



ARIB TR-B14  
Version 6.7-E1

## **ENGLISH TRANSLATION**

OPERATIONAL GUIDELINES FOR  
DIGITAL TERRESTRIAL TELEVISION BROADCASTING

# **ARIB TECHNICAL REPORT**

ARIB TR-B14 Version 6.7  
(Fascicle 2)

Established on January 24th, 2002

Revised on September 28th, 2020 Version 6.7

**Association of Radio Industries and Business**

## General Notes to the English Translation of ARIB Standards and Technical Reports

### 1. Notes on Copyright

- The copyright of this document is ascribed to the Association of Radio Industries and Businesses (ARIB).
- All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, without the prior written permission of ARIB.

### 2. Notes on English Translation

- ARIB Standards and Technical Reports are usually written in Japanese. This document is a translation into English of the original document for the purpose of convenience of users. If there are any discrepancies in the content, expressions, etc. between the original document and this translated document, the original document shall prevail.
- ARIB Standards and Technical Reports, in the original language, are made publicly available through web posting. The original document of this translation may have been further revised and therefore users are encouraged to check the latest version at an appropriate page under the following URL:  
<https://www.arib.or.jp/english/index.html>.

## **Foreword**

The Association of Radio Industries and Businesses (ARIB) investigates and summarizes the basic technical requirements for various radio systems in the form of “ARIB Standards”. These standards are developed with the participation of and through discussions amongst radio equipment manufacturers, telecommunication operators, broadcasting equipment manufacturers, broadcasters and users.

ARIB Technical Reports contain the concrete measurement methods, detailed explanation and remarks in respect to the operation and maintenance of the radio equipment and broadcasting equipment in order to ensure their compatibility and adequate quality, based on the ARIB Standards deriving from “governmental technical regulations” (mandatory standard) and “private technical standards” (voluntary standards).

This ARIB Technical Report is developed for the operation at digital terrestrial television broadcasting stations and the functional specifications for digital terrestrial television receiver units. In order to ensure fairness and transparency in the defining stage, the report was set by consensus at the ARIB Standard Assembly with the participation of both domestic and foreign interested parties from radio equipment manufacturers, telecommunication operators, broadcasting equipment manufacturers, broadcasters and users.

ARIB sincerely hopes that this ARIB Technical Report will be widely used by radio equipment manufacturers, telecommunication operators, broadcasting equipment manufacturers, broadcasters and users.



## Integrated Contents

Foreword

Digital Terrestrial Television Broadcasting

General Information of the Technical Report ..... Fascicle 1

Volume 1 Digital Terrestrial Television Broadcasting

Provisions for Download Operation ..... Fascicle 1

Volume 2 Digital Terrestrial Television Broadcasting

Function Specifications for the Receiver..... Fascicle 1

Foreword

Volume 3 Digital Terrestrial Television Broadcasting

Specifications for Data Broadcasting Operations(Part 1) ..... Fascicle 2

Foreword

Volume 3 Digital Terrestrial Television Broadcasting

Specifications for Data Broadcasting Operations(Part 2) ..... Fascicle 3

Foreword

Volume 4 Digital Terrestrial Television Broadcasting

Provisions for PSI/SI Operations ..... Fascicle 4

Foreword

Volume 5 Digital Terrestrial Television Broadcasting

Provisions for Conditional Access System (CAS)

Operations and Specifications on Receiver Units..... Fascicle 5

Volume 6 Digital Terrestrial Television Broadcasting

Provisions for Interactive Data Broadcasting Services ..... Fascicle 5

Volume 7 Digital Terrestrial Television Broadcasting

Provisions for Carrier Operations..... Fascicle 5

Volume 8 Digital Terrestrial Television Broadcasting

Provisions for Contents Protection..... Fascicle 5

Volume 9 Digital Terrestrial Television Broadcasting

Provisions for Transmission Operations ..... Fascicle 5



Vol. 3

DIGITAL TERRESTRIAL TELEVISION  
BROADCASTING

Specifications for Data Broadcasting Operations  
(Part 1)





## Contents

[Section 1] Common specifications for different types of reception.....	1
1 Introduction .....	1
2 References .....	1
3 Definitions .....	2
4 Type of operational specifications specified in Vol. 3 .....	12
4.1 C-profile corresponding to different reception types .....	12
4.2 Relationship between assumed receiver units and each profile.....	12
4.3 Basic functions and optional functions.....	12
[Section 2] Operational provisions related to Profile A.....	13
1 Functions that basic receiver units should be equipped with to receive data broadcasting .....	13
1.1 Components of receiver units .....	13
1.1.1 Hardware components.....	13
1.1.2 Receiver unit reference model.....	14
1.2 Presentation functions .....	17
1.2.1 Pixel size/restrictions of each plane composing the display screen .....	17
1.2.2 Combination and restrictions of presentable planes .....	20
1.2.3 Relationship between mono-media encoding and the presentation plane.....	23
1.2.4 Audio playing function.....	27
1.2.5 Fonts .....	28
1.3 Remote control .....	29
1.3.1 Keys used in data broadcasting .....	29
1.3.2 Key masks.....	29
1.4 Memory that should be installed in receiver units.....	29
1.4.1 RAM .....	30
1.4.2 NVRAM .....	30
1.5 Communication functions .....	31
1.6 Character entry function .....	31
1.6.1 Function specifications.....	31
1.6.2 Character type.....	34
1.6.3 Kana Kanji conversion function .....	35

1.7	Bookmark list display function by receiver's native application (Optional).....	36
1.7.1	Start/End bookmark list by receiver's native application.....	36
1.7.2	Bookmark list functions to be equipped by receiver's native applications.....	37
1.7.2.1	Display function of recorded bookmark titles.....	37
1.7.2.2	Functions that allow viewers to select desired bookmarks from a list and to select stations in accordance with specifications of the link destination URI.....	38
1.7.2.3	Delete management function.....	39
1.7.3	Startup of the bookmark list by receiver's native application by function.....	40
1.8	BML browser.....	40
1.8.1	Display by browser features.....	40
1.9	Registration transmission function (Optional).....	41
2	Operation of the data transmission method.....	43
2.1	PSI/SI.....	43
2.1.1	Types of data broadcasting services.....	43
2.1.1.1	Data programs and TV programs.....	43
2.1.1.2	Types of data broadcasting service programs.....	43
2.1.1.3	"service_type" of channels that operate data broadcasting programs.....	44
2.1.2	Configuration of contents of data broadcasting service and operation of components ....	44
2.1.2.1	Contents and local contents.....	44
2.1.2.2	Relationship between local contents and ES.....	45
2.1.2.3	Operation of component tags.....	45
2.1.2.4	Identification of entry components.....	45
2.1.2.5	Restrictions on the number of ES.....	46
2.1.2.6	Detailed specifications of section data transmission.....	46
2.1.2.7	Default maximum bit rate of data broadcasting programs.....	46
2.1.2.8	Operation of video and audio components in data broadcasting services.....	46
2.1.3	Series reservation operation in data broadcasting services.....	48
2.1.4	Operation of the PMT specific to data broadcasting services.....	48
2.1.5	Operation of the Data Component Descriptor in the PMT.....	48
2.1.6	Operation of the Target Resion Descriptor in the PMT.....	49
2.1.7	Operation of the H-EIT, M-EIT Data Contents Descriptor.....	49
2.1.8	Operation of the Data Contents Descriptors in H-EIT[schedule],M-EIT[p/f after].....	52
2.1.9	Operation of EIT's specific to data broadcasting services.....	52
2.1.10	Performance of related receiver.....	52
2.1.10.1	BML-engine startup.....	52

2.1.10.2	Receiver operation when data broadcasting programs start .....	53
2.1.10.3	Receiver behaviour when updating the PMT .....	54
2.1.10.4	Handling of the dButton .....	56
2.1.10.5	Pixel size and aspect control in data broadcasting programs.....	57
2.1.10.6	Receiver operation when selecting stations .....	57
2.1.10.7	Reservation related behaviors of data broadcasting programs(Guidelines).....	58
2.1.10.8	Specifications when the partial transport stream is output/input.....	59
2.1.10.9	the desired display of EPG, etc. ....	59
2.2	Operation of the independent PES transmission method.....	59
2.3	Data carousel transmission method/event message transmission method.....	60
2.3.1	Data carousel transmission operation .....	60
2.3.1.1	Introducon of data events and local contents .....	60
2.3.1.2	Operation of data events .....	60
2.3.1.3	Start/End of local contents.....	60
2.3.1.4	Introduction of the return flag.....	61
2.3.1.5	Local contents and the Data Contents Descriptor .....	61
2.3.1.6	Operation of empty carousels.....	63
2.3.1.7	Basic behavior of receiver during data broadcasting program presentation.....	63
2.3.1.8	Transmission of root certificates .....	65
2.3.1.9	Transmission operation related to broadcasting integrated HTML5 applications (phase 1 [see Chapter 8]) .....	67
2.3.2	Operation of “DownloadInfoIndication”(DII)message.....	67
2.3.3	Operation of “DownloadDataBlock”(DDB)message .....	70
2.3.4	Operation of event messages.....	70
2.3.4.1	Purpose for operating event messages .....	70
2.3.4.2	Transmission of event messages .....	71
2.3.4.3	Transmission of general event messages .....	71
2.3.4.4	Transmission of the NPT reference message .....	74
2.3.4.5	Event message process in receiver.....	76
2.3.4.6	Operation of the DSMCC_section( ).....	80
2.3.4.7	Operation of General Event Descriptor.....	80
2.3.4.8	Operation of the NPT Reference Descriptor.....	80
2.3.5	Operation of IIT .....	81
2.3.6	Performance of related receiver .....	81
2.3.6.1	Filtering resources used when receiving data broadcasting.....	81
2.4	Temporary services and data broadcasting services .....	81

2.5	Multi-view operations and data broadcasting services .....	82
2.6	Interaction channel transmission protocol used for data broadcasting services.....	82
3	Operation of mono-media encoding .....	83
3.1	Video encoding .....	83
3.1.1	MPEG-1 Video .....	83
3.1.1.1	Restrictions of encoding parameters .....	83
3.1.1.2	Synchronized playback with audio (MPEG-2 AAC) .....	83
3.1.1.3	Other restrictions .....	83
3.1.2	MPEG-2 Video .....	83
3.1.2.1	Restrictions of encoding parameters .....	84
3.1.2.2	Other restrictions .....	86
3.1.3	MPEG-4 Visual.....	86
3.2	Still images and bitmap figure encoding .....	86
3.2.1	JPEG .....	86
3.2.1.1	Encoding parameters.....	86
3.2.1.2	Scaling .....	87
3.2.1.3	Other restrictions .....	87
3.2.1.4	Operated markers and marker segments .....	87
3.2.2	PNG.....	88
3.2.2.1	Encoding parameters.....	88
3.2.2.2	Chunks operated by PNG .....	88
3.2.2.3	Other restrictions .....	88
3.2.3	MNG .....	88
3.2.3.1	Chunks operated by the MNG .....	88
3.2.3.2	Restrictions of MNG operation.....	89
3.3	Audio encoding .....	90
3.3.1	MPEG-2 AAC.....	90
3.3.1.1	Encoding parameters.....	90
3.3.1.2	Transmission of MPEG-2 AAC .....	90
3.3.1.3	Restrictions in data carousel distribution .....	90
3.3.1.4	Data format of the AAC audio file .....	90
3.3.2	AIFF-C.....	91
3.3.2.1	Encoding parameters.....	91
3.3.2.2	Maximum data volume.....	91
3.3.2.3	Other restrictions .....	91

3.3.3	MPEG-4 Audio .....	91
3.3.4	Additional sounds.....	91
3.3.5	Built-in sound .....	91
3.3.6	Audio synthesis by receivers .....	92
3.3.6.1	Mixing balance.....	92
3.3.6.2	Simultaneous playable encoding method .....	92
3.3.6.3	Audio when the receiver uses a broadcasting integrated HTML5 application .....	93
3.4	Character encoding .....	93
3.4.1	8-bit character codes (including EUC-JP) .....	93
3.4.1.1	Restrictions on the character coding function.....	94
3.4.1.2	Character sets used in data broadcasting .....	99
3.4.1.3	Initialization of character codes (optional) .....	100
3.4.2	Universal multi-octet coded Character Set.....	100
3.5	Description command figure encoding .....	100
3.5.1	Geometric.....	100
4	Operation of caption and superimpose encoding .....	101
4.1	Scope of service and definitions.....	101
4.2	Composition and transmission operation.....	101
4.2.1	Specification for composition and transmission .....	101
4.2.2	PES transmission methods used in caption.....	103
4.2.3	PES transmission methods used in superimpose.....	104
4.2.4	Operation of data groups .....	105
4.2.5	Operation of caption management data .....	106
4.2.5.1	Caption management data used in caption .....	106
4.2.5.2	Caption management data used in superimpose .....	107
4.2.6	Operation of caption statement data .....	108
4.2.7	Operation of data units.....	109
4.2.8	OPERATION OF PSI/SI .....	109
4.2.8.1	Operation of component tags .....	109
4.2.8.2	Operation of the PMT .....	109
4.2.8.3	Stream format identification .....	109
4.2.8.4	Descriptor operation .....	109
4.2.8.5	Data Component Descriptor .....	110
4.2.8.6	Target Region Descriptor.....	110
4.2.8.7	Data Contents Descriptor .....	110

4.3	Visual pixel size and display format of caption/superimpose.....	111
4.3.1	Display format .....	111
4.3.2	Display area .....	111
4.3.3	Initial operation location .....	112
4.4	Characters used in caption/superimpose .....	112
4.4.1	Character codes.....	112
4.4.2	Character fonts.....	112
4.4.3	Character size controls .....	113
4.4.4	Display zone.....	115
4.4.5	Non-spacing characters .....	125
4.5	Control codes used in caption/superimpose .....	125
4.5.1	Control codes .....	125
4.5.1.1	Operation of color specifications.....	130
4.5.2	Operation of flashing.....	135
4.5.2.1	Restricted matters.....	135
4.5.3	Raster color control .....	136
4.5.4	Operation of highlighting .....	136
4.5.5	Operation of underlines .....	136
4.5.6	Operation of outlines.....	137
4.5.7	Operation of scrolling.....	137
4.5.7.1	One line lateral direction scroll.....	137
4.5.7.2	Control codes.....	138
4.5.7.3	Scrolling speed .....	138
4.5.8	Priorities of the display function .....	138
4.6	Operation of the DRCS.....	138
4.7	Operation of initialization .....	139
4.7.1	Initialization by caption management .....	139
4.7.2	Initialization by caption statements.....	139
4.7.3	Initialization by data units in this document.....	139
4.7.4	Initialization by character control codes .....	140
4.8	Mono-media used in caption and superimpose .....	140
4.8.1	Operation of geometric .....	140
4.8.2	Operation of bit map data .....	140
4.8.3	Operation of alert sounds .....	140
4.8.4	Operation of additional sound.....	140
4.9	Expected performance of the receiver units .....	140

4.9.1	Start/end of caption display .....	141
4.9.2	Start/end of superimpose display .....	141
4.9.3	Set-up fields etc. in the receiver units.....	141
4.10	Roll-up mode (Optional).....	142
4.10.1	Declaration of the roll-up mode .....	142
4.10.2	Operation of caption management data in the roll-up mode.....	142
4.10.3	Operation of caption statement data in the roll-up mode.....	142
4.10.3.1	Operation of the text data unit of caption statement data .....	142
4.10.4	Initialization of the roll-up mode .....	142
4.10.5	Presentation of the roll-up mode .....	143
4.10.6	Restrictions of PES packet transmission of roll-up mode .....	143
4.10.7	Compatibility with receiver units not equipped the roll-up mode .....	143
4.11	Caption outscreen display function (Optional).....	144
5	Operation of multimedia encoding .....	145
5.1	Introduction.....	145
5.2	Operation of NVRAM used commonly in Multimedia services in Digital Terrestrial Television Broadcasting.....	145
5.2.1	Allocation of NVRAM area .....	146
5.2.1.1	Allocation of the A-profile memory area for the affiliation .....	147
5.2.1.2	Allocation of the A-profile memory area for the specified broadcaster and the A-profile memory area of communication purpose for the specified broadcaster .....	147
5.2.2	Identification of the A-profile memory area for the all broadcasters .....	148
5.2.3	Identification of the broadcaster affiliates area.....	148
5.2.4	Identification of the A-profile memory area for the specified broadcaster .....	149
5.2.5	Identification of the A-profile memory area of communication purpose for the specified broadcaster .....	149
5.2.6	Identification of the memory area for bookmark service .....	149
5.2.7	Use of the viewer residential area information from the Multimedia service .....	150
5.2.8	Operation of the A-profile memory area for the all broadcasters of receiver unit NVRAM .....	151
5.2.8.1	Purpose of the A-profile memory area for the all broadcasters .....	151
5.2.8.2	Format of the A-profile memory area for the all broadcasters .....	151
5.2.9	Specification relating to writing in the A-profile memory area for the all broadcasters....	158
5.2.10	Specification relating to reading from the A-profile memory area for the all broadcasters .....	159

5.2.11	Specification regarding customer registration, changed contents .....	159
5.2.12	Specification when registering customer information to the center server .....	160
5.2.13	Contents description guidelines of NVRAM access .....	161
5.3	Use of remote control keys from Multimedia services .....	161
5.3.1	Values that used-key-list specification could take .....	161
5.3.2	Handling of remote control keys, key codes, and access keys .....	162
5.3.3	Guidelines of contents using selection by color keys.....	163
5.4	Operation of the BML version .....	163
5.5	Operation of character codes .....	163
5.5.1	Transmission of DRCS pattern data.....	163
5.6	Operation area of media type and mono-media .....	163
5.7	Operation of BML elements .....	164
5.7.1	Entity .....	164
5.7.2	Operational guidelines of BML elements.....	164
5.7.2.1	Restrictions on the order of elements within head elements .....	167
5.7.3	Attributes .....	167
5.7.4	Operation relating to beitem elements .....	171
5.7.5	Operation of the BML elemnts extension module (interruption event) .....	171
5.8	Operation of CSS.....	173
5.9	Operation guidelines relating to procedure description .....	173
5.9.1	Operation area of DOM .....	173
5.9.2	Operation area of embedded objects .....	173
5.9.3	Operation area of extended objects for broadcasting .....	173
5.9.4	Operation area of browser pseudo-objects .....	173
5.9.5	Operation range of navigator pseudo-objects .....	178
5.9.6	Extended functions that have been added in the digital terrestrial broadcasting .....	178
5.9.6.1	Print function (Optional).....	178
5.9.6.2	Receiver application identification function (Optional).....	178
5.9.6.3	Function to acquire root CA certificate information.....	178
5.9.6.4	Cache set-up function of communication contents (Optional) .....	179
5.9.6.5	Receiver’s native application startup function (Optional).....	179
5.9.6.6	Specifiable values in arguments of “getBrowserSupport( )” .....	179
5.10	Restrictions in the BML document description.....	179
5.11	Presentation control of BML document.....	179
5.12	Behavior of BML browsers.....	180
5.12.1	Operational guidelines relating to presentation .....	180



5.12.2	Operational guidelines relating to the performance of external characters .....	181
5.12.3	Behavior of the DOM .....	181
5.12.4	Operation of script language.....	181
5.12.5	Extended objects for broadcasting .....	181
5.12.6	Browser pseudo-objects .....	182
5.12.6.1	Operation of Ureg .....	182
5.12.6.2	Operation of Greg .....	182
5.12.6.3	Operation of EPG functions .....	182
5.12.6.4	Operation of series reservation function .....	183
5.12.6.5	Operation of the NVRAM functions.....	184
5.12.6.6	Interaction channel function-Operation of TCP/IP .....	184
5.12.6.7	Interaction channel function-Operation of the function that acquires the line connection status .....	184
5.12.6.8	Interaction channel function-Operation of large volume call acceptance service .....	184
5.12.6.9	Operation of operational control function.....	184
5.12.6.10	Operation of bookmark function.....	200
5.12.6.11	Operation of IPTV linkage function.....	201
5.12.6.12	Operation of inquiry function .....	201
5.12.6.13	Operation of the AIT-controlled application linkage function .....	201
5.12.7	Embedded objects .....	201
5.12.8	Other specifications .....	201
5.13	Transmission of contents and name spaces .....	202
5.13.1	Mapping of scope to transmission systems .....	202
5.13.2	Restrictions related to reference of mono-media, etc. over media .....	202
5.13.3	Operation of name spaces.....	202
5.13.4	Operation relating to communication contents.....	204
5.14	Operation of communication contents .....	204
5.14.1	Operation guidelines relating to the presentation of communication contents.....	204
5.14.2	Operation guidelines relating to the performance of external characters of communication contents.....	204
5.14.3	Performance of DOM of communication contents .....	204
5.14.4	Operation of ECMA script execution for the script language of communication contents .....	205
5.14.5	Expansion object for the broadcasting of communication contents.....	205
5.14.6	Operation of browser pseudo-objects of communication contents.....	205
5.14.6.1	Ureg related functions.....	205

5.14.6.2	EPG functions.....	205
5.14.6.3	Series reservation function .....	206
5.14.6.4	NVRAM functions .....	206
5.14.6.5	Interaction channel function-Basic Procedure .....	207
5.14.6.6	Interaction channel function-TCP/IP .....	207
5.14.6.7	Interaction channel function/Function to acquire line connection status.....	213
5.14.6.8	Interaction channel function-Mass calls reception service.....	213
5.14.6.9	Operational control function.....	214
5.14.6.10	Receiver unit audio control .....	218
5.14.6.11	Timer function .....	219
5.14.6.12	External character function .....	219
5.14.6.13	Other functions .....	220
5.14.6.14	Closed caption display control function .....	220
5.14.6.15	Bookmark control function .....	220
5.14.6.16	Print related functions .....	221
5.14.6.17	IPTV linkage function.....	221
5.14.6.18	Inquiry function .....	222
5.14.6.19	AIT-controlled application linkage function .....	222
5.14.7	Mapping the scope of communication contents .....	222
5.14.8	Operation guideline for transmission of communication contents .....	225
5.14.9	Restrictions related to mono-media reference from communication contents .....	231
5.14.9.1	Reference from broadcasting contents to communication contents .....	231
5.14.9.2	Reference from communication contents to broadcasting contents .....	232
5.14.10.	Name space of communication contents .....	232
5.14.10.1	Restricted matters relating to URI.....	232
5.14.10.2	Operation of name spaces for servers.....	232
5.14.10.3	Operation of name spaces when referring to communication contents from broadcasting contents.....	233
5.14.10.4	Operation of name spaces when referring to broadcasting contents from communication contents .....	233
5.14.11	Guidelines of communication contents operation .....	234
5.14.11.1	Precautions during the operation of communication contents .....	234
5.14.11.2	Guidelines of communication contents descriptions.....	235
5.14.12	Guidelines of receiver unit performance when receiving communication contents .....	237
5.14.12.1	Guidelines of the receiver unit status transition performance.....	237
5.14.12.2	Guidelines for when receiver unit performance fails.....	242

5.14.12.3	Guidelines of receiver unit performance when receiving error responses .....	242
5.14.13	Guidelines for connecting/disconnecting .....	242
5.14.13.1	Receiver performance guidelines for disconnecting and re-connecting .....	242
5.14.13.2	Receiver unit performance guidelines when connecting by scripts .....	245
5.14.13.3	Receiver unit performance guidelines for auto-connection.....	245
5.14.13.4	Receiver unit performance guidelines when disconnected.....	246
5.14.13.5	Connection guidelines from receiver's native application .....	247
5.14.13.6	Display guidelines of confirmation messages when connecting or sending data .....	247
5.14.14	Operation of root CA certificates .....	248
5.14.14.1	Operation of the memory area for root CA certificates .....	248
5.14.14.2	Storage of root CA certificates of general purpose to receiver units .....	249
5.14.14.3	Guidelines of receiver unit behavior for updating the root CA certificate of general purpose .....	250
5.14.14.4	Guidelines of receiver unit behavior when receiving root CA certificates of limited purpose by broadcasters .....	252
5.15	Operation of the bookmark service.....	253
5.15.1	Service type of the bookmark list service .....	253
5.15.2	Operation of the pre-list .....	253
5.15.3	Registration by bookmark type .....	254
5.15.4	Bookmark list display function by receiver's native application (Optional) .....	254
5.15.5	General address definitions relating to the bookmark service .....	254
5.15.6	Operation of the expansion data section of the memory area for bookmark service .....	254
5.16	Operation of registration transmission .....	254
5.16.1	Operation of the memory area for registration transmissions.....	255
5.16.2	Identification of the memory area for registration transmission .....	255
5.16.3	Operation of the bookmark function for registration transmission .....	255
5.16.3.1	Operation of the "writeBookmarkArray" function in registration transmission.....	255
5.16.3.2	Operation of the "readBookmarkArray" function in registration transmission .....	256
5.16.3.3	Operation of the "deleteBookmarkArray" function in registration transmission.....	257
5.16.3.4	Operation of the "lockBookmarkArray" function in registration transmission .....	257
5.16.3.5	Operation of the "unlockBookmarkArray" function in registration transmission .....	257
5.16.3.6	Operation of the "getBookmarkInfo2" function in registration transmission .....	257
5.16.3.7	Format of registration transmission information.....	257
5.16.3.8	Basic data section of registration transmission information.....	257
5.16.3.9	Expansion data section of registration transmission information.....	259
5.16.3.10	Operation of each field of registration information .....	261

5.16.4	Guidelines of registration transmission operation by contents .....	264
5.16.4.1	Operation when registration transmission is set up .....	264
5.16.4.2	Operation of registration transmission by contents.....	265
5.16.5	Registration transmission using registration transmission information by receiver's native application (Optional).....	266
5.16.5.1	Registration of registration transmission information.....	266
5.16.5.2	Operation of registration transmission by receiver's native application .....	266
5.16.5.3	Reservation transmission function by receiver features .....	267
5.16.5.4	Guidelines for sending messages by receiver's native application.....	268
6	Operation of extended service (Optional) .....	270
6.1	Display of HTML contents (Optional).....	270
6.2	Print function.....	270
6.2.1	Expansion API group .....	270
6.2.2	Print data format.....	271
6.2.3	Supplementary information regarding each print related API .....	272
6.2.4	Regarding presentation by receiver.....	273
6.3	Broadcasting integrated IPTV service type 1 (optional).....	274
6.3.1	Activation and display of the IPTV browser.....	274
6.3.2	Returning to BML browser from IPTV browser .....	275
6.3.3	Confirmation of the IPTV browser function from BML content .....	275
6.3.4	Return values of the getResidentAppVersion() function.....	275
6.3.5	Positioning of the IPTV activation function startExtraBrowser() .....	275
6.4	Broadcasting integrated IPTV service type 2 (optional).....	276
6.5	External device linkage function (optional) .....	276
6.5.1	Service profile and the operation.....	276
6.6	AIT-controlled application linkage function (optional) .....	280
6.7	AIT-controlled application external activation function (optional) .....	280
6.7.1	Operation at the time of AIT-controlled application external activation .....	281
6.7.2	Operation at the time of termination of AIT-controlled application with external activation .....	281
7	IPTV download/VOD (integrated service type 2) function .....	282
7.1	Overview of the IPTV function used in BML documents.....	282
7.1.1	Overview of the IPTV download/VOD (integrated service type 2) function .....	282
7.1.1.1	IPTV download function (integrated service type 2) .....	282
7.1.1.2	VOD function (integrated service type 2) .....	283

7.1.2	Internet scope and CDN scope .....	283
7.2	IPTV download/VOD (integrated service type 2) function in the Internet scope.....	283
7.2.1	Specification range .....	283
7.2.2	Functions to be implemented in receivers that support the IPTV download/VOD (integrated service type 2).....	283
7.2.2.1	Classification of the functions of receivers that support the IPTV download/VOD (integrated service type 2) .....	284
7.2.2.2	IPTV download/VOD common functions.....	284
7.2.2.3	IPTV download (integrated service type 2) function .....	285
7.2.2.4	VOD (integrated service type 2) function .....	287
7.2.3	Operation of the BML browser in receivers with the IPTV download/VOD (integrated service type 2) function .....	288
7.2.3.1	Implementation of the IPTV download/VOD (integrated service type 2) function .....	288
7.2.3.2	Operation of browser pseudo-objects related to the IPTV download/VOD (integrated service type 2) function.....	289
7.2.3.3	Disabling the return flag .....	292
7.2.3.4	Identification of IPTV download/VOD (integrated service type 2) function (contents guidelines).....	292
7.2.4	Operation of IPTV download (integrated service type 2) function .....	292
7.2.4.1	Execution sequence of IPTV download .....	292
7.2.4.2	Operation of browser pseudo-objects involving IPTV download.....	293
7.2.4.3	About download control information.....	294
7.2.4.4	Operation of BML contents for the IPTV download function .....	294
7.2.5	Operation of VOD (integrated service type 2).....	295
7.2.5.1	Operation of browser pseudo-objects involving VOD (integrated service type 2).....	295
7.2.5.2	Execution sequence to activate the VOD function .....	297
7.2.5.3	Specifications for the operation when starting the VOD function.....	297
7.2.5.4	Specifications for the operation while replaying VOD contents .....	297
7.2.5.5	Sequence to terminate the VOD function .....	298
7.2.5.6	Specifications for the operation when the VOD function is completed .....	300
7.2.5.7	Operation of BML contents for VOD (integrated service type 2) services.....	300
7.3	IPTV download/VOD (integrated service type 2) function for CDN scope services.....	303
8	AIT-controlled application .....	304
8.1	Receiver functions .....	304
8.1.1	Receiver model .....	304

8.1.2	Communication function .....	305
8.1.3	Media type operation range <phase 1>.....	305
8.2.1	Operation of PMT.....	305
8.2.1.1	Operation of activation priority information of the PMT.....	305
8.2.1.2	Operation of the data component descriptor <phase 1>.....	306
8.2.1.3	Operation of the PMT when transmitting an AIT as a section.....	306
8.3.1	Description format of AIT .....	306
8.3.2	AIT transmission operation.....	306
8.3.2.1	Operation of transmitting an XML-format AIT via communication.....	307
8.3.2.2	Operation of transmitting an AIT using a data carousel <phase 1>.....	307
8.3.2.3	Operation of transmitting an AIT using a section .....	309
8.3.3	Operation of an AIT that is transmitted via communication .....	310
8.3.3.1	Operation of an XML-format AIT that is transmitted via communication.....	310
8.3.3.2	Character coding scheme for an AIT that is transmitted via communication .....	315
8.3.3.3	Operation of XML declaration and name space.....	315
8.3.3.4	Operation of permissionBitmap for an AIT that is transmitted via communication.....	315
8.3.4	Operation of a section-format AIT that is transmitted using a data carousel <phase 1>.....	316
8.3.4.1	Operation of application identifier .....	317
8.3.4.2	Operation of application descriptor .....	317
8.3.4.3	Operation of transport protocol descriptor .....	318
8.3.4.4	Operation of simple application location descriptor .....	318
8.3.4.5	Operation of application boundary and permission descriptor.....	319
8.3.4.6	Operation of autostart priority descriptor.....	320
8.3.4.7	Operation of cache control info descriptor .....	321
8.3.4.8	Operation of randomized latency descriptor .....	321
8.3.4.9	Operation of external application control descriptor.....	321
8.3.4.10	Operation of playback application descriptor .....	321
8.3.4.11	Operation of simple playback application location descriptor .....	321
8.3.4.12	Operation of application expiration descriptor.....	321
8.3.4.13	Operation of the lengths of character strings for an AIT .....	321
8.3.5	AIT Monitoring.....	322
8.3.5.1	Monitoring an AIT that is transmitted via communication .....	322
8.3.5.2	Monitoring an AIT that is transmitted using a data carousel <phase 1>.....	322
8.3.5.3	Monitoring an AIT that is transmitted using a section .....	325
8.4	Operation for transmitting the broadcasting integrated HTML5 application using a data carousel <phase 1> .....	325

