International Telecommunication Union

ITU-R

Radiocommunication Sector of ITU

Recommendation ITU-R BT.1699-2

(01/2013)

Harmonization of declarative application formats for interactive TV

BT Series

Broadcasting service (television)

Foreword

The role of the Radiocommunication Sector is to ensure the rational, equitable, efficient and economical use of the radio-frequency spectrum by all radiocommunication services, including satellite services, and carry out studies without limit of frequency range on the basis of which Recommendations are adopted.

The regulatory and policy functions of the Radiocommunication Sector are performed by World and Regional Radiocommunication Conferences and Radiocommunication Assemblies supported by Study Groups.

Policy on Intellectual Property Right (IPR)

ITU-R policy on IPR is described in the Common Patent Policy for ITU-T/ITU-R/ISO/IEC referenced in Annex 1 of Resolution ITU-R 1. Forms to be used for the submission of patent statements and licensing declarations by patent holders are available from http://www.itu.int/ITU-R/go/patents/en where the Guidelines for Implementation of the Common Patent Policy for ITU-T/ITU-R/ISO/IEC and the ITU-R patent information database can also be found.

Series of ITU-R Recommendations

(Also available online at http://www.itu.int/publ/R-REC/en)

Series	Title
BO	Satellite delivery
BR	Recording for production, archival and play-out; film for television
BS	Broadcasting service (sound)
BT	**Broadcasting service (television)**
F	Fixed service
M	Mobile, radiodetermination, amateur and related satellite services
P	Radiowave propagation
RA	Radio astronomy
RS	Remote sensing systems
S	Fixed-satellite service
SA	Space applications and meteorology
SF	Frequency sharing and coordination between fixed-satellite and fixed service systems
SM	Spectrum management
SNG	Satellite news gathering
TF	Time signals and frequency standards emissions
V	Vocabulary and related subjects

Note: *This ITU-R Recommendation was approved in English under the procedure detailed in Resolution ITU-R 1.*

Electronic Publication
Geneva, 2017

RECOMMENDATION ITU-R BT.1699-2*

Harmonization of declarative application** formats for interactive TV

(Questions ITU-R 131/6 and ITU-T 4/9)

(2005-2009-2013)

Scope

This Recommendation is intended to harmonize the application environment for declarative applications for interactive TV. It specifies common elements, media types, and APIs at the syntactic level of the declarative application environment[1].

The ITU Radiocommunication Assembly,

considering

a) that digital broadcasting services (satellite, terrestrial and cable) are becoming widely available and offer multimedia applications;

b) that multimedia applications comprising video, audio, still-picture, text, graphics, etc. associated with interactive features have been developed;

c) that multimedia applications planned or deployed in some Regions are using the declarative application environment;

d) that common application formats are desirable for production and international exchange of multimedia applications;

e) that Recommendation ITU-T J.200 defines, in addition to the definition above, the high-level architecture for a harmonized set of interactive application formats and application programming interfaces (APIs) and identifies the structure of application environment comprising the procedural application environment and the declarative application environment for digital television services;

f) that Recommendation ITU-T J.202 specifies the common core of procedural application formats in the procedural application environment for interactive TV applications;

g) that specification of harmonized declarative content formats in the declarative application environment is also required for interactive TV applications,

recommends

that for interactive TV applications in the declarative application environment, the harmonized declarative application formats specified in Annexes 1-7 should be used.

---

* Radiocommunication Study Group 6 made editorial amendments to this Recommendation in October 2017 in accordance with Resolution ITU-R 1.

** Recommendation ITU-R BT.1889 provides the definition for "declarative application": An application which primarily makes use of declarative information to express its behaviour; an XML document instance is an example of a declarative application.

1 "Declarative application environment" is defined as an environment for applications and/or content based on markup languages to enable their execution, typically to specify on screen presentation.

Annex 1

Common core of the declarative application formats for interactive TV

1 Introduction

This Recommendation identifies functional commonality among the declarative application environments for interactive TV application specifications ACAP-X[2], BML and DVB-HTML. Elements which are common to these three standards are identified as a "Common Core". The value of the Common Core is to assist program authors to exchange declarative applications internationally using these standards. This Recommendation also notes features outside of the Common Core of the covered standards. The goal of this Recommendation is to note these differences to encourage efforts toward increasing commonality between the standards to further improve functionality and enhance economies of scale.

