgroup`.	Yes	
`<tbody>`	The `tbody` element represents a block of rows that consist of a body of data for the parent `table` element, if the `tbody` element has a parent and it is a `table`.	Yes	
`<thead>`	The `thead` element represents the block of rows that consist of the column labels (headers) for the parent `table` element, if the `thead` element has a parent and it is a `table`.	Yes	
`<tfoot>`	The `tfoot` element represents the block of rows that consist of the column summaries (footers) for the parent `table` element, if the `tfoot` element has a parent and it is a `table`.	Yes	
`<tr>`	The `tr` element represents a row of cells in a `table`.	Yes	
`<td>`	The `td` element represents a data cell in a `table`.	Yes	
`<th>`	The `th` element represents a header cell in a `table`.	Yes	
`<form>`	The `form` element represents a collection of form-associated elements, some of which can represent editable values that can be submitted to a server for processing.	Yes	
`<fieldset>`	The `fieldset` element represents a set of form controls optionally grouped under a common name.	Partial	Support for the `name` and `type` attributes is optional.
`<legend>`	The `legend` element represents a caption for the rest of the contents of the legend element's parent `fieldset` element, if any.	Yes	
`<button>`	The `button` element represents a button.	Yes	
`<optgroup>`	The `optgroup` element represents a group of option elements with a common label.	Yes	

Element name	Description	Mandatory	Comment
\<script\>	The script element allows authors to include dynamic script and data blocks in their documents. The element does not represent content for the user.	Partial	Support for the async attribute is optional.
\<noscript\>	The noscript element represents nothing if scripting is enabled, and represents its children if scripting is disabled. It is used to present different markup to user agents that support scripting and those that do not support scripting, by affecting how the document is parsed.	No	Since support for scripting is mandated by this profile, support for this element is not needed.
\<span\>	The span element does not mean anything on its own, but can be useful when used together with the global attributes, e.g. class, lang, or dir. It represents its children.	Yes	
\<title\>	The title element represents the document's title or name.	Yes	
\<ul\>	The ul element represents a list of items, where the order of the items is not important — that is, where changing the order would not materially change the meaning of the document.	Yes	
\<section\>	Represents a generic document or application section. It can be used together with the h1, h2, h3, h4, h5, and h6 elements to indicate the document structure.	Yes	
\<article\>	Represents an independent piece of content of a document, such as a blog entry or newspaper article.	Yes	
\<aside\>	Represents a piece of content that is only slightly related to the rest of the page.	Yes	
\<hgroup\>	Represents the header of a section.	Yes	
\<header\>	Represents a group of introductory or navigational aids.	Yes	
\<footer\>	Represents a footer for a section and can contain information about the author, copyright information, etc.	Yes	
\<nav\>	Represents a section of the document intended for navigation.	Yes	
\<figure\>	Can be used to associate a caption together with some embedded content, such as a graphic or video:	Yes	
\<figcaption\>	Provides the caption for the \<figure\> element's contents.	Yes	
\<video\>	Represents a video or movie. It is a media element whose media data is ostensibly video data, possibly with associated audio data.	Partial	Support for crossorigin, mediagroup, and controls content attributes is optional.
\<audio\>	Represents a sound or audio stream. It is a media element whose media data is ostensibly audio data.	Partial	Support for crossorigin, mediagroup, and controls content attributes is optional.
\<track\>	The track element allows authors to specify explicit external timed text tracks for media elements. It does not represent anything on its own.	No	Not required as only support for in-band tracks is required.
\<source\>	Allows authors to specify multiple media resources for media elements. It does not represent anything on its own.	Yes	