8.4.1	Operation for transmitting auto start applications.....	326
8.4.1.1	Component to transmit the auto start application .....	326
8.4.1.2	Name of a module to transmit the auto start application.....	326
8.4.1.3	Method to store resources in a module for the application .....	326
8.4.1.4	Capacity of autostart.zip .....	326
8.4.1.5	Auto start application caches .....	327
8.4.2	Operation of transmitting applications other than auto start applications.....	327
8.4.3	Name space of an application that is transmitted using a data carousel .....	327
8.4.4	Operation of the application boundary and access right for an application transmitted using a data carousel.....	329
8.5	Activation and termination of AIT-controlled applications.....	329
8.5.1	Activation and termination of AIT-controlled applications using an AIT that is not constantly monitored .....	329
8.5.1.1	Operation when activating using a BML document.....	329
8.5.1.2	Operation when an application is changed .....	330
8.5.1.3	Operation when the AIT-controlled application terminates by itself .....	330
8.5.1.4	Operation when activating a BML document .....	330
8.5.1.5	Operation when receiving a return flag .....	330
8.5.1.6	Operation when switching a broadcasting service .....	331
8.5.1.7	Operation when broadcasting is suspended .....	331
8.5.2	Activation/termination of an AIT-controlled application based on an AIT that is constantly monitored <Phase 1> .....	331
8.5.2.1	Operation when selecting a channel .....	331
8.5.2.2	Operation when updating a PMT .....	334
8.5.2.3	Operation when updating a constantly monitored AIT .....	334
8.5.2.4	Operation when activating using a BML document.....	336
8.5.2.5	Operation when using another application.....	336
8.5.2.6	Operation when an AIT-controlled application is terminated by itself .....	336
8.5.2.7	Operation when activating a BML document .....	337
8.5.2.8	Operation when a return flag is received .....	337
8.5.2.9	Operation when switching broadcasting service .....	337
8.5.2.10	Operation when a monitored AIT is removed.....	337
8.5.2.11	Operation when broadcasting is suspended .....	338
8.5.2.12	Operation when the application deviates from the application boundary range.....	338
8.5.2.13	Operation when a URI that is out of the application boundary is specified to the argument of replaceApplication .....	338

8.6	Referring broadcasting resources using a broadcasting integrated HTML5 application ...	338
8.6.1	Operation of a Ureg value using a broadcasting integrated HTML5 application.....	338
8.6.2	Operation of Greg using a broadcasting integrated HTML5 application .....	339
8.6.3	Referring video and audio components using a broadcasting integrated HTML5 application .....	339
8.6.4	Acquisition of information related to EIT [p/f].....	339
8.6.5	Operation of NVRAM .....	339
8.6.6	Referring the general event descriptor .....	340
8.6.7	Referring the NPT reference descriptor <phase 1> .....	340
8.6.8	Referring the built-in sound of a receiver .....	340
8.6.9	Operation of return flag <phase 1>.....	340
8.7	Displaying broadcasting video and audio and displaying video and audio via communication .....	340
8.7.1	Using VOD contents .....	340
8.7.2	Using audio via communication.....	341
8.7.2.1	Mixing broadcasting audio and audio via communication.....	341
8.8	Operation of browser pseudo-objects.....	342
8.8.1	Operation of operation control function .....	342
8.8.2	Operation of AIT-controlled application linkage functions.....	342
8.9	Broadcasting the integrated HTML5 application linkage function used on a BML document .....	343
8.9.1	Operation of the BML browser used in receivers that support the broadcasting integrated HTML5 application linkage function .....	343
8.9.1.1	Operation of functions used for the AIT-controlled application linkage function when obtaining an AIT via communication .....	343
8.9.1.2	Operation of functions used for the AIT-controlled application linkage function when obtaining an AIT through broadcasting <phase 1> .....	345
8.9.1.3	Activation of a broadcasting integrated HTML5 application from data broadcasting ..	347
8.9.1.4	Returning to the BML browser from a broadcasting integrated HTML5 application....	347
8.9.1.5	Confirmation of the broadcasting integrated HTML5 application linkage function from a BML document.....	347
8.9.1.6	Operation of a BML document using the broadcasting integrated HTML5 application linkage function .....	348
9	AIT operation for external application control .....	349
Appendix-1	Receiver unit common fixed colors.....	350



Appendix-2	Module compression format .....	354
Appendix-3	Clip function in video plane.....	355
Appendix-4	Regarding the scaling of videos .....	357
Appendix 4-1.	“Height”, “width” attributes that are specifiable in object elements .....	357
Appendix 4-2.	Display example of videos .....	359
Appendix-5	DTD for checking the operation area for basic services .....	363
Appendix 5-1	DTD .....	363
Appendix 5-2	Description of the DTD declaration section .....	368
Appendix-6	Precautions for Non-volatile memory access .....	369
Appendix-7	Root Certificate Descriptor .....	370
Appendix-8	getBookmarkInfo2 function.....	371
Appendix-9	Operation of argument of “getBrowserStatus()” .....	372
Appendix-10	Precautions at the time of switching video pixel size.....	373
Appendix-11	Operation guidelines of information on data broadcasting non-volatile memory .....	375
Appendix-12	Interaction channel communication based on connection status - Guidelines of return values for TCP/IP related functions .....	379
Appendix-13	Timing to initialize closed caption .....	380
Appendix-14	Arguments and return values for confirming whether the receiver implements the IPTV browser and its functions .....	381
Appendix 15	Operation when a required function is changed to an optional function due to revision .....	382
Appendix 16	Method to confirm the implementation of the BASIC procedure.....	383
Appendix 17	How to realize the VOD (integrated service type 2) function .....	385
Appendix 18	Operation of a CorrelatingID.....	388
Appendix 19	Operation of a TVsetID .....	389
Appendix 20	Governing organization.....	391

Appendix 21	Operation example of AIT module transmission among affiliate stations .....	392
[Section 3]	Operational specifications related to profile B (T.B.D.).....	397
[Section 5]	Operational provisions of common digital receiver units for terrestrial/BS/wide bandwidth CS .....	399
1	Introduction .....	399
2	Functions required in common receiver units .....	399
2.1	RAM .....	399
2.2	NVRAM .....	399
3	Contents operation guidelines assuming common receiver units.....	401
3.1	Judgment of common receiver units .....	401
3.2	NVRAM access for other media.....	401
3.3	Channel selection of other media services .....	401
3.4	Viewing reservation and programmed recording of other media services .....	401
3.5	Information sharing between media.....	401
[Section 6]	Operational provisions of common digital receiver units for terrestrial/BS/wideband CS/Advanced BS/Advanced wideband CS .....	403
1	Introduction .....	403
2	Functions required for common receiver units.....	404
2.1	Functions installed in common receiver units .....	404
2.2	RAM .....	404
2.3	NVRAM for digital broadcasting.....	404
2.4	Nonvolatile storage area for advanced wideband satellite digital broadcasting.....	406
2.5	Character code conversion .....	406
2.5.1	Conversion from EUC-JP to UCS.....	406
2.5.2	Conversion from UCS to EUC-JP.....	413
3	Contents operation guidelines assuming common receiver units.....	414
3.1	Judgment of common receiver units .....	414
3.2	Greg access.....	414
3.3	NVRAM access.....	416
3.4	Channel selection of advanced wideband satellite digital broadcasting services .....	416
3.5	Viewing reservation and programmed recording of advanced wideband satellite digital	

broadcasting services .....416



## [Section 1] Common specifications for different types of reception

### 1 Introduction

Data broadcasting services for Digital Terrestrial Television Broadcasting will be executed in accordance with the Ministry of Public Management's ordinances/announcements(including the former Ministry of Posts and Telecommunications' ordinances/announcements)and standards outlined by the Association of Radio Industries and Businesses(ARIB)in the "Data coding and transmission specification for digital broadcasting"(ARIB STD-B24), and "Service information for digital broadcasting system"(ARIB STD-B10). However, in order to widely utilize such regulations, it will be necessary to create separate specifications regarding the operations in detail, so this document, "Digital Terrestrial Television Broadcasting Specifications for Data Broadcasting Operations", was created.

The operational specifications presented herein are based on preserving the flexibility of data broadcasting program arrangement for each respective broadcaster and expandability for data broadcasting service development in the future, and its' objective is to be a guideline for signal transmission and reception specifications for the sound operation of data broadcasting services in Digital Terrestrial Television Broadcasting.

Broadcasters of data broadcasting programs shall follow these transmission specifications for Digital Terrestrial Television.

Digital Terrestrial Television Broadcasting receiver units should be able to receive signals which are transmitted according to these guidelines. Also, adequate care should be taken to avoid malfunctions that might be caused by transmission of signals that stray from these guidelines.

### 2 References

The contents of this volume define the operation of data broadcasting in Digital Terrestrial Television Broadcasting based on stipulations specified in the following standards.

- (1) "Receiver for digital broadcasting" standard ARIB STD-B21
- (2) "Service information for digital broadcasting system" standard ARIB STD-B10
- (3) "Transmission system for digital terrestrial television broadcasting" standard ARIB STD-B31
- (4) "Data coding and transmission specification for digital broadcasting" standard ARIB STD-B24
- (5) "Video coding, audio coding and multiplexing specifications for digital broadcasting" standard ARIB STD-B32
- (6) "Conditional access system specifications for digital broadcasting" standard ARIB STD-B25

Documents related to 6.3 "Broadcasting integrated IPTV service type 1 (optional)" and 7 IPTV "Download/VOD (integrated service type 2) function" are as follows.

- IPTV Forum "IPTV Standard VOD Specifications" IPTVFJ STD-0002
- IPTV Forum "IPTV Standard Download Specifications" IPTVFJ STD-0003
- IPTV Forum "IPTV Standard Internet Scope Service Approach Specifications" IPTVFJ STD-0007
- IPTV Forum "IPTV Standard Specifications of IPTV Service Approach by Integration with Broadcasting" IPTVFJ STD-0008

Documents related to 8 "AIT-controlled Application" are as follows.

- IPTV Forum "IPTV Standard Integrated Broadcast-Broadband System Specification" IPTVFJ STD-0010
- IPTV Forum "IPTV Standard HTML5 Browser Specification" IPTVFJ STD-0011

### 3 Definitions

16 : 9	Horizontal to vertical ratio of the display screen: 16 horizontal by 9 vertical
4 : 3	Horizontal to vertical ratio of the display screen: 4 horizontal by 3 vertical
8-bit character encoding	Lower overhead for character set switching compared to 7-bit codes; encoding system with improved transmission efficiency.
ARIB	Association of Radio Industries and Businesses. An association of broadcasters, telecommunications carriers and product manufacturers created to standardize use of radio-based technology within Japan.
BASIC PROCEDURE	Basic Mode Data Transmission Control Procedure: Communication procedure developed for basic hosting of data transmission control procedures and for terminal-terminal use. It features communication procedures to minimize inaccurate transmission of data.
CLUT	Color Look Up Table: Table to convert color information from an index value to a physical value.
DAVIC	Digital Audio-Visual Council: Name of the association whose objective was to define a standard method to transmit MPEG-digitized information reciprocally.
DRCS	Dynamically Redefinable Character Sets: Method to send external characters used in the standards of superimpose broadcasting and data broadcasting character encoding in patterns.
ES	Elementary Stream: Basic stream. It corresponds to encoded video, audio, independent data in a PES packet. One ES is transmitted by the PES packet that has the same stream ID.
EUC-JP	Japanese character code encoded in accordance with ISO 2022

HTTP	HyperText Transfer Protocol: Application layer protocol. This protocol (RFC2616) is used for the transfer of data over the World Wide Web.
I frame	Intra Frame: Frames built from compressed data embedded within an initial frame.
IP	Internet protocol: Network layer protocol which defines Internet addressing and distribution processing of data.(RFC791)
ISO	International Organization for Standardization
Multimedia service	Data broadcasting service based on XML-based multimedia encoding methods.
MNG	Multiple-image Network Graphics: File format for animation graphics. It is pronounced "MING". It includes multiple PNG images of which sequential displaying and repetition are possible.
MPEG-1	Moving Pictures Expert Group –1: MPEG1. Data compression coding technology including video and audio, which is standardized by the International Organization for Standardization (ISO/IEC 11172)
MPEG-2	Moving Pictures Expert Group –2: MPEG-2. Data compression coding technology including video and audio, which is standardized by the International Organization for Standardization (ISO/IEC 13818).
NPT	Normal Play Time: Absolute coordinates on time axis that show the positional relationship of events in a stream.
PES	Packetized Elementary Stream: Packetized stream. Packeted video, audio, and independent data of variable lengths.
PID	Packet Identifier: Packet ID (identifier). Shows the attributes of each individual stream of the corresponding packet in 13-bit stream identifying information.
PNG	Portable Network Graphics: Graphics file format succeeding GIF. It is pronounced "PING" and is capable of lossless compression. The file format is comprised of an 8-byte signature followed by a series of "chunks".
RFC	Request for Comments: Technological information made public to the Internet community by the Network Working Group.
SBR	Spectral Band Replication: Low bit spectrum expansion technology by AAC.
TCP	Transmission Control Protocol: Protocol for the transport layer in end-end. Offers highly reliable connection transmission that includes error detection and correction. (RFC793)
TLS	Transport Layer Security: One of the protocols used to send/receive encrypted information via the Internet. This can prevent wire tapping, tampering of data, and Web spoofing by using a combination of security technologies such as public key encryption and secret key encryption, and digital certification.
TS	Transport Stream: Transport stream standardized by the MPEG system standard(ISO/IEC 13818-1)
V.22bis	Modulation method for all double-layered telephone modems up to 2400bit/s specified by an ITU-T advisory
Kana-Kanji conversion	Process to convert input kana characters to appropriate kanji
Event	Program. Event in ARIB STD-B10

Entry component, entry carousel	Component whose component tag value is specified as 0x40 in the 2nd loop of the PMT is called “entry component”. Also, the data carousel transmitted in this component is called “entry carousel”.
Startup module	Module where moduleID=0
Startup document	Among all BML documents transmitted by the data carousel, this is the document that is presented first by default. Included in the startup module.
Stream format identification	Stream format identification specified in ISO/IEC 13818-1
Chunk	Blocked information found in the PNG and MNG file formats.
Data event	Period of time during which a BML document or a group of BML documents are transmitted in a component. Unrelated to SI events. Data events are switched based on the updating of the DII “data_event_id” transmitted in the component.
Data carousel	Method specified by ISO/IEC 13818-6 whose objective is to distribute data repeatedly in order to download various data via broadcasting.
TV programs with additional data	TV programs where additional data is broadcast along with an event in which video/audio are primary. Here, primarily audio programs are also considered “TV Programs”.
Visual component of TV programs with additional data	Parts other than data in TV programs with additional data.
BML-engine	Receiver software which receives and interprets multi-media data (BML documents) in order to present it to viewers.
Data broadcasting reception status	Status where receiver is receiving data broadcasting and playing it back.
Transport stream	Refer to the TS section.
Hash function	Mathematical function used to map large (very large in some cases) areas to smaller areas. Quality functions need to be simultaneously interaction channel and collision free.
Partial transport stream	Specially selected bit stream obtained by eliminating one or more transport packets not related the program from MPEG transport packets.
Font	Set of printed characters. Classified by style and size.
Font size	Same as design frame.
Bookmark list service	Service to broadcast “bookmark list display contents” which have the following features. Display bookmarked items as a list. When viewers select a desired bookmark from the list, and selects the channel to broadcast detailed information obtained in the URI from NVRAM bookmark information. Provides functions which manage bookmarks for viewers. The bookmark list service is not a service restricted by service type, but it refers to the service specified by bookmark list display contents.
Bookmark list data service	Service broadcast by service type 0xAA whose primary contents are “Bookmark list display contents”.



Bookmark writing service	Service to broadcast “bookmark write contents” with the following features. Present bookmark icons with the timing that has been specified in the contents, following the broadcast contents. According to the buttons pressed by the viewer, write the URI etc. of the communication site providing related information to contents that are currently displayed to the bookmark area of NVRAM. In C-profile, refer to the “TVlink”.
Broadcasting bookmark *1	This is a function to write link information to a communication sites to provide information related to programs which have been included in BML contents according to buttons pressed by user, and is used in C-profile. It refers to “TVlink”.
Pre-list service	Service to broadcast “pre-list display contents” with the following features. List of services that provide the bookmark list service (organized channels). When the viewer selects the desired channel from the list, the specified service (composing channel) should be selected and the bookmark list service displayed.
Base URI directory	Identifier to recognize communication contents as one document group.
Multi-section	Format used to include and transmit more than one section into a single TS packet.
Multi-view	Multi-view TV. Multiple video and audio are broadcast within one service. Combinations of video/audio intended by broadcasting stations can be switched by using this method.
Real-time viewing	To acquire data for display from the data carousel in real time (i.e. while viewing).
Link state	Status where receiver units can receive/play both BML contents from a server managed by a broadcasting station and video/audio resources which are being broadcast, together.
Local Contents	Data carousel transmitted in a data event of a given component.
General header	Header added to messages, which is used with request/response messages.
VIDEO PES	Data component used to transmit encoded video as standardized in ISO/IEC 13818-1.
Response header	Header added to response messages, which is used only for response messages.
Response message	Unit of response received by HTTP/1.1 client.
Audio PES	Data component used to transmit encoded audio as standardized in ISO/IEC 13818-1.
Related data broadcast	Collective term for the additional data area of TV programs with additional data, and is a data broadcasting service assumed to be viewed alongside video.
Common key encryption	Also called secret key encryption/symmetrical encryption. Using common key encryption owned secretly by the sender/receiver, encryption on the sender side and decryption on the receiver side. The common key itself needs to be shared via other methods.
Common fixed color	Color specified as common on the receiver unit color palette for things such as logo display.

Public key encryption	Also called asymmetrical encryption. There are different keys for encryption (public key) and decryption (secret key). By managing the public key as public and secret key as secret, encrypted communication is possible without common secret information. Some public key encryption schemes (RSA encryption and ECC encryption) have a signature function.
Viewing reservation	Used to make reservations to view programs by event, based on SI information.
Broadcaster color setup	Colors per broadcaster that can be set by CLUT index values and color combinations.
Closed caption	Service related to visual contents that display captions on top of the television broadcasting image.
Entity-header	Header added in order to indicate data attributes in case data is included in request/response messages.
Receiver dependent color	Colors per receiver that can be set by CLUT index values and color combinations.
Optional feature	When featured functions and performance depend on the receiver unit or product.
Uplink	Line used to connect to center equipment by using modems, etc.
2 displays of horizontal vertical picture element	Displays 1 pixel data in a 4 pixel area (horizontally and vertically).
Server	Web server capable of handling HTTP/1.1 on TCP/IP networks.
First (2nd, 3rd, 4th) level kanji	Standard levels of character codes. Specified by JIS X0208, etc.
Communication contents	BML contents located on servers
Independent data program	Data program principally involving multi-media data. In some cases, the video/audio components of TV programs are shared.
Unlinked status	Status where the receiver unit only receives and plays communication contents. In this state, receiver units cannot refer to broadcast video or audio resources. This function can be optionally implemented in a receiver.
Additional data	Data section of TV programs with additional data.
Superimpose	Closed caption service not synchronized with main video, audio and data. It is used for news flashes, changes in air times, time signals, earthquake early warning, etc.
Request header	Header given to request messages used only for request messages.
Request message	Unit of request for HTTP/1.1 servers.
Special service	This service is prepared for broadcasting on a temporary basis using a service_id that is different from the one used for normal service. This service is not regularly provided but temporarily provided.
Recording	To record broadcasting services in the transport stream or partial transport stream format on D-VHS or HDD. Recording function is optional for basic receiver units. (In case of referring to analog recording, write "analog recording" explicitly)
Programmed recording	To reserve (program) recording of programs by event unit, based on SI information.
Fringe area	Area located near the outer limits of the service area.
One-touch button	Key used for one-touch channel selection. Specifications specified in Volume II of this document.

Roll-up mode	Closed caption service to display additional, prepared closed caption data sent as page data into areas of 3 lines, gradually line-by-line. Rolls up vertically at line breaks.
Closed caption out-screen display function	Function to reduce the video size in order to prevent overlapping of closed caption with captions within the broadcast image.
Root certificate	When encrypting communication by TLS, format pursuant to X.509 which is used to send and store public keys. In particular, certificates to authenticate the authorizer are called root certificates.
General-purpose root certificate	Root certificate that is stored in the receiver unit and is sent by a data carousel. When performing encrypted communication, the receiver unit refers to the general-purpose root certificate.
Root certificate exclusively for broadcasters	Root certificate transmitted by a data carousel which is used temporarily by receiver units.
Module for storing root certificates	Modularised root certificate used when transmitted by a data carousel. In the module for root certificate storage, a maximum of two root certificates are stored.
General-purpose root certificate storage area	Area allocated to NVRAM of receiver units to store general-purpose root certificates.
General-purpose root certificate storage number	Number to specify storage location of general-purpose root certificates and is location specified for storing general-purpose root certificates transmitted by data carousels.
General-purpose root certificate ID	ID to recognize the different types of general-purpose root certificates. The values are set by the certificate management group stated in Vol. 6.
General-purpose root certificate version	Value to indicate the version of general-purpose root certificates. The values are set by certificate management group stated in Vol. 6.
Registration transmission	Function to store/save telegrams in receiver units and send stored/saved telegrams per user instruction.
Registration transmission information	Term to refer to telegrams stored/saved in registration transmission and related information.
Reservation transmission	A function of registration transmission. Sends telegrams stored/saved by the registration transmission function at the time specified by the receiver unit. Registration (programming) of reservation transmission is done by the viewer.
BML browser	Data broadcasting browser which works based on this volume. If the receiver conforms to A-profile, the data broadcasting browser operates when the receiver is receiving a data broadcasting service and in linked and unlinked (these functions can be optionally implemented in the receiver) status. The browser displays the contents received through broadcasting signals and communication lines. In C-profile, it works when receiving data broadcasting and in linked status, and received contents are presented via broadcasting signals and communication lines.
Communication browser	General term to refer to the browser for the C-profile contents on the internet and the carrier's proprietary browsers. Mixed display with broadcast contents and execution of some scripts will be restricted.

C-profile communication browser	Communication browser which works based on the C-profile specifications in this volume. Execution of some scripts will be prohibited.
Carrier's proprietary browser	Browser implemented based on specifications defined by the carrier.
HTML browser	Browser with a feature to view HTML contents.
Broadcast contents	Broadcasting materials of video, audio, and data received via broadcasting signals.
8-bit character codes for C-profile	The character encoding method for closed caption used in C-profile, this is subset by restricting the use of control codes based on the "8-bit character codes" used in Profile A.
C-profile communication contents	General term for BML contents located on servers specified in C-profile. Classified as C-profile linked contents and C-profile unlinked contents.
C-profile linked contents	Communication contents written under the assumption that a receiver unit in accordance with C-profile and with a linked status will present them.
C-profile unlinked contents	Communication contents written under the assumption that a receiver unit in accordance with C-profile and with an unlinked status will present them from a Browser for the C-profile contents on the internet.
Broadcaster contents	General term for broadcast contents and C-profile linked contents.
Communication carrier specification contents	Contents located on a server, scripted based on specifications specified by the communication carrier.
Mobile phone	A "specific mobile terminal equipment" as prescribed by the ministerial ordinance (Regulations for Enforcement of the Telecommunications Business Law) provided by telecommunications carriers.  (Note: From the viewpoint of protecting personal information, mobile phone is assumed to be a terminal compatible with TLS/SSL and root certificates.)
Low-frame-rate and low-resolution picture	Compared to those used for television services, this video's pixel size and frame rate is much lower.
Virtual plane	Plane to locate contents of data broadcasting. The virtual plane is defined by specifications and it is not required to be implemented by receiver units.
Color formatter	A method for converting pixel color information (such as converting YCbCr components into RGB).
H.264/MPEG-4 AVC	Advanced encoding/decrypting technology co-developed by the Moving Pictures Expert Group (MPEG) of the International Organization for Standardization/International Electro-technical Commission U.S., Inc.(ISO/IEC) and the Video Encoding Expert Group (VCEG) of the International Telecommunication Union(ITU).
NVRAM	NVRAM is Non-Volatile Random Access Memory. It is called a non-volatile memory area and it is an area that does not get turned off even when the power is turned off.
Affiliation	Network created in order for broadcasters to interoperate programs. It consists of a broadcaster called "key station" and multiple broadcasters called "net stations". A net station with multiple parent key stations is called a "crossnet station".

NIT	Network Information Table: Sends information related to transmission path information such as frequency and arranged channel, and all of arranged channel ID numbers included in one distribution system are addressed here.
PI	Processing Instruction. This is used to include instructions for applications in documents.
Mono-media	Independently presented media such as video, audio, characters, still images, etc.
Multi-media	Presented media composed from multiple mono-media. Involves mutual interaction between mono-media and often involves interaction with users.
DTD	Document Type Definition. Defines what kind of elements and attributes are used in a document.
Focus	Indicates the focal point of users in data broadcasting. Generally, a palette is highlighted or boxed or underlined to indicate where the focal point of the user is.
Module	One of the data units transmitted after being divided into blocks by a data carousel.
Resource	Resource often refers to mono-media, which is a component element of multi-media. However, it is called a resource when referred to as a raw material.
Selector(CSS)	Format to select for which element corresponding CSS characteristics are used.
Default style sheet	Refers to the CSS characteristics list which is implicitly assumed to have values specified within the receiver units. It has the lowest priority and it gets overwritten by values specified by external style sheets and style attributes.
Event	Refers to events in the computer world. This is differentiated by the word "event" in broadcasting, which refers to programs.
Simultaneous display	To display multiple different contents and statuses, etc. on one screen simultaneously.
Mixed display	When a content provider is intentionally associating its contents with other contents from a different provider and displaying them simultaneously, to have influences to display etc., in order to display contents from multiple different providers as if they are from the same source and thereby misleading the viewer.
Full screen view	Status where a part of a broadcaster's contents is displayed in full screen without displaying any other contents at the same time. Method to switch to full screen view is different for each terminal.
User-Agent	One of the HTTP Request headers that are sent to communication destinations from terminals that have communication functions.
IPTV download	A process for downloading on-demand contents through the communication via an IP network.
AV player	An AV player is a receiver application that controls playback of videos and audio contents in conformity with the <i>Integration with Broadcasting Specifications</i> , which is a part of the IPTV Forum Japan Technical Specifications. AV players include players to playback stream-type contents and also those to playback contents stored in a storage.
Playback control information	This refers to the supplementary information necessary for replaying contents.

Download control information	Download control information is the meta-information necessary for acquiring contents and the supplementary information.
VOD	Video on Demand: VOD is generally a service that distributes video contents upon the request of a receiver. For the standard of these guidelines, this service refers to streaming video contents distribution service.
ECG	Electronic Contents Guide: ECG is a resident application to navigate contents.
Broadcasting integrated IPTV service type 1	IPTV service that can be used by activating the IPTV browser installed in the receiver, from the BML using the startExtraBrowser( ) function.
Broadcasting integrated IPTV service type 2	IPTV service that is executed by downloading VOD/IPTV directly from the BML.
IPTV download/VOD (integrated service type 2) function	A function to provide services that conforms to the broadcasting integrated IPTV service standard type 2.
IPTV download/VOD (integrated service type 2) compliant receiver	A receiver that supports the broadcasting integrated IPTV service standard type 2.
CDN	Contents Delivery Network: An IP communication network that may be directly connected to the user's reception environment through an access network in consideration of the conditions such as communication quality.
CDN scope	A CDN scope is IPTV service that is provided based on an IPTV standard and part of the IPTV Forum Japan Technical Specification, <i>CDN scope service approach specifications</i> .
Internet scope	IPTV service that is provided in conformity with the <i>Internet Scope Service Approach Specifications</i> , which is a part of the IPTV Forum Japan Technical Specifications.
DRM	Digital Rights Management: A general term for technologies that prevent unauthorized copying or distribution of the contents (images, music, and videos) using such means as encryption.
DRM client	A functional entity inside a receiver that acquires and manages licenses and supplies appropriate keys for using contents.
DRM server	A server that issues and manages licenses.
RTP	Realtime Transport Protocol: A protocol for transmitting audio and video streaming data.
RTSP	Real Time Streaming Protocol: A control protocol for real-time distribution of video and audio contents over an IP network.
Streaming	Streaming is a method of presenting data at the same time as receiving the data through an IP network.
ECG metadata	ECG metadata is an XML document that contains attribute information of contents, packages, and licenses that are used by the ECG.
Playback control metafile	This is an information file used by receivers to receive streaming data or replay contents over an IP network. The metafile consists of information for reception and playback control, information on DRM, etc.
License ID	This is an identifier to uniquely identify a license.
RMPI	Rights Management and Protection Information: RMPI is the conditions for using contents.

CorrelatingID	A CorrelatingID is an ID allocated for a receiver that is used as hardware for associating the registration information for IPTV service with the receiver.
External device linkage function	A function to send text data from BML contents to external devices, such as tablets and smartphones.
Application control information	Signals that control the activation and termination of an application and its access to a broadcast resource. These signals are transmitted through broadcasting and communication paths.
AIT	Application Information Table: AIT refers to application control information.
AIT-controlled application	A general term for applications whose execution including activation and termination is controlled by the AIT specified in ARIB STD-B24.
AIT-controlled application linkage function	This is a function to activate the AIT-controlled application from data broadcasting using a mathematical function.
AIT-controlled application external activation function	This is a function to activate directly AIT-controlled application from external devices such as smartphones.
Broadcasting integrated HTML5 application	This application is one of the AIT-controlled applications and conforms to the communication-broadcast integrated systems and HTML5 browsers provided in the IPTV Forum Japan Technical Specifications.
Auto start application	An application that is one of the broadcasting integrated HTML5 applications described in an AIT that is constantly monitored, with an application_control_code value of the AIT that is 0x01 (identification name: AUTOSTART).
External application	An application that activation and termination are controlled by means other than AIT specified in ARIB STD-B24. This application may be controlled by AIT except for activation and termination.

- (\*1) Trademark rights for "BOOKMARK/ブックマーク" (Japanese Trademark No. 3281163) belong to Sony Corporation. Contact the governing organization described in [Section 4] Appendix 13 regarding the terms and conditions for the use of these trademarks.

## 4 Type of operational specifications specified in Vol. 3

### 4.1 C-profile corresponding to different reception types

In Digital Terrestrial Television Broadcasting, car TVs and portable type receiver units are considered as targets in addition to stationary type TVs and STBs. In each service, there are different presentation models of receiver units and different operational styles of mono-media as the contents target. Therefore, in this chapter, profiles are specified by the service type, and detailed specifications are specified in relation to each profile.