2 Overview

This Recommendation is intended to harmonize the application environment for declarative applications for interactive TV. It specifies common elements, media types and APIs at the syntactic level of the declarative application environment to satisfy regional application requirements for the three standards ACAP-X, BML and DVB-HTML as specified in the normative references below. This Recommendation is divided into seven annexes. Annex 2 describes the Common Core of the three standards. Annex 3 describes additional functionality outside the Common Core for BML. Annex 4 describes additional functionality outside the Common Core for ACAP-X. Annex 5 describes additional functionality outside the Common Core for DVB-HTML.

The format described in Annex 6 is an intermediate format for translation between formats including the Common Core and the standards covered in this Recommendation. The format described in Annex 7 is a framework to bind content authored in multiple formats into single content.

It is noted that there are other declarative formats such as ETSIMHEG-5, which are not covered in this Recommendation. However, the migration from environments in use to the harmonized environment is assisted by the identification of a Common Core and the translation using the intermediate format.

3 References

3.1 Normative references

[1] BML	ARIB STD-B24 V5.3
[2] ACAP-X	ATSC A/101
[3] DVB-HTML	ETSI TS 102 812 V1.2.2
[4] wTVML	ETSI TS 102 322 V1.1.1
[5] NCL	ABNT NBR 15606-2 V2

[2] The acronym ACAP stands for "Advanced Common Application Platform". It was an ATSC Standard (A/101A) (withdrawn in 2013) that specified an application environment and terminal specifications for television receivers to support both procedural and declarative interactive applications.

Users of this Recommendation are encouraged to investigate the possibility of applying the most recent editions of the references listed above, whose maintenance is the responsibility of the issuing standard bodies. Content authors should refer to the cited documentation to ensure conformity with the semantics provided by those elements, media types and APIs.

NOTE 1 – BML, ACAP-X, DVB-HTML, wTVML and NCL standards are available via the links in Appendix 1.

NOTE 2 – By agreement between ITU-R, ABNT, ATSC, ARIB and ETSI, the versions listed in § 3.1 were authorized for use by ABNT, ATSC, ARIB and ETSI, and accepted by ITU-R for inclusion in this Recommendation. Any subsequent versions of these standards which have not been accepted and approved by ITU-R are not part of this Recommendation.

3.2 Informative references

[1] ETSI-MHEG	ETSI TS 202 184 V1.1.1
[2] J.202	ITU-T J.202
[3] J.200	ITU-T J.200

3.3 Terms and definitions

See the normative references listed in § 3.1.

Annex 2

Common Core

1 Overview

Methodology for Common Core, Common Core of Media types, XML Markup, Stylesheet Markup, Monomedia and Behavioural APIs, which are based on the commonality between ACAP-X, BML and DVB-HTML are described below. Note that BML has four content profiles. Except where otherwise noted, all four profiles of BML are assumed.

1.1 Methodology

1.1.1 Layer model

Graphics layer should lie on top of other layers such as video or text plane.

1.1.2 Application life cycle

There should be a mechanism to destroy an application from outside the application itself.

1.2 Media type

Common media types are listed in Table 1.

TABLE 1

Common media type

Image/jpeg
Image/png
Text/css
Application/xhtml+xml

1.3 Schema

Common schema is listed in Table 2.

TABLE 2

Common schema

http://
https://

1.4 XML markup

Common XML markups are listed in Table 3.

TABLE 3

Common XML markup module

Structure
Text
Hypertext
List
Presentation
Bidirectional text
Forms
Image
Client Side Image Map
Object
Frames
Target
Meta Information
Scripting
Stylesheet
Style Attribute
Link
Base

Common XML markups for BML for basic services (fixed terminal profile), ACAP-X and DVB-HTML are listed in Table 4.

TABLE 4

Common XML markup for BML for basic services, ACAP-X and DVB-HTML

Common attributes		
Core attributes		id class
Style attributes		
		style
Core modules		
Structure module		
	body	%Core.attrib;
	head	
	title	
Text module		
	br	%Core.attrib
	div(1)	%Common.attrib
	p(1), (2)	%Common.attrib
	span	%Common.attrib
Hypertext module		
	a	%Common.attrib accesskey href
Forms module		
	input(1)	%Common.attrib accesskey disabled readonly maxlength type value
Object module		
	object(1)	%Common.attrib data type
Metainformation module		
	meta	name content
Scripting module		
	script	
Stylesheet module		
	style	

(1) Only these elements can be a child element of .