Element name	Description	Mandatory	Comment
`<embed>`	It is is used for plug-in content. It represents an integration point for an external (typically non-HTML) applications or interactive content.	Yes	
`<mark>`	Represents a run of text in one document marked or highlighted for reference purposes, due to its relevance in another context.	Yes	
`<progress>`	Represents a completion of a task such as downloading, or when performing a series of expensive operations.	Yes	
`<meter>`	Represents a scalar measurement within a known range, or a fractional value; for example: disk usage the relevance of a query result the fraction of a voting population to have selected a particular candidate	Yes	
`<time>`	Represents either a time on a 24 h clock, or a precise date in the proleptic Gregorian calendar, optionally with a time and a time zone. It provides an API for accessing the date/time as a Date object.	No	
`<ruby>`	Allows one or more spans of phrasing content to be marked with ruby annotations. Ruby annotations are short runs of text presented alongside base text, primarily used in East Asian typography as a guide for pronunciation or to include other annotations. In Japanese, this form of typography is also known as furigana.	No	
`<rt>`	Marks the ruby text component of a ruby annotation.	No	
`<rp>`	Can be used to provide parentheses around a ruby text component of a ruby annotation, to be shown by user agents that do not support ruby annotations.	No	
`<wbr>`	Represents a line-break opportunity where phrasing content is expected. For example, someone is quoted as saying something which, for effect, is written as one long word. However, to ensure that the text can be wrapped in a readable fashion, the individual words in the quote are separated using a `<wbr>` element.	Yes	
`<canvas>`	Represents a resolution-dependent bitmap canvas, which can be used for rendering graphs, game graphics, or other visual images on the fly.	Partial	For details on support of Canvas APIs, see A.4.3
`<command>`	Represents a command that the user can invoke.	No	
`<details>`	Represents additional information or controls which the user can obtain on demand.	No	
`<summary>`	Represents a summary, caption, or legend for the rest of the contents of the summary element's parent details element, if any.	No	

Element name	Description	Mandatory	Comment
`<datalist>`	Represents a set of `<option>` elements that represent predefined options for other controls. The contents of the element represents fallback content for legacy user agents, intermixed with `<option>` elements that represent the predefined options. In the rendering, the `<datalist>` element represents nothing and it, along with its children, should be hidden. The `<datalist>` element together with the new `list` attribute for input can be used to make comboboxes.	Yes	
`<keygen>`	Represents a key pair generator control. When the control's form is submitted, the private key is stored in the local keystore, and the public key is packaged and sent to the server.	Yes	
`<output>`	Represents some type of output, as from a calculation done through scripting.	Yes	
`<textarea>`	Represents a multiline plain text edit control for the element's raw value.	Yes	
`<select>`	Represents a control for selecting amongst a set of options.	Yes	
`<input>`	The `<input>` element represents a typed data field, usually with a form control to allow the user to edit the data. The `<input>` element `type` attribute has the following values. "text" "url" "email" "datetime" "date" "month" "week" "time" "datetime-local" "number" "range" "tel" "search" "color" "checkbox" "image" "file"	Partial	The following values for the `type` attribute shall be supported: "text", "url", "email", "number", "range", "tel", "search", "checkbox", "date"; all other values are optional to support (note that some are marked as feature at risk in W3C Candidate Recommendation, HTML5). Support for the `autocomplete` attribute is optional.
`<a>`	The `<a>` element *without* an `href` attribute now represents a **"placeholder link"**. It can also contain flow content rather than being restricted to phrase content.	Partial	Support for the IDL attribute `relList` is optional.
`<address>`	The `<address>` element is now scoped by the new concept of sectioning. It represents the contact information for its nearest article or body element ancestor. If that is the body element, then the contact information applies to the document as a whole.	Yes	