**Table 4-1 Profiles for different types of receiver units**

Profile	Contents
Profile A	Basic operation profile mainly targeting fixed receiver units (Stationary TVs, STBs, Portable TVs, etc.)
Profile B (T.B.D)	Basic operation profile of data broadcasting services mainly targeting transportable receiver units(car TVs, portable TVs, PDAs, etc.)
C-profile	Basic operation profile of data broadcasting services mainly targeting portable receiver units(mobile devices, etc.)

### 4.2 Relationship between assumed receiver units and each profile

**Table 4-2 Relationship between profiles and receiver units**

	Profile A	Profile B	C-profile	Comments
Fixed receiver units	O	Δ	Δ	
Transportable receiver units	T.B.D.	T.B.D.	T.B.D.	
Mobile receiver units	X	X	O	

O: Presentation is required for receiver units that feature data broadcasting functions

Δ: Optional

X: Reception is optional

### 4.3 Basic functions and optional functions

- Basic functions refer to the functions that receiver units with data broadcasting playback functions should be equipped with.
- All the functions other than the basic functions are optional; however, minimized operational specifications that should be followed in case the optional functions are realized will also be specified in this document.
- In Vol. 3, when it is written “should”, it refers to implementations (specifications) that are “not required” but are “preferable” from the broadcaster point of view.



## [Section 2] Operational provisions related to Profile A

### 1 Functions that basic receiver units should be equipped with to receive data broadcasting

In this chapter, the functions that basic receiver units should be equipped with to receive data broadcasting are defined.

#### 1.1 Components of receiver units

Basic receiver units are in compliance with Class A indicated in "2. Example of receiver construction" in ARIB STD-B24 "Data coding and transmission specification for digital broadcasting" Vol. 1 Part 1, Informative explanation. The provisions for each process part of basic receiver units from the hardware component perspective are defined as well as resources inside the receiver units by the reference decoder.

##### 1.1.1 Hardware components

Basic receiver unit hardware components are shown in figure 1-1.

The digital broadcasting signal input into a basic receiver unit is converted into a transport stream by a tuner and demodulation. The demodulated transport stream is divided into video, audio, and other data by a transport stream decoding process, after which the video stream is output to a video decoding process and the audio stream is output to an audio decoding process. By going through the above process, playback of normal video and audio are performed by basic receiver units.

When receiving data broadcasting, data needs to be transferred to main memory or nonvolatile memory once to be processed by the CPU. Also, in addition to the playback process of normal video and audio, when presenting characters, data in main memory may be transferred to the video and audio decoder at the same time to perform the playback process of video and audio. Also, more interactive performance is expected compared to traditional TV viewing with the use of an uploading connection, etc. From the perspective of process performance of the above hardware, the following provisions need to be stated.

- (1) Data receivable by transport decoders
- (2) Playback of the stream system and accumulated audio data
- (3) Playback of the stream system and accumulated video data.
- (4) Presentation of video, still pictures, texts and graphics, etc.
- (5) Interaction channel communication function using communication line
- (6) Data size that can be saved indefinitely

- (7) Data size that should be equipped in receiver units such as fonts.
- (8) Memory size for obtaining and decoding data.
- (9) Guidelines regarding remote control operation

In particular, (1) is defined as a TS decoder function, (2),(3),(4),(7) are defined as presentation functions, (5) is defined as a communication function,(6),(7),(8) are defined as memory volume, and (9) is defined as a remote control function.

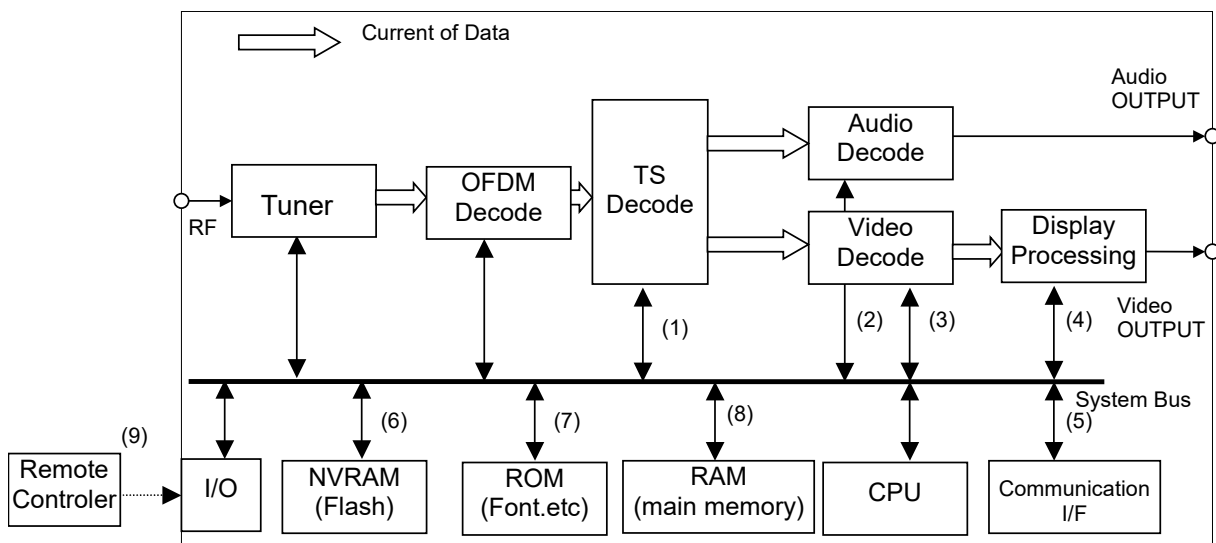


Figure 1-1 Hardware components of basic receiver units

### 1.1.2 Receiver unit reference model

By clarifying the resources inside the receiver units with a receiver unit reference model, the receivable limit of data broadcasting in a basic receiver unit is defined. The receiver unit reference model is shown in figure 1-2.

This reference model was created in compliance with provision items of this operation specification based on the decoder model indicated in DAVIC 1.4 Part 9. All units should be in compliance with definitions in DAVIC unless otherwise specified in subsequent sections of this document.

The received MPEG-2 TS is classified in each ES using the PID filter and elementary video and audio streams transmitted through the PES is stored in B<sub>n</sub>, which is the primary buffer, through T<sub>Bn</sub>, which is the transport buffer. On the other hand, elementary stream of MM contents transmitted by the data carousel is “section filtered” and stored in B<sub>contents</sub> through T<sub>Bn</sub> after PID filtering.

Multimedia contents data received in this way are started by the receiver unit user, etc. The

Multimedia engine follows this startup instruction and reads data inside Bcontents and performs execution of MM contents with Bwork as the memory for execution. In such case, mono-media content transmitted by a data carousel is given to each decoder from Bcontents, and mono-media content transmitted by streaming is given to each decoder from Bn. Mono-media content for audio systems is presented through speakers, which are the presentation device after decoding. On the other hand, after mono-media contents for video systems are decoded separately for video, still pictures, texts and graphics, and caption, they are buffered and combined into plane display memory once and presented through the monitor, which is the presentation device. Combinations of each plane are modeled as follows. In other words, effective pixels are switched separately for the video area and still picture area. This combined image of video and still pictures are mixed with the text and graphic plane by multiplication of the alpha value and its output is mixed with the subtitle plane by multiplication of the alpha value in the same way to compose the presentation screen.

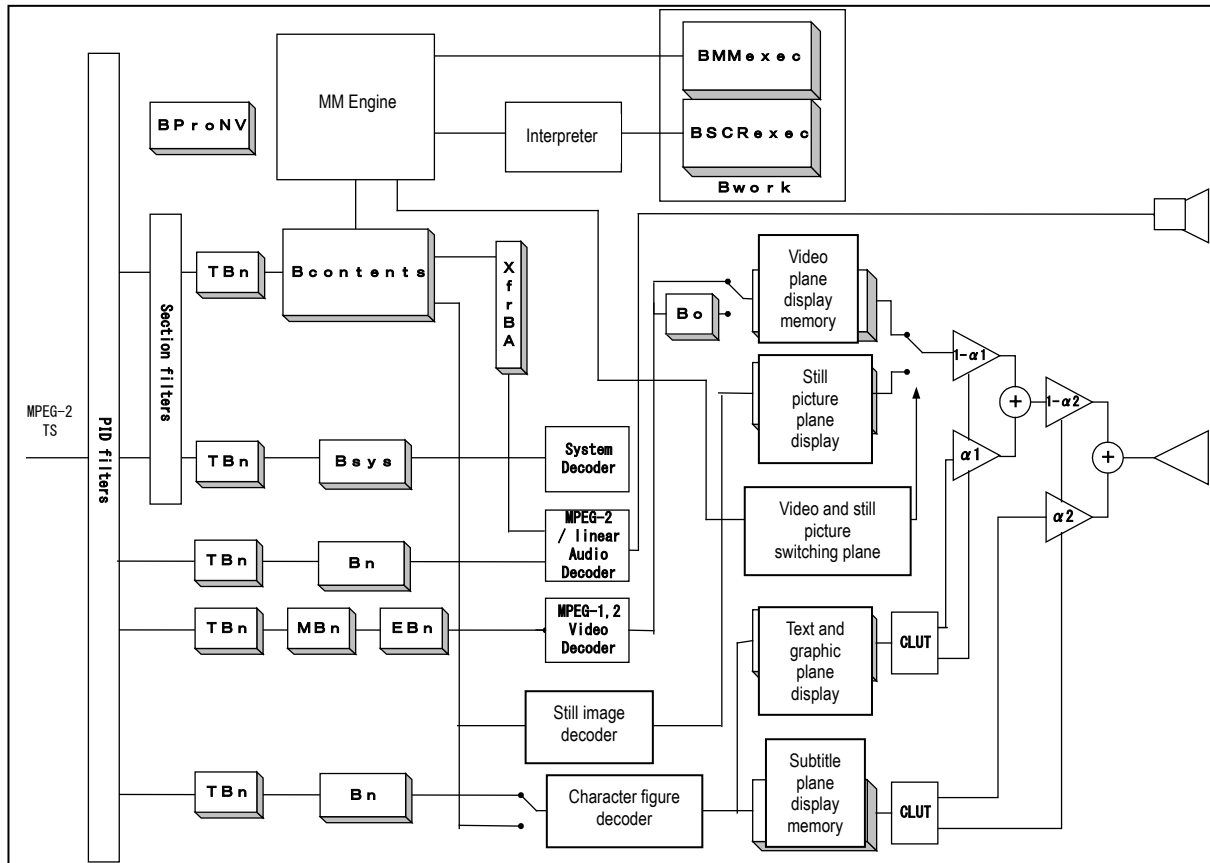


Figure 1-2 Reference model of basic receiver units

TBn	Transport buffer of the elementary stream $n$
Bn	Primary buffer of the elementary stream inside the decoder.
Bcontents	Buffer of multi-media contents data transmitted by the data carousel. When the module is transmitted with compression (Compression Type Descriptor of DII is placed), both data before and after compression (compressed, expanded) are buffered.
XfrBA	Buffer for transferring audio contents in file format to the audio decoder.
BMMexec	Execution memory of multi-media code
BSCRexec	Execution memory of script.
Bwork	Multi-media contents execution memory that is a combination of BMMexec and BSCRexec
BproNV	Non-volatile memory to store unique information for each receiver user and broadcaster.
Other buffers	Refer to ISO/IEC13818-1, DAVIC 1.4 Specification Part 9 for definitions.

## **1.2 Presentation functions**

Presentation functions of a basic receiver unit are in compliance with presentation functions of receiver units indicated in Vol. 1, Part 1 of ARIB STD-B24 "Data coding and transmission specification for digital broadcasting".

### **1.2.1 Pixel size/restrictions of each plane composing the display screen**

Provisions relating to the pixel size of each plane composing the display screen are shown in table 1-1.

**Table 1-1 Pixel size of screen planes**

Item	Description of provision	
Video plane	pixel size	1920x1080x16, YCbCr(4:2:2), 16:9
		720x 480x16, YCbCr(4:2:2), 16:9
		720x 480x16, YCbCr(4:2:2), 4:3
Still picture plane	pixel size	1920x1080x16, YCbCr(4:2:2), 16:9
		720x 480x16, YCbCr(4:2:2), 16:9
		720x 480x16, YCbCr(4:2:2), 4:3
text and graphic plane	pixel size	960x 540x8, 16:9 (* )Display size is 1920x1080(each pixel is rendered twice both horizontally and vertically)
		720x 480x8, 16:9
		720x 480x8, 4:3
	CLUT	CLUT Number:1 Common fixed colors :17 colors(Appendix-1 index value 0-16) Broadcaster setup colors :207 colors (Index value 17-223) Receiver unit dependent colors:32 colors (Index value 224-255) However, regarding index value of 0-127, each time of BML document presentation, the value indicated in Appendix-I should be set as the initial value by the receiver unit.
	Presentation	Present 8-bit index value of CLUT after converting it to $\alpha$ value of 4 bits and YCbCr(4:2:2). (*1)
Subtitle plane	pixel size	960x 540x8, 16:9 (* )Display size is 1920x1080 (each pixel is rendered twice both horizontally and vertically)
		720x 480x8, 16:9
		720x 480x8, 4:3
	CLUT	CLUT Number:1 Common fixed colors :128 colors (Refer to Appendix-1) Receiver unit dependent colors: 32 colors
	Presentation	Present 8-bit index value of CLUT after converting it to $\alpha$ value of 4 bits and YCbCr(4:2:2). (*1)
Video and still picture switching plane	pixel size	960x 540x1, 16:9 (* ) Size is 1920x1080 (picture element is divided into four equal-sized zones)
		360x 240x1, 16:9 (* ) Size is 720x 480 (picture element is divided into four equal-sized zones)
		360x 240x1, 4:3 (* ) Size is 720x 480 (picture element is divided into four equal-sized zones)

(\*1) When setting a color map with 8-bit alpha value to the CLUT, the most significant 4 bits of the alpha value in the color map should be mapped to the 4 bits of the alpha value in the CLUT.

For a combination display of the text and graphic plane and the subtitle plane, nonlinearity in alpha blending is allowed. However, the objective is to allow a degree of flexibility in the design of blending circuit, and it is required to realize the presentation effect according to the 16 levels of alpha value.

Provisions relating to presentable mono-media code, the presentation position of mono-media content, and size, etc. are shown in table 1-2 as restrictions relating to each plane.

**Table 1-2 Presentation restrictions on the screen plane**

Item	Description of provision	
Video plane	Presentable mono-media code	MPEG-2 (* )Number of videos presented at once is 1 regardless of the encoding method.
		MPEG-1 (* )Number of videos presented at once is 1 regardless of encoding method.
	Presentation position	From even number picture elements to odd number picture elements of planes for both x and y coordinates. (*1)
	Size	Even number picture element for both x and y coordinates.
	Overlapping	Videos do not overlap with each other.
	Clipping	Only possible in the direction of x coordinates. (refer to appendix-3)
Still picture plane	Presentable mono-media code	JPEG
	Presentation position	From even number picture elements to odd number picture elements of planes for both x and y coordinates.
	Size	Even number picture element for both x and y coordinates
	Overlapping	No restrictions (*2)
Text and graphic plane	Presentable mono-media code	8-bit character codes (*4) including EUC-JP (optional)
		PNG
		MNG
	Presentation position	From any picture element to any picture element for both x and y coordinates
	Size	Any picture element count for both x and y coordinates
Overlapping	No restrictions(*2)	
Subtitle plane	Presentable mono-media code	8-bit character codes
		Bit map data(*3)
	Presentation position	From any picture element to any picture element for both x and y coordinates
	Size	Any picture element count for both x and y coordinates
Video and still picture switching plane	Switching position	From any picture element for both x and y coordinates.
	Size	Any picture element count for both x and y coordinates
Presentation switching effect	Process of presentation switching effect is implementation dependent.	

- (\*1) ARIB STD-B24 is followed for the definition of picture element.
- (\*2) Production of applications with no (or little) need to re-render still pictures, texts and graphics due to changes and moving of overlapping sequence is recommended. For receiver units, however, indication should not fail due to re-rendering. Please refer to section 1.2.2 for restrictions on overlapping of still pictures.
- (\*3) Refer to ARIB STD-B24 Vol. 1 Chapter 6 Presentable combinations of planes and restrictions
- (\*4) The receivers can optionally use 8-bit character codes. However, do not externally refer to 8-bit character codes included in BML documents.

## 1.2.2 Combination and restrictions of presentable planes

As indicated in the reference model, the presentation screen is composed of a combination of various planes. Provisions for the combination of each plane are shown in table 1-3.

**Table 1-3 Combination and restrictions of presentable planes**

Item	Description of provision
Pixel size	For presentation of combinations of planes of video, still picture, text and graphic, subtitle, only those with the same pixel size and the same aspect ratio can be presented in combination. The text and graphic/subtitle plane in 960 x 540 is, however, it is recognized as 1920 X 1080. (*1)(*2)(*3)
	In the video and still picture switching plane, video to be switched/still picture plane pixel size in 1/2x1/2 pixel size is recognized as the same pixel size as the switching target plane. (*4)
Area specification in video and still picture switching plane	Area specification of either video or still pictures is rectangular. (*5)
Maximum area set up of video, still picture	When the rectangular area is video, the maximum area setup number is 1. When the rectangular area is all still pictures, the maximum area setup number is 4. (*5)

- (\*1) For presentation of contents with only video as the display target, a receiver unit that can perform presentation without composing the combination of above planes is conceivable. For such receiver units, the video display format indicated in Part 1 Chapter 5 of ARIB STD-B32 should be enforced for the transparent contents defined below. However, this will not interfere with the ability of receiver units to uniquely choose the effective areas of this provision. For example, in case the pixel size specified by the document is 960x540 16:9 and the pixel size of the video is 720x480 16:9, then in order to avoid blackouts at the time of transition from transparent contents to non-transparent, choosing to display transparent contents to be up-converted and presented as 1920x1080 16:9 video should be allowed. "Transparent contents" refer to BML documents where the "invisible" attribute of body elements are set as "invisible". Even in such cases, "Precautions when switching video pixel size" indicated in Appendix-10 should be taken into consideration.
- (\*2) Even when aspect ratio of video is changed suddenly due to insertion of emergency broadcasting, etc., and the status becomes different from the data broadcasting pixel size, a combined display of TV broadcasting and data broadcasting should be continued. Display behavior here is implementation dependent. Refer to Chapter 2 – "Operation of the data transmission method" for guidelines on receiver unit operation.
- (\*3) Combinations of text and graphic, subtitle, video, and still picture planes that can be used in data broadcasting simultaneously are called data broadcasting pixel size, and are defined as follows.



		Data broadcasting pixel size		
		960x540 (16:9) (*)	720x480 (16:9)	720x480 (4:3)
Text and graphic plane pixel size	960x540 (16:9)	○		
	720x480 (16:9)		○	
	720x480 (4:3)			○
Subtitle plane pixel size	960x540 (16:9)	○		
	720x480 (16:9)		○	
	720x480 (4:3)			○
Video plane pixel size	1920x1080 (16:9)	○		
	720x480 (16:9)		○	
	720x480 (4:3)			○
Still picture plane pixel size	1920x1080 (16:9)	○		
	720x480 (16:9)		○	
	720x480 (4:3)			○

- (\*) Text and graphic plane in data broadcasting pixel size 960x540 (16:9) is overlapped with video and still picture respectively with rendering each pixel twice horizontally and rendering each line twice vertically.
- Data broadcasting pixel size that can be combined with 720p video should be 960x540(16:9).
  - In case new MPEG pixel size is defined along with the revisions of ARIB STD-B21, then inspection will take place as needed.
- (\*4) Full pixel size for both video and still pictures is YCbCr(4:2:2) and the effective unit for switching is 2 picture elements. This is to execute switching of video/still pictures by a video/still picture switching plane of 1/2x1/2 pixel size of video, still picture plane.
- (\*5) There are two visual patterns for display combinations of video plane and still picture plane.  
The first pattern is the pattern where video is placed over a still picture as shown in figure 1-3. In this case, the rectangular area is a video area and the number of areas that can be set is 1.  
The second pattern is where still pictures are placed over a full screen view video as shown in figure 1-4. In this case, the rectangular area is a still picture area and the number of areas that can be set is 4 or less. Also, as an example where multiple still picture areas are set as shown in figure 1-6, a case where the area is set with still picture 1 and 2 overlapping each other is conceivable. In this case, however, the area is not rectangular, so it is against these provisions and therefore not implementable. On the other hand, still pictures 3 and 4 in figure 1-5 do not form a rectangle, but in this case, this is understood as two rectangular areas that are sharing a border, and is recognized as in compliance with these provisions.

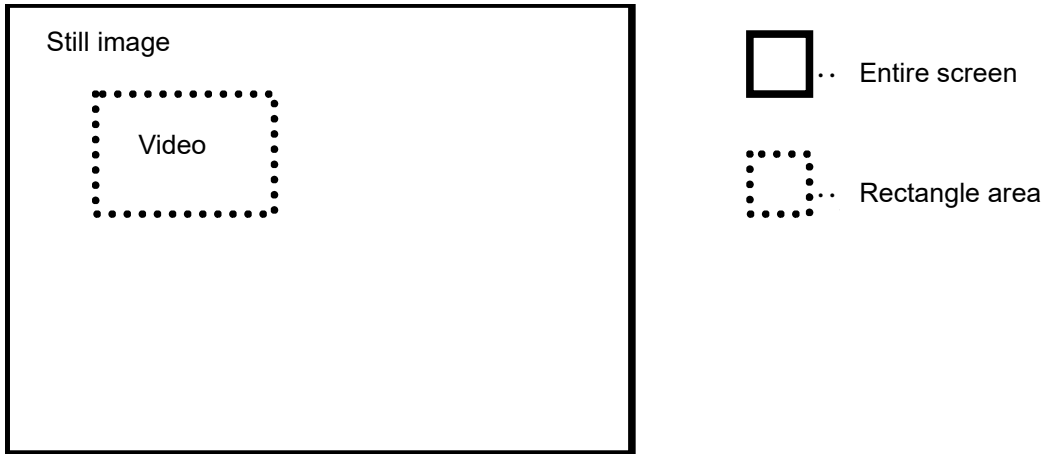


Figure 1-3 Display example 1 of possible combination of video and still pictures

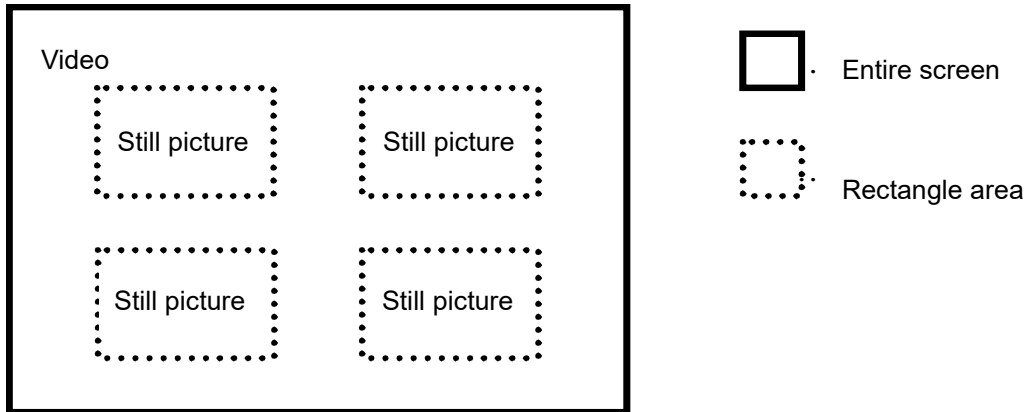


Figure 1-4 Display example 2 of a possible combination of video and still pictures

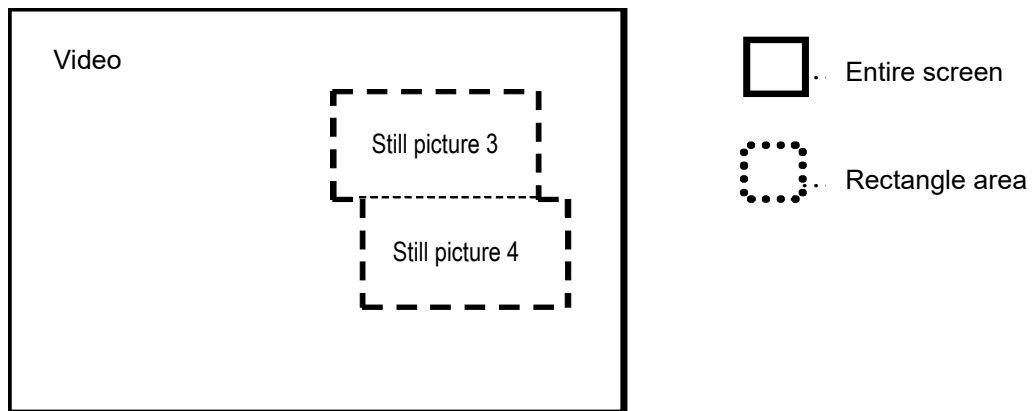
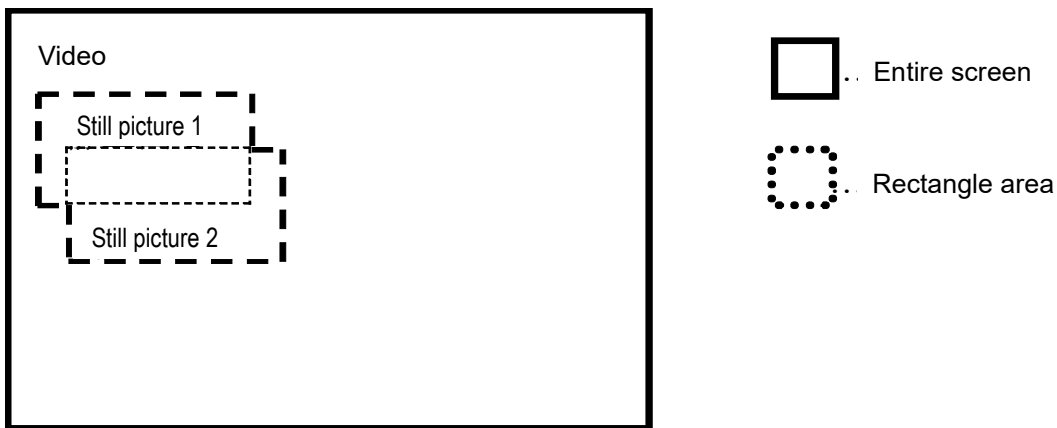


Figure 1-5 Display example 3 of a possible combination of video and still pictures



**Figure 1-6 Display example of a impossible combination of video and still pictures**

### **1.2.3 Relationship between mono-media encoding and the presentation plane**

An overview of restriction conditions for mono-media coding presented in each plane of the above is described in Table 1-4. This is assuming the broadcaster will not be sending mono-media encoding that is not included in these provisions, or mono-media data encoded in non-standard operations. A detailed description of contents of provisions regarding each respective encoding method will be defined in Chapter 3.

**Table 1-4 Overview of restriction conditions for mono-media encoding presented in each screen plane**

Encoding method		Description of provision		
Video encoding	MPEG-2 (*1)	Transmission method	Screen image PES (*2);Stream format identification = 0x02	
		Image size(*3)	1920x1080 (16:9), 1440x1080 (16:9) 1280x720(16:9), 1920x1080 (4:3), 1440x1080 (4:3) 1280x720(4:3), 720x480 (16:9), 544x480 (16:9), 480x480 (16:9), 352x240 (16:9), 720x480 (4:3), 544x480 (4:3), 480x480 (4:3), 352x240 (4:3)	
		Scaling	256/128,192/128,160/128,128/128,112/128,96/128,80/128,64/128,48/128,32/128(*4)	
	MPEG-1	Transmission method	Screen image PES (*2);Stream format identification =0x01	
		Image size	352x240(4:3, 16:9),176x120(4:3, 16:9) (*3)	
		Scaling	256/128,192/128,160/128,128/128,112/128,96/128,80/128,64/128,48/128,32/128(*4)	
Still picture encoding	JPEG	Transmission method	Data carousel; Stream format identification = 0x0D	
		Image size	Any desired size from a horizontal vertical 16 picture element image to a full size image	
		Scaling	128/128(*5)	
		Others	Presentation in 4:2:0 pixel sizes is assumed. However, receiver unit indication should not fail by 4:2:2 input.	
Text and graphic encoding	PNG	Transmission method	MM ENCODING	Data carousel; Stream format identification = 0x0D
			Closed caption	Independent PES; Stream format identification = 0x06
		Image size	Any desired size from a horizontal vertical 2 picture element image to a full size image.	
	Scaling	128/128		
	MNG	Transmission method	Data carousel; Stream format identification = 0x0D	
		Image size	Any desired size from a horizontal vertical 2 picture element image to a full size image.	
		Scaling	128/128	
	8-bit character codes(*6) Including EUC-JP (optional)	Transmission method	Independent PES;Stream format identification = 0x06 Data carousel; Stream format identification = 0x0D	

(\*1) Possible to pseudo- display still pictures by transmitting only intra-frames. Restrictions for all such cases are in the category of video encoding. Therefore, in this case, simultaneous decoding with other types of video (MPEG-1, 2) is not possible. Also, when transmitting intra-frames above, the placement of a Video Decode Control Descriptor is required. The operation of a Video Decode Control Descriptor is in compliance with Vol. 4.

(\*2) Stream packetized by PES packets, which was multiplexed with MPEG-2 TS

(\*3) The physical picture element of planes has been defined as pixel size in this document. In case screen size of mono-media is not in accordance with this, mapping the plane is performed according to the following policy.

MPEG pixel size should be defined from Vertical Size, Horizontal Size, and Aspect Ratio in the “sequence header” in data encoded by MPEG. MPEG pixel size of video data presented in each respective video plane pixel size is as follows. The aspect ratio 4:3 of HD video data indicates the value of sent stream in the sequence header, and there is no difference from the 16:9 aspect ratio in appearance of the display.

MPEG VIDEO		MPEG pixel sizes	Video plane pixel sizes				
			1920x1080 (16:9)	720x480 (16:9)	720x480 (4:3)		
HD video data		1920x1080 (16:9)	O	X	X		
		1440x1080 (16:9)	O	X	X		
		1280x720 (16:9)	O	X	X		
		1920x1080 (4:3)	O	X	X		
		1440x1080 (4:3)	O	X	X		
SD video data	MPEG 2 VIDEO	1280x720 (4:3)	O	X	X		
		720x480 (16:9)	O	O	X		
		544x480 (16:9)	O	O	X		
		480x480 (16:9)	O	O	X		
		352x240 (16:9)	O	O	X		
		720x480 (4:3)	O	X	O		
		544x480 (4:3)	O	X	O		
		480x480 (4:3)	O	X	O		
		352x240 (4:3)	O	X	O		
		MPEG 1 video		352x240 (16:9)	O	O	X
				176x120 (16:9)	O	O	X
				352x240 (4:3)	O	X	O
				176x120 (4:3)	O	X	O

(O:Presentable X:Not presentable)

(\*4) Scaling factors are defined as follows by combinationsof MPEG pixel size and plane.(Appendix-4)

- HD video data presentation in video planes of 1920x1080 pixel size or scaling ratio when presenting SD video data in 720x480(16:9,4:3) = consideration of video data presentation by 128/128(100%) is defined as follows:

- 1) Quantize video data in each respective picture element of MPEG pixel size
- 2) Quantized video data is presented in the following number of horizontal/vertical pixels on the video plane:
  - Number of vertical direction picture elements is the same as vertical direction picture elements for video data.
  - The number of horizontal picture elements is the aspect ratio of the number of vertical picture elements x video data.