(2) Only these elements and CDATA can be a child element of <p>.

1.5 Stylesheet

1.5.1 Common stylesheet properties

Common stylesheet properties are listed in Table 5.

TABLE 5

Common stylesheet properties

Background	Clear	Outline-color
Background-attachment	Clip	Outline-style
Background-color	Color	Outline-width
Background-image	Content	Overflow
Background-position	Counter-increment	Padding
Background-repeat	Counter-reset	Padding-bottom
Border	Display	Padding-left
Border-bottom	Float	Padding-right
Border-bottom-color	Font	Padding-top
Border-bottom-style	Font-family	Position
Border-bottom-width	Font-size	Right
Border-color	Font-style	Text-align
Border-left	Font-variant	Text-decoration
Border-left-color	Font-weight	Text-indent
Border-left-style	Height	Text-transform
Border-left-width	Left	Top
Border-right	Letter-spacing	Vertical-align
Border-right-color	Line-height	Visibility
Border-right-style	List-style	White-space
Border-right-width	List-style-image	Width
Border-style	List-style-position	Word-spacing
Border-top	List-style-type	Z-index
Border-top-color	Margin	Nav-index
Border-top-style	Margin-bottom	Nav-left
Border-top-width	Margin-left	Nav-right
Border-width	Margin-right	Nav-up
Bottom	Margin-top	Nav-down
Caption-side	Outline	

Common stylesheet properties for BML for basic services, ACAP-X and DVB-HTML are listed in Table 6.

TABLE 6

Common stylesheet properties for BML for basic services, ACAP-X and DVB-HTML

@media
Margin
Padding-top
Padding-right
Padding-bottom
Padding-left
Border-width
Border-style
Position
Left(1)
Top(1)
Width(1)
Height(1)
Z-index
Line-height
Display
Visibility
Overflow
Background-image
Background-repeat
Font-family
Font-size
Font-weight
Text-align
Letter-spacing
White-space

(1) The elements <input>, <object>, , and <p> must have these property values. The elements
, <a>, must not have these property values.

Furthermore, the following restrictions should be applied:

– Display property
Only block element can be applied for <p>, , <body>, <input> and <object>.
Only inline values can be applied for
, <a> and .

– Position property
Only absolute values can be applied for <p>, , <input> and <object>.
Only static values can be applied for
, and <a>.

1.5.2 Common CSS selectors

Common CSS selectors are listed in Table 7.

TABLE 7

Common CSS selectors

Universal
Type
Descendant
Class
Id
:first-child pseudo-class
:link pseudo-class
:hover pseudo-class
:active pseudo-class
:focus pseudo-class
:lang pseudo-class
:pseudo-elements (:first-child, :first-letter, :before, :after)

Common CSS selectors for BML for basic services, ACAP-X and DVB-HTML are listed in Table 8.

TABLE 8

Common CSS selectors for BML for basic services, ACAP-X and DVB-HTML

Universal
Type
Dynamic(:focus and :active)
Class
Id

1.6 Scripting language

Common scripting language is ECMAScript 2nd Edition with the following restriction:

– Number type supports integer operation only.

Common native objects for BML for basic services, ACAP-X and DVB-HTML are listed in Table 9.

TABLE 9

Common native objects for BML for basic services, ACAP-X and DVB-HTML

Object	Methods, properties
(global)	NaN parseInt(string, radix) isNaN(number)
Object	All
Object.prototype	All
Function	prototype length
Function.prototype	All
Array	All
Array.prototype	All
String	All
String.prototype	All
Boolean	All
Boolean.prototype	All
Number	Prototype MAX_VALUE MIN_VALUE NaN Number([value]) New number([value])
Number.prototype	All
Date	prototype Date([year [, month [, date [, hours [, minutes [, seconds [, ms]]]]]]]) new Date([year [, month [, date [, hours [, minutes [, seconds [, ms]]]]]]])
Date.prototype	toString() getFullYear() getUTCFullYear() getMonth() getUTCMonth() getDate() getUTCDate() getDay() getUTCDay() getHours() getUTCHours() getMinutes() getUTCMinutes() getSeconds()