Element name	Description	Mandatory	Comment
\<b>	The \<b> element represents a span of text to be stylistically offset from the normal prose without conveying any extra importance, such as key words in a document abstract, product names in a review, or other spans of text whose typical typographic presentation is boldened.	Yes	
\<cite>	The \<cite> element now solely represents the title of a work (e.g. a book, a paper, an essay, a poem, a score, a song, a script, a film, a TV show, a game, a sculpture, a painting, a theatre production, a play, an opera, a musical, an exhibition, a legal case report, etc). Specifically the example in HTML4 where it is used to mark up the name of a person is no longer considered conforming.	Yes	
\<hr>	The \<hr> element represents a paragraph-level thematic break, e.g. a scene change in a story, or a transition to another topic within a section of a reference book.	Yes	
\<i>	The \<i> element represents a span of text in an alternate voice or mood, or otherwise offset from the normal prose, such as a taxonomic designation, a technical term, an idiomatic phrase from another language, a thought, a ship name, or some other prose whose typical typographic presentation is italicized.	Yes	
\<label>	For the \<label> element the browser should no longer move focus from the label to the control unless such behavior is standard for the underlying platform user interface.	Yes	
\<menu>	The \<menu> element represents a list of commands and is redefined to be useful for toolbars and context menus.	Partial	\<menu> element of type list shall be supported. Support for other types is optional. Support for the label attribute is optional.
\<small>	The \<small> element now represents small print (for side comments and legal print).	Yes	
\<strong>	The \<strong> element now represents importance rather than strong emphasis.	Yes	
\<head>	The \<head> element no longer allows the object element as child.	Yes	

A.2.2   Global attributes

Table A.2 lists the global attributes.

Table A.2 – Global attributes

Attribute name	Description	Mandatory	Comment
accessKey	The accessKey attribute's value is used by the user agent as a guide for creating a keyboard shortcut that activates or focuses the element.	No	
class	Every HTML element may have a class attribute specified. The attribute, if specified, shall have a value that is an unordered set of unique space-separated tokens representing the various classes that the element belongs to. The classes that an HTML element has assigned to it consist of all the classes returned when the value of the class attribute is split on spaces.	Yes	
dir	The dir attribute specifies the element's text directionality (left-to-right, or right-to-left). The attribute is an enumerated attribute with the keyword ltr ((left-to-right) mapping to the state ltr, and the keyword rtl (right-to-left) mapping to the state rtl. The attribute has no defaults.	Yes	
id	The id attribute represents its element's unique identifier. The value shall be unique in the element's home subtree and shall contain at least one character. The value shall not contain any space characters.	Yes	
lang	The lang attribute (in no namespace) specifies the primary language for the element's contents and for any of the element's attributes that contain text. Its value shall be a valid BCP 47 language code, or the empty string.	Yes	
style	All HTML elements may have the style content attribute set. If specified, the attribute shall contain only a list of zero or more semicolon-separated (;) CSS declarations. In user agents that support CSS, the attribute's value shall be parsed when the attribute is added or has its value changed, with its value treated as the body (the part inside the curly brackets) of a declaration block in a rule whose selector matches just the element on which the attribute is set. All URLs in the value shall be resolved relative to the element when the attribute is parsed. For the purposes of the CSS cascade, the attribute shall be considered to be a style attribute at the author level. Documents that use style attributes on any of their elements shall still be comprehensible and usable if those attributes were removed.	Yes	

Attribute name	Description	Mandatory	Comment
`tabindex`	The `tabindex` content attribute specifies: If the element is focusable. If it can be reached using sequential focus navigation. The relative order of the element for the purposes of sequential focus navigation. The name "tab index" comes from the common use of the `tab` key to navigate through the focusable elements. The term "tabbing" refers to moving forward through the focusable elements that can be reached using sequential focus navigation. If it is specified, it shall have a value that is a valid integer, and it shall be parsed using the rules for parsing integers.	Yes	
`title`	The `title` attribute represents advisory information: For an element, such as would be appropriate for a tooltip. On a link, this could be the title or a description of the target resource. On an image, it could be the image credit or a description of the image. On a paragraph, it could be a footnote or commentary on the text. On a citation, it could be further information about the source. etc. The value is text.	Yes	
`contenteditable / isContentEditable`	The `contenteditable` attribute is an enumerated attribute whose keywords are the `empty string`, `true`, and `false`. The empty string and the true keyword map to the true state. The false keyword maps to the false state. In addition, there is a third state, the `inherit` state, which is the missing value default (and the invalid value default). The `true` state indicates that the element is editable. The `inherit` state indicates that the element is editable if its parent is. The `false` state indicates that the element is not editable.	No	
`designMode`	Documents have a designMode, which can be either enabled or disabled. When enabled, the document is editable.	No	
`contextmenu`	The `contextmenu` attribute gives the element's context menu. The value shall be the ID of a menu element in the DOM. If the node that would be obtained by the invoking the `getElementById()` method using the attribute's value as the only argument is `null` or not a menu element, then the element has no assigned context menu. Otherwise, the element's assigned context menu is the element so identified.	No	