However, video data with an MPEG pixel size of 352x240 gives side panels of 4 picture element size on both the left side and right side respectively, and is presented on the video plane with a horizontal direction total of 360 picture elements. Video data with an MPEG pixel size of 176 x 120 gives a 2 picture element size for side panels and is presented on the video plane with a horizontal total of 180 picture elements. When side panels are given by receiver units this way, the display format will be implementation dependent. Therefore, it is recommended to hide the side panel parts using another plane in multi-media content.
  - Video data of 1280x720(16:9) will be a scaling ratio = 128/128 by mapping (vertical picture element count 1080, horizontal picture element count 1920) in the video plane of 1920x1080 (16:9) pixel size.
- HD video data presented in a video plane of 1920x1080 or consideration of video data presentation by a scaling ratio =  $n/128$  in case of presenting SD video data in 720x480(16:9,4:3) is as follows.
  - 1) Determine video plane pixel size. Determination method of the video plane pixel size is as follows;
    - Determine data broadcasting pixel size based on “document\_resolution” specified by the BML document.
    - Determine video plane pixel sizes corresponding to data broadcasting pixel size
  - 2) Quantize video data to the picture element of each respective MPEG pixel size.
  - 3) In accordance with the logic written previously, calculate vertical/horizontal picture element count in case of scaling ratio = 128/128.
  - 4) Present quantized video data in the following vertical horizontal picture element count on the video plane.
    - Vertical direction picture element count is the same as vertical picture element count of video data specified in height property for object elements in the BML document.
    - Horizontal picture element count is vertical picture element count x aspect ratio of video data. If the horizontal vertical direction becomes a non-even picture element count at this point, then the picture elements are rounded by truncating. Truncating is done from picture elements with a larger number of picture elements (lower right) for both horizontal and vertical. (Refer to ARIB STD-B24 for the picture element count) However, presentation where horizontal picture elements do not match the value specified in the width property of the object element is implementation dependent.
- Logic for scaling ratio when displaying SD video data on a video plane of 1920x1080 pixel size is defined as follows:

Define with flexibility to change to a video pixel size of 960x540 data pixel size and also with consideration to secure image quality of the video.

Definition to secure flexibility to video pixel size change:  
First of all, define the scaling ratio =128/128 as follows:

  - 1) Quantize video data in each respective MPEG pixel size picture element.
  - 2) Map the quantized video data for the vertical horizontal picture elements on the following video plane:
    - Vertical picture elements are converted to vertical picture elements on the video plane.
    - Horizontal picture elements are converted to a picture element count that maintains the roundness ratio against the vertical picture elements above.

Next, in case the scaling ratio =  $n/128$ , use the above scaling ratio = 128/128 as the standard, and the vertical picture element count is the same as the vertical picture elements for video data specified by the height property for object elements in the BML document.

- The horizontal picture elements are the vertical picture elements x aspect ratio of video data. If the horizontal vertical direction becomes non-even picture elements at this time, then the picture elements are rounded by truncating. Truncating is done from picture elements with a larger number of picture elements (lower right) for both horizontal and vertical. (Refer to ARIB STD-B24 for information on picture elements) However, the presentation of horizontal picture elements, which do not match the value specified in the width property of object elements is implementation dependent.

Definition with considerations to secure the picture quality of video:

First of all, define the scaling ratio =128/128 as follows:

1) Quantize video data to picture element of each MPEG pixel size.

2) Map quantized video data to the following vertical horizontal picture elements on the video plane.:

- Vertical picture elements are vertical picture elements on the video plane.
- The horizontal picture element count is converted to picture elements that maintain the roundness ratio against the above vertical picture elements.

Next, in case the scaling ratio = n/128, then use the above scaling ratio = 128/128 as the standard and set the vertical picture element count to the same number as the vertical picture element count of video data specified by the height property for object elements in the BML document.

- The horizontal picture element count is the vertical picture element count x aspect ratio of video data. If the horizontal vertical direction becomes a non-even picture element count at this point, then the picture elements are rounded by truncating. Truncating is done from picture elements (lower right) with a larger number of picture elements for both horizontal and vertical. (Refer to ARIB STD-B24 for information on the picture element count) However, the presentation for when horizontal picture elements do not match the value specified in the width property of object elements is implementation dependent.

However, in either case, video data in 352x240 MPEG pixel size should be given a 4 picture element size side panel on both the left and right sides to make the horizontal size 360 picture elements, and also video data in 176x120 MPEG pixel size should be given a 2 picture element sized side panel on both the left and the right sides, and after editing the horizontal length total of 180 picture elements, apply the above definitions.

(\*5) 256/128 scaling is operated only when transmitting pictures of a 960x540 pixel size and presenting it in a picture element divided into 2 equal-size displays as 1920x1080 size pictures.

(\*6) A receiver can optionally use 8-bit character codes that are transmitted by data carousels : using a stream type identification=0x0D. However, do not externally refer to 8-bit character codes included in BML documents.

#### 1.2.4 Audio playing function

Regulations regarding playback of audio are shown in Table 1-5. This assumes that mono-media coding which is not described here, and mono-media data which is encoded by operations outside these provisions will not be sent or operated by broadcasters. Detailed descriptions of contents of the provisions regarding respective encoding methods are defined in Chapter 3.

**Table 1-5 Audio playing function**

Encoding method	Description of provision	
AAC-LC	Transmission method	Audio PES;Stream format identification = 0x0F Data carousel; Stream format identification = 0x0D
	Sampling frequency	48kHz,32kHz (*1)
	Maximum file size possible for continuous play	512KB
	Others	(*1)(*2)
AIFF-C	Transmission method	Data carousel; Stream format identification = 0x0D
	Sampling frequency	12kHz(*) 1/4 of main audio stream
	Maximum file size possible for continuous play	96KB
	Others	(*1)(*2)
Closed caption warning sounds	Transmission method	Built-in sound(*3)
	Sampling frequency	12kHz
	File size possible for continuous play	48KB

(\*1) For restrictions, such as whether or not simultaneous decoding of video data and/or additional audio data is possible, see chapter 3.

(\*2) For decoding file format data transmitted by data carousels, allows silent time when the sound repeats.

(\*3) Size of total ROM for built-in sound is 480KB.

### 1.2.5 Fonts

Regarding fonts, considering the size of ROM featured in receiver units, restrictions have been added without affecting practicality. Font specifications are shown in Table 1-6.

**Table 1-6 Fonts**

Item	Description of provision
Number of font styles	Number of styles:3(round gothic, bold round gothic, angle gothic) (*1)shared for 960x540 and 720x480 Proportional font: none
Character type	Kanji (1st, 2nd level), Hiragana, Katakana, alphanumeric, symbols, etc. (*1)
	External characters: YES
Character size controls (pixel)	Round gothic 16, 20, 24, 30, 36
	Bold round gothic 30
	Angle gothic 20, 24
Grayscale font	4 shades

(\*1) Refer to 3.4.1.2 for details on character types

(\*2) Character decoration such as hemming is assumed to be software implemented, but it is not mandatory. Details on implementation including whether or not to use exclusive fonts, is outside of the scope of this document.



### 1.3 Remote control

#### 1.3.1 Keys used in data broadcasting

The types of remote control keys used in data broadcasting and guidelines for content production are shown in Table 1-7. In order to avoid user confusion, multiple meanings should not be assigned to one button. When assigning multiple meanings to one button, operation content should be explicitly explained to the user within the contents. Having a bookmark button is preferred for remote control used in data broadcasting. Featuring a bookmark button is optional.

**Table 1-7 Remote control keys used for data broadcasting**

Key type	Guidelines
↑、↓、←、→ (up, down, left, right keys)	To move up, down, left, right.
0 - 9 (number keys)	To input numbers
Enter	Separator of operation (enter)
Return	Cancel operation
	Back space of user input character (or bulk erase)
	Disconnection of a call to a communication server (*):During connection, receiver units will take the instruction; after connection, instruction is carried out in the contents. (A display to the effect that the connection will be terminated is desirable when the back key is pressed.)
	(*):It is okay to use BML documents for the purpose of going back. However, whether or not there is something after returning should be considered.
d	Data button. Switches display/non-display of multi-media data broadcasting.
Blue, red, green, yellow(color keys)	Selection of operation (execution) (*):Location of buttons on the remote control should be in order of blue, red, green, yellow from the left and each button should have the corresponding words "blue", "red", "green" and "yellow" displayed.
Bookmark (Optional)	Recording of bookmark.

#### 1.3.2 Key masks

Multi-media contents are in compliance with ARIB STD-B24 and are able to perform masks for keys. However, keys related to selecting stations (one-touch select button, channel up/dedicated button, screen image key) should not be masked by contents except during on-line communication. Masks on number keys (one-touch select button) should not be performed unless number input is necessary. Masks should be released once the input is over.

### 1.4 Memory that should be installed in receiver units

Refer to section 1.1.2 for the definition and location within the receiver units for each memory in basic receiver units

### 1.4.1 RAM

As shown in the reference model, a receiver unit will feature various memories. In this section, among all the memories that are assumed to use RAM, Bcontents, XfrBA will be defined in particular. Each RAM size is shown in Table 1-8. Refer to section 1.1.2 for details.

**Table 1 -8 RAM**

Item	Description of provision
Bcontents	More than 5MB
XfrBA	512KB

For the buffer size for caption and superimpose, refer to chapter 4.

### 1.4.2 NVRAM

As primary nonvolatile memory featured in receiver units related to data broadcasting, there is BproNV which stores unique information on each receiver unit user and broadcaster which is indicated in the reference model. The area type and capacity that BproNV should have is shown in Table 1-9.

**Table 1-9 Type and capacity of BproNV**

Type	Capacity
A-profile memory area for the all broadcasters	2KB
A-profile memory area for the affiliation	More than 32KB
A-profile memory area for the specified broadcaster	More than 48KB
A-profile memory area of communication purpose for the specified broadcaster	More than 24KB
Memory area for bookmark service	More than 16KB
Memory area for root CA certificates	24KB
Memory area for registration transmission	More than 4.5KB

For details of BproNV, refer to 5.2 or 5.15. The area indicated in Table 1-9 should, from the confidentiality perspective, be constructed so that viewers cannot read or write using other functions of receiver units not defined in standards of related documents or in provisions of this document or by using devices connected to receiver units. In order to satisfy the above restrictions, the corresponding areas should be located in the memory of the NVRAM, etc. receiver unit.

Refer to 1.1.2 “Receiver unit reference model” for details.

A device with limits on the number of times of writing for NVRAM equipped in receiver units is used. Function failure will occur on those devices once the writing times exceed its limit, and as a result, the lifespan of this receiver unit becomes shorter. Therefore, adequate attention should be paid to avoid exceeding the amount of writing to NVRAM. Regarding this issue, it is explained in Appendix-6.

## 1.5 Communication functions

The communication functions assumed in interaction channel communication for data broadcasting services are shown in Table 1-10.

**Table 1-10 Communication functions**

Item	Description of provision
Protocol	<ul style="list-style-type: none"> <li>TCP/IP communication protocol is used. Refer to Vol. 6, Chapter 6 for details on the TCP/IP communication protocol,</li> <li>HTTP1.1(RFC2616) is used as the application layer.</li> <li>Receiver units equipped with modems should support the BASIC system procedure (Code Independent Mode) (optional)</li> </ul>
Security function	In case a scheme is specified by http: for BML contents, then TLS security should be followed. Refer to Vol. 6, Chapter 8 for details on the TLS operation.*
Communication speed	Speed faster than 28.8kbit/s is preferred.

\* It should be noted that the TLS1.2 operation has been added in version 5.5.

## 1.6 Character entry function

The character entry function, assuming there is a software keyboard, etc. for the purpose of supporting character entry to BML contents by viewer operation, is defined as a resident application. Functions that this application should be equipped with as standard are described in the following sections. Functions not defined in this document such as the *kana kanji* conversion function are implementation dependent.

### 1.6.1 Function specifications

This application is to support character entry to “input” elements in BML documents. An “input” element of which “readonly” attribute is not effective must be focused on for this application to start up, and this application must behave as specified by the attributes of the input element, described as below..

- “inputmode” attribute

Specifies whether or not text input frontend application should be started, and also specifies the format of the startup.

In case “direct” or “indirect” is specified in this attribute, whether or not “click”, “keyup”, “key down” events will occur by pressing the “enter” key is implementation dependent.

**Table 1-11 “inputmode” attribute**

Attribute value	Contents
direct	The “input” element is focused on and after application of the focus style and occurrence of the focus event, the text input frontend application is started immediately. (*1)(*2)
indirect	The “input” element is focused on and after application of the focus style and occurrence of the focus event, the text input frontend application is started upon pressing of the enter key.(*2)(*3)
none	Text input frontend application is not started.(Behaviour of the input element is in compliance with ARIB STD-B24)

- (\*1) In case a focus transition to the “input” element has occurred by pressing down a key, (i.e. the focus is moved according to "nav-index" property or the focuses() method is applied to the input element by “onkeydown”, “onclick” event handler)whether or not the processes defined in [6] of ARIB STD-B24 Vol. 2 Appendix 1 5.4.2.1 is executed is implementation dependent. The "onkeyup" event handler should not be specified for “input” elements for which the “inputmode” attribute is specified. For details, refer to section 5.7.3.
- (\*2) In case a focus transition to the “input” element has occurred by the access key for the “input” element, whether the key event processes defined after [6] of ARIB STD-B24 Vol. 2 Appendix 1 5.4.2.3 will occur or not is implementation dependent. In other words, in case “indirect” attribute is specified for the “input” element, there may exist receiver units starting up the application immediately by the pseudo "click" event, as well as receiver units that will not startup the application. Also, in case “direct” attribute is specified, the behaviour of receiver units relating to the key event after [6] of the above specification is implementation dependent. Therefore, the behaviour of receiver units in case the access key is specified for an “input” element for which “inputmode” attribute is specified is implementation dependent. For those reasons, contents should not set up the access key for “input” elements for which “inputmode” attribute is specified.
- (\*3) The above provision is not a restriction for the text input frontend application to be started by a key other than "enter". In other words, receiver units equipped with buttons exclusively for the character entry function for users to control the start and/or end of the text input frontend application may be designed to start the text input frontend application by one of those buttons, as long as startup by the “enter” key is supported. Whether or not to employ such buttons is outside of the scope of this document.

- “charactertype” attribute

Specifies the character type available for input.

This attribute is valid only when “direct” or “indirect” is specified in the “inputmode” attribute.

In case an external keyboard, etc. is equipped, it should be reflected in the operation. For details of character type, refer to the next section.

Also, guidelines for the character type specifications during contents writing is defined as follows.

When “password” is specified in the “type” attribute of the “input” element, either one of “number”, “alphabet”, “hankaku” should be specified in the “charactertype” attribute of the corresponding “input” element.

**Table 1-12 “charactertype” attribute**

Attribute value	Contents
all	No restrictions
number	Only single byte numbers
alphabet	Single byte(alphabet + symbols)
hankaku	Single byte characters(alphanumeric, symbols)
zenkaku	Double byte characters (Hiragana, katakana, alphanumeric, symbols)(*1)
katakana	Double byte characters (katakana + symbols)
hiragana	Double byte characters (hiragana + symbols)

(\*1) For the selection of *kana*, it is recommended to make “*hiragana*” as the default.

For the above attributes, DOMBML interface is defined as the following in addition to ARIB STD-B24 Vol. 2 provisions.

**Table 1-13 Operation of attributes in the DOMBML interface**

Interface	Attribute/method	Operation	Comments
BML Input Element			
	style	-	
	normalStyle	O	note 1)
	focusStyle	O	note 1)
	activeStyle	O	note 1)
	inputMode	-	
	characterType	-	

Note 1) Refer to ARIB STD-B24 Vol. 2 Appendix 2

Ending of this application should be executed prior to the occurrence of the “change” event when focus is shifted from the corresponding “input” element. For details of focus control, refer to ARIB STD-B24 Appendix 1.

Furthermore, in case there are restrictions in character entry by dynamic changes of the “readonly” attribute, or in case invisible attributes becomes valid, or in case the browser finishes, this application ends. In these cases it is implementation dependent whether or not the text that has been entered to the application is implemented to the “value” attribute.

Texts entered by users must be implemented to the “value” attribute in EUC-JP encoding when this application is finished, before deciding on whether or not a change event occurs on the input element. It is implementation dependent whether or not incorporation to the “value” attribute while this application is running.

The event handling and script execution must be progressed while this application is running, except in the case of the provision of key event described later. Therefore, contents authors may expect that script executions by the event handler by bevent is processed even while this application is running. This means that, for example, processes from the disposal of documents

to re-presentation in case of the data event update defined in chapter 2 of this document must be executed by the browser even while this application is running.

Details on the user interface of text input frontend application , including graphical design, remote control operation and whether or an external keyboard is assumed or not, is outside of the scope of this document unless the following provisions are satisfied.

- Two startup methods in accordance with the value of “inputmode” attribute must be able to be selected for startup of the application.
- Type of input characters must be able to be regulated by the “charactertype” attribute
- All characters described in the next section must be supported.
- All key events necessary to the application must be passed to it, regardless of the keygroup specified by the “used-key-list” property in the BML document that is currently presented, and those key events should not be used for purposes other than character entry.
- In case “password” is specified for the “type” attribute of the “input” element, it is recommended that the application displays the entered text in a non-plain form, just as in the “input” element itself.
- Sizeable display area of the application is, regardless of data broadcasting pixel size, the area where y coordinate value is 110 or greater. Outside of this areas, displays to interrupt contents presentation are not allowed.

### **1.6.2 Character type**

Character types that the text input frontend application should be supported are described as follows. Supporting of character types other than those is implementation dependent.



only when the value of the “charactertype” attribute is “all” in accordance with the restrictions of input characters by the “charactertype” attribute.

### 1.7 Bookmark list display function by receiver’s native application (Optional)

It is implementation dependent whether a receiver has the bookmark list function as a resident application. However, when the convenience of viewers who use bookmarks is considered, the bookmark list function for receiver’s native applications should be equipped as a simple selection method of communication contents assuming the TCP/IP communication protocol introduced in digital terrestrial broadcasting. The minimized functions assuming installation of the bookmark list function of receiver’s native application are indicated in the following. Functions and contents not defined in this document are implementation dependent.

Regarding the bookmark list by receiver’s native application, refer to ARIB STD-B24 Vol. 2 Appendix 1, “8.4 Guidelines on bookmarks”, “Informative Explanation 5 Bookmark services”.

(Note) Trademark rights of the “BOOKMARK/ブックマーク” regarding general electronics (registration of trademark in Japan No. 3281163)

#### 1.7.1 Start/End bookmark list by receiver’s native application

- Startup of the bookmark list by receiver’s native application is done by executing the receiver features for remote control buttons, and functions of multi-media contents (startResidentBookmarkList). Regarding this function, it is explained later in this document.
- In case of starting the bookmark list by receiver’s native application during the presentation of multi-media contents, the display of multi-media contents can be terminated. However, the performance of multi-media contents should be continued.
- In the bookmark list screen display method, there are, for example, a method to display by shrinking the screen image and a method to  $\alpha$ -blend and overlap the bookmark list screen on top of the screen image. In case of a data broadcasting service screen (excluding superimpose) rather than a screen image, then the graphic character display of the data broadcasting service and bookmark list could conflict in some cases. Therefore, for data broadcast services when the bookmark list screen is displayed, the display of the data broadcast service screen may be terminated in some cases within a reasonable degree.
- While bookmark list screen is displayed, remote control, etc. key input is obtained by the bookmark list.
- The bookmark list is closed at the time of transition to linked destination by selection of viewer.
- When presentation of multi-media contents is paused and the bookmark list is displayed, the bookmark is selected by the viewers and probably will not move to other contents, and



when the bookmark list is closed, the paused multi-media contents for display are re-displayed and performance should be continued. Key input also returns to multi-media contents.

- When the bookmark list is started during presentation of multi-media contents, since the data broadcasting browser continues its performance without synchronizing with the bookmark list, the screen presented when multi-media contents are re-displayed after the bookmark list is closed depends upon the performance of the data broadcasting browser executed during bookmark list presentation. For example, if the data event is changed during bookmark list display, then the display contents of data broadcasting and display contents at the time of closing the bookmark list are different. Even when the data screen is displayed, data broadcasting may not be presented after the bookmark list is closed in some cases.
- User interface and design are implementation dependent.

### **1.7.2 Bookmark list functions to be equipped by receiver's native applications**

The bookmark list by receiver's native application should be in accordance with ARIB STD-B24 Vol. 2 Appendix 1 "8.4.3.5 Operation of bookmark list service".

In order to realize the bookmark list by receiver's native application, the following functions are necessary.

- Display function of recorded bookmark titles
- Function for viewers to select desired bookmarks from a list and to select stations and transit in accordance with specifications of the linked destination URI and URL.
- Delete management function

The minimum contents that receiver units should satisfy for respective functions are listed below. Contents that are not described in this section are implementation dependent.

#### **1.7.2.1 Display function of recorded bookmark titles**

The minimum items and contents displayed by the bookmark list for bookmark information recorded in the area for bookmark service of NVRAM are described in this section. Also, display of items other than the ones listed here, bookmark list, display method and user interface are implementation dependent.

The area for bookmark service of NVRAM is divided into a basic data section and an expansion data section. The minimum items that should be displayed in the bookmark list are the following items.

**Table 1-15 Items that receiver application should present**

Minimum items that should be presented	
Basic data section	Title
Expansion data section	Bookmark description("bookmark_information")
	Linkdestinationchannel information("channel_information") Or memorandum("memorandum_information")

Among the items above, explanation of bookmark, linked site information and memo information are assumed to be displayed after viewers select bookmarks from the list using buttons, however, in such an event, it is recommended that all information be displayed on the same screen.

The relationship between the bookmark type and items are as follows:

- Present the title of basic data section regardless of bookmark type.
- Bookmark type (bm Type) that presents the expansion data section is as follows.
  - Favorite channel B(bmType=001)
  - Favorite site B(bmType=003)
  - Related link service B (bmType=005)
  - Related link site B(bmType=007)
- Memorandum (bmType=008)
- Reserved area I for common bookmark types(bmType=009~099)
- Reserved area I for proprietary bookmark types(bmType=200~299)
- For a receiver's application(900 - 999)

In bookmark types that belong to reserved area I for common bookmark types or Reserved area I for proprietary bookmark types, the component of the sections equivalent to “bookmark description” and “link destination channel information” in expansion data section should be fixed.

Also, regarding reserved area II for common bookmark types(bmType=100~198), reserved area II for proprietary bookmark types(bmType=300~899), since components of the expansion data section are not fixed, the display of the expansion data section of bookmarks of this type by the resident bookmark list function is implementation dependent.

### **1.7.2.2 Functions that allow viewers to select desired bookmarks from a list and to select stations in accordance with specifications of the link destination URI**

Based on bookmark information provided by the above display functions, selection of or transition to the service, component and communication contents (link state, non-link state) specified by the link destination URI (dstURI) of the basic data section of the area for bookmark

service is possible. The URI specified by the link destination URI (dstURI) may be up to module and resource specified, but the bookmark list function by receiver's native application interprets only up to component\_tag as shown below. In case transition to contents of specification for the module below is desired, read the bookmark information in the startup document in the transit destination, and transition is possible.

- arib://<original\_network\_id>.<transport\_stream\_id>.<service\_id>
- arib-dc://<original\_network\_id>.<transport\_stream\_id>.<service\_id>  
  [;<content\_id>][.<event\_id>]/<component\_tag>
- Name space that begins with http, https

Selecting the station and transition performance for URI's other than the above are receiver unit-dependent.

The desired operations in receiver's native applications at the time of selecting stations and transition are indicated in Table 1-16.

**Table 1-16 Transit action from the bookmark list**

Transit destination	Desired performance
Service	In compliance with selecting station operation of the receiver units
Component	In compliance with selecting station operation of the receiver units
Communication contents of link state	Transition to communication contents is always in non-link state; therefore transition to communication contents from receiver's native applications in link state is not possible. (When a transition to link state is expected, it is done by processing on the contents side) For example, the destination of transition recorded in a bookmark is set as the component of data broadcasting, which is the originating link, and transit to desired communication contents from the startup document of the corresponding destination of transition in link state. In such an event, the URL of the desired communication contents obtains the bookmark number from the contents of Greg block 0 written by the receiver's native application, and reads and uses the URL information written in the bookmark.
Communication contents of non-link state	In case of reception condition of data broadcasting, the link state or non-link state, the performance is in compliance with 5.14.12 Guidelines of receiver unit performance when receiving communication contents and figure 5-6 transition D, transition F, transition H. Other cases will be receiver unit-dependent, however, presented status should be presented assuming non-link state.

Note: The receivers can optionally support the unlinked state.

### 1.7.2.3 Delete management function

Regarding the delete management function, functions indicated in ARIB STD-B24 Volume 2 Appendix 1 "8.4 Guidelines on bookmarks" need to be able to (1) perform manual deletion of

bookmark information, (2) perform automatic deletion of bookmarks, (3) turn ON/OFF the flag to prohibit deletion by viewer's instruction. For details, refer to ARIB STD-B24, Vol. 2 Appendix 1 "8.4.3.5 Operation of bookmark list service."

### **1.7.3 Startup of the bookmark list by receiver's native application by function**

Regarding startup of the bookmark, other than starting by receiver units, it can be started by calling the receiver's native application from contents using functions. Startup by receiver units is outside of the scope of this document, however, for startup by contents, the bookmark list can be directly started by executing "startResidentBookmarkList()". A receiver's having the resident bookmark list function implies the support of this function, but it is optional, therefore, supporting of the function should be confirmed using getBrowserSupport( ) before calling this function. For details on this function, refer to 5.6.9 of this document.

## **1.8 BML browser**

Regarding functions that BML browsers should be equipped with, hardware resources and software resources are defined in this chapter. Starting from the next chapter, transmission, mono-media, multimedia encoding will be defined, however, the specifications that are not included in those will be defined in this section.

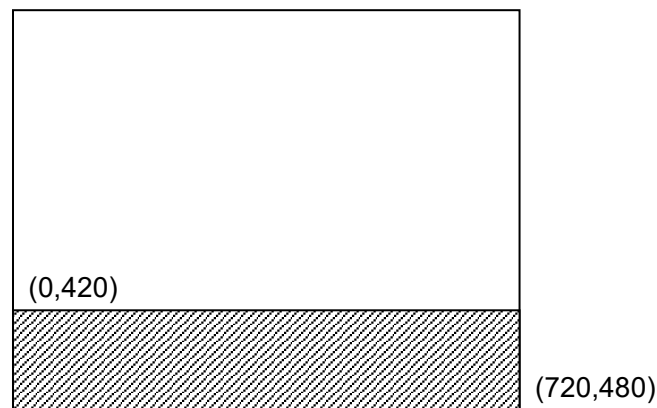
### **1.8.1 Display by browser features**

Browsers should not display anything that may possibly cover up the presentation that contents intend to be. This is a restriction to avoid missing a part of contents with display by browser features such as buttons for users to control the browser and/or the URL frame, which are common in browsers on PCs. However, this does not mean the restriction of the receiver's resident application used for data broadcasting such as text input frontend at the time of starting the browser or displaying channel banners at the time of startup of the browser. Also, in case of presenting communication contents in non-link state (\*1), browsers should display something of its own explicitly in the areas shown in figure 1-7 and 1-8 indicating that the receiver unit is in non-link state. Details on the expression and display methods are implementation dependent, but they should be obvious enough for viewers to understand that contents they are viewing are outside the responsible area of broadcasting contents.

(\*1)For the non-link state, refer to the definitions and chapter 5.



**Figure 1-7 Display when the data broadcasting resolution is 960x540**



**Figure 1-8 Display when the data broadcasting resolution is 720x480(16:9, 4:3)**

### 1.9 Registration transmission function (Optional)

The registration transmission function is comprised of functions to send superimpose recorded in the area for registration transmission by Multimedia services and functions to set up superimpose in the area for registration transmission by Multimedia services, and contents which use registration transmission function or receiver's native applications with registration transmission function of Multimedia services in case superimpose that are necessary during programs that cannot be sent due to congestion or restrictions on the number of reception lines at the interaction channel center, or in case of sending superimpose in bulk in interaction channel programs. The details of provisions relating to receiver's native applications providing registration transmission functions are defined in section 5.16.5 of this document.

The registration transmission function by receiver's native application is optional. However, it should be implemented for the following reasons.

- Since the registration transmission function by Multimedia service (contents) requires the viewer to be watching the registered channel at the time of specifying the call, and during program viewing, the viewer needs to operate the remote control for calling, the ratio of actual calls may be extremely limited, and it may be unavoidable for viewers to feel annoyance and dissatisfaction. Therefore, we should conclude that relying on only the registration transmission function by contents is not appropriate for the objective to have as many viewers as possible use the interaction channel function.
- Registration transmission by the Multimedia service is expected to be operated differently for each broadcasting station, and if there are a lot of cases where calling does not happen because of contents problems, then some trouble involving viewers is assumed.
- On the other hand, the registration transmission function by receiver features can be called by targeting the time zone the viewer is not viewing, and even if the call fails once, it is possible to retry by judgement of receiver unit, therefore, the possibility of a successful call is high and it fits the benefit of viewers. Also, operations by viewers at the time of the call are not necessary, and the usability for viewers is high as well. Furthermore, in case of calling by receiver features, it has nothing to do with defects of contents and there will be no problems caused by different operations of registration transmission by each transmission broadcaster.

## 2 Operation of the data transmission method

In this chapter, based on the contents of specifications in applicable documents listed below, for newly specified items, differences and restrictions for operation related to the transmission method of the XML-based multimedia encoding method will be mentioned. In this chapter, data broadcasting services refer to multimedia data broadcasting services by the XML-based multimedia encoding method unless specified otherwise. For items without a specific description, a description of specifications in applicable documents is applied as they are.

- ARIB STD-B10 "Service information for digital broadcasting"
- ARIB STD-B24 "Data coding and transmission specification for digital broadcasting"

### 2.1 PSI/SI

#### 2.1.1 Types of data broadcasting services

##### 2.1.1.1 Data programs and TV programs

Data programs and TV programs are distinguished as follows.

TV programs: Programs without a Data Component Descriptor located in the components, which is described first in PMT 2<sup>nd</sup> loop

Data programs: Programs with a Data Component Descriptor located in the components, which is described first in PMT 2<sup>nd</sup> loop

Furthermore, TV programs are classified as "TV program with additional data", which has additional data in the same event and "normal TV program" without additional data. In this document, "TV program" refers to "normal TV program".

Regarding data programs, only "independent data program", which is meant to be viewed on its own is specified.

##### 2.1.1.2 Types of data broadcasting service programs

Program types for programs to operate data broadcasting services are classified as follows.

**Table 2-1 Data broadcasting service program types**

Program types	Definitions
TV program with additional data	Among TV programs, programs with a Data Component Descriptor located in components other than the first PMT 2 <sup>nd</sup> loop.
Independent data program	Broadcast as a service of service_type=0xC0( data service), 0xAA(bookmark list data service), and programs with a Data Component Descriptor located in the first PMT 2 <sup>nd</sup> loop component.

### 2.1.1.3 “service\_type” of channels that operate data broadcasting programs

“service\_type” of service that broadcasts data programs is indicated below.

- 1) TV programs with additional data are broadcast by service\_type=0x01 (TV service) channels.
- 2) Independent data programs are broadcast by service\_type=0xC0, 0xAA(bookmark list data service)channels.

## 2.1.2 Configuration of contents of data broadcasting service and operation of components

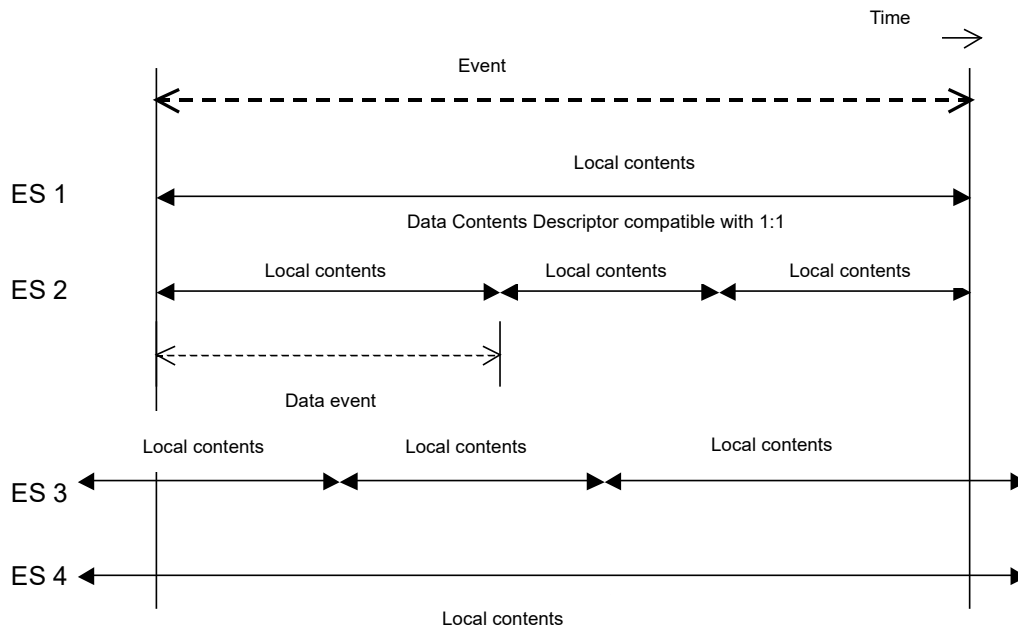
### 2.1.2.1 Contents and local contents

The relationship between events and definitions of contents and local contents are shown in Table 2-2 and Figure 2-1.

**Table 2-2 Definitions of contents and local contents**

	Definitions	Operations
Contents	see the group of local contents that are transmitted during certain event periods in certain components. Identified by a Data Contents Descriptor.	Operation is never specified for the contents.
Local Contents	Group of BML documents that are transmitted in certain data events of certain components.	Sometimes they are broadcast by sequentially switching multiple local contents along with the passing time within a component.





**Figure 2-1 Local contents and events**

### 2.1.2.2 Relationship between local contents and ES

One location contents is transmitted by one data carousel (ES) and local contents may refer to ES, which transmits video, audio, and event messages in some cases. The stream ID of components that transmit data carousels and event messages is always 0x0D.

### 2.1.2.3 Operation of component tags

Operation of the component tag value used in data broadcasting including closed caption and superimpose shall be in accordance with Vol. 7, section 5.1.1 (1). However, reference of the ACC audio component in a 24 kHz sampling frequency, which is transmitted by the partial reception layer, is optional.

### 2.1.2.4 Identification of entry components

The default component of data broadcasting programs is always transmitted by component with tag value of 0x40. The identification of entry components, which transmit startup documents on tuning, is not done by the entry\_point\_flag of the Data Component Descriptor but it is done by this tag value. Entry components transmit the entry module (moduleId=0x0000) and the entry module always includes one startup document.

In case of multi-view programs, a component with a tag value of 0x40 is always the entry component.

### 2.1.2.5 Restrictions on the number of ES

Considering the restrictions on the hardware resources of receiver, the total number of ES's that can exist in one service (1 PMT) is 16 ES's. (For example, in case of video 1, audio 2, closed caption 1 and superimpose 1, then 11 can be used for the multi-media data component.) However, 16 ES's includes ES's that transmit PCR's as well.

Also, the number of ES's that transmits data carousels is 8 ES's and the maximum number of ES's that transmit only event messages is 4 ES's.

Also, in 1 service, the maximum number of ES's that have set a valid ECM PID is 12.

At the time of multi-view operation, this restriction is applied to each component group specified in the Component Group Descriptor.

### 2.1.2.6 Detailed specifications of section data transmission

- Multi section transmission (transmission of more than two sections in 1 TS packet) is not done in data carousel and event message transmission.
- The following standards are set for the transmission bit rate when transmitting data of each section of data carousels and event messages.
  - Do not send more than 6 packets continuously using the same PID.
  - The total bit rate of components received (maximum 7 PID, refer to 2.3.6.1) when receiving one content is less than 4Mbit/s. This includes DII, DDB and event messages.
  - Additionally, the maximum bit rate per one sub-table is less than 2Mbit/s (8KB±100% per 32 milliseconds)
  - When transmission exceeding those standards is executed, then the reception efficiency of sections may decrease for some receiver, and the time required to obtain necessary sections may become longer, so it is recommended that the above standards are not exceeded for transmission.

### 2.1.2.7 Default maximum bit rate of data broadcasting programs

Shall be in accordance with Vol. 7

### 2.1.2.8 Operation of video and audio components in data broadcasting services

In the following descriptions, video/audio streams in 1PMT are classified as follows.

TV video/audio:

The video and audio stream is specified in Vol. 7. It becomes the target for playback by EPG, etc. and switching by video/audio buttons. A component\_tag of 0x00~0x0F(video)or 0x10~0x2F(audio)is given and when tuning an event that includes this stream, EPG, etc. will play those video/audio. TV video/audio may be used in independent data programs in some cases.

Video/audio referred to only from data broadcasting:

The video/audio stream is referred to and played only from data broadcasting contents, component\_tag of 0x50~0x7F is given. EPG, etc. will not play and it is not a target of switching functions by the video and audio buttons.

- For either TV programs with additional data or independent data programs, as the AV stream referral method, methods to directly specify the component\_tag value (direct tag specification) and to specify -1 as the component\_tag and refer to the stream that EPG, etc. has chosen (default specification) are both operable.
- Based on the model indicated in ARIB STD-B24 Vol. 2 Informative Explanation 2, when referring to TV video/audio component from data broadcasting contents, the default specification is performed as a basic rule. However, in the following cases, TV video/audio switching operation by remote control key and such receiver functions would not contradict data broadcasting content specifications, therefore, referral by direct tag specification is possible.
  - 1) In case only one TV video component is included in event, direct tag specification can be referred to for the TV video component.
  - 2) In case only one TV audio component is included in an event and if it is not dual mono, then direct tag specification can be referred to for the TV audio component.
- Relating to the above, the following operational restrictions are instituted.
  - 1) At the time of starting the startup documents for independent data programs including TV programs with additional data and TV video/audio, only default specifications can be used. Reason: To prevent the BML-engine from switching video/audio that is being played by EPG, etc. at the time of starting the presentation of startup documents.
  - 2) Referral of TV video/audio from data broadcasting contents in multi-view programs is default specifications only. (Because multiple components of TV video are included)
  - 3) When change to TV video/audio status from the status where video/audio is being played by direct tag specification, both the default specification and direct tag specification can be used. In case there is dual mono TV audio, only default specifications are possible and since the selection status of dual mono of TV audio changed by default specifications is receiver dependent, the channel ID is specified if necessary.

- Operation of related receiver is specified as follows.
  - 1) In case TV video/audio exists, EPG, etc. performs video/audio playback process at tuning in the same way as TV programs with additional data for independent data programs.
  - 2) In case there is only one selection for TV audio, audio is not switched even if the audio button is pressed. It is the same for TV video.
  - 3) When the BML-engine ends, the default TV video/audio is played. However, if data broadcasting contents do not switch the video/audio by direct tag specification (in case of carrying out default specifications only), then the video/audio component that is being played at the time of completion will continue to be played.

### **2.1.3 Series reservation operation in data broadcasting services**

- Refer to 5.9 for operation of the series reservation function from multi-media contents

### **2.1.4 Operation of the PMT specific to data broadcasting services**

- Refer to 2.1.10.3 and 2.1.10.6 for receiver performance on tuning or updating the PMT.
- The only components equipped in the PMT that allow the non-existence of ES are the following 3.
  - Closed caption component
  - Superimpose component
  - Component that transmits event messages only

### **2.1.5 Operation of the Data Component Descriptor in the PMT**

The Data Component Descriptor is located in the following components.

- Components that transmit closed caption and superimpose
- Refer to (chapter 4) the chapter on closed caption and superimpose for operation details of the Data Component Descriptor located in components of closed caption and superimpose.
- Components that transmit data carousels
- Do not locate the Data Component Descriptor in any other components. However, it can be located in components that transmit event messages only.
- Operation of Data Component Descriptor is shown in Table 2-3.

**Table 2-3 Operation of the Data Component Descriptor**

Flag	Operation
data_component_id	0x000C
Contents of additional_data	component_info(additional_arib_bxml_info( ))
transmission_format	It is 00(Data carousel transmission method and event message transmission method).
entry_point_flag	<ul style="list-style-type: none"> <li>• It is always 1 only for component of component_tag= 0x40 (component of component_tag= 0x40 that transmit modules including documents that should be started first when data broadcasting programs are selected.) It is 0 for any other components.</li> <li>• When receiver is tuned, obtain and present startup documents of data carousels transmitted by components of component_tag= 0x40.</li> </ul>
auto_start_flag	<p>It is operated. Follows the specifications of 2.1.10.2 and 2.1.10.3. It is always auto_start_flag=1 in independent data programs.</p>
document_pixel_size	<p>The pixel size and aspect ratio of BML contents are displayed. Only the following 3 types of parameters are operated. 0011: 960X540 0100: 720X480(16:9) 0101: 720X480(4:3) However, refer to 2.1.10.5 for pixel size and aspect control of data broadcasting programs.</p>
use_xml	It is 0.(XML using application dependent tags is not transmitted.)
default_version_flag	It is always 0.
independent_flag	It is always 1.
style_for_tv_flag	Operated as specified. If this flag is specified as 0 (does not have TV as the style and only includes contents that cannot be laid out by TV receiver) then, it is judged as “viewing is impossible”.
bml_major_version, bml_minor_version	Operated as specified. When these fields are arranged, basic receiver assess this as “viewing is possible” when bml_major_version is 3.
ondemand_retrieval_flag	It is always 1.
file_storable_flag	It is always 0.
start_priority	Operated as specified.*
Operation of additional_arib_carousel_info( )	
data_event_id	Not operated in the PMT. Value is fixed to 0xF(1111).
event_section_flag	It is always 1.

\* See 8.2.1.1 for the operation of receivers that support phase 0 operation specifications. Receivers that do not support the AIT-controlled application shall ignore this value.

Phase 0 operation specifications are the specifications for operating with the broadcasting integrated HTML5 application linkage function. For a detailed definition, see Chapter 8.

### 2.1.6 Operation of the Target Resion Descriptor in the PMT

The Target Resion Descriptor is not used.

### 2.1.7 Operation of the H-EIT, M-EIT Data Contents Descriptor

- Refer to chapter 4 for operation of the Data Contents Descriptor regarding closed caption and superimpose.
- In H-EIT, multiple Data Contents Descriptor may exist for one event.

- Even there is an event to execute multi-media data broadcasting services, sometimes the Data Contents Descriptor is not placed.
- Data Contents Descriptor is not always placed for all broadcasting contents.
- The maximum number of Data Contents Descriptors for one event in H-EIT is 32.

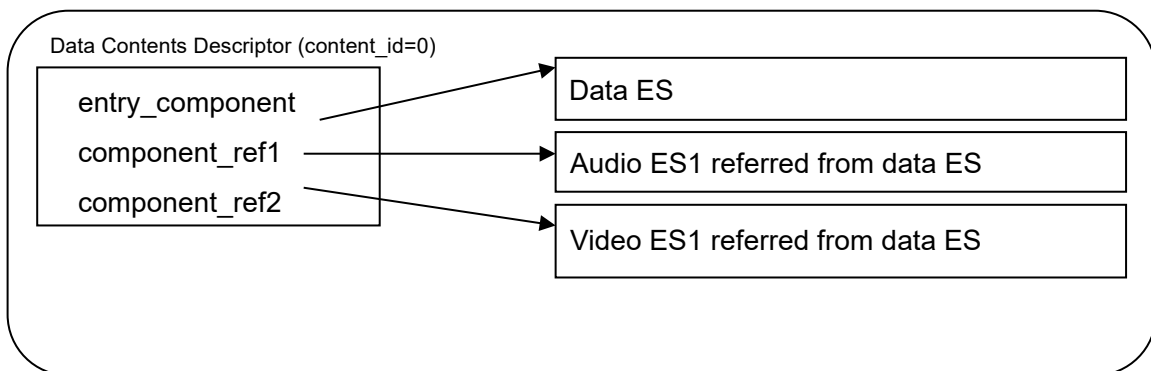
Operation of the H-EIT and M-EIT Data Contents Descriptor is shown in Table 2-4, and the relationship between contents and the ES is shown in Figure 2-2.

**Table 2-4 Operation of Data Contents Descriptor**

Field	Operation
entry_component	Specify components including startup module of corresponding local contents.
num_of_component_ref	“num_of_component_ref” of H-EIT[schedule]M-EIT[p/f after] does not mean anything to receiver, so it is not referred to.
component_ref	Describes component to transmit carousels other than the entry carousel and the component to transmit referred video/audio streams and event messages. H-EIT[p/f], M-EIT[p/f] have established values. Receiver do not have to refer to this field for H-EIT[schedule], M-EIT[p/f after] when recording reservations and viewing reservations are made. Execution is executed with the H-EIT[p/f], M-EIT[p/f] values.
ISO_639_language_code	Fixed in jpn(Japanese).
text_length	The maximum value is 80(bytes). However, in case of displaying the title of contents in receiver, then at least 40 bytes worth will be displayed.
text_char	Describes the title of contents displayed in EPG.
Operation of arib_bxml_info()	
transmission_format	It is 00(Data carousel transmission method and event message transmission method).
auto_start_flag	“auto_start_flag” of EIT is not used. It is always 0.
document_pixel_size	Presents the pixel size and aspect ratio of the corresponding contents. Only the following 3 types of parameters are used. 0011: 960X540 0100: 720X480(16:9) 0101: 720X480(4:3) However, for pixel size and aspect control of data broadcasting programs, refer to 2.1.10.5.
use_xml	It is 0. (XML using application dependent tags is not transmitted.)
default_version_flag	It is always 0.
independent_flag	It is always 1.
content_id_flag	Operated as specified. If “content_id” and “content_version” are located in this descriptor then 1, if not then 0.
associated_contents_flag	Operated as specified. When the receiver displays that data broadcasting associated with TV programs will be executed by EPG, etc., then execute only for the events where the Data Contents Descriptors, which are specified this flag as 1, exist.

Field	Operation
style_for_tv_flag	Operated as specified. When this flag is specified as 0, then it is judged as impossible to view with a basic receiver. (does not have TV as style and only includes contents that cannot be laid out by TV receiver)
update_flag	“update_flag“ is not operated. It is always 0.
ISO_639_language_code	Operated as specified.
content_id	Operated as specified.
content_version	“content_version” is used. However, “content_version” does not change during event time.
bml_major_version, bml_minor_version	Operated as specified. When those fields are arranged, and if “bml_major_version” is 3, then basic receiver will assess this as possible to view.
ondemand_retrieval_flag	It is always 1.
file_storable_flag	It is always 0.
Operation of arib_carousel_info( )	
num_of_carousels	When the number_of_carousels=0, then it shows that all of the following carousel information is undecided. Confirmed information (num_of_carousels≠0)is input in H-EIT[p/f], M-EIT[p/f]. This value is not referred to during recording reservation and viewing reservation.
component_tag	Operated as specified. When contents are selected from EPG, etc., the startup documents of the components specified by this tag value are obtained and presented.
event_section_flag	It is always 1.
component_size_flag	Operated as specified. If the “component_size” field is not located for reasons such as operation is not certain, then it is 0.
default_transaction_id_flag	It is 0(transaction_id is not encoded). That is, to obtain desired “transaction ID” without locating “transaction_id” in this descriptor.
default_timeoutDII_flag	Operated as specified. When this flag is 0, the “timeout_vaule_DII” field may not be located.
default_leak_rate_flag	It is 0(“leak_rate” field is not located). The “leak_rate” is not operated and transmission rate of data carousel is specified in 2.1.2.6.
component_size	Operated as specified and only the confirmed values are located. Confirmed values do not increase or decrease as a basic rule and the change in the direction of the value increase is rejected. When operation is not confirmed, then sometimes location is omitted.
transaction_id	(Not located)

Field	Operation
timeout_value_DII	If the location is omitted, or 0xFFFFFFFF is specified (there's no recommended time out value), then it should be interpreted as 5000 milliseconds is specified. If a carousel with a DII transmission period cycle exceeding 5000 milliseconds is included in this corresponding event, then locate the appropriate value that is suitable for the DII transmission period cycle for the carousel with the longest DII transmission period cycle. Also in some cases, the broadcaster sends out a DII at fixed period cycle regardless of the carousel length and operates timeout_value_DII value as fixed.
leak_rate	(Not located. For the transmission rate of section data, refer to 2.1.2.6)



**Figure 2-2 Relationship between contents and ES**

### 2.1.8 Operation of the Data Contents Descriptors in H-EIT[schedule],M-EIT[p/f after]

- Since the num\_of\_component\_ref in H-EIT[schedule],M-EIT[p/f after] may have unconfirmed values in some cases, do not refer to the receiver.
- In H-EIT[schedule],M-EIT[p/f after], num\_of\_carousel=0 may be set in some cases.

### 2.1.9 Operation of EIT's specific to data broadcasting services

- Information to select or to reserve contents is written in the Data Contents Descriptor. However, some programs that perform data broadcasting services do not have Data Contents Descriptor located.
- Descriptions regarding components of Data Contents Descriptors are fixed during the event time. For operation of various EIT, refer to Vol. 4.

### 2.1.10 Performance of related receiver

#### 2.1.10.1 BML-engine startup

When the data broadcasting program is selected and the auto\_start\_flag of the Data Component Descriptor located in the PMT entry component (component\_tag=0x40) is equal to 0 (Only TV program with additional data is specifiable), then the broadcasting engine should not start immediately, and the startup process of the BML-engine starts when the viewer presses



the d button. If the auto\_start\_flag=1, then the BML-engine startup process starts without waiting for the button to be pushed by the viewer. For the flow up to the start of the BML-engine, refer to the next section.

### 2.1.10.2 Receiver operation when data broadcasting programs start

Prior condition of data transmission operational specifications

- “data\_event\_id” located in the PMT Data Component Descriptor is fixed (0xF) and is not used.
- Component of the “component\_tag=0x40” is set as an entry component, and data carousels transmitted by entry components are called entry carousels.
- An entry component exists at all times. PID values do not change except for exceptional cases. However, the following exceptions are assumed. Including cases other than the following example, performance of receiver when PID values are changed is specified in 2.1.10.3.
  - When changing from a multiple channel arrangement to a single channel arrangement in case of a spotted arrangement
- One entry component exists per one event.

Basic operation of receiver when selecting channels (Guidelines)

- 1) When ECM is specified in PMT 1st loop, this process is carried out. Perform the following process only if the program becomes viewable by confirming the contract status.
- 2) If TV video/audio are included in the PMT 2nd loop (TV video :component\_tag=0x00-0x0F, TV audio :component\_tag=0x10-0x2F), then it will be played.
- 3) Entry components(component of component\_tag=0x40)are identified from components located in the PMT 2nd loop.
- 4) If the data\_component\_id of the Data Component Descriptor for the entry component is 0x000C, then it is judged as a data broadcasting program by the XML-based multimedia encoding method, and the process below is followed. If it is a data encoding method that the receiver is not compatible with, then the data broadcasting service does not start.
- 5) Refer to the “auto\_start\_flag” of the Data Component Descriptor for the entry component and if it is 1, then execute the following process immediately. If the “auto\_start\_flag” is 0 then, the following process is done when the viewer presses the dButton.
- 6) Perform the following assessment process from 7 to 10 and judge whether or not the BML-engine startup should be performed.
- 7) Based on the BML / B-XML version number specified in the Data Component Descriptor of the entry component, judge whether or not the presentation of data is possible. If viewing is judged as impossible, then the presentation of the data broadcasting service is not carried out.

- 8) If the presentation of data broadcasting service is judged as possible in 7, then the BML-engine is started and after initialization of the BContents and Ureg is carried out, the startup documents of the entry component are obtained and presented. Also, the root certificate should be re-obtained. (Refer to 5.14.14.3)
- 9) However, after the BML-engine is started, if the entry component is an empty carousel, then perform the following. (Refer to 2.3.1.6 for empty carousels)
  - When the “auto\_start\_flag=0”, the engine start process is terminated as soon as the entry carousel is detected as empty, and it returns to standby to wait for the viewer to press the d Button. (Step 5 above)
  - When the “auto\_start\_flag=1”, then keep monitoring the data event switch of the entry component, and obtain and present the startup document as soon as the data event switch occurs and the startup module appears.

Receiver behaviour in case the component group (sub-channel) of multi-view programs is switched. (Guideline)

- Presentation of data broadcasting does not change even if the group is switched. (Presentation of the document that was presented shall continue even after switching)

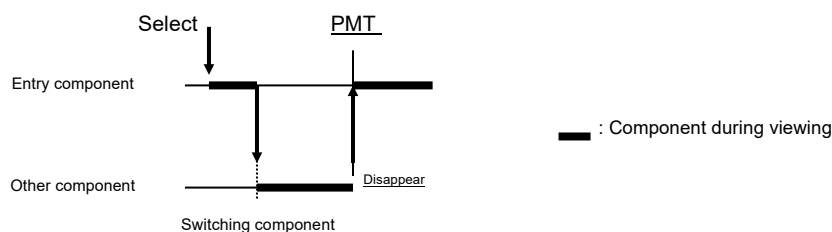
Receiver behaviour when contents are selected from EPG, etc. and to view or to make a viewing reservation. (Guidelines)

- 1) Based on the Data Contents Descriptor of the EIT, EPG, etc. present content candidates to viewers. In such an event, based on the “on demand viewing possible/impossible information” located in the Data Contents Descriptor, whether the corresponding data broadcasting program is viewable or not by the receiver is judged, and when it is impossible to view, display an explanation and do not use it as a selection target.
- 2) In case of programs that are currently broadcasting, the component specified by the “entry\_component” of contents selected by the viewer is set as the starting component and the viewing of data broadcasting program is started. In other words, the BML version and limited reception for corresponding components is judged and if the corresponding component can be viewed, the startup documents are obtained and presented.
- 3) In case of starting the viewing of programs that were reserved for viewing, the performance of 2 above from the start time of the program is executed.

### 2.1.10.3 Receiver behaviour when updating the PMT

Receiver behaviour when updating the PMT occurs during the viewing of data broadcasting programs:

- In case the components during viewing disappear.
  - Discard the documents that are currently presented and start the presentation of the startup document of the entry carousel.



**Figure 2-3 Behaviour when the components during viewing disappear due to the PMT update.**

- In case the entry component has disappeared.
  - Since it is no longer data broadcasting program, the BML-engine ends.
- In case the PID of the component to transmit the data carousel (including empty carousel) during viewing or return flag monitoring has changed.
  - The receiver judges the contents of data program as changed and performs a process that is equivalent to re-selecting station for its own self-services described below.
    - Discard documents that are currently presented.
    - Delete all resources that are already acquired from BContents memory.
    - Re-acquisition and presentation of startup documents of the entry component.
- In case the value of the “auto\_start\_flag” is changed from 0 to 1.
  - If the BML-engine has not been started, then item 4 and the following items of “Basic operation of receiver when selecting channels” described in 2.1.10.2 should be performed.
    - (\*) However, depending on the display status of the receiver, the BML-engine may not start.
- In case the value of “auto\_start\_flag” is changed from 1 to 0.
  - The status of BML-engine does not change.

Receiver performance in case the update of the PMT occurs while viewing TV programs other than TV programs with additional data:

- In case the entry component appears
  - It is interpreted that the data broadcasting program has started, and item 4 and the following items of “Basic operation of receiver when selecting channels” described in 2.1.10.2 should be performed. However, depending on display status of the receiver, the BML-engine does not have to start.

#### 2.1.10.4 Handling of the dButton

There are two patterns of use for the dButton which is equipped in remote controllers.

- BML-engine startup instruction

For TV programs with additional data where a forced start is not specified, (auto\_start\_flag=0 is specified by the Data Component Descriptor of the entry component) it is used by viewers to start the data broadcasting service. When such a program is tuned, the BML-engine startup process is started by pressing the dButton.

- Contents control after BML-engine startup

After BML-engine startup, events of pressing the dButton are all processed by the BML-engine and in BML documents, it is obtainable as an event (type attribute is "DataButtonPressed") if needed. Processes when the dButton is pressed are scripted as multi-media contents. However, the usage of the dButton is as a basic rule a toggle operation for the ON/OFF display of data broadcasting services related to viewing TV programs. This is a restriction on contents production.

For reasons listed above, the dButton cannot have functions to force exiting of the BML-engine. Whether or not such methods should be realized as a separate receiver features or remote control button depends on a model.

An example of using the assumed dButton is shown below:

(Example of assumed usage 1)

Contents of data broadcasting services are not displayed immediately after tuning, and contents viewing is started when the viewer presses the dButton.

- 1) In the Data Component Descriptor of the entry component, "auto\_start\_flag"=0 is specified
- 2) After selecting a TV program with additional data, the BML-engine startup process is not immediately performed
- 3) Once the pressing of the dButton by the viewer is detected, then the receiver performs the BML-engine startup process and after the BML-engine is started, obtains and presents the startup document.
- 4) In the startup document, the menu display of multi-media contents, etc. is carried out and the presentation of the data broadcasting service starts.
- 5) Pressing of the dButton during the presentation of the data broadcasting service is communicated to multi-media contents and all character shapes will be deleted by contents control or icons, etc. will be displayed.
- 6) When pressing of the dButton is detected again, it returns to the data broadcasting service presentation status before deletion by contents control.

(Example of assumed usage 2)

In case of displaying data broadcasting service contents immediately after the station is selected.

- 1) "auto\_start\_flag=1" should be specified in the Data Component Descriptor of the entry component.
- 2) After selecting the station of the data broadcasting program, the BML-engine startup process is performed and the startup document is obtained and presented.
- 3) In the startup document, the menu display of multi-media contents, etc. is executed and the presentation of data broadcasting service starts.
- 4) From here on, the equivalent processes for the above usage examples 1-5 and after are executed.

(Example of assumed usage 3)

In the case BML-engine is started immediately after selecting the station, but contents of the data broadcasting service are not displayed on the screen.

- 1) "auto\_start\_flag=1" should be specified in the Data Component Descriptor of the entry component.
- 2) After selecting the station of the data broadcasting program, the BML-engine startup process is executed and the startup document is obtained and presented.
- 3) At the initial status of the startup document, only TV video and audio are presented in full size, and it enters standby to wait for the viewer to press the dButton.
- 4) Once the pressing of the dButton by the viewer is detected, the menu display of multi-media contents by contents control is performed and presentation of the data broadcasting service starts.
- 5) From here on, the equivalent processes of the above usage examples 1-5 and after are executed.

#### **2.1.10.5 Pixel size and aspect control in data broadcasting programs**

The receiver controls the pixel size and aspects based on "document\_resolution" specified in the BML document instead of "document\_resolution" of the Data Component Descriptor of the PMT or Data Contents Descriptor of the EIT. For details, follow 1.2.2 and 1.2.3.

#### **2.1.10.6 Receiver operation when selecting stations**

The receiver behavior list when tuning is shown in Table 2-5.

However, data broadcasting programs should be viewable by the judgment of the BML version.

**Table 2-5 Receiver behavior when selecting stations**

Selection method	Operation selection	Receiver performance
Channel selection of remote control	TV program with additional data is selected.	In case of "auto_start_flag=1", presentation of the startup document of the entry component of TV programs with additional data is started. In case of "auto_start_flag=0", presentation of the startup document is started after waiting for the viewer to press the dButton.
	Independent data program is selected.	The selected independent data program is started.
Select from EPG (Selection of current broadcasting program)	Specify the contents and select.	Regardless of the value of auto_start_flag, the specified contents are started. When displaying the title of the contents, display at least 40 bytes.
When executing viewing reservation	The same operation as channel selection.	

#### 2.1.10.7 Reservation related behaviors of data broadcasting programs(Guidelines)

- When the viewing reservation is set
 

In the same way as normal programs which do not conduct data broadcasting services, execute the reservation registration for viewing the start time, etc. of the event. In case the Data Contents Cescrptor is located in H-EIT[schedule]M-EIT[p/f after] at the time of the reservation, the following functions can be specified to the viewer.

  - Contents selection function
 

Set up function to present selected contents first instead of the entry component of the PMT at the time of reservation execution.
- When the recording reservation is set
 

The bit rate of required events in the mode selection of recording media or spectrum securement is determined and based on specifications of 2.1.2.7. Others will be at the same time as when viewing reservation is setup.
- When viewing reservation and recording reservation are conducted
 

At the starting time of the reserved event, the presentation and recording of data broadcasting based on PMT is started. If the contents are not specified at the time of setting up the viewing

reservation, then presentation of startup document of the entry component is started. If specified, presentation of the startup document for specified contents starts.

### 2.1.10.8 Specifications when the partial transport stream is output/input

When the receiver output data components(component tag value 0x40~0x7F) to the partial transport stream, the descriptor shown in Table 2-6 should be output along with it. (As specified in Vol. 2)

**Table 2-6 Descriptor outputting to partial TS**

Descriptor name	Written table	Location standard
Data Component Descriptor	PMT 2 <sup>nd</sup> loop	Required
Data Contents Descriptor	SIT 2 <sup>nd</sup> loop	Required
Extended Broadcaster Descriptor	SIT 2 <sup>nd</sup> loop	Required
Broadcast ID Descriptor	SIT 2 <sup>nd</sup> loop	Required

- Operation when functions that require SI information which is not included in the partial transport stream while multi-media contents are being played with partial transport stream as input depend on models. For example, the following are examples of extended functions for broadcasting.
  - EPG functions: Selecting station(epgTune), reservations(epgReserve, epgRecReserve)
  - Operation control function: Confirm whether or not it is during broadcasting(isBeingBroadcast)

### 2.1.10.9 the desired display of EPG, etc.

- Whether or not the data program display will co-exist with the radio TV section or whether it will have an EPG exclusively for the data program depends on a model.
- The desired EPG display related to data broadcasting services is shown in Table 2-7.