TABLE 9 (*end*)

	getUTCSeconds() getMilliseconds() getUTCMilliseconds() getImtezoneOffset() setMilliseconds(ms) setUTCMilliseconds(ms) setSeconds(sec [, ms]) setUTCSeconds(sec [, ms]) setMinutes(min, [, sec [, ms]]) setUTCMinutes(min, [, sec [, ms]]) setHours(hours, [,(min, [, sec [, ms]])] setUTCHours(hours, [,(min, [, sec [, ms]])] setDate(date) setMonth(mon [, date]) setUTCMonth(mon [, date]) setFullYear(year [, mon [, date]]) setUTCFullYear{year [, mon [, date]]} toLocaleString() toUTCString()

For BML for basic services, the length to represent signed integer is 32 bits including sign.

1.7 DOM API

Common DOM APIs in DOM level 1 are listed in Table 10.

TABLE 10

Common DOM level 1 APIs

Core fundamental	DOMException
	DOMImplementation
	DocumentFragment
	Document
	Node
	NodeList
	NamedNodeMap
	CharacterData
	Attr
	Element
	Text
	Comment

Common DOM level 1 APIs for the BML for basic services, ACAP-X and DVB-HTML are listed in Table 11. Interfaces listed in Table 11 that have no specified attributes or methods cover all attributes and methods of the interfaces.

TABLE 11

Common DOM level 1 APIs for BML for basic services, ACAP-X and DVB-HTML

	Interface	Attributes, Methods
Core fundamental	DOMImplementation	
	Document	implementation documentElement
	Node	parentNode firstChild lastChild previousSibling nextSibling
	CharacterData	data length
	Element	tagName
	Text	

Annex 3

Additional elements, media types and APIs for BML

Elements, media types and APIs for BML in addition to those listed in Annex 2 are described below. Items marked "BD)" are common to BML and DVB-HTML. Items marked "BA)" are common to BML and ACAP-X.

1 Additional BML media types

Additional BML media types are listed in Table 12.

TABLE 12

Additional BML media types

Multipart/mixed
Text/xml[BD)]
Text/xsl
Text/html
Text/plain[BD)]
Text/css
Text/X-arib-bml;charset="euc-jp"
Text/X-arib-bml;charset="UTF-16"
Text/X-arib-bml;charset="Shift_JIS"
Text/X-arib-bml;charset="UTF-8"

TABLE 12 (*continued*)

Text/X-arib-jis8text
Text/X-arib-ecmascript;charset="euc-jp"
Text/X-arib-ecmascript;charset="UTF-16"
Text/X-arib-ecmascript;charset="Shift_JIS"
Text/X-arib-ecmascript;charset="UTF-8"
Image/gif
Image/X-arib-png
Image/X-arib-mng
Image/X-arib-mpeg2-I
Image/X-arib-mpeg4-I-simple
Image/X-arib-mpeg4-I-core
Image/X-arib-H264-I-baseline
Image/X-arib-H264-I-main
Audio/X-arib-mpeg2-aac
Audio/X-arib-mpeg2-bc
Audio/X-arib-mpeg4
Audio/X-arib-aiff
Audio/X-arib-additional
Audio/X-arib-romsound
Application/X-arib-stream-text;charset="euc-jp"
Application/X-arib-stream-text;charset="UTF-16"
Application/X-arib-stream-text;charset="Shift_JIS"
Application/X-arib-stream-text;charset="UTF-8"
Application/X-arib-stream-jis8text
Application/X-arib-stream-png
Application/X-arib-stream-jpeg
Application/X-arib-stream-mpeg2-I
Application/X-arib-stream-mpeg4-I-simple
Application/X-arib-stream-mpeg4-I-core
Application/X-arib-mpeg2-tts
Application/X-arib-bmlclut
Application/X-arib-btable
Application/X-arib-drcs
Application/X-arib-PDI
Application/X-arib-resourceList
Application/X-arib-stream-H264-I-baseline
Application/X-arib-stream-H264-I-main
Application/X-arib-mpeg2-ts
Application/X-arib-rootcertificate

TABLE 12 (*end*)

Application/X-arib-contentPlayContrl
Application/X-arib-streamControlInfo
Application/X-arib-meta+xml;charset="UTF-8"
Application/X-arib-meta+xml;charset="UTF-16"
Video/X-arib-mpeg1
Video/X-arib-mpeg2
Video/X-arib-mpeg4-simple
Video/X-arib-mpeg4-core
Video/X-arib-H264-baseline
Video/X-arib-H264-main

2 Additional BML XML markup

Additional BML XML markups are listed in Table 13.