Attribute name	Description	Mandatory	Comment
`data-*`	A custom `data` attribute is an attribute in a no namespace whose name starts with the string "data-", has at least one character after the hyphen, is XML-compatible, and contains no characters in the range U+0041 .. U+005A (LATIN CAPITAL LETTER A .. LATIN CAPITAL LETTER Z). Custom data attributes store custom data private to the page or application, for which there are no more appropriate attributes or elements. These attributes do not use software that is independent of the site that uses the attributes. Every HTML element may have any number of custom data attributes specified, with any value. They refer to the DOM Dataset API.	Yes	
`draggable`	All HTML elements may have the `draggable` content attribute set. It is an enumerated attribute, and has three states: The first state is true and it has the keyword `true`. The true state means the element is draggable. The second state is false and it has the keyword `false`. The false state means that it is not draggable. The third state is `auto`; it has no keywords but it is the missing value default. The auto state uses the default behavior of the user agent.	No	
`dropzone`	All HTML elements may have the dropzone content attribute set.	No	
`hidden`	All HTML elements may have the `hidden` content attribute set. The hidden attribute is a *boolean* attribute. When specified on an element, it indicates that the element is not yet, or is no longer, relevant. User agents should not render elements that have the hidden attribute specified.	Yes	
`spellcheck`	The `spellcheck` attribute is an enumerated attribute whose keywords are the `empty string`, `true` and `false`. The empty string and the `true` keyword map to the true state. The `false` keyword maps to the false state. Please note that the "default" state has been removed. The true state indicates that the element is to have its spelling and grammar checked. The false state indicates that the element is not to be checked.	No	

A.2.3   Web applications APIs

A.2.3.1   General

HTML5 introduces a number of APIs that help with creating applications, as shown in Table A.3.

Table A.3 – Web applications APIs

API name	Description	Mandatory	Comment
Media elements	An API for playing of video and audio which can be used with the new `<video>` and `<audio>` elements.	Partial	See profile in A.2.3.2
Application cache	An API that enables offline Web applications.	No	This feature is potentially useful but marked as feature at risk in W3C Candidate Recommendation, HTML5 and likely to change in the near future.
Custom scheme and content handlers	An API that allows a Web application to register itself for certain protocols or media types.	No	Not relevant for this profile. Also, marked as feature at risk in W3C Candidate Recommendation, HTML5.
Custom search provider	An API that allows a Web application to register itself for certain search provider.	No	Not relevant for this profile. Also, marked as feature at risk in W3C Candidate Recommendation, HTML5.
Drag and drop	This API works in combination with a `draggable` attribute.	No	
History	An API that exposes the history and allows pages to add to it to prevent breaking the back button.	Yes	
Base64 utility methods	An API that allow authors to transform content to and from the base64 encoding.	Yes	

A.2.3.2   Media APIs

HTML Media APIs shall be supported as profiled in Table A.4.