**Table 2-7 Desired EPG display**

Display to show that related data broadcasting services exist.	Display of programs with additional data	Based on the associated_contents_flag located in Data Contents Descriptor of the EIT, display to indicate that there are additional data associated with TV programs in the TV program chart or display while stations are selected.

## 2.2 Operation of the independent PES transmission method

Refer to chapter 4 for the transmission method of closed caption and superimpose.

Independent PES transmission will not be executed for other components.

## 2.3 Data carousel transmission method/event message transmission method

### 2.3.1 Data carousel transmission operation

- Maximum number of modules sent per 1 data carousel is 256.
- Configuration of Module of data carousels may change during events with time. (Module may be increased/decreased) in some cases. When it changes, then the DII version is updated.
- Module that composes the data carousel may have a different transmission frequency depending on the module.

#### 2.3.1.1 Introduction of data events and local contents

- The concept of data events which do not have any direct time relationships with events is introduced in order to make switching of contents possible regardless of during or in between programs, and the switching of contents presentation is performed in the data event unit. Contents transmitted within 1 data event are called local contents. (Refer to figure 2-5)
- Data events are identified by “data\_event\_id” of DII.

#### 2.3.1.2 Operation of data events

- “data\_event\_id” is updated when local contents are switched. In other words, “data\_event\_id” is different before and after switching local contents. (It is not always increased by 1) (Figure 2-4)
- When ES is paused (Component Descriptor disappears from PMT), “data\_event\_id” is not always updated before or after the pause. Starting of ES means the beginning of new local contents. “data\_event\_id” before ES pause does not have to be recorded in receiver. (Figure 2-4)
- Manage/update “data\_event\_id” by each component. “data\_event\_id” is operated by values other than 15 (0xF).

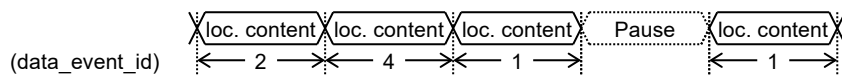


Figure 2-4 Local Contents and “data\_event\_id”

#### 2.3.1.3 Start/End of local contents

- When “data\_event\_id” of DII is updated, it is recognized as switching of local contents and the document that is currently presented is destroyed, and the startup document of new local contents is obtained and presented. (Refer to 2.3.1.7)
- Start and end of local contents occurs in sync with the update of “data\_event\_id” in the DII of the carousel that is currently being viewed as a basic rule.



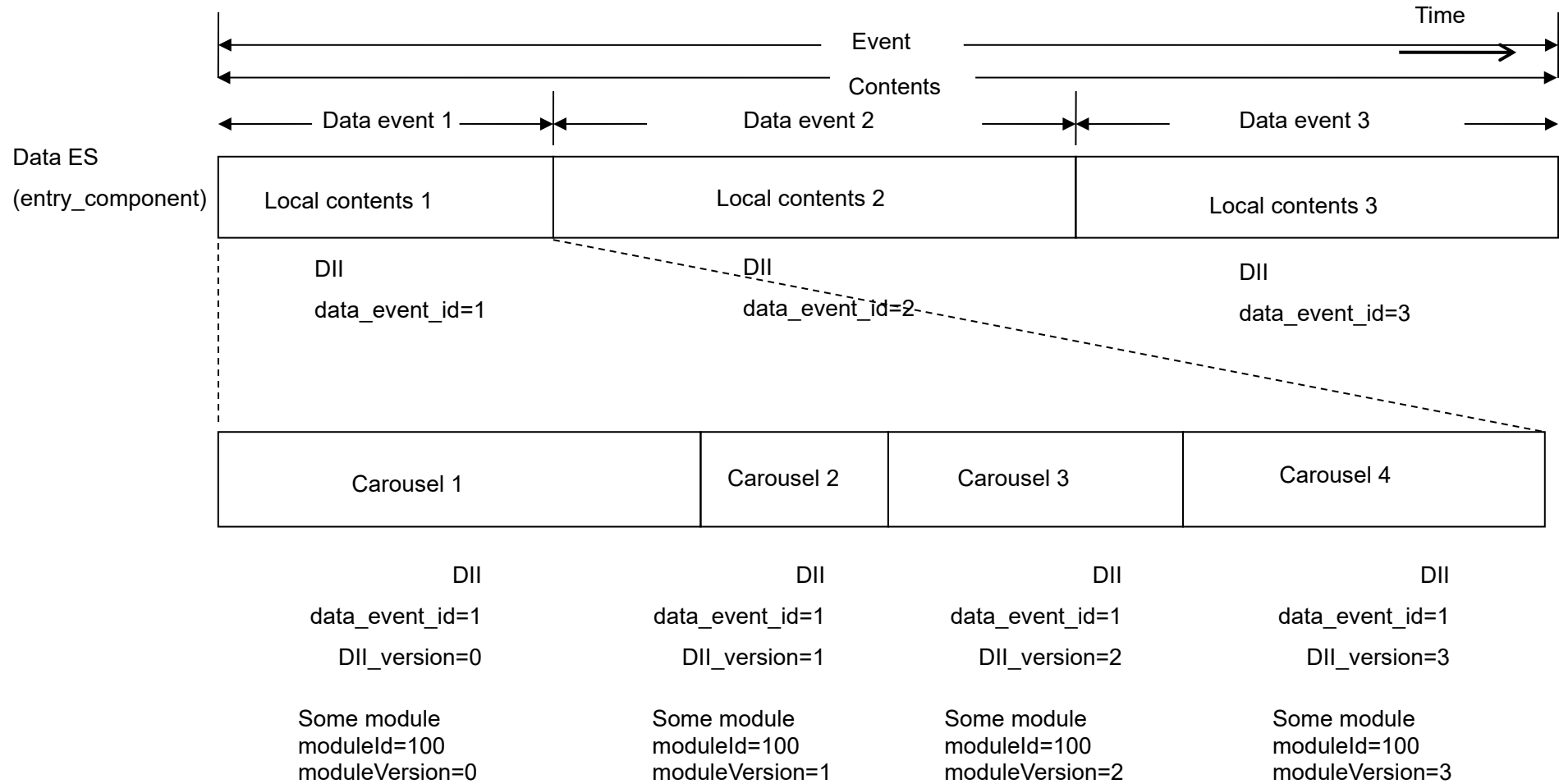
#### **2.3.1.4 Introduction of the return flag**

- In sync with the local contents switch of the entry component, there are some cases in which it should be forced to move to local contents of entry regardless of viewing local contents. For that purpose, a flag (return flag: "return\_to\_entry\_flag") to notify this to the personal information area of the DII of the entry carousel should be located and receiver will monitor this regardless of which local contents of the component are presented.
- The return flag is not located in the DII of the data carousel transmitted by components other than the entry component.

See 7.2.3.3 for the operation of the return flag in the case of a receiver that supports the IPTV download/VOD (integrated service type 2) function.

#### **2.3.1.5 Local contents and the Data Contents Descriptor**

In case multiple contents are transmitted during the event period, the Data Contents Descriptor describe the information for a group of multiple local contents.



- If any one of the modules in DII is updated (module version is upgraded, or increased/decreased), then the version of DII is updated.
- Cut line of the data event does not always occur at the cut line of event.
- For local contents whose transmission period matches the event period, data contents descriptors are handled 1:1.

**Figure 2-5 Data events and local contents**

### 2.3.1.6 Operation of empty carousels

- Data carousels composed only of DII's whose "numberOfModules" field is 0, and do not include DDB's are called empty carousels.
- Empty carousels are used only to switch the data sending/stopping of components without changing the ES description in the PMT in case certain components are needed only for a specific event period etc.
- When non-empty data carousels are switched with empty carousels, the "data\_event\_id" is switched.
- The minimum transmission frequency specification of DII (section 2.3.2) is applied even when sending empty carousels.
- A version up of DII may occur while sending empty carousels. Also, an update of the data event may occur.
- When empty carousels are detected during tuning(when "auto\_start\_flag"=1) or by a switching data event, then the update of the DII is monitored without any errors, and the presentation of the startup document begins when the startup document appears. In case the entry component is an empty carousel when the BML-engine is started by the pressing of the dButton by a viewer in TV programs with additional data of "auto\_start\_flag"=0, then the BML-engine startup is aborted and returns to the standby status to wait for the dButton to be pressed.
- When carousels including documents that are currently being viewed are switched to empty carousels during the presentation of multi-media contents, an event ("DataEventChanged" event, identified by status=1) to indicate "to be changed to an empty carousel" will occur for the BML document. This performance is the same whether it's entry component or not, or regardless of the value of the auto\_start\_flag.
- When empty carousels in components transmitting specified BML documents are transmitted in the document launch where BML documents for the transition destination are specified just as launchDocument(), then they are processed as errors equivalent to when reference modules are not included in normal carousels.
- When empty carousels are transmitted by specified components in document transition where only components are specified without specifying the BML documents of transit destination just like the execution of "epgTuneToComponent()" or transition to entry component by return flag, then the same operation as detecting empty carousels by switching the data event is executed. (However, "DataEventChanged" does not occur.)

### 2.3.1.7 Basic behavior of receiver during data broadcasting program presentation

- Receiver monitor DII of carousels transmitted by the following component at all times.
  - 1) DII of components including modules currently being viewed.

2) DII of entry components

The DII of entry components is monitored at all times to judge whether or not forced process to return to the entry carousel should be performed at the time of switching local contents of the entry component.

Receiver monitor the DII of carousels transmitted by the following components.

3) DII of components transmitting modules specified in "lockModuleOnMemoryEx()"

4) DII of components transmitting modules subscribing to "ModuleLocked" or "ModuleUpdated" event.

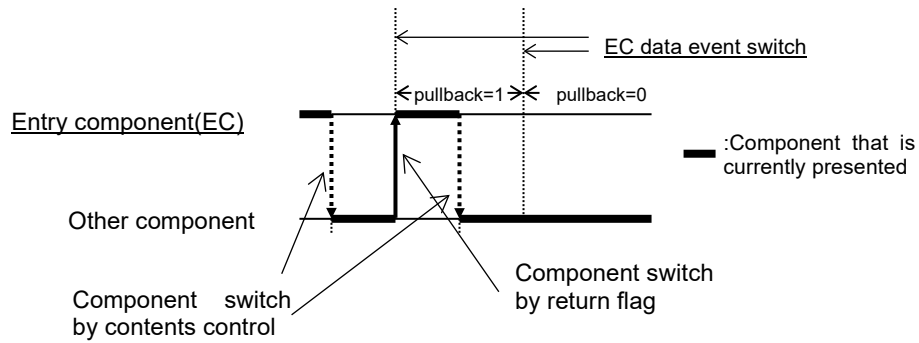
Components that can become the target of module lock by "lockModuleOnMemoryEx()" are limited to components with a "component\_tag" value of 0x40, 0x50, 0x60. (Refer to section 5.12.6.9) Also, components that can become the monitoring target of "ModuleLocked" / "ModuleUpdated" event are limited to components that are currently presented and components with a "component\_tag" value of 0x40, 0x50, 0x60. (Refer to section 5.13.3). Therefore, the receiver need to be able to obtain DII and modules in parallel with a maximum of 4ES (components with a "component\_tag" value of 0x40, 0x50, 0x60 and components that are currently presented).

- Receiver processing when the data event that is currently viewed is switched.

The BML-engine will make the data event message "DataEventChanged" occur for documents that are currently being presented. And then, documents that are currently presented are destroyed and the startup documents included in moduleID=0x0000 module are obtained and presented. When there is an empty carousel after the switching data event, then monitor the update of DII of corresponding components after discarding the document that is currently being presented, and as soon as it is no longer an empty carousel after the data event is switched again, the startup document is obtained and presented. However, synchronized/non-synchronized events that have occurred after the occurrence of "DataEventChanged" are discarded except for the "unload" event. Also, if "epgTune" / "epgTuneToComponent" / "launchDocument" / "launchDocumentRestricted" / "reloadActiveDocument" function is used in the "DataEventChanged" event handler then do not obtain the startup document and follow the specifications of the function.

- When "data\_event\_id" of the monitored entry component is updated while viewing the desired data component, then process as follows.
- When the return flag of DII of entry component is 1, then execute the process to return to entry component. In other words, the module that's currently presented is destroyed and the process to obtain and present the startup document included in the module of moduleID=0x0000 for the entry carousel is executed.(Figure 2-6) In such an event, the "DataEventChanged" event does not occur.

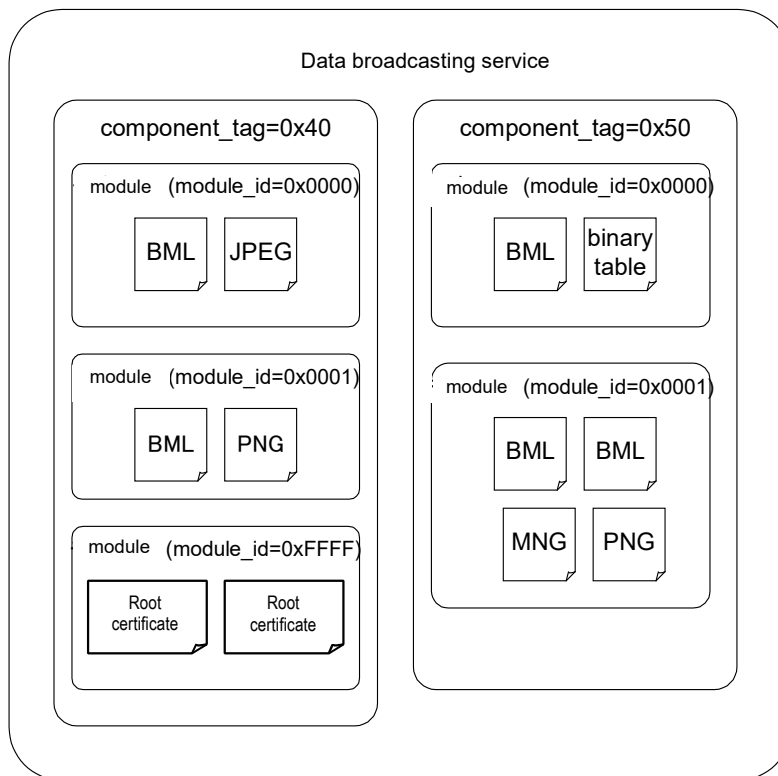
- When the return flag of DII of the entry component is 0 then continue to present the document that is currently presented.(Figure 2-6)
- In all cases, when viewing documents of the entry component, perform the “receiver process when the data event of components that are currently being viewed switches” in the above.



**Figure 2-6 Update of “data\_event\_id” and return flag of entry components**

### 2.3.1.8 Transmission of root certificates

Root certificates are transmitted by data carousels as one of the resources that comprise data broadcasting services. Figure 2-7 is a composition example of a data broadcasting service when transmitting a root certificate. The figure is an example of transmitting 2 root certificates, however, the number of root certificates can be 1.



**Figure 2-7 Transmission of root certificates**

Broadcasters who provide programs that include interaction channel services using TLS or SSL should attach a root certificate at all times when broadcasting corresponding data broadcasting programs. Particularly for programs that register bookmarks or register calling using TLS or SSL should have root certificates attached for transmission. Also, it is recommended to check whether the required certificate is stored in receiver or not before registration. This is, however, except for cases where empty carousels are sent out by components of component\_tag=0x40. Also, considering the receiver performance indicated in section 5.14.14.3, it is recommended to perform operations to store appropriate and the most recent general-purpose root certificates within receiver.

When transmitting root certificates, carry out the following steps.

- Use module\_id=0xFFFF modules for component\_tag=0x40 components to transmit root certificates. Store the resources which display root certificates in this module in an entity format. For details on module components refer to Vol. 6.
- In order to make the loading volume of the filtering process of root certificates lighter, a Root Certificate Descriptor is located in the DII module information area(moduleInfoByte).

### **2.3.1.9 Transmission operation related to broadcasting integrated HTML5 applications (phase 1 [see Chapter 8])**

When the receiver supports broadcasting integrated HTML5 applications, the AIT is transmitted using an arbitrary component module with a module\_id=0x9FFF. For data broadcasting, a module\_id=0x9FFF cannot be used for any purposes other than AIT transmission. The operation of a receiver when data for data broadcasting is contained in a module with such module ID shall be a matter of product planning.

### **2.3.2 Operation of “DownloadInfoIndication”(DII)message**

- Refer to figure 2-5 for the relationship between the DII and the local contents/data event.
- Due to restrictions on the processing ability of receiver, the minimum transmission interval of each DII component is constantly 100 milliseconds (excluding when the carousel is switched, etc.).
- “DSMCC\_section” to transmit the DII is operated as specified.
- Module information stored in the DII message is stored in an ascending order for “moduleId”. (However, “moduleId” is not always continuous.)
- Operation of “userNetworkMessage( )” is indicated in Table 2-8.

**Table 2-8 DII: Operation of “userNetworkMessage( )”**

Field	Operation	Comment
dsmccMessageHeader( )		
protocolDiscriminator	Operated as specified.	0x11
dsmccType	Operated as specified.	0x03
messageId	Operated as specified.	0x1002
transaction_id	Operated as specified. Transaction Number(the bottom 30 bits of “transaction_id”) are updated in the following cases. <ul style="list-style-type: none"> <li>• When data events are switched.</li> <li>• At least one module that composes the carousel is updated.</li> <li>• When the module number that compose the carousel is changed. (including vicinity of “numberOfModules=0”)</li> </ul> It is recommended that the value be incremented by 1. (When the value is changed, then the receiver should judge that the contents of the DII have been changed, this is not limited to when it is incremented by 1)	“contents_id” of the Data Contents Descriptor of the EIT is not always updated even when the transaction_id is changed.
dsmccAdaptionHeader( )		
Not operated.		
downloadId	Operated as specified. Updated at the time of switching the data event. bit31-28 data_event_id bit27-0 all 1 Operated for identification and switching of local contents and handling of event messages and local contents.	“Data_event_id” is operated to switch data events and to avoid the wrong reception of event messages in local contents that are next to each other time wise.
blockSize	Operated with a fixed value(4066).	4066
windowSize	Operated as specified.	0
ackPeriod	Operated as specified.	0
tcDownloadWindow	Operated as specified.	0
tcDownloadScenario	Operated. Among the modules that compose the carousel, the module with the longest sending cycle period is written.	The timeout time setup based on this value depends on a model.
compatibilityDescriptor( )	As specified in case of operating without contents.	compatibilityDescriptorLength=2 descriptorCount=0
numberOfModules	The maximum number of modules transmitted by 1 data carousel is 256. Also, to specify that this is an empty carousel, “numberOfModules=0” may be operated. For empty carousels, refer to 2.3.1.6.	
moduleId, moduleVersion	Operated as specified.	
moduleSize	The maximum value of the module size is 1MB. Refer to 2.3.3 for details.	
Module information area	Descriptor described later is located.	



Field	Operation	Comment
Private data area	<p>DII transmitted by the entry component of an event (tag value 0x40) may locate and operate the arib_bxml_private_data_descriptor( ) as specified. This descriptor is not located in DII's transmitted by other components.</p> <p>In case the (1) data event of the entry carousel is switched and (2) this descriptor exists and (3) "return_to_entry_flag" is 1, then the receiver should discard the document currently viewed and the startup document of the entry carousel should be presented.</p> <p>Refer to 2.3.1.4.</p>	<p>descriptor_tag 0xF0 descriptor_length 1</p>
Descriptor stored in the module information area		
Type Descriptor	In case 1 resource is mapped directly in 1 module, placement is required. It is not necessary for modules to store resources in an entity format.	
Name Descriptor	Not operated.	
Info Descriptor	Not operated.	
Module_Link Descriptor	Not operated.	
CRC Descriptor	Not operated.	
Estimated Download Time Descriptor	Operated in some cases. When operated, specify the maximum transmission cycle period of the corresponding module.	Handling by basic receiver is optional.
Expire Descriptor	Not operated.	
ActivationTime Descriptor	Not operated.	
CompressionType Descriptor	<p>Module may be compressed for transmission and in such case, it should be located. When module compression is not executed, then it is not located.</p> <p>Refer to 2.3.3 for details of module compression</p> <p>Operation of compression_type</p> <p>0: Compress module in the zlib format</p>	
Control Descriptor	Operated as specified. When the BML version of the BML document matches the specification of the PMT, then this descriptor is not located. Also, use_xml is 0 at all times.	
RootCertificate Descriptor	Operated as specified.	Refer to Appendix-7

### 2.3.3 Operation of “DownloadDataBlock”(DDB)message

- The “DSMCC\_section” that transmits DDB messages is operated as specified.
- Maximum size of modules transmitted by DDB messages is 1MB. To be more precise, the number of the “DSMCC\_section” transmitting DDB is a maximum of 256.(Maximum module size is 4066 x 256=1040896 byte)
- Module may be compressed in zlib format for transmission. In this case, the “Compression Type Descriptor” is located in the module information area of the DII handling the corresponding module and the “compression\_type” is set as 0. For details on the compression format, refer to Appendix-2.
- In case of compressed transmission of modules, the total module size before compression and after compression should not exceed the maximum module size above.
- Regarding the operation of the DDB(downloadDataMessage( )),
- Indicated in Table 2-9.

**Table 2-9 DDB: Operation of “downloadDataMessage( )”**

Field	Operation	Comments
dsmccDownloadDataHeader( )		
protocolDiscriminator	Operated as specified.	0x11
dsmccType	Operated as specified.	0x03
messageId	Operated as specified.	0x1003
downloadId	Operated as specified.	The same value as “downloadId” of DII is stored.
adaptationLength	In case of a multi-media service, “dsmccAdaptationHeader( )” is not operated.	0
moduleId	The value of “moduleId” is not particularly specified.	
moduleVersion	Operated.	When it is updated, the value is not guaranteed to be incremented by 1.
blockNumber	Operated as specified.	Value obtained from moduleSize/blockSize

### 2.3.4 Operation of event messages

#### 2.3.4.1 Purpose for operating event messages

- Event messages including the General Event Descriptor (hereinafter referred to as general event message) are operated to transmit data simultaneously attached to the event by generating events in multimedia contents specified by non-synchronized periods or time.

- In case of using an NPT as time specification in an general event message, or in case of using an NPT in multi-media contents, an event message including an NPT Reference Descriptor (referred to as NPT reference message from hereafter) is operated in order to notify the receiver of the relationship between the NPT and STC.
- Objective of NPT operation
  - Even when time shifts occur due to baseball game extensions or emergency news, etc., this is used this as a time specification that does not require correction/re-compilation of data broadcasting service contents. (It cannot be handled in relative time from the starting time or MJD)
  - It is the same when there are changes in CM duration during re-runs. (It cannot be handled in relative time from the starting time of the program or MJD)
  - Time precision may need to be 1/10 seconds for quiz shows, etc.(The relative time from the starting time of the program or MJD is not so precise.)
  - Time axis used in accumulated contents is necessary. (It cannot be handled by relative time from the starting time of the program or MJD)

#### **2.3.4.2 Transmission of event messages**

- General event messages are transmitted by components that transmit local contents, which use it in some cases, and may be transmitted by different components in other cases.
- NPT reference messages may be transmitted by multiple-components per one event.
- NPT reference messages and data carousels are not transmitted by the same component.
- “Event\_section\_flag” located in the Data Component Descriptor of the PMT and the Data Contents Descriptor of the EIT is not used. It is always 1.
- In “DSMCC\_section” that transmits event messages, either the General Event Descriptor or NPT Reference Descriptor is located.
- The “last\_section\_number” of the “DSMCC\_section” that transmits event messages is always 0. In other words, the subtable that transmits one event message is always transmitted by 1 section.

#### **2.3.4.3 Transmission of general event messages**

- From BML documents, general event messages transmitted by the same component as local contents which transmit corresponding documents or general event messages transmitted by different components can be used. The “component\_tag” of general event messages that can be specified by 1 BML document is up to 2 different types.
- “private\_data\_byte” is operated.
- The maximum number of General Event Descriptors that can be located in 1 DSMCC\_section is 8.

- In case of executing setup to acquire the desired message\_id while the message\_id=255 or the message\_id is omitted, the “message\_id” is not set up at the same time as explicit specification of message\_id(0~254). Also, in this case, (0~254) should be omitted (or 255 should be specified).
- In case of explicitly specifying the message\_id, the maximum number of general event messages that can be subscribed simultaneously is 16.
- In order to avoid missed messages, a subtable with the same contents may be sent multiple times in general event messages.
- In this case, transmission intervals and the number of transmissions are not specified.
- In consideration of messages missed by receiver, it is recommended that the update interval of the DSMCC\_section, which transmits general event messages within the same ES, be secured for more than 100ms. This indicates the recommended value of the interval between sending of the first DSMCC\_section of any version to the DSMCC\_section of next updated version. This is not to define the sending interval for the DSMCC\_section of different versions that are adjacent to each other at the time of update.
- In general event messages that perform time specification by using the NPT, the sending of event message specified outside the effective period specified in the next section or after the NPT discontinue time specified as firing time NPT are not executed. Regarding NPT discontinuity, refer to the next section.

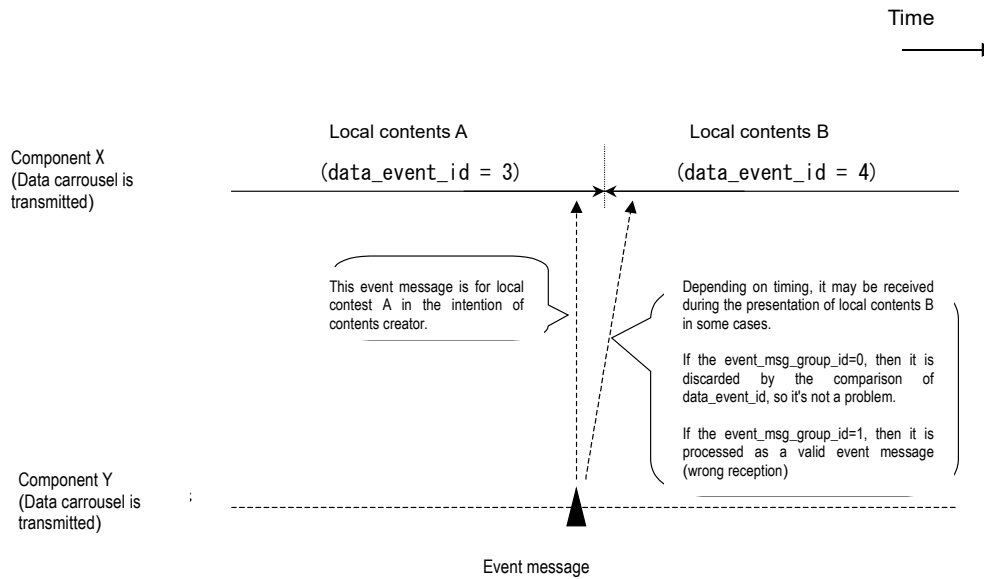
Operation of “data\_event\_id”, “event\_msg\_group\_id” of general event messages

0 or 1 is used for the value of general event messages. The receiver will perform as follows.

- Regarding general event messages with the event\_msg\_group\_id = 0, only when the value of the data\_event\_id of the event message is the same as the data\_event\_id of local contents that are currently presented, they are processed as a valid event message. If the data\_event\_id value is different, then the event message is ignored.
- Regarding general event messages with the event\_msg\_group\_id = 1, regardless of the data\_event\_id value, they are processed as a valid event message.

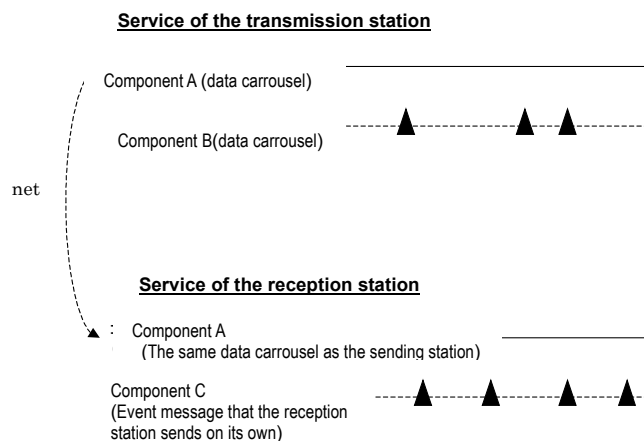
The above definitions are applied in the same way when general event messages are transmitted by the same component as local contents, which it uses, or when they are transmitted by different components.

Normally, specifying the event\_msg\_group\_id = 0 is recommended. This is, for example, just as in figure 2-8, when the event message is sent at a timing that is close to the switching of local contents, etc., adjoining local contents might receive it by mistake when event\_msg\_group\_id=1 is specified.



**Figure 2-8 Receiving general event messages by mistake**

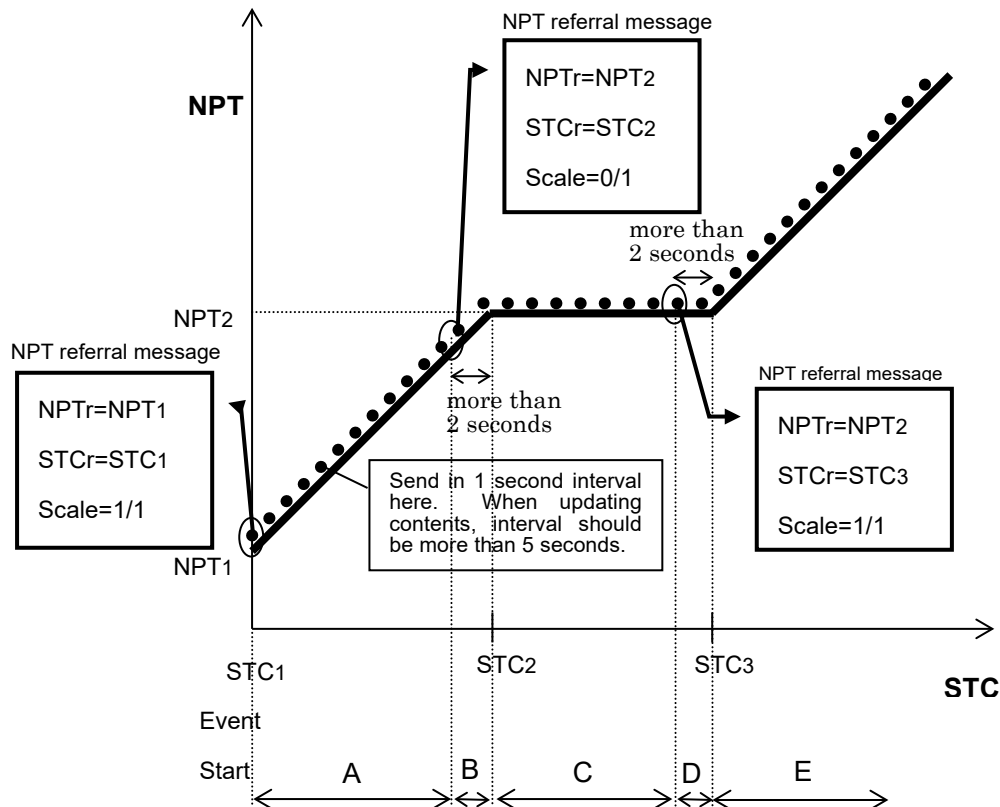
Specification of “event\_msg\_group\_id=1” is used when it is difficult to match the value of the “data\_event\_id” such as, for example, when the data carousel of the transmission station is used and the event message is sent by receiving station. (Figure 2-9)



**Figure 2-9 Example when the value of data\_event\_id cannot be matched.**

#### 2.3.4.4 Transmission of the NPT reference message

- The “table\_id\_extension” of “DSMCC\_section” of the NPT reference message is fixed to 0xFFFF (“event\_msg\_group\_id” = 0xFFF, “data\_event\_id” = 0xF). “event\_msg\_group\_id”=0xFFF is used only for NPT reference messages.
- In case of sending NPT reference messages, the transmission interval is set as 1 second as specified, however, normally the minimum update interval of NPT reference messages is 5 seconds.
- In case of NPT stepping changes, an NPT reference message that notifies stepping changes is transmitted by 2 seconds before it happens.
- Specifications that make the NPT value go around in the middle of the data event should not be performed. However, sending NPT reference messages that make the NPT discontinue during data events is permitted. In this case, the timing of the discontinuity is assumed to be the time of event update, etc. and the frequent occurrence of discontinuity within a short time is not assumed. Regarding NPT related performance in case discontinuity occurs in referred NPT reference messages, refer to the next section.
- In case of using NPT in multi-media contents (in case of event firing by NPT or using getNPT), monitoring of the NPT reference message in multi-media contents is declared explicitly. (Written as event with an “NPTReferred” type attribute)
- Regarding transmission of the NPT reference message method, it is shown in figure 2-10.