TABLE 13

Additional XML markups

Module	Tag
Table[BA)]	All
Intrinsic events[BA)]	All
Name identification[BA)]	All
Applet	All
Basic forms	All
Basic table[BD)]	All
Server side image map	All
Iframe[BD)]	All
Legacy	All
BML extension	Bml, bevent, beitem, body&, div&, p&, span&, a&, bdo&, object&

3 Additional BML CSS properties

Additional BML CSS properties are listed in Table 14.

TABLE 14

Additional CSS properties

Clut(1)
Color-index(1)
Background-color-index(1)
Border-color-index
Border-top-color-index(1)
Border-right-color-index(1)
Border-bottom-color-index(1)
Border-left-color-index(1)
Outline-color-index
Resolution(1)
Display-aspect-ratio(1)
Grayscale-color-index(1)
Used-key-list(1)
nav-index(1)
nav-up(1)
nav-down(1)
nav-left(1)
nav-right(1)
-wap-marquee
-wap-marquee-style
-wap-marquee-loop
-wap-marquee-dir
-wap-marquee-speed
-wap-accesskey
-wap-input-format
-wap-input-required

(1) These attributes are employed for BML for basic services.

4 Additional BML DOM APIs

Additional BML DOM level 1 APIs are listed in Table 15.

TABLE 15

Additional BML DOM level 1 APIs

Core extension[BA)]	CDATASection
	DocumentType
	Notation
	Entity
	EntityReference
	ProcessingInstruction
HTML	HTMLCollection[BA)]
	HTMLDocument[BA)]
	HTMLElement[BA)]
	HTMLAnchorElement[BA)]
	HTMLFormElement[BA)]
	HTMLInputElement[BA)]
	HTMLOptionElement[BA)]
	HTMLSelectElement[BA)]
	HTMLTextAreaElement[BA)]
	HTMLImageElement[BA)]
	HTMLObjectElement[BA)]
	HTMLBodyElement[BA)]
	HTMLBlockquoteElement
	HTMLPreElement
	HTMLHeadingElement
	HTMLHRElement
	HTMLDivElement[(1)]
	HTMLParagraphElement[(1)]
	HTMLQuoteElement
	HTMLBRElement[(1)]
	HTMLModElement
	HTMLBaseElement
	HTMLLinkElement
	HTMLDListElement
	HTMLOlistElement
	HTMLUListElement
	HTMLLIElement
	HTMLButtonElement
	HTMLFieldSetElement
	HTMLLabelElement
	HTMLLegendElement
	HTMLOptGroupElement

TABLE 15 (*end*)

HTML (*cont.*)	HTMLTableCaptionElement
	HTMLTableColElement
	HTMLTableElement
	HTMLTableSectionElement
	HTMLTableCaptionElement
	HTMLTableColElement
	HTMLTableElement
	HTMLTableSectionElement
	HTMLTableCellElement
	HTMLTableRowElement
	HTMLAreaElement
	HTMLMapElement
	HTMLParamElement
	HTMLFrameSetElement
	HTMLFrameElement
	HTMLIFrameElement
	HTMLMetaElement(1)
	HTMLTitleElement(1)
	HTMLScriptElement(1)
	HTMLStyleElement(1)
	HTMLHeadElement(1)
	HTMLHtmlElement(1)

(1) These elements are employed for BML for basic services.

BML extensions of DOM APIs are listed in Table 16.

TABLE 16

Additional BML extensions

BML extension	BMLDocument(1)
	BMLCSS2Properties(1)
	BMLEvent(1)
	BMLIntrinsicEvent(1)
	BMLBeventEvent(1)
	BMLDocument(1)
	BMLElement
	BMLBlockquoteElement
	BMLPreElement
	BMLHeadingElement
	BMLHRElement

TABLE 16 (*end*)

BML extension (*cont.*)	BMLDivElement(1)
	BMLSpanElement(1)
	BMLParagraphElement(1)
	BMLQuoteElement
	BMLBRElement(1)
	BMLModElement
	BMLAnchorElement(1)
	BMLLinkElement
	BMLDListElement
	BMLOListElement
	BMLUListElement
	BMLLIElement
	BMLButtonElement
	BMLFieldSetElement
	BMLFormElement
	BMLInputElement(1)
	BMLLabelElement
	BMLLegendElement
	BMLOptGroupElement
	BMLOptionElement
	BMLSelectElement
	BMLTextAreaElement
	BMLTableCaptionElement
	BMLTableColElement
	BMLTableElement
	BMLTableSectionElement
	BMLTableCellElement
	BMLTableRowElement
	BMLImageElement
	BMLAreaElement
	BMLMapElement
	BMLObjectElement(1)
	BMLFrameSetElement
	BMLFrameElement
	BMLIFrameElement
	BMLBodyElement(1)
	BMLBmlElement(1)
	BMLBeventElement(1)
	BMLBitemElement(1)

(1) These elements are employed for BML for basic services.

5 Additional functions capable of integrated broadcast-broadband services

5.1 Markup language switch

A function added to ECMAScript to launch another declarative application environment such as an HTML browser to access IP service portals is listed in Table 17.

TABLE 17

Markup language switch function

Number startExtraBrowser(input String browserName, input Number showAV, input String returnURI, input String uri)

Argument:

browserName	Name of an extra browser to be started up
showAV	Flag that specifies whether or not the current playback of a TV programme (video and sound) is allowed to continue when the resident application software has been started up
	1: The playback is allowed to continue
	0: The playback is not allowed to continue
returnURI	URI of a component that is rendered first when the BML browser is restarted after the resident application software that was started up by the function has been quitted. To specify no component, returnURI must contain an empty string. This argument is designed to help a receiver to work. It is not required that any receiver depends on the argument to work properly.
uri	URI that is rendered first when the extra browser is started up

Return values:

1	Success
NaN	Failure

Description:

This function starts up an extra browser, as specified in browserName. Once this function is executed, no script parts following this function are executed.

5.2 Content download

Two functions added to ECMAScript for content download are listed in Tables 18 and 19.

TABLE 18

Initiation of content download

Number startDlcDownload(input String src_path)

Argument:

src_path URI which represents control information of content to be downloaded

Return values:

1 Success

–1 Invalid parameters

–4 Failure due to incapability of accept of request

NaN Failure due to other reasons

Description:

This function starts acquisition of control information of content to be downloaded described as 'src_path'. This function returns immediately without waiting for completion of acquisition of the control information. Control information of content to be downloaded is meta-information which contains information related to the content including location, license information, etc. Because the control information depends on each IP service which offers the content, format of the control information is not part of this Recommendation and is not defined in the BML standard. A receiver which allows execution of this function is expected to acquire the content and its related information, as instructed by the control information.

TABLE 19

Acquisition of download status

Number getDlcDownloadStatus()

Argument:

None

Return values:

1 Request acceptable

–4 Request unacceptable

NaN Failure

Description:

This function returns a status indicating whether request of control information of content to be downloaded is acceptable by calling startDlcDownload().

5.3 VOD playback

A function added to ECMAScript for VOD content playback is listed in Table 20.

TABLE 20

VOD content playback

Number startVOD(input String metafile_uri [, input Array option])

Argument:

metafile_uri URI of playback control information file for VOD content

Return values:

1 Success

NaN Failure

Description:

This function starts a resident application to acquire and play VOD content, and gives required information to the application and the receiver. Actual acquisition and presentation of the VOD content is carried out by the resident application.

Annex 4

Additional elements, media types and APIs for ACAP-X

Elements, media types and APIs for ACAP-X in addition to those listed in Annex 2 are described below. Items marked "AD)" are common to ACAP-X and DVB-HTML. Items marked "AB)" are common to ACAP-X and BML.

1 Additional ACAP-X media types

Additional ACAP-X media types are listed in Table 21.

TABLE 21

Additional ACAP-X media types

Application/acap-j
Application/acap-certificate
Application/acap-digest
Application/acap-permission
Application/acap-signature
Application/acap-x
Application/acap-x-metadata
Application/font-tdpfr
Application/java

TABLE 21 (*end*)

Application/zip
Application/xhtml+xml
Audio/ac3
Audio/basic
Audio/mpeg[AD)]
Image/mpeg[AD)]
Text/ecmascript[AD)]
Video/mng
Video/mpeg
Video/mpv

2 Additional ACAP-X XML markup

Additional ACAP-X XML markups are listed in Table 22.

TABLE 22

Additional ACAP-X XML markups

Module	Tag
Table[AB)]	All
Intrinsic Events[AB)]	All
Name Identification[AB)]	All

3 Additional ACAP-X CSS properties

Additional ACAP-X CSS properties and selectors are listed in Table 23.

TABLE 23

Additional ACAP-X CSS properties and selectors

Properties	Atsc-dynamic-refresh
Selectors	Child
	Adjacent sibling
	Attribute and attribute values

4 Additional ACAP-X stylesheet attributes

Additional ACAP-X stylesheet attributes are CSS level 2, CSS-BOX, CSS-COLOR, CSS-TV, CSS-UI and their related DOM APIs.

5 Additional ACAP-X DOM APIs

Additional ACAP-X DOM level 2 APIs are listed in Table 24.

TABLE 24

Additional ACAP-X DOM level 2 APIs

Core fundamental[AD)]	DOMException
	DOMImplementation
	DocumentFragment
	Document
	Node
	NodeList
	NamedNodeMap
	CharacterData
	Attr
	Element
	Text
	Comment
Core extension[AB)]	CDATASection
	DocumentType
	Notation
	Entity
	EntityReference
	ProcessingInstruction
HTML[AB)]	HTMLAnchorElement
	HTMLBodyElement
	HTMLCollection
	HTMLDocument
	HTMLElement
	HTMLFormElement
	HTMLInputElement
	HTMLObjectElement
	HTMLOptionElement
	HTMLSelectElement
	HTMLTextAreaElement
	HTMLImageElement
View	AbstractView
	DocumentView

TABLE 24 (*end*)

Style sheets[AD)]	DocumentStyle
	LinkStyle
	MediaList
	Stylesheet
	StylesheetList
CSS	Counter
	CSSCharsetRule
	CSSFontFaceRule
	CSSImportRule
	CSSMediaRule
	CSSPageRule
	CSSPrimitiveValue
	CSSRule
	CSSRulesList
	CSSStyleDeclaration
	CSSStyleRule
	CSSStyleSheet
	CSSUnknownRule
	CSSValue
	CSSValueList
	DocumentCSS
	DOMImplementationCSS
	ElementCSSInlineStyle
	Rect
	RGBColor
Event[AD)]	ViewCSS
	Event
	EventException
	EventListner
	EventTarget
EventSet	KeyEvent
	KeyModifiers
	MouseEvent[AD)]
	MutationEvent[AD)]
	UIEvent[AD)]
	VirtualKeys

ACAP-X extensions of DOM APIs are listed in Table 25.

TABLE 25

Additional ACAP-X extensions

ACAP-X Extension	DOMExceptionExt
	HTMLAnchorElementExt
	HTMLDocumentExt
	HTMLImageElementExt
	HTMLFormElementExt
	HTMLObjectElementExt
	HTMLTriggerObjectElementExt
	HTMLOptionsCollection
	DocumentViewExt

Annex 5

Additional elements, media types, and APIs for DVB-HTML

Elements, media types and APIs for DVB-HTML in addition to those listed in Annex 2 are described below. Items marked "DB)" are common to DVB-HTML and BML. Items marked "DA)" are common to DVB-HTML and ACAP-X.

1 Additional DVB-HTML media types

Additional DVB-HTML media types are listed in Table 26.

TABLE 26

Additional DVB-HTML media types

Application/xml
Application/dvbj
Application/dvb.pfr
Audio/mpeg[DA)]
Image/gif
Image/mpeg[DA)]
Text/ecmascript[DA)]
Text/plain[DB)]
Text/css
Text/xml[DB)]
Text/dvb.utf8
Multipart/dvb.service
Video/dvb.mpeg.drip

2 Additional DVB-HTML XML markups

Additional DVB-HTML XML markups are listed in Table 27.

TABLE 27

Additional XML markups

Basic Table[DB)]
Iframe[DB)]

3 Additional DVB-HTML CSS properties

Additional DVB-HTML CSS properties and selectors are listed in Table 28.