Table A.4 – Media APIs

Interface	Attribute name	Mandatory	Comment
HTMLVideoElement	width	Yes	
	height	Yes	
	videoWidth	Yes	
	videoHeight	Yes	
	poster	Yes	
HTMLAudioElement		Yes	
HTMLSourceElement	src	Yes	
	type	Yes	
	media	Yes	
HTMLTrackElement		No	
HTMLMediaElement	error	Yes	
	src	Yes	
	currentSrc	Yes	
	crossOrigin	No	
	networkState	Yes	
	preload	Yes	
	buffered	Yes	
	load()	Yes	
	canPlayType()	Yes	
	readyState	Yes	
	seeking	Yes	
	currentTime	Yes	
	duration	Yes	
	startDate	No	
	paused	Yes	
	defaultPlaybackRate	Yes	
	playbackRate	Yes	
	played	Yes	
	seekable	Yes	
	ended	Yes	
	autoplay	Yes	
	loop	Yes	
	play()	Yes	
	pause()	Yes	
	mediaGroup	No	
	controller	No	
	controls	No	
	volume	Yes	
	muted	Yes	
	defaultMuted	Yes	
	videoTracks	No	
	audioTracks	Yes	

Interface	Attribute name	Mandatory	Comment
	textTracks	Yes	Only support for in-band tracks is required
	addTextTrack()	No	
MediaError	code	Yes	
AudioTrackList	length	Yes	
	getter(index)	Yes	
	getTrackByID()	Yes	
	onchange	Yes	
	onaddtrack	No	
	onremovetrack	No	
AudioTrack	id	Yes	
	kind	Yes	
	label	Yes	
	language	Yes	
	enabled	Yes	
VideoTrackList		No	
VideoTrack		No	
TextTrackList		Yes	
TextTrack	Kind	Yes	
	Label	Yes	
	Language	Yes	
	onMetadataTrackDispatchType	Yes	
	Mode	Yes	
	Cues	No	
	activeCues	No	
	addCue	No	
	removeCue	No	
	Oncuechange	No	
TextTrackCueList		No	
TextTrackCue		No	
MediaController		No	
TimeRange	Length	Yes	
	Start	Yes	
	End	Yes	
TrackEvent		Yes	

A.2.3.3   Media element events support

Table A.5 lists the media element events.

Table A.5 – Media element events

Events	Mandatory
loadstart	Yes
progress	Yes
suspend	Yes
abort	Yes
error	Yes
emptied	Yes
stalled	Yes
loadedmetadata	Yes
loadeddata	Yes
canplay	Yes
canplaythrough	Yes
Playing	Yes
Waiting	Yes
Seeking	Yes
Seeked	Yes
Ended	Yes
durationchange	Yes
timeupdate	Yes
play	Yes
pause	Yes
ratechange	Yes
volumechange	Yes

A.3   CSS3 profile

A.3.1   General

This clause contains support tables for various CSS specifications.

A.3.2   CSS basic user interface

A.3.2.1   Properties and values

Table A.6 lists the CSS basic user interface properties and values.

Table A.6 – CSS basic user interface properties and values

Property	Values	Description	Mandatory	Comment
appearance	normal | <appearance> | inherit	This property can be used to make an element look like a standard user interface element on the platform. It is a shorthand for "appearance", "color", "font", and "cursor".	No	
box-sizing	padding-box | content-box | border-box | inherit	This property specifies a border/padding value in relation to a fluid length element.	Partial	Support for property value padding-box is optional (marked as feature at risk in W3C Candidate Recommendation, HTML5)
content	icon	This property is used with the:before and:after pseudo-elements to generate content in a document.	No	Marked as feature at risk in W3C Candidate Recommendation, HTML5
cursor	[[<uri> [<x> <y>]?,]* [auto | default | none | context-menu | help | pointer | progress | wait | cell | crosshair | text | vertical-text | alias | copy | move | no-drop | not-allowed | e-resize | n-resize | ne-resize | nw-resize | s-resize | se-resize | sw-resize | w-resize | ew-resize | ns-resize | nesw-resize | nwse-resize | col-resize | row-resize | all-scroll | zoom-in | zoom-out]] | inherit	This property specifies the type of cursor to be displayed for the pointing device.	Yes	
font	<appearance> | status-bar | message-box | caption | small-caption | inherit	The shorthand "font" property has the effect of setting all of the elemental "font-*" properties.	Yes	
icon	auto | <uri> [, <uri>]* | inherit	The "icon" property give the author the ability to style any arbitrary element with an iconic equivalent.	No	Feature at risk in W3C Candidate Recommendation, HTML5
nav-index	auto | <number> | inherit	This property is an input-method-neutral way of specifying the sequential navigation order, also known as "tabbing order".	Yes	
nav-up	auto | <id> [current | root | <target-name>]? | inherit	This property allows User agents for devices with directional navigation keys to respond by navigating the nav-up focus.	Yes	