**Figure 2-10 Transmission of the NPT reference message method**

- Operations below are executed in order to handle 0 revolutions of the STC.

Regarding the NPT Reference Descriptor to be sent or the NPT event message, send out the ones with STC value specified by the NPT Reference Descriptor or STC value corresponding to NPT that is specified by the NPT event message for the valid period range below at all times. (Refer to figure 2-11). This area is to specify the time range to the current STC value (STCc) and the valid period on the STC time axis is changed as needed with the following formula.

Valid period:  $STCc + STC\_PreWindow \sim STCc - STC\_PostWindow$

However, on the actual STC time axis, the valid area will be  $STCc \sim STC_{max}$  and  $0 \sim (STCc + STC\_PreWindow) - (STC_{max} + 1)$  when the calculation value of  $(STCc + STC\_PreWindow)$  exceeds the STC maximum value ( $STC_{max}$ ), and  $0 \sim STCc$  and  $(STC_{max} + 1) - (STC\_PostWindow - STCc) \sim STC_{max}$  when calculation value of  $(STCc - STC\_PostWindow)$  becomes negative.

It is set as

$STC\_PreWindow = 3888000000$  (12 hours),  $STC\_PostWindow = 3888000000$  (12 hours)

(Hereafter,  $STC\_PreWindow$  is also written as  $W_{pre}$ , and  $STC\_PostWindow$  as  $W_{post}$ )

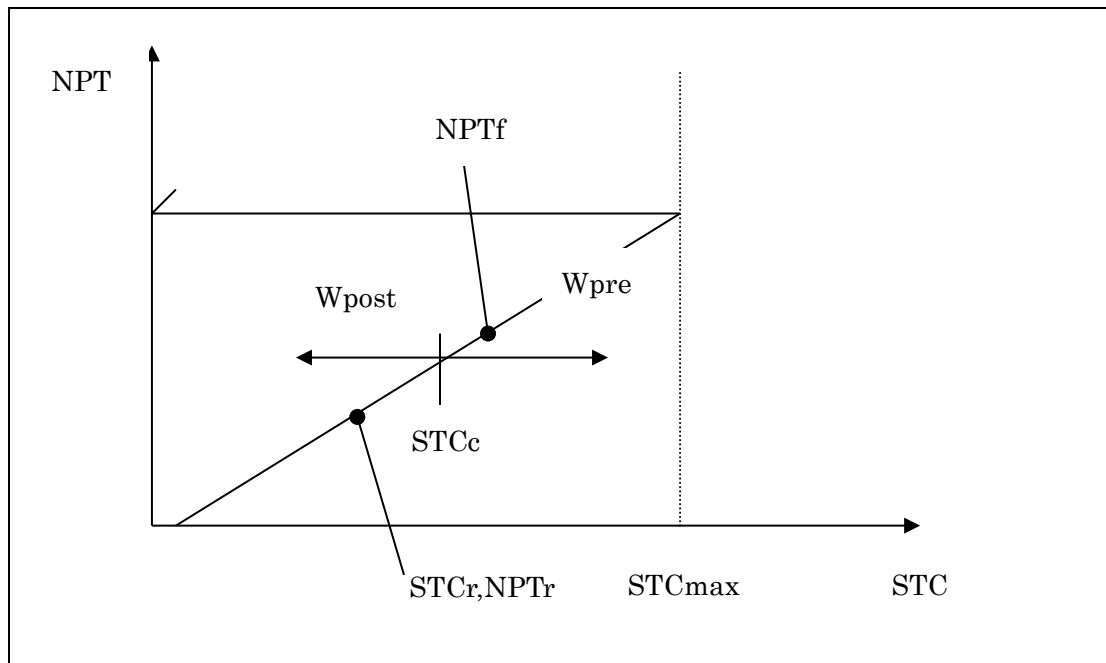


Figure 2-11 NPT valid period

#### 2.3.4.5 Event message process in receiver

##### (1) General event messages

- When an event with a specification of “type” attribute as "EventMessageFired" is written in the BML document, then filtering of the DSMCC\_section which transmits general event messages that match one of the following conditions should be carried out.

##### (Condition 1)

- “component\_tag” specified in “es\_ref” attribute
- “event\_msg\_group\_id”=0x000
- “data\_event\_id” of local contents that are currently presented.

##### (Condition 2)

- “component\_tag” specified in “es\_ref” attribute
- “event\_msg\_group\_id”=0x001
- In case “es\_ref” is omitted, it should be interpreted as a component of local contents that is currently presented.
- At the time of initial acquisition of the DSMCC\_section or version upgrade detection of the DSMCC\_section, the event as specified in the multimedia encoding method based on message\_id/message\_version specified in the BML document and event\_msg\_id of the General Event Descriptor included in the corresponding DSMCC\_section is generated.



- Regarding operated time-mode  
0x00(immediate firing), 0x02(NPT time) are operated as the time mode specifying firing time.
- For firing times (if time mode 0x00, then event message receiving time) specified by general event messages, the receiver will generate events in multi-media contents as quickly as possible after the specified time. A rough standard is 100 milliseconds for both immediate firing and NPT specification.
- When multiple general event messages are received where the time mode is specified as 0 (immediate firing), then they will be fired in the order received. Timing for the process in this case is not specified.
- In receiver, queuing of a maximum of 8 general event messages before firing at the same time is possible.
- When multiple general event messages with immediate firing specifications that are transmitted by the DSMCC\_section are the target, firing is in the order of the descriptors that are located in the DSMCC\_section.
- When the specified NPT time is already passed at the time of reception, it will be fired immediately.
- The valid period of the general event message subscription is the period presented in the document that specified subscribe.  
(2) NPT reference message
- When an interruption event with the "NPTReferred" attribute is described in the BML document, filtering of the "DSMCC\_section" that transmits the NPT reference messages will be performed in the following condition.
  - "component\_tag" specified by "es\_ref" attribute
  - "event\_msg\_group\_id"=0xFFF
  - "data\_event\_id"=0xF
- The above events are generated only when an NPT reference message is acquired for the first time, and events will not be generated even when a version upgrade of the DSMCC\_section that transmits NPT reference messages takes place after that.
- When new NPT Reference Descriptors or General Event Descriptors specified by new NPT's are incoming, then re-calculation of the valid period is done.
- When an NPT Reference Descriptor is received, check whether or not the STC value in the NPT Reference Descriptor is within the valid period. In case it is in the valid period, it is recognized as reception of the NPT Reference Descriptor. When receiving new NPT Reference Descriptors, re-calculation of the valid period is done.)
- After the event of "NPTReferred" is generated, if the NPT reference message that makes discontinuity of stepping for NPT stepping specified in the NPT reference message, which

generated the same event was received, then the NPT related performance in the BML document depends on a model of receiver after that. Even when discontinuity is generated in NPT stepping as above, if an event with an “NPTReferred” attribute is subscribed after updating the BML documents, then execute the process specified in this section not as discontinuity of NPT stepping but as receiving new NPT reference messages. Therefore, for contents production, in order to secure stable performance of contents, transit of documents needs to be performed immediately after discontinuity occurs in NPT stepping. In case of using NPT related functions, NPTReferred interruption needs to be re-subscribed from the document after transition. As a method to transit documents after occurrence of the above NPT discontinuity, for example, event messages of immediate firing can be sent out immediately after occurrence of NPT discontinuity and reception of the event message written in the BML document and execution, etc. of transition along with the firing is assumed.

- For the calculation of NPT, a stepping calculation is done assuming STC 0 revolutions occur just once within the valid period.
- Firing process of NPT event message with firing time specified within the valid period is the same as it has been.(When new General Event Descriptors specified by NPT are incoming, then re-calculate the valid period)
- Regarding event message firing with specified NPT times and firing of events, while taking STC discontinuity into consideration, the implementation substituted by a timer process, etc. is possible.
- For the NPT value returned by getNPT, the return value corresponding to NPT stepping changes.
- Receiver record the most recently acquired NPT\_Reference(NPT<sub>r</sub>), STC\_Reference(STC<sub>r</sub>) and scale of NPT reference message as a set. At the time that the NPT value becomes necessary, based on this information and the STC value (STC<sub>c</sub>) acquired then, the NPT valid period and NPT value (NPT<sub>c</sub>) are calculated using the following algorithm.

NPT value calculation algorithm

```

STCc>STCr and STCc-STCr ≤ Wpost or
STCc<STCr and STCc+STCmax-STCr ≤ Wpost and
  Scale=1/1
    NPTc=mod((STCc+mod((STCmax+NPTTr-STCr),STCmax)),STCmax)
                                                    // Equivalent to sections A-E of Figure 2-10
  Scale=0/1
    NPTc=NPTTr
                                                    // Equivalent to section C of Figure 2-10

STCc>STCr and STCr+STCmax-STCc ≤ Wpre or
STCc<STCr and STCr-STCc ≤ Wpre and
  Scale=1/1
    NPTc=NPTTr
                                                    // Equivalent to section D of Figure 2-10
  Scale=0/1
    NPTc=mod((STCc+mod((STCmax+NPTTr-STCr),STCmax)),STCmax)
                                                    // Equivalent to section B of Figure 2-10

Only if STCmax=0x1FFFFFFF

```

- After presenting the document, the receiver performance in case the NPT related process is generated before the first NPT reference message is acquired(until the "NPTReferred" event is generated)is specified as follows.
  - At the reception time the general event message which performs the firing specification by NPT time
    - The corresponding general event message is ignored and monitoring performance of the general event message is continued.
  - Event set up(bevent) by NPT time specification.
    - Interruption event is not executed.
  - GetNPT
    - Returns an error value
- The valid period of NPT reference message subscription is set as the period presented in the document specifying subscription.

### 2.3.4.6 Operation of the DSMCC\_section( )

**Table 2-10 Operation of DSMCC\_section( ) transmitting event message**

Field	Operation	Comments
table_id	As specified.	0x3D
section_syntax_indicator	As specified.	1
private_indicator	As specified.	0
dsmcc_section_length	As specified.	
data_event_id	General event message: In case the event_msg_group_id=0x000, the value is the same as the data_event_id of corresponding local contents. In case the event_msg_group_id=0x001, it is 0x0. NPT reference message: it is 0xF.	
event_msg_group_id	General event message: 0x000 or 0x001. NPT reference message: It is 0xFFF.	
version_number	In the same subtable which is identified by the combination of table_id, data_event_id, event_msg_group_id, the continuity of +1 is guaranteed.	
current_next_indicator	Always 1.	

### 2.3.4.7 Operation of General Event Descriptor

**Table 2-11 Operation of the General Event Descriptor**

Field	Operation	Comment
event_msg_group_id	The same value as the event_msg_group_id of the DSMCC_section.(0x000 or 0x001)	
time_mode	Operate the following modes 0x00: Event is fired immediately after reception 0x02: Event is fired in accordance with NPT time data	
event_msg_type	Always 1.	
event_msg_id(message_id, message_version)	Operated as specified.	
private_data_byte	Operated as specified.	Maximum 244bytes

### 2.3.4.8 Operation of the NPT Reference Descriptor

**Table 2-12 Operation of the NPT Reference Descriptor**

Field	Operation	Comment
postDiscontinuityIndicator	Only 0 is operated.	
dsm_contented	Not operated.(It is always 0.)	
scaleNumerator/scaleDenominator	"0/1" and "1/1" are operated.	

### 2.3.5 Operation of IIT

Still image transmission (still image carousel) by video PES whose objective is interactive playback is not operated. Therefore IIT is not operated.

### 2.3.6 Performance of related receiver

#### 2.3.6.1 Filtering resources used when receiving data broadcasting

##### Necessary filtering resource for data carousels

Prior condition of transmission operation

- DII of the entry carousel in addition to the DII of the carousel that is currently presented is monitored. (Refer to section 2.3.1.4)
- The entry ES, and ES with a component\_tag value of 0x50 or 0x60 can be locked by “lockModuleOnMemoryEx()”.(Refer to section 5.12.6.9.)

**Table 2-13 Necessary filtering resources for data carousels**

Target ES	Number of PID filters	Number of section filters	
		DII	DDB
ES that is currently presented	1	1	1
Entry ES	1	1	1
ES with component_tag=0x50,0x60	2	2	2
Cash target ES	1	1	1

##### Necessary filtering resources for event messages

Assumed transmission operation

- General event messages are transmitted by a maximum of 2 ES other than the ES that is currently presented. Also, since an “event\_msg\_group\_id” with a value of 1 and 0 can be sent by each ES, 2 section filters are necessary respectively.
- A NPT reference message referred from a single BML document is transmitted with only one ES.

**Table 2-14 Necessary filtering resources for event messages**

Target ES	Number of PID filters	Number of section filters
General event message	2	4
NPT reference message	1	1

### 2.4 Temporary services and data broadcasting services

- Regarding the operation of temporary data broadcasting services, refer to Vol. 7.

## **2.5 Multi-view operations and data broadcasting services**

- Broadcasting of TV programs with additional data is possible even during multi-view operation. However, it is data broadcasting services that are common for all component groups.
- The combinations of video, audio, data are described using the Component Group Descriptor.
- All component groups should include the same data broadcasting components. (Component transmitting data carousel or event message) In other words, all data broadcasting components should be included in all component groups during multi-view operation.
- Presentation of data broadcasting will not change even if the component group is switched. (Presentation of documents that were presented before switching will be continued after switching)
- ES limit number specified in 2.1.2.5 is applied to each component group.
- Return flag may be operated during multi-view. Refer to 2.3.1.7.
- When referring to the AV stream from the BML document in multi-view programs, only "-1" can be specified as the component\_tag. (Refer to 2.1.2.8.)

## **2.6 Interaction channel transmission protocol used for data broadcasting services**

- When performing interaction channel communication in data broadcasting services, the TCP/IP communication protocol is used. For TCP/IP communication protocol details, conform with Vol. 6.
- HTTP1.1 (RFC2616) is operated as an application layer. For operation of HTTP1.1, refer to 5.14.8.
- Basic system procedure is operated in receiver equipped with modems (optional).

### 3 Operation of mono-media encoding

#### 3.1 Video encoding

##### 3.1.1 MPEG-1 Video

##### 3.1.1.1 Restrictions of encoding parameters

**Table 3-1 MPEG-1 Video Encoding parameter restriction**

"Sequence Header" restrictions				Other parameters
vertical_size	horizontal_size	pel_aspect_ratio	picture_rate	
240	352	6, 12	4	Condition of "constrained parameters"
120	176			

Meaning of each code number for the MPEG-1 Encoding parameters in Table 3-1	
pel_aspect_ratio	6 = 16:9 display (525line)      12 = 4:3 display (525line)
picture_rate	4 = 30/1.001 Hz

##### 3.1.1.2 Synchronized playback with audio (MPEG-2 AAC)

Synchronized playback of MPEG-1 transmitted by video PES and MPEG-2 AAC transmitted by PES is possible.

##### 3.1.1.3 Other restrictions

MPEG-1 Video is transmitted by video PES (stream\_type 0x01).

#### 3.1.2 MPEG-2 Video

- Vol. 7 section 4.1 shall be followed.
- It is possible to pseudo-display still images by transmitting intraframes only, however, restrictions in such case are specified in the video encoding category and is written as MPEG-2I.

3.1.2.1 Restrictions of encoding parameters

Table 3-2 MPEG-2 Video encoding parameter restrictions

Restrictions of the "Sequence Header" which are restrictions for the encoding parameter				Sequence extension restrictions	"Sequence display extension" restrictions			Other parameters
vertical_size_value	Horizontal_size_Value	Aspect_ratio_information	frame_rate_code	progressive_sequence	color_primaries	transfer_characteristics	matrix_coefficients	
1080	1440, 1920	2,3	4	0	1	1	1	Value specified for MP@HL
720	1280	2,3	7	1				
480	720	2,3	7	1				
480	480, 544, 720	2, 3	4	0				
240	352			1				

Meaning of each code number of MPEG-2 video encoding parameters in Table 3-2	
Aspect_ratio_information	1 = square pixel      2 = 4:3 display      3 = 16:9 display
Frame_rate_code	4 = 30/1.001 Hz,      7 = 60/1.001 Hz
Progressive_sequence	0 = interlaced scanning system      1 = progressive scanning system
Color_primaries	1 = Specified value of Rec.ITU-R BT.709(BT.1361)
Transfer_characteristics	1 = Specified value of Rec.ITU-R BT.709(BT.1361)
Matrix_coefficients	1 = Specified value of Rec.ITU-R BT.709(BT.1361)



**Table 3-3 MPEG-2 I restrictions**

“Sequence Header” restrictions				“Sequence extension” restrictions	“Sequence display extension” restrictions			Other parameters (note5)	
vertical_size_value	horizontal_size_value	aspect_ratio_information	frame_rate_code <sup>(note2)</sup>	Progressive_sequence	low_delay	color_primaries	transfer_characteristics		matrix_coefficients
1080 <sup>(note 1)</sup>	1920	2,3	4	0 (note 3)	1 <sup>(note 4)</sup>	1	1	1	Value specified for MP@HL
480	720	2,3	7	1					Value specified for MP@H14L
		2, 3	4	0 <sup>(note 3)</sup>					Value specified for MP@ML

Meaning of each code number of MPEG-2 I frame encoding parameters in Table 3-3	
aspect_ratio_information	1 = square pixel      2 = 4:3 display      3 = 16:9 display
frame_rate_code	4 = 30/1.001 Hz      7 = 60/1.001 Hz
Progressive_sequence	0 = interlaced scanning system      1 = progressive scanning system
low_delay	1 = B picture is not included.
color_primaries	1 = Rec.ITU-R BT.709(BT.1361)
Transfer_characteristics	1 = Rec.ITU-R BT.709(BT.1361)
matrix_coefficients	1 = Rec.ITU-R BT.709(BT.1361)

(note 1) 1088 is actually encoded in the MPEG-2 encoding method(ITU-T H.262). 8 lines worth of imaginary video data (dummy data) is added under the effective line in the encoder, and it will be encoded and processed as 1088 video data in reality. In the decoder, among the 1088 video data, the first 1080, in other words, the 1080 video signals with effective lines excluding dummy data are output.

(note 2) Timing of decoding and displaying is controlled by the time stamp value in the PES header and the vbv\_delay value is 0xFFFF.

(note 3) In case the “progressive\_frame=0”(there is a timing difference in 2 fields within the frame caused by the interlaced scan), then a freeze screen of the field should be displayed, and in case the progressive\_frame=1(2 fields within the frame is the same timing), a freeze screen of the frame should be displayed.

(note 4) In case the low\_delay=1, the time stamp for decoding and displaying will be the same value (DTS=PTS). Only PTS can be attached to a still image I frame.

(note 5) In case the sequence\_display\_extension is not transmitted, then each value for “color\_primaries”, “transfer\_characteristics”, “matrix\_coefficients” are assumed to be equal to “1” and processed on the receiver

side. The values specified in ISO/IEC 13818-2 for each level of the main profile are adapted to as the values of “vbv\_buffer\_size\_value”, etc. However, the “bit\_rate\_value” is the max value of each level and the maximum capacity of MP@LL is 4Mbit/s, for MP@ML is 15Mbit/s, and for MP@H14L and MP@HL it is the maximum capacity that can be transmitted in Digital Terrestrial Television Broadcasting.

### 3.1.2.2 Other restrictions

- MPEG-2 Video is transmitted by video PES(stream\_type 0x02)
- Scaling of MPEG-2I is not conducted.
- The following guidelines show the relationship in presentation timing control of MPEG -2I frame. Sending of the  $n$ -th +1 frame data shall start before the expected time to start presentation of  $n$ -th frame.

(Supplementary explanation)

“lowdelay” mode is specified when transmitting MPEG-2I.

The timing for starting display in receivers when in the “lowdelay” mode shall be in accordance with the PTS as a basic rule, however, for cases where the “sequence\_end\_code” is received after the PTS time, then the reception time of the “sequence\_end\_code” becomes the time to start the display. From the same perspective, the time expecting the above presentation can be understood as the PTS time or “sequence\_end\_code” receiving time. However, considering the general component of the receiver decoder, data from succeeding frames needs to be sent in order for this “sequence\_end\_code” to be recognized by the decoder (data is pushed out from the receiving buffer and arrives at the decoder itself). In this case, since the depth of this buffer is implementation dependent, data should be transmitted in bursting way without affecting the buffer control.

Also, the same process is needed for the presentation of the final frame, so after transmitting the final frame, data shall be transmitted at the expected timing for the start of presentation. In the meantime, considering the fact that above buffer size is different for each receiver, the same data as the final frame should be transmitted.

### 3.1.3 MPEG-4 Visual

Encoding methods using MPEG-4 Visual are not operated.

## 3.2 Still images and bitmap figure encoding

### 3.2.1 JPEG

- Shall be in compliance with the baseline method of ISO/IEC 10918-1(ITU-T T.81).
- Regarding the colorimetry of JPEG, B-24 Vol. 1, Part 1 7.2 “Colorimetry” is applied.

#### 3.2.1.1 Encoding parameters

- Encoding sequence: Interleave method is used.

- Baseline method with loss.
- Sampling factor: It is set as  $YCbCr=4:2:0$ . However, it shall not collapse when receiving the 4:2:2 format. Also, the JPEG for the 4:2:0 format refers to the ones with the values of (2, 2), (1, 1), (1, 1) for the SOF0 marker information (H1,V1), (H2,V2), (H3,V3) respectively, and similarly the 4:2:2 format only refers to the ones with (2, 1), (1, 1), (1, 1) as the values of (H1,V1), (H2,V2), (H3,V3) respectively.

### 3.2.1.2 Scaling

Scaling is 128/128 only.

However, only when 960x540 pixel size images are transmitted and presented as 1920x1080 size images by expanding twice, horizontally and vertically on the receiver side, is 256/128 scaling operated.

### 3.2.1.3 Other restrictions

Presented screen size is less than the full screen size of a still picture plane.

Progressive mode is not operated.

### 3.2.1.4 Operated markers and marker segments

Operated markers and marker segments by JPEG are shown in Table 3-4.

**Table 3-4 Marker/marker segments operated by JPEG**

Marker	Comments	Receiver process
SOI	Starting of image	Normal process
DQT	Definition of the quantizing table	Normal process
DRI	Definition of the restart interval	Normal process <sup>(note 1)</sup>
SOFn	Starting of the frame Only SOF0(FFC0) becomes the target for decoding.	Normal process
DHT	Definition of the Huffman table.	Normal process
SOS	Start the scan.	Normal process
RSTm	End of the restart interval.	Normal process <sup>(note 1)</sup>
EOI	End of the image.	Normal process
COM	Comment	Normal process
APPn	To be used in the application	Ignore
DNL	Special size specification	Ignore

(note 1) Handling of DRI and RSTm when an error has occurred is implementation dependent.

Only up to the above SOI-DNL will appear in the baseline method, markers other than the above are handled as errors.

### 3.2.2 PNG

The PNG shall be in compliance with the specifications of ARIB STD-B24 Vol. 1 Part 2 5.3.

#### 3.2.2.1 Encoding parameters

colortype=3

bitdepth=1, 2, 4, 8

When operating bitdepth 1, 2, 4, only the area of index 0-1, 0-3, 0-15 (fixed colors) can be specified for the color specifications of CLUT.

Image compression type =0 (zlib)

#### 3.2.2.2 Chunks operated by PNG

Chunks operated by PNG are shown in Table 3-5. When chunks other than Table 3-5 are operated, the receivers will ignore them.

**Table 3-5 Chunks operated by PNG**

Chunk names	Operation details	
IHDR	Bit depth	1, 2, 4, 8
	Color type	3 only (palette specification)
	Compression method	0 only (deflate/inflate less than 32KB)
	Filtering method	0 only
	Interlace method	0 only (no interlace)
IDAT	Filter type	0 only (none)
IEND		

- PLTE is not operated.
- PLTE of the PNG chunk is not used for the specification of the color palette. Refer to the CLUT.

#### 3.2.2.3 Other restrictions

The data storage format is non-interlace.

The presented figure size is less than the full screen size of the presented character shapes plane.

### 3.2.3 MNG

The MNG shall be in compliance with the specifications of ARIB STD-B24 Vol. 1 Part 2 5.4.

#### 3.2.3.1 Chunks operated by the MNG

Chunks operated by the MNG are shown in Table 3-6. When other chunks are used, the receivers will ignore them.

**Table 3-6 Chunk operated by MNG**

Chunk names	Operation details.
MHDR	Required.
MEND	Required.
IHDR,PNG chunks,IEND	In compliance with operational specifications of PNG.
TERM	In compliance with ARIB STD-B24.
FRAM	In compliance with ARIB STD-B24.
DEFI	In compliance with ARIB STD-B24.

### 3.2.3.2 Restrictions of MNG operation

(1) Total data size

256KB

The total data size of the MNG file means the total data volume when different MNG's are deployed. Also, the data volume of each respective MNG is the horizontal picture element count x vertical picture element count x bit depth x PNG number.

(2) Total number of PNG images

64 images

The total number of PNG images refers to the total number of PNG images that are possessed by the MNG.

(3) PNG update cycle specification value

Minimum : 100 milliseconds

Maximum :5000 milliseconds

Specification unit:100 milliseconds

(4) Repeat

- In case of specifying an infinite repeat, then 0x7FFFFFFF shall be specified as the repeat count.
- In case of specifying a limited number, then the PNG cycling period x number of PNG image x repeat count shall not exceed 120 seconds.

(5) Display size

The maximum total PNG data size in 1 screen: 256KB

Total display area in one second shall not exceed 256 KB.

(6) Others

- The display location of the MNG cannot be changed during playback. ( When streamStatusAttribute=play)
- PNG object size during MNG does not change.
- Location on the CLUT of the complete range of transparent colors is fixed.

- Even when there is a delay in the PNG update that should occur at the same time with other imaging executions, pixel skipping of PNG image is not executed and the display of PNG in order is executed. In case the “framing mode 0” is specified in the first frame, it shall be handled as “framing mode 1” regardless of the number of cyclic times.

### 3.3 Audio encoding

#### 3.3.1 MPEG-2 AAC

In accordance with Vol.7, section 4.2.

##### 3.3.1.1 Encoding parameters

**Table 3-7 Encoding parameters of MPEG-2 AAC**

Sampling frequency	Bit length
48kHz, 32kHz	16 bit

##### 3.3.1.2 Transmission of MPEG-2 AAC

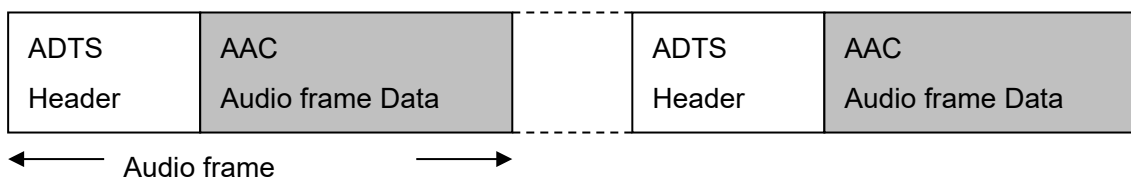
- Audio encoded in MPEG-2 AAC is transmitted by the audio PES(stream\_type 0x0F)and the data carousel(stream\_type 0x0D).
- Refer to 3.3.1.4 for the file format when transmitting by data carousel.

##### 3.3.1.3 Restrictions in data carousel distribution

- File size is 512KB or less.
- When the stop control is specified, the audio currently being played can be stopped.
- Simultaneous playback with MPEG-2Video is not possible.

##### 3.3.1.4 Data format of the AAC audio file

- It is in MPEG-2 AAC Elementary Stream Format.
- As shown in Figure 3-1, audio frames composed of audio data corresponding to the ADTS header are considered as “one unit”, and it is composed from this single unit or multiple units of those. (1 audio frame is 1024 sampling unit in PCM, so in 48kHz sampling, it is approximately 21.3 milliseconds)



**Figure 3-1 Data format of the AAC audio file**

### 3.3.2 AIFF-C

#### 3.3.2.1 Encoding parameters

**Table 3-8 Encoding parameters of AIFF-C**

Sampling frequency	Bit length
12kHz	16 bit

#### 3.3.2.2 Maximum data volume

The maximum capacity is 96KB or less.

#### 3.3.2.3 Other restrictions

- Audio encoded in AIFF-C is transmitted by data carousels(stream\_type 0x0D).
- Basic receivers do not have to handle "Private\_chunk". (Chunks other than "Format\_Version\_Chunk", "Extended\_Common\_Chunk" or "Sound\_Data\_Chunk".)
- Number of channels is 1 channel.
- It does not have to support the repeat playback function. (Seamless play is not possible.)

### 3.3.3 MPEG-4 Audio

Audio encoding method using MPEG-4 is not operated.

### 3.3.4 Additional sounds

Additional sounds are not operated.

### 3.3.5 Built-in sound

The encoding method for sound built-in receivers is AIFF-C. The specifications of 3.3.2.1 shall be followed. However, depending on the implementation of the receiver, other encoding methods for equivalent functions can be used. Built-in sound sources are allocated in Table 3-9.

**Table 3-9 Allocation of sound built-into receivers**

0:Quick report chime 1	1: Quick report chime 2	2: Quick report chime 3	3: Quick report chime 4
4: Quick report chime 5	5: Button operation sound 1	6: Button operation sound 2	7: Button operation sound 3
8: Button operation sound 4	9: Button operation sound 5	10:Button operation sound 6	11:Button operation sound 7
12:Button operation sound 8	13:Alert sound	14:	15:

Numbers in the table indicate the sound\_id in case they are specified from multi-media code, and they indicate the built-in sound specification in case of using expansion control codes PRA of 8-bit character codes.

The total capacity of receivers that use built-in sound sources is 480KB.

### **3.3.6 Audio synthesis by receivers**

#### **3.3.6.1 Mixing balance**

When mixing audio distributed in different codes, the volume should be mixed in the ratio of 1:1.