TABLE 28

Additional DVB-HTML CSS properties and selectors

Properties	Direction
	Unicode-bidi
	Min-width
	Max-width
	Min-height
	Max-height
	Font-stretch
	Font-size-adjust
	Table-layout
	Empty-cells
	Speak-header
	Opacity
	Nav-first
	Clip-video
	Compose-rule
Selectors	Child
	Adjacent sibling
	Attribute and attribute values

4 Additional DVB-HTML DOM APIs

4.1 Additional DVB-HTML DOM level 1 APIs

Additional DVB-HTML DOM level 1 APIs are listed in Table 29.

TABLE 29

Additional DVB-HTML DOM level 1 APIs

HTML	DVBHTMLCollection
	DVBHTMLDocument
	DVBHTMLElement
	DVBHTMLAnchorElement
	DVBHTMLButtonElement
	DVBHTMLFormElement
	DVBHTMLInputElement
	DVBHTMLOptionElement
	DVBHTMLSelectElement
	DVBHTMLTextAreaElement
	DVBHTMLImageElement
	DVBHTMLAreaElement
	DVBHTMLMapElement
	DVBHTMLObjectElement
	DVBHTMLFrameSetElement
	DVBHTMLFrameElement
	DVBHTMLIFrameElement

4.2 Additional DVB-HTML DOM level 2 APIs

Additional DVB-HTML DOM level 2 APIs are listed in Table 30.

TABLE 30

Additional DVB-HTML DOM level 2 APIs

Core fundamental[DA)]	DOMException
	DOMImplementation
	DocumentFragment
	Document
	Node
	NodeList
	NamedNodeMap
	CharacterData
	Attr
	Element
	Text
	Comment

TABLE 30

View	AbstractView
	DocumentView
Style sheets[DA)]	DocumentStyle
	LinkStyle
	MediaList
	Stylesheet
	StylesheetList
Event[DA)]	DocumentEvent
	Event
	EventException
	EventListener
	EventTarget
EventSet[DA)]	MouseEvent
	MutationEvent
	UIEvent

Annex 6

Presentation interoperability through translation

Some service providers may find the core functionality a little limiting for their purposes, though they still wish to target the multiple presentation engines identified in this Recommendation.

As a supplement to the core functionality, the Worldwide Television Markup Language (wTVML) specified in ETSI TS 102 322 defines a format to author such interactive services, which can then be mechanically translated to any desired presentation markup language. The wTVML format uses an XML data structure with declarative behaviour and little or no scripting, and as such is easier to translate to other markup languages. Because wTVML expresses the author's intent, rather than implementation, the richer non-core features of each supported markup become available for use.

In addition, wTVML can also be used as a native presentation language.

When using wTVML as the intermediate format for format translation of the declarative application, the following may require careful consideration for translation of the original application to wTVML in particular organization of the application:

– broadcast message signal, such as that carried by DSM-CC stream event;
– additional functions in scripting language, such as that for cache control.

Annex 7

Presentation interoperability by management framework of multiple formats for declarative applications

Some service providers may wish to employ multiple formats identified in this Recommendation including the common core. Usage of the multiple formats can be in many ways, such as concurrent use, switch of one format to another, and so on. This means management framework for content employing multiple formats is required.

As a framework for management of multiple declarative application formats, Nested Context Language (NCL) specified in ABNT NBR 15606-2 defines a format to bind content authored in multiple formats into single content. NCL is a glue language based on XML that holds media objects together in a multimedia presentation, no matter which is each object type.

When using NCL as the framework to bind content authored in different declarative application formats, following may request careful consideration to organize binding content:

– System time base can only be in NCL, not in each media object. In particular, time based event such as an event triggered by normal play time should be handled in NCL. LuaScript is one of the processing mechanisms for this kind of behaviour in NCL.

Appendix 1

Standards

BML	http://www.arib.or.jp/tyosakenkyu/kikaku_hoso/hoso_kikaku_number.html
ACAP-X	http://www.atsc.org/cms/standards/a_101a.pdf
DVB-HTML	http://www.etsi.org/deliver/etsi_ts/102800_102899/102812/01.03.01_60/ts_102812v010301p.pdf
wTVML	http://webapp.etsi.org/workprogram/Report_workitem.asp?WKI_ID=19886
NCL	http://abnt.iso.org/livelink/livelink/fetch/2000/2827/7589984/8699711/8727725/ABNTNBR15606%2D2_2007Ing_2008Vc2_2009.pdf