Property	Values	Description	Mandatory	Comment
`nav-right`	auto | <id> [current | root | <target-name>]? | inherit	This property allows User agents for devices with directional navigation keys to respond by navigating the nav-right focus.	Yes	
`nav-down`	auto | <id> [current | root | <target-name>]? | inherit	This property allows User agents for devices with directional navigation keys to respond by navigating the nav-down focus.	Yes	
`nav-left`	auto | <id> [current | root | <target-name>]? | inherit	This property allows User agents for devices with directional navigation keys to respond by navigating the nav-left focus.	Yes	
`outline`	[<"outline-color"> || <"outline-style"> || <"outline-width">] | inherit	This property allows authors who want to create outlines around visual objects such as buttons, active form fields, image maps, etc., to make them stand out.	Yes	
`outline-color`	<color> | invert | inherit	This property controls the border color of a dynamic outline.	Yes	
`outline-offset`	<length> | inherit	This property controls the border offset of a dynamic outline.	Yes	
`outline-style`	auto | <border-style> | inherit	This property controls the border style of a dynamic outline.	Partial	At this time `outline-style` shall be supported as defined by CSS2.1 specification, so support for value `auto` is not required.
`outline-width`	<border-width> | inherit	This property controls the border width of a dynamic outline.	Yes	
`resize`	none | both | horizontal | vertical | inherit	This property allows an author to specify whether or not an element is resizable by the user, and if so, along which axis/axes.	No	
`ime-mode`	auto | normal | active | inactive | disabled | inherit	The 'ime-mode' CSS property controls the state of the input method editor for text fields.	No	Feature at risk in W3C Candidate Recommendation, HTML5

Property	Values	Description	Mandatory	Comment
`text-overflow`	(clip | ellipsis |){1,2} | inherit	This property specifies rendering when inline content overflows its block container element ("the block") in its inline progression direction that has 'overflow' other than 'visible'.	Partial	Support for the `text-overflow` property value <string> and the `text-overflow` property 2-value syntax and definition is optional (marked as feature at risk in W3C Working Draft, CSS3 Basic User Interface Module.

A.3.2.2   User interface pseudo classes

Table A.7 lists the user interface pseudo classes.

Table A.7 – User interface pseudo classes

Pseudo class	Mandatory	Comment
`:active`	Yes	
`:checked`	Yes	
`:default`	Yes	
`:disabled`	Yes	
`:enabled`	Yes	
`:focus`	Yes	
`:hover`	Yes	
`:in-range`	Yes	
`:indeterminate`	Yes	
`:invalid`	Yes	
`:optional`	Yes	
`:out-of-range`	Yes	
`:read-only`	Yes	
`:read-write`	Yes	
`:required`	Yes	
`:valid`	Yes	
`:visited`	Yes	

A.3.3   CSS image values and replaced content

Terminals shall support gradients as defined in W3C Candidate Recommendation, CSS Image Values and Replaced Content. Support for other features is optional.

A.3.4   CSS backgrounds and borders

Terminals shall support all features in W3C Candidate Recommendation, CSS Backgrounds and Borders Module Level 3, with the exception of the `border-image` set of properties.

A.3.5   CSS fonts module level 3

Terminals shall support the W3C Working Draft, CSS Fonts Module Level 3 specification, as profiled below: