

Digital terrestrial television broadcasting – Data Coding and transmission specification for digital broadcasting

Televisão digital terrestre – Codificação de dados e especificações de transmissão para radiodifusão digital

Televisión digital terrestre — Codificación de datos y especificaciones de transmisión para radiodifusión digital

Digital terrestrial television - Data Coding and transmission specification for digital broadcasting

地上デジタルテレビジョン放送の伝送方式

Foreword

This document is the result of the joint efforts of the ABNT, ARIB and SBTVD Forum under the standardization and technical cooperation activities of the Brazil-Japan Digital Television Joint Working Group.

The Brazilian Association for Standardization (ABNT) is the organism responsible for technical standardization in Brazil, providing essential support for Brazilian technical development. It is a private, non-profit organization, recognized as the only National Standardization Body. It provides Brazilian society with systematic knowledge, through normative documents, enabling the production, commercialization and use of goods and services, in a competitive and sustainable manner, in the internal and external markets, contributing to scientific and technological development, environmental and consumer's protection.

The Association of Radio Industries and Businesses (ARIB) was designated as “the Center for Promotion of Efficient Use of the Radio Spectrum” and “the Designated Frequency Change Support Agency” by the Minister of Internal Affairs and Communications (MIC) of Japan under the provisions of the Radio Law. Under this designation, ARIB conducts studies and R&D, establishes standards, provides consultation services for radio spectrum coordination, cooperates with other overseas organizations and provides frequency change support services for the smooth introduction of digital terrestrial television broadcasting. These activities are carried out in cooperation with and/or participation by telecommunication operators, broadcasters, radio equipment manufacturers and related organizations as well as under the support by MIC.

The Brazilian Digital Terrestrial Television Forum (SBTVD Forum) is a non-profit entity, created with the objective of aiding and stimulating the development and implementation of best practices, with the aim of making systems reality for digital broadcasting of images and sounds in Brazil a success. Since the creation of the SBTVD Forum in February, 2007, its members have endeavored to establish standards of technical quality which permit deployment of digital television in Brazil. The Technical Module has contributed to the preparation of standards, with active participation by universities, research centers, related industry organizations and broadcasters.

This document does not describe the industrial property rights mandatory to these standards.

This document has no standardization value. Its purpose is to serve as a reference for characterizing the specificities of Brazilian and Japanese digital terrestrial television standards within the scope of the Brazil-Japan Digital Television Joint Working Group.

This document is drafted in accordance with the rules established in the ISO/IEC Directives, Part 2.

No reference is made to the domestic policies of the countries.

1 Scope

This document addresses the standard for the data coding and transmission specification for digital terrestrial television broadcasting in Brazil and Japan.

The document refers to the middleware presentation environment, as specified in ARIB STD-B24 and in ABNT 15606 (volumes 1,2,3, and 5) and the harmonization of the middleware execution environment, as specified in ARIB STD-B23 and in ABNT 15606-4.

2 Normative references

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

ABNT NBR 15606-1:2010, *Digital Terrestrial TV – Data Coding and transmission specification for digital broadcasting – Part 1: Data Coding*

ABNT NBR 15606-2:2011, *Digital Terrestrial TV – Data Coding and transmission specification for digital broadcasting – Part 2: Ginga-NCL for fixed and mobile receivers: XML application language for application coding*

ABNT NBR 15606-3:2010, *Digital Terrestrial TV – Data Coding and transmission specification for digital broadcasting – Part 3: Data Transmission Specification*

ABNT NBR 15606-4:2010, *Digital Terrestrial TV – Data Coding and transmission specification for digital broadcasting – Part 4: Ginga-J - The environment for the execution of procedural applications*

ABNT NBR 15606-5:2011, *Digital Terrestrial TV – Data Coding and transmission specification for digital broadcasting – Part 5: Ginga-NCL for portable receivers: XML application language for application coding*

ARIB STD-B24 V5.3(2009), *Data Coding and transmission specification for digital broadcasting*

ARIB STD-B23 V1.2(2009), *Application execution engine platform for digital broadcasting*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ABNT NBR 15606-1:2010, ABNT NBR 15606-2:2011, ABNT NBR 15606-3:2010, ABNT NBR 15606-4:2010, ABNT NBR 15606-5:2011, ARIB STD-B23 V1.2:2008 and ARIB STD-B24 V5.3:2009 apply.

4 Abbreviated terms

For the purposes of this document, the abbreviated terms given in ABNT NBR 15606-1:2010, ABNT NBR 15606-2:2011, ABNT NBR 15606-3:2010, ABNT NBR 15606-4:2010, ABNT NBR 15606-5:2011, ARIB STD-B23 V1.2:2008 and ARIB STD-B24 V5.3:2009 apply.

5 Commonalities and Differences in the monomedia data coding

The common parts and differences of ABNT NBR 15606-1:2009, ARIB STD-B24 V5.3 and ARIB STD-B23 V1.2 and how they correspond are described in Table 1.

Table 1 — Correspondence among ABNT NBR 15606-1:2009, ARIB STD-B24 V5.3 and ARIB STD-B23 V1.2 standards

Description	ABNT NBR 15606-1	ARIB STD-B24	ARIB STD-B23
PNG with restriction	YES	YES	YES
PNG without restriction	YES	YES	YES
GIF with restriction	YES	YES	YES
MPEG-4 "I – VOP"	YES	YES	YES
H.264 / MPEG-4 AVC "I - Picture"	YES	YES	YES
JPEG with restriction	YES	YES	YES
JPEG without restriction	YES	YES	YES
MNG with restriction	YES	YES	YES
MNG without restriction	YES	YES	YES
MPEG-2 audio AAC LC/BC	YES	YES	YES
PCM ^a	YES	YES	YES
MPEG-4 audio AAC-LC	YES	YES	YES
Monomedia format for audio <i>clips</i> MPEG-1 audio (<i>Layers 1 and 2</i>)	YES	YES	YES
MPEG-1 audio <i>layer 3</i> (MP3)	YES	NO (NOT DEFINED)	NO (NOT DEFINED)
MPEG-1 video <i>clips – Part 2</i>	YES (Transmission Method defined as File Format or PES)	NO (Transmission Method defined only as PES)	NO (Transmission Method defined only as PES)
MPEG-2 video	NO (NOT DEFINED)	YES (Transmission Method defined only as PES)	YES (Transmission Method defined only as PES)
MPEG-2 video <i>drips</i>	YES	YES	YES
MPEG-4 video <i>clips</i>	YES (Transmission Method defined as File Format or PES)	NO (Transmission Method defined only as PES)	NO (Transmission Method defined only as PES)
H.264 / MPEG-4 AVC <i>clips</i>	YES (Transmission Method defined as File Format or PES)	NO (Transmission Method defined only as PES)	NO (Transmission Method defined only as PES)
8 bits Character code	NO	YES	YES (Code mapping to UCS defined)
<i>Universal multi-octet coded character set (UCS)</i>	YES	YES	YES
<i>Shift-JIS</i> Character code	NO (NOT DEFINED)	YES	YES (Code mapping to UCS defined)

6 Commonalities and Differences in the XML based multimedia coding scheme

6.1 NCL and the Ginga-NCL presentation environment

NCL (Nested Context Language) is an XML application that allows authors to write interactive multimedia presentations. Using NCL, an author can describe the temporal behavior of a multimedia presentation, associate hyperlinks (user interaction) with media objects, define alternatives for presentation (adaptation), and describe the layout of the presentation on multiple devices. All these functionalities provided using only the declarative approach

Ginga-NCL is the logical subsystem of the Ginga system that processes NCL declarative applications (NCL documents) (see [ABNT NBR 15606-2 and ABNT NBR 15606-5] for the NCL profiles for the Brazilian DTV System). A key component of Ginga-NCL is the declarative content decoding engine (NCL formatter or NCL user agent). Another important module is the Lua engine, which is responsible for interpreting imperative NCLua objects, that is, media objects with Lua code (see ABNT NBR 15606-2 for Lua reference manual). Lua is the scripting language of NCL.

Ginga-NCL deals with applications collected inside a data structure known as private base. A Private Base Manager component is in charge of receiving NCL document editing commands and maintaining the NCL documents being presented. In Ginga-NCL, an application can be generated or modified on the fly, using Ginga-NCL editing commands [NCL Live Editing Commands, 2006].

6.2 Ginga-NCL harmonization with BML

NCL has a stricter separation between document's (or application's) content and structure. NCL does not define any media itself. Instead, it defines the glue that holds media together in multimedia presentations.

An NCL document only defines how media objects are structured and related, in time and space. As a glue language, it does not restrict or prescribe media-object content types.

In this sense, it is possible to have image objects (GIF, JPEG, etc.), video objects (MPEG, MOV, etc.), audio objects (MP3, WMA etc.), text objects (TXT, PDF, etc.), imperative objects (Xlet, Lua, ECMAScript, etc.), declarative hypermedia objects (SVG, XHTML, other embedded NCL application, etc.), as NCL media objects. Which media objects are supported depends on the media players that are coupled in the NCL formatter. One of these players is the main video and audio decoder/player, usually implemented in hardware in a DTV receiver. In this way, the main video and audio are treated like all other media objects that may be related using NCL.

Another NCL media object that is required in a Ginga-NCL implementation is the XHTML-based media object. Therefore, NCL does not substitute but embed XHTML-based documents (or objects). As with other media objects, which XHTML-based language will have support in an NCL formatter is an implementation choice, and therefore it will depend on which XHTML browser will act as a media player coupled in the NCL formatter.

As a consequence, it is possible to have BML browsers embedded in an NCL document player, or to receive a XHTML browser code through datacasting and install it as a plug-in (usually as Lua objects).

It is also possible to have the ITU harmonization browser (see Rec. ITU-R BT.1699-1) implemented, and receiving the complementary part, if needed, as a plug-in, in order to convert the XHTML player into one for the several profiles of standardized XHTML-based (e.g., BML) browsers.

Note thus that, in the extreme case, an NCL document may be reduced to having only one BML media object. In this case, the NCL document player will act nearly like a BML browser.

Any XHTML-based media object implementation in conformance with Ginga-NCL shall at least support all common XML markups and stylesheet properties for the BML for basic services ("fixed terminal profile"), ACAP-X and DVB-HTML, as defined in ITU Recommendation [Rec. ITU-R BT.1699-1].

Common features of ECMAScript native objects and DOM APIs are not required. However, when ECMAScript engine is implemented, it should support the common native objects for BML for basic services, ACAP-X and DVB-HTML, as defined in ITU Recommendation [Rec. ITU-R BT.1699-1]. Likewise, the DOM APIs, when implemented, should support the common DOM level 1 APIs for BML for basic services, ACAP-X and DVB-HTML, as defined in ITU Recommendation [Rec. ITU-R BT.1699 1].

No matter the case, the XHTML-based user agent implementation should be a consequence of the following requirements:

- robustness;
- alignment with W3C specifications;
- rejection of non-conformant content;
- compatibility with the Ginga security model;
- minimization of the redundancy with existing NCL facilities;
- precise content layout control mechanisms;
- support of different pixel aspect ratios.

Although an XHTML-based browser is required to be supported in NCL, the use of XHTML elements to define relationships (including XHTML links) is not recommended when authoring NCL documents. Structure-based authoring should be emphasized for the well-known reasons largely reported in the literature.

It must also be noted that, during the exhibition of media-object contents, several events are generated. Examples of events are the presentation of marked segments of a media-object content, the selection of a marked content segment, etc. Events may generate actions on other media objects, like to start or stop their presentations. Hence, events shall be reported by media players to the NCL formatter that, in its turn, can generate actions to be applied to these or other players. Ginga-NCL defines a generic API [ABNT NBR 15606-2 and ABNT NBR 15606-5] to standardize the interface between the Ginga-NCL formatter and each specific player.

Therefore, when any media player, in particular a BML browser, is integrated to the Ginga-NCL formatter, it shall support the generic API. For XHTML-based browsers, an adapter module can be necessary to accomplish the integration.

Finally, for live editing, Ginga-NCL has also defined NCL stream events and editing commands [NCL Live Editing Commands, 2006; ABNT NBR 15606-2; and ABNT NBR 15606-5]. Again, although an XHTML-based browser shall be supported, the use of XHTML elements to define relationships (including those triggered by stream events) should be dissuaded in authoring NCL documents, for the same motivation: structure-based authoring should be emphasized for the well-known reasons largely reported in the literature.

The general differences between Ginga-NCL and ARIB-BML are:

- Based on NCL (Nested Context Language): *declarative support* for defining spatiotemporal synchronization, content and presentation adaptability; presentation on multiple exhibition devices and live editing of non-linear TV programs
- Support for Lua and Java imperative objects
- Lua as the NCL scripting language

In opposition, general similarities between Ginga-NCL and ARIB-BML are:

- NCL is a glue language that does not restrict or prescribe any media-object content type. NCL treats an HTML document as one of its possible objects, depending only on the XHTML player implementation. This player can optionally be a BML engine.
- Any XHTML-based hypermedia object implementation in conformance with this Ginga shall at least support all common XML markups and stylesheet properties for the BML for basic services. Common features of ECMAScript native objects and DOM APIs, for the BML for basic services, can also be supported, but they are optional.

Table 2 — Correspondence between ABNT NBR 15606-2:2009 and ARIB STD-B24 V5.32009 standards

Features	ABNR NBR 15606-2	ARIB STD-B24
XML application language	YES	YES
XML-based language	Structure-based	Media-based
Support to spatiotemporal synchronizations without viewer interactions	YES, by declarative element and by dynamic event handling	YES, by dynamic event handling
<i>Declarative</i> support to viewer interactions	YES	YES
<i>Declarative</i> support to content adaptation and content presentation adaptation	YES	NO
Support to multiple exhibition devices	YES	YES, but only by ECMAScript
<i>Declarative</i> support for handling local variables	YES	Partial, only CSS properties
Support for handling global variables	YES, by declarative element and by Lua scripts	YES, by pseudo object of ECMAScript
Declarative key navigation (remote control, etc) support	YES	YES
Reuse	Large support for layout, media object, and structure reuse	Only of layout (CSS properties)
<i>Declarative</i> support for object animation	YES	NO
<i>Declarative</i> support for transition effects	YES	YES
<i>Declarative</i> support for live editing	Yes, through DSM-CC events	NO
Support to Lua objects	YES	NO
Support to Java objects	YES	NO
Support to ECMAScript	Optional, embedded in HTML documents	YES
Bridge with imperative environment	YES	NO
Support to HTML	YES (required at least elements as in BML for Basic services - "fixed terminal profile")	YES (required elements depend on the language profile)
Support to CSS	YES (required at least properties as in BML for Basic services - "fixed terminal profile")	YES (required properties depend on the language profile)
Support to DOM-HTML	Optional	YES (required properties depend on the language profile)
Stream events	Ginga editing commands	Broadcast events (bevent)
Storage, management, and retrieval of the presentation status control	YES, both through NCL and through Lua scripts	YES, by use of NVRAM with ECMAScript

7 Commonalities and Differences in the data transmission specification

The common parts and differences of ABNT NBR 15606-3:2009 and ARIB STD-B24:2005 and how they correspond are described in Table 3.

Table 3— Correspondence between ABNT NBR 15606-3:2009 and ARIB STD-B24 V5.32009 standards

Description	ABNT NBR15606-3	ARIB STD-B24
Object Carousel	YES	NO (Defined only Data Carousel)
Data Carousel	YES	YES
DII Message	YES	YES
DDB Message	YES	YES
Descriptor of Module information area and private area (ISO)	YES (ISO/IEC 13818-6)	YES (But, there are some specifics ARIB descriptors)
Event message transmission protocol	YES	YES
Stream Descriptor	YES	YES

8 Commonalities and Differences in the Java based multimedia coding scheme

8.1 Commonalities between ARIB-J and Ginga-J

ARIB-J and Ginga-J are Java based procedural interactive content format. They share the same core functionalities and technologies defined and harmonized by Recommendation ITU-T J.202. The latest revision of Rec. ITU-T J.202 considers JavaDTV as parts of the core functionalities recommended for all middleware execution engines, on the same level as DVB-GEM.

The application environment for both ARIB-J and Ginga-J formats employs JavaME, Connected Device Configuration (CDC), Foundation Profile (FP), Personal Basis Profile (PBP) and JavaTV 1.1. The application model of both formats is similar, including stored application and unbounded application.

Both standards also define APIs to provide to the interactive content the following functionalities:

- Application Management and Application Catalogue access API
- DSMCC control and Stream Events Support API
- Section Filter API
- Rich graphical components library
- Media control extensions for the subset of the JMF framework included in JavaTV 1.1
- Resource Management
- Protocol dependent Service Information API
- Smart Card access API
- High level access to tuner devices
- API for setting or querying screen configuration parameters (resolution, aspect ratio, pixel depth, etc)
- Support for downloadable fonts
- Handling of remote control keys and other kind of inputs

8.2 Differences between ARIB-J and Ginga-J

ARIB-J is an extension to DVB-GEM that includes a considerable number of DAVIC, DVB and HAVi defined APIs. Ginga-J is JavaDTV based content format that provides functional equivalents to DAVIC, DVB and HAVi defined APIs. UI functionalities defined in JavaDTV are based on LWUIT API.

ARIB-J defines PVR APIs that allows the application to control recording and playback of broadcast program.

Ginga-J extends several basic core JavaDTV APIs to:

- Allow access to Screen Planes (Still Picture Plane, Still/Video Switching Plane) other than the Text and Graphics Plane or the Video Plane
- Enhanced tuning control over multiple tuner devices
- Ginga-J defines APIs to allow bi-lateral communication between the presentation engine (Ginga-NCL) and execution engine (Ginga-J).

Table 4— Correspondence between ABNT NBR 15606-4:2010 and ARIB STD-B23 V1.2 standards

Features	ABNT NBR 15606-4	ARIB STD-B23
Java Platform and Java General Functionalities		
Java Core	Connected Device Configuration (CDC) 1.1 Foundation Profile (FP) 1.1 Personal Basis Profile (PBP) 1.1	Connected Device Configuration (CDC) 1.1 Foundation Profile (FP) 1.1 Personal Basis Profile (PBP) 1.1
Broadcast specific functionalities		
Broadcast specific Core functionalities	JavaTV 1.1	JavaTV 1.1
Broadcast specific Extended functionalities	<i>JavaDTV 1.3</i>	<i>DAVIC and DVB API defined in DVB-GEM 1.2</i>
Protocol-independent SI	JavaTV 1.1	JavaTV 1.1
Protocol-dependent SI	<i>ABNT-NBR 15606-4 (br.org.sbtvd.net, br.org.sbtvd.net.si) compatible with STD-B10</i>	<i>ARIB-J (jp.or.arib.tv.si)</i>
Media Control Core	JavaTV 1.1 (javax.tv.media.* javax.media.*)	JavaTV 1.1 (javax.tv.media.* javax.media.*)
Media Control Extensions	<i>JavaDTV 1.3</i>	<i>DVB-GEM 1.2</i>
Tuning control and Service Selection	JavaTV 1.1 (high level service selection API)	JavaTV 1.1 (high level service selection API)
	<i>ABNT-NBR 15606-4 (br.org.sbtvd.rc)</i>	<i>ARIB-J (jp.or.arib.tv.net.tuning)</i>
DSMCC control and Stream Events Support API	<i>JavaDTV 1.3</i>	<i>DVB-GEM 1.2</i>
Service Selection Scheduling	<i>Not supported</i>	<i>ARIB-J (jp.or.arib.tv.service.selection)</i>
Resource Management	<i>JavaDTV 1.3</i>	<i>DAVIC API defined in DVB-GEM 1.2</i>

Table 4— Continuation

User Interface		
Basic Graphics Functionalities	Based on AWT from PBP1.1	Based on AWT from PBP1.1
Graphics Components Library	<i>Based on specific LightWeight User Interface (LWUIT) framework customized for DigitalTV and defined as part of JavaDTV 1.3</i>	<i>HAVi UI level2 defined in DVB-GEM1.2</i>
Other UI extension packages	<i>ABNT-NBR 15606-4 (com.sun.dtv.ui.event, com.sun.dtv.ui.event)</i>	<i>ARIB-J (jp.or.arib.tv.ui)</i>
Screen Planes	<i>Based on ARIB-B23 5 planes mode ABNT-NBR 15606-4 (br.org.sbtvd.ui) Support for Still Picture and Still Picture and Video Switching Plane</i>	<i>Based on DVB-GEM 1.2 3 planes model</i>
Other Functionalities		
Application Management and Application Catalogue access API	<i>JavaDTV 1.3</i>	<i>DVB-GEM 1.2</i>
Return channel with asynchronous messaging	<i>ABNT 15606-4 (br.org.sbtvd.rc)</i>	<i>Not supported</i>
SmartCard access	<i>API for J2ME (SATSA) 1.0.1</i>	<i>DAVIC API</i>
Bridge with Presentation Engine	<i>ABNT 15606-4 (br.org.sbtvd.bridge)</i>	<i>Not supported</i>
Device integration	<i>Ginga-J's Profile C Draft (br.org.sbtvd.interactiondevices)</i>	<i>ARIB-J (jp.or.arib.tv.peripheral, jp.or.arib.tv.peripheral.protocol, jp.or.arib.tv.perpheral.stream)</i>
PVR API	<i>Not supported</i>	<i>GEM-PVR (ETSI TS 102 817)</i>
Java Security Extensions	<i>Java Cryptographic Extensions 1.0.1 Security and Trust Services API 1.0.1 for Smartcard access</i>	<i>DVB-GEM 1.2 (Some are localized by ARIB)</i>
Return Channel Security	<i>Java Sockets Security Extensions 1.0.1</i>	<i>Secure http (https) is supported</i>