#### **3.3.6.2 Simultaneous playable encoding method**

Simultaneous playback of multiple audio is only possible for the combination of O symbols in Table 3-10. ( ) indicates audio with a higher priority to be played when simultaneous playback is not possible. "AIFF-C file (quick report)" and "quick report super sound" refer to the built-in sound played by superimpose that specify automatic display.

If the quick report super sounds and other sounds are duplicated, then the quick report super sound has priority. (The quick report super sound shall be played continuously.)

If duplicated playback of AIFF-C is specified, then whichever is specified later has priority as a basic rule.

If the playback of MPEG-2 AAC files and AIFF-C are specified at the same time, then MPEG-2 AAC has priority for play.



**Table 3-10 Audio encoding methods that support simultaneous play**

	AAC-LC Stream (Main line)	AAC-LC file (storage)	AIFF-C file (storage)	AIFF-C file (Built-in sound)	AIFF-C file (Quick report)
AAC-LC Stream (Main line)	X	X (AAC stream is prioritized)	○ <sup>(1)</sup> (2)(3)	○ <sup>(1)</sup> (2)(3)	○ <sup>(1)</sup> (2)(3)
AAC-LC File (storage)		X (Whichever is written later is prioritized)	X (AAC is prioritized)	X (AAC is prioritized)	X (Quick report is prioritized)
AIFF-C file (storage)			X (whichever is written later is prioritized)	X (whichever is written later is prioritized)	X (Quick report is prioritized)
AIFF-C file (built-in sound)				X (Whichever is written later is prioritized)	X (Quick report is prioritized)
AIFF-C file (Quick report)					X (Whichever is written later is prioritized)

(1) While performing composite output of MPEG-2 AAC audio PES and AIFF-C, if the MPEG-2 AAC audio PES disappears, then playback of AIFF-C is not guaranteed.

(2) When AIFF-C is being played independently (without composite), it cannot be combined with the MPEG-2 AAC audio PES in the middle of playback.

(3) When the sampling frequency of MPEG-2AAC-LC (AAC-LC stream main line) that is transmitted by audio PES is 32kHz, simultaneous playback is not possible. Playback of MPEG-2AAC\_LC transmitted by the audio PES is prioritized.

### 3.3.6.3 Audio when the receiver uses a broadcasting integrated HTML5 application

The simultaneous audio playback and ordering of the priority when a broadcasting integrated HTML5 application is used shall be in accordance with the IPTV Forum operational specifications.

## 3.4 Character encoding

### 3.4.1 8-bit character codes (including EUC-JP)

JIS compatible *kanji* first-plane group, second-plane group, additional symbol group are not used.

### 3.4.1.1 Restrictions on the character coding function

Whether or not the character coding function used as mono-media which is referred from multi-media code are usable/non-usable is shown in Table 3-11. To find out if the character code function in caption/superimpose is usable/non-usable, refer to 4.5 "Control codes used in caption/ superimpose" in this document.

**Table 3-11 Character code function usable/non-usable**

(1) C0 control groups

C0 control codes	Control function	Usable in multi-media codes	Restricted matters and supplements
NUL	Empty	O	
BEL	Bell	X	
APB	Active position backwards	O	
APF	Active position forwards	O	
APD	Active position down	O	
APU	Active position up	O	
APR	Active position return	O	
PAPF	Parameterized active position forward	O	
APS	Active position set	O	
CS	Clear screen	X	
CAN	Cancel	X	
ESC	Escape	O	
LS1	Locking shift 1	O	
LS0	Locking shift 0	O	
SS2	Single shift 2	O	
SS3	Single shift 3	O	
RS	Data separator	X	
US	Unit separator	X	

O:Usable Δ:Usable with some restrictions X:Non-usable

(2) C1 control groups

C1 control codes	Control function	Usable/non-usable in multi-media codes	Restricted matters and supplements
BKF	Foreground color is black and specify color map lower address.	O	0 is specified in color map lower address of foreground color.
RDF	Foreground color is red and specify color map lower address.	O	1 is specified in color map lower address of foreground color.
GRF	Foreground color is green and specify color map lower address.	O	2 is specified in color map lower address of foreground color.
YLF	Foreground color is yellow and specify color map lower address.	O	3 is specified in color map lower address of foreground color.
BLF	Foreground color is blue and specify color map lower address.	O	4 is specified in color map lower address of foreground color.
MGF	Foreground color is magenta and specify color map lower address.	O	5 is specified in color map lower address of foreground color.
CNF	Foreground color is cyan and specify color map lower address.	O	6 is specified in color map lower address of foreground color.
WHF	Foreground color is white and specify color map lower address.	O	7 is specified in color map lower address of foreground color.
COL	Color specification	Δ	17 colors of receivers common fixed color and 207 colors of broadcaster setup color 207 can be used. Specifications by COL specifies index value (0-223) of CLUT with upper 4 bit by palette number and specifies lower 4 bit by CMLA. Example: 1) palette number = 0, CMLA = 0, then index value of CLUT = 0. 2) Palette number = 10 and CMLA = 5 indicate that 0xA5 and index value of CLUT = 165.

C1 control codes	Control function	Usable/non-usable in multi-media codes	Restricted matters and supplements
POL	Pattern polarity	Δ	Only POL 04/0(normal polarity)and POL 04/1(inverse polarity1) are operated. The process when normal polarity is set as inverse polarity is, background color → foreground color, half foreground color → half background color, half background color → half foreground color, foreground color → background color.
SSZ	Small size	O	Not specified for DRCS.
MSZ	Medium size	O	Not specified for DRCS.
NSZ	Normal size	O	
SZX	Specification size	Δ	Enlargement by doubled vertical, doubled horizontal or doubled vertical/horizontal doubled can be used.
FLC	Flashing control	X	
CDC	Conceal control	X	
WMM	Change writing mode	X	
TIME	Time control	X	
MACRO	Macro specification	X	Macro specification of the control code is not used and only the default macro is used.
RPC	Character repeat	O	
STL	Start underline and mosaic separation	Δ	Only underline is operated. Refer to 4.5.5 for operation of underline.
SPL	Finish underline and mosaic separation	O	Only underline is operated.
HLC	Highlighting character block	Δ	Refer to 4.5.4 for operation of boxes.
CSI	Control sequence introducer	O	

O:Usable Δ:Usable with some restrictions X:Non-usable

(3) Expansion control codes(CSI)

Character	Control function	Usable/non-usable in multi-media codes	Restricted matters and supplements
SWF	Set writing format	Δ	Specification is possible only before appearance of characters that require display performance and control codes after initialization of the display screen. Only one code parameter is used and parameters that can be specified to P11.... Pli are 7(960x540, horizontal writing) and 8(960x540 vertical writing) when pixel size of the character shapes plane is 960x540, and 9(720x480 horizontal writing) and 10(720x480 vertical writing) when pixel size of the character figure is 720x480.
CCC	Composite character composition	X	
RCS	Raster color command	O	Specification is possible only before the appearance of characters that require display performance and control codes after initialization of the display screen. For type of colors, receiver common fixed color and broadcaster setup color (index value 0~223) can be specified. Refer to 4.5.3 for the operation of raster color control. However, areas that are raster color controlled as rectangles are specified in CSS style of the object element of BML.
ACPS	Active coordinate position set	X	
SDF	Set display format	X	
SDP	Set display position	X	
SSM	Character composition dot designation	Δ	Only 16x16,20x20,24x24,30x30,36x36 dots can be specified. (note 1)
PLD	Partial line down	X	
PLU	Partial line up	X	
SHS	Set Horizontal spacing	O	
SVS	Set vertical spacing	O	
GSM	Character deformation	X	
GAA	Coloring block	X	Colored area is full display zone only.

Character	Control function	Usable/non-usable in multi-media codes	Restricted matters and supplements
SRC	Raster color designation	X	
TCC	Switch control	X	
CFS	Character font setup	Δ	3 font styles specified in chapter 4, which are(1. round gothic(P1:03/01), 2. angle gothic(P1:03/02), 3. bold round gothic(P1:03/03)) can be specified. <sup>(note 1)</sup>
ORN	Ornament control	Δ	Character decoration cannot be used, only outlines can be used. Refer to 4.5.6 for the operation of outlines.
MDF	Font	X	
PRA	Built-in sound replay	X	
XCS	External character set	X	

O: Usable Δ:Usable with some restrictions X:Non-usable

(Note 1) Only the combination of character fonts and font sizes specified in Chapter 1 can be specified.

### 3.4.1.2 Character sets used in data broadcasting

Table3-12 Character sets used in data broadcasting

Character sets	BML Document *1	8-bit character codes to reference external BML documents(optional) *2	Closed caption/ Superimpose *3
Alphanumeric set (1byte code) specified in ARIB STD-B24	O	O	O
<i>Hiragana</i> set (1byte code) specified in ARIB STD-B24	X	O	O
<i>Katakana</i> set (1byte code) specified in ARIB STD-B24	X	O	O
<i>Kanji</i> set (2 byte code, Row 1-94) specified in ARIB STD-B24 *7	O	O	O
DRCS character set (1 byte code) specified in ARIB STD-B24	X	X	O
DRCS character group (2 byte code) specified in ARIB STD-B24	Δ*4	Δ*5	O
Macro code set (1byte code) specified in ARIB STD-B24	X	O*6	O*6
JIS compatible <i>kanji</i> first plane set specified in ARIB STD-B24	X	X	X
JIS compatible <i>kanji</i> second plane set specified in ARIB STD-B24	X	X	X
Additional symbol set specified in ARIB STD-B24	X	X	X

\*1 EUC-JP is used as specified in 4.1 "Character Codes" in Appendix 2, Vol. 2 of ARIB STD-B24.

\*2 Code table, expansion techniques, invocation of code element, designation of graphic sets and final byte of various code sets are operated as specified in ARIB STD-B24 Vol. 1, Part 2, 7.1. However, the JIS compatible *kanji* first plane set, JIS compatible *kanji* second plane set, additional symbol set, and DRCS character sets are excluded. For specifications on the display format etc, refer to Table 4-14. It is optional whether or not a receiver supports the external reference to 8-bit character codes included in documents. However, do not externally refer to 8-bit character codes included in BML documents.

\*3 Code table, expansion techniques, invocation of code element, designation of graphic sets and final byte of various code sets are operated as specified in ARIB STD-B24 Vol. 1, Part 2, 7.1. However, the JIS compatible kanji first plane set, JIS compatible kanji second plane set, and additional symbol set are excluded. For specifications on the display format etc, refer to Table 4-14.

\*4 As specified in the specifications of ARIB STD-B24 Vol. 2 Appendix 2 4.1 Character codes, Row 87 and Row 88 of the kanji set are used as the DRCS area.

\*5 Operate in accordance with the specifications specified in ARIB STD-B24 Vol. 2 Appendix 2 4.1 Character codes. BML documents and 8-bit character codes, which are externally referenced from BML documents share Row 87-88 of the kanji set as the DRCS area.

\*6 Use default macros only.

\*7 For details, ARIB STD-B3 Appendix-(3) should be followed. The recommended company for the fonts in Row 90-91 is a VICS implemented foundation.

### **3.4.1.3 Initialization of character codes (optional)**

Operations to initialize character codes shall be executed when starting presentation for each object included in BML documents that refer to 8-bit character codes. It is optional whether or not a receiver supports the external reference to 8-bit character codes included in BML documents.

The initial value is written horizontally and pixel size is the same as BML documents. The initialization of invocation and designation of the code, operation, and state shall be in accordance with the specifications of ARIB STD-B24 Vol. 1, Part 3, Chapter 8 Initializing operation. The initial values of character size controls, interval between characters, interval between lines are in accordance with Table 4-15. Also, the initial values for the character font, foreground additive colors and backgrounds additive color are round gothic, index value = 15, index value = 30 for each. The initial raster value color is transparent (index = 8).

### **3.4.2 Universal multi-octet coded Character Set**

Universal multi-octet coded Character Set is not operated.

## **3.5 Description command figure encoding**

### **3.5.1 Geometric**

Geometric is not operated.



## 4 Operation of caption and superimpose encoding

### 4.1 Scope of service and definitions

The following two services of caption and superimpose are performed in Digital Terrestrial Television Broadcasting.

**Caption** : Caption Services (i.e.: translation caption) synchronized with main video, audio, and data.

**Superimpose** : Superimpose services (i.e., news flashes, changes in air times, time tones, earthquake early warnings, etc.)

There are normal services and roll-up mode services for caption. The roll-up mode is a mode where caption data sent as page data are additionally displayed in units of lines and roll-up performance is assumed each time a line is added. The roll-up mode is classified as a separate display mode from normal caption in caption management data, and it can only be written horizontally and scrolled vertically.

Implementation of the roll-up mode is optional; however, even for receiver units without implementation, it is required for display as normal mode.

Details of operation and limitations of the roll-up mode are described in 4.10.

### 4.2 Composition and transmission operation

#### 4.2.1 Specification for composition and transmission

##### (1) Transmission methods

Caption/superimpose are transmitted by the independent PES Transmission method(stream format identification0x06).

##### (2) Composition

Each caption and superimpose is transmitted by a separate ES respectively. Also, they are transmitted by the same PMT at the same time with the main service, and caption data is not distributed within the same program or before the program starts.

##### (3) Number of ES

The maximum number of ESs that can be transmitted simultaneously for the same service is 1 for captions and 1 for superimpositions when the component group descriptor is not transmitted. When the component group descriptor is transmitted, the maximum number of ESs for captions is 1 and the maximum number of ESs for superimpositions is 1 for each component group.

##### (4) ES in case of multi-view

The maximum number of ES's for caption is 1 and the maximum number of ES's for superimpose is 1 for each component group.

Caption/superimposed character ES that appear in the component\_group\_id=0(default component group that are played while selecting stations) are the default ES of caption and superimpose respectively. For caption/superimposed character ES that appear in component groups where the component\_group\_id is not 0, and do not appear in component groups where the component\_group\_id=0, a fixed operation is not executed.

(5) ES of temporary services

The maximum number of ES's for caption is 1 and for superimpose is 1 for each temporary service.

(6) Transmission of multiple languages.

The maximum number of languages that can be transmitted at the same time is 2 languages per one ES, Language identification is done by caption management data in the ES.

(7) Bit map data

Bit map data can be used for superimpose.

(8) Usable display mode

For caption, only the "Automatic display when received/selection of display when recording and playback" and "selection of display when received/selection of display when recording and playback" can be operated. For superimpose, only "Automatic display when received/automatic display when recording and playback" and "Automatic display when received/selection of display when recording and playback" and "selection of display when received/selection of display when recording and playback" can be operated. In case of transmitting multiple languages, display mode of those languages will be the same. On the other hand, receiver performance in case of transmitting against the above is implementation dependent, however, automatic display is prioritized.

(9) Operation of alert sounds/additional sounds

Limited to built-in sound of receiver units, alert sounds can be operated for caption/superimpose. Additional sounds are not operated for caption/superimpose.

(10) Data Contents Descriptor

The Data Contents Descriptor of the EIT is not described since superimpose are not associated with events. 1 descriptor is written for caption per 1 ES. However, in case parameters do not match the Data Component Descriptor of the PMT or the set up of caption management data, then the Data Component Descriptor of the PMT or the set up of caption management are prioritized for each parameter in the display mode, number of languages and language code in receiver unit performance.

(11) Transmission frequency of caption management data

Since the caption management data includes information that is necessary to display caption and superimpose, caption texts cannot be displayed until the caption management data is

received. Therefore, considering the time for the channel selection, the caption management data is sent in the following intervals at the time of sending normal caption/superimpose.

The maximum sending frequency: 1 time / 0.3 second

The minimum sending frequency: 1 time / 5.0 seconds

However, it may be interrupted due to commercials, etc.

#### 4.2.2 PES transmission methods used in caption

A synchronous PES transmission method is applied and synchronization of timing is initiated by the PTS. Parameters set in PES packets are indicated in Table 4-1.

- Setup parameter : Refer to Table 4-1  
The maximum number of ESs sent to the same layer simultaneously for the same service: 1 when the component group descriptor is not transmitted; 1 for each component group when the component group descriptor is transmitted.
- The maximum number of languages per 1ES : 2 languages
- The constitutional unit of PES : 1 data group
- The maximum size of PES : 32KB
- The minimum sending interval of PES packets : 100 milli-seconds
- The maximum ES rate : 256Kbit/s
- Receiving buffer : More than or equal to 64KB(In both cases of 1 language and 2 languages)

Additionally, 16KB is necessary for DRCS, refer to 4.6.

**Table 4-1 Setup parameters of PES packets for caption**

Field	Operation
Stream_ID	0xBD(private_stream_1)
PES_packet_length	Continuing number of byte in PES packet *1
data_identifier	0x80
private_stream_id	0xFF
PES_data_packet_header_length	Indicates the field length of PES_data_private_data_byte. Normally, input 0x00.*2
PES_data_private_data_byte	This field may be skip read. *2
Synchronized_PES_data_byte	Data of data group of caption is stored.

\*1 Insert 0 in this value to prohibit operations without a defined PES packet length.

\*2 The current length of PES\_data\_private\_data\_byte should be specified in PES\_data\_packet\_header\_length in case of operating PES\_data\_private\_data\_byte.

The following restrictions are stipulated regarding the sending of PES packets.

- Sending sequence of PES packets and time sequence of the PTS should not be reversed.

- At PTS time of  $n$ th PES packet, the total information Vol. of the PES packet, which started sending after the  $n$ th PES packet should not exceed the receiving buffer capacity (64KB).
- The sending of PES packets should be completed before  $T_d$  from PTS time.  $T_d$  refers to the time between completion of receiving and completion of presentation. The rough standard for this is 0.5 seconds.
- For intervals of sending PES packets of the caption text data group, the interval between PTS time of the  $n$ th PES packet in the sending sequence and PTS time of the  $n-1$ st PES packet should be bigger than  $T_d$  of  $n$ th data. When the total delay  $T$  of the video satisfies the following conditions, it is possible for the sender side to send in sync with video images.

$$T > LX8 / R + T_d$$

$L$  refers to the maximum PES packet length and  $R$  refers to ES bit rate in this section.

- Performance of receiver units in case it exceeds the receiving buffer is implementation dependent.

#### 4.2.3 PES transmission methods used in superimpose

A non-synchronous PES transmission method is applied. Parameters set in the PES packet are indicated in Table 4-2.

- Setup parameters : Refer to Table 4-2  
The maximum number of ESs sent to the same layer simultaneously for the same service: 1 when the component group descriptor is not transmitted; 1 for each component group when the component group descriptor is transmitted.
- The maximum number of languages per 1 ES : 2 languages
- The constitutional unit of PES : 1 data group
- The maximum size of PES : 32KB
- The minimum sending interval of PES packets: 100 milli-seconds
- The maximum ES rate : 256Kbit/s
- Receiving buffer : More than or equal to 64KB(In both cases of 1 language and 2 languages)

Other than the above, 16KB is required for the DRCS, refer to 4.6.

**Table 4-2 Setup parameters of PES packets for superimpose**

Field	Operation
Stream_id	0xBF(private_stream_2)
PES_packet_length	Number of succeeding bytes during PES packet. *1
data_identifier	0x81
private_stream_id	0xFF
PES_data_packet_header_length	Indicates the field length of PES_data_private_data_byte. Normally 0x00 should be inserted. *2
PES_data_private_data_byte	This field can be skip-read. *2
Asynchronized_PES_data_byte	Data of data group of caption is stored.

\*1 Insert 0 in this value to prohibit operation without a defined PES packet length.

\*2 In case of operating PES\_data\_private\_data\_byte, the correct length of PES\_data\_private\_data\_byte should be specified in the PES\_data\_packet\_header\_length.

The following restrictions are stipulated regarding the sending of PES packets.

- Bit map data units can be used only when the TMD is free.
- Regarding the interval of PES packet sending for the caption/statement data group, the interval between  $n$ th PES packet in the sending sequence and  $n+1$ st PES packet should be bigger than the Td of  $n$ th data. The rough standard of time from completion of receiving to completion of presentation is 0.5 seconds for text only, in case of bit map data of 32KB, it is 3 seconds.
- For the completion time of sending the  $n$ th PES packet + Td time, the total information Vol. of the PES packet should not exceed the receiving buffer Vol. (64KB).
- Performance of receiver units in case the receiving buffer is exceeded is implementation dependent.

#### 4.2.4 Operation of data groups

Along with the update of caption management data, a “data\_group\_id” should be sent after switching the data group from group A to group B and from group B to group A. However, in case caption management data is not sent for more than 3 minutes, either one of group A or group B is sent out regardless of which group was sent earlier. The “data\_group\_version” is not used. When the caption management data is group A, the receiver unit processes only the caption texts (this statement, bitmap data, DRCS) of group A and when the caption management data is group B, the receiver unit only processes caption texts of group B.

In case the caption management data is the same as the caption management data of the group already received, it is processed as caption management data that was re-sent and the initialization operation by caption management data is not executed. When receiving the same caption texts as already received caption management data multiple times, each caption text is processed as the new caption text.

**Table 4-3 Data group parameters**

Field	Operation
data_group_id	Operated as defined.
data_group_version	It is not operated.
data_group_link_number	0x00
last_data_group_link number	0x00
data_group_size	Operated as defined. However, 1 PES packet should not exceed 32KB.
data_group_data_byte	Data group data (Caption management data, caption text data) is stored.
CRC_16	Error detection is operated by CRC16. When an error is detected, the receiver unit will discard the corresponding data group.

#### 4.2.5 Operation of caption management data

Within the same caption management data, data units for the same or different data unit parameters can be located multiple times. In case multiple data units exist within the caption management data, then it is processed in the appearing sequence of the data unit.

However, data that can be written in this document is only the control codes of SWF, SDF, SDP, SSM, SHS, SVS and the character code group that requires screen display cannot be written.

Any changes in line intervals and character intervals and format setup will be done immediately after CS.

In case of specifying SVS, SHS and SWF in the texts of management data, SCS, SHS and SWF should be specified in the text body of the caption texts. Also, for the roll-up mode, the changes in line intervals and character intervals and format setup should be done immediately after changing the group from A/B.

##### 4.2.5.1 Caption management data used in caption

Caption management data should be sent at least every 3 minutes. In case caption management data is not received for more than 3 minutes, then the initialization operation of receiver unit at the time of selecting stations is done. Parameters that can be specified in caption management data used in caption are shown in Table 4-4.

**Table 4-4 Parameters of caption management data for caption**

Fields	Operations
TMD	'00'(Free)
num_languages	1-2
language_tag	0-1
DMF	'0010'(Automatic display when received/Selection of display when recording and playback) '1010'(Selection of display when received/Selection of display when recording and playback)
ISO 639 language code	Language codes to be used
Format	'1000'(Written horizontally in 960x540) '1001'(Written vertically in 960x540) '1010'(Written horizontally in 720x480) '1011'(Written vertically in 720x480)
TCS	'00'(8-bit character codes)
rollup_mode	'00' (Non-roll up) '01' (Roll up) '10' (Reserved) '11' (Reserved)
data_unit_loop_length	Operated as defined. However, 1 PES packet should not exceed 32KB.
data_unit	Data unit(this sentence, DRCS)is stored.

#### 4.2.5.2 Caption management data used in superimpose

Considering the superimposition of time, real time setup of the TMD on top of a free setup is possible to execute synchronization of time by STM. In case caption management data is not received for more than 3 minutes, then the initialization operation of the receiver unit while selecting stations is done. Parameters that can be specified in caption management data used in superimpose are indicated in Table 4-5.

**Table 4-5 Parameters of caption management data for superimpose**

Fields	Operations
TMD	'00'(Free) '01'(Real time) Free and real time cannot be mixed when presenting.
num_languages	1-2
language_tag	0-1
DMF	'0000'(Automatic display when received/Automatic display when recording and playback) '0010'(Automatic display when received/Selection of display when recording and playback) '1010'(Selection of display when received/Selection of display when recording and playback)
ISO_639_language_code	Language codes to be used.
Format	'1000'(Written horizontally in 960x540) '1001'(Written vertically in 960x540) '1010'(Written horizontally in 720x480) '1011'(Written vertically in 720x480)
TCS	'00'(8-bit character codes)
data_unit_loop_length	Operated as defined. However, 1 PES packet should not exceed 32KB.
data_unit	Data unit(this text and DRCS)is stored.

#### 4.2.6 Operation of caption statement data

Multiple data units of the same or different data unit parameters can be located within the same caption statement data. In case multiple data units exit from the same caption statement data, it is processed in the appearance sequence of the data units.

Parameters that can be set in caption statement data are indicated in Table 4-6.

**Table 4-6 Parameters of caption statement data**

Field	Operation
TMD	'00'(Free) '01'(Real time): Superimpose only However, the same value as caption management data should be set within the same program.
STM	Operated as defined. Valid only when, in Data Component Descriptor of PMT, timing='10'(Time synchronization)
data_unit_loop_length	Operated as defined. However, 1 PES packet should not exceed 32 KB.
data_unit	Data unit(this text and DRCS、bit map data)is stored.



#### 4.2.7 Operation of data units

Parameters that can be set in data units are indicated in Table 4-7.

**Table 4-7 Parameters of data units**

Fields	Operations
unit_separator	It is 0x1F as defined.
data_unit_parameter	0x20(Texts) 0x35(Bit map data) 0x30(1 byte DRCS) 0x31(2 byte DRCS)
data_unit_size	Operated as defined. However, 1 PES packet should not exceed 32KB.
data_unit_data_byte	Data unit data is stored.

#### 4.2.8 OPERATION OF PSI/SI

##### 4.2.8.1 Operation of component tags

Component tag value of caption ES is 0x30-0x37 in case of transmitting through layers other than the partial reception layer, in case of transmitting through the partial reception layer, it is 0x87. As for component tag value of superimpose ES, it is 0x38-0x3F when transmitting through layers other than the partial reception layer, in case of transmitting through the partial reception layer, set value within the range of 0x88. However, for component tag values of default ES of caption, set 0x30 or 0x87, for component tag value of default ES of superimpose, set 0x38 or 0x88.

##### 4.2.8.2 Operation of the PMT

Updating the PMT is basically to add/delete ES information when starting/ending caption and superimpose, however, operations to write ES information at all times is possible, too.

##### 4.2.8.3 Stream format identification

Stream\_type of caption/superimpose ES is 0x06(Independent PES\_packet)

##### 4.2.8.4 Descriptor operation

Descriptor operation of the PMT and EIT for caption/superimpose is indicated in Table 4-8.

**Table 4-8 Descriptor operation of the PMT and EIT**

Descriptors	PMT	EIT
Stream Identifier Descriptor	Required	-
Data Component Descriptor	Required	-
Data Contents Descriptor	-	H-EIT[p/f]、 M-EIT[p/f]: Required  H-EIT[schedule]、 M-EIT[p/f after]: Arbitrary  (superimpose are not written)

#### 4.2.8.5 Data Component Descriptor

The data\_component\_id of the Data Component Descriptor is 0x0008 for both caption and superimpose. Also, parameters set in additional information ID are indicated in Table 4-9.

**Table 4-9 Setup parameters of additional information ID of the Data Component Descriptor**

Field	Operations
DMF	'0011'
Timing	Caption : '01'(Program synchronization) Superimpose : '00'(Non-synchronization) or '10'(Time synchronization)

#### 4.2.8.6 Target Region Descriptor

The Target Region Descriptor is not used.

#### 4.2.8.7 Data Contents Descriptor

Parameters that can be set in the Data Contents Descriptor and its selector area are indicated in Tables 4-10 and 4-11. However, in case the setup parameter value does not match the PMT Data Component Descriptor and caption management data during the same event, set values of Data Component Descriptor and caption management data are prioritized.

**Table 4-10 Setup parameters of the Data Contents Descriptor for caption**

Fields	Operations
data_component_id	0x0008
entry_component	"component_tag" value of corresponding caption ES.
num_of_component_ref	0 should be specified.
component_ref	Not necessary because "num_of_component_ref=0".
ISO_639_language_code	Fixed to jpn(Japanese).
text_length	The maximum value is 16(byte).
text_char	Contents of caption displayed in EPG are written.

**Table 4-11 Setup parameters in the selector area of the Data Contents Descriptor for caption.**

Fields	Operations
num_languages	The same value as caption management data.
DMF	The same value as the Data Component Descriptor.
ISO_639_language_code	The same value as caption management data.

### 4.3 Visual pixel size and display format of caption/superimpose

#### 4.3.1 Display format

The usable display format is 960x540 and 720x480 in horizontal and vertical writing. Also, pixel size and the display format of caption/superimpose of the video plane should be in the combinations in Table 4-12 for horizontal and vertical writing respectively. When displaying in 720x480, it should be in the same display format regardless of the aspect ratio of video, and in consideration of the aspect ratio for display, it is corrected by the sender side.

**Table 4-12 Combinations of video display format and pixel size**

Pixel size of video plane	Display format of caption and superimpose
1920X1080	960X540
720X480	720X480

For data broadcasting programs, it is the same as above.

Even in case of combining HD data broadcasting for SD video, the caption display should be continued. The display position in this case is implementation dependent.

#### 4.3.2 Display area

The display area refers to the horizontal/vertical picture element count specified by the control code SDF and the area specified by coordinates from the upper-left corner of the caption plane specified by the control code SDF. (Refer to Figure 4-1) The starting point (0,0) of the coordinates of display area is the upper left of the "caption plane" regardless of the direction (vertical or horizontal) of writing.

For caption and superimpose, the display area that can be set simultaneously is 1. Also, the display area is valid for bit map data. The priority order of the display area is,

- (1) Values specified by SDF and SDP in the texts of caption statement data
- (2) Values specified by SDF and SDP in the texts of updated caption management data
- (3) Initial value based on the display format specified by the updated caption management data header. The initial value of the display area (display composition dot) is as shown in Table 4-13. Also, the initial values of character size controls, intervals between characters and lines are specified in Table 4-15.

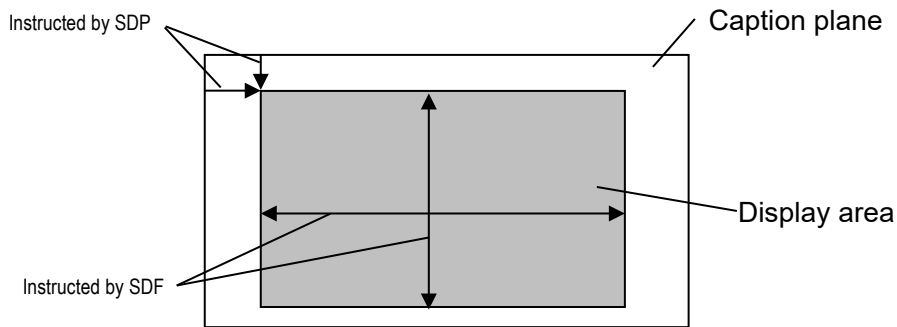


Figure 4-1 Caption plane and display area

Table 4-13 Initial values of the display area and display location

Display format	Display area	Display location
960X540	960X540	0, 0
720X480	720X480	0, 0

### 4.3.3 Initial operation location

The initial operation location is the first location of first line by the size of characters in the initial condition. The first location of the first line is the upper left corner of the display area in case of horizontal writing and the upper right corner of the display area in case of vertical writing. Also, the performance direction is towards the right for horizontal writing and downwards for vertical writing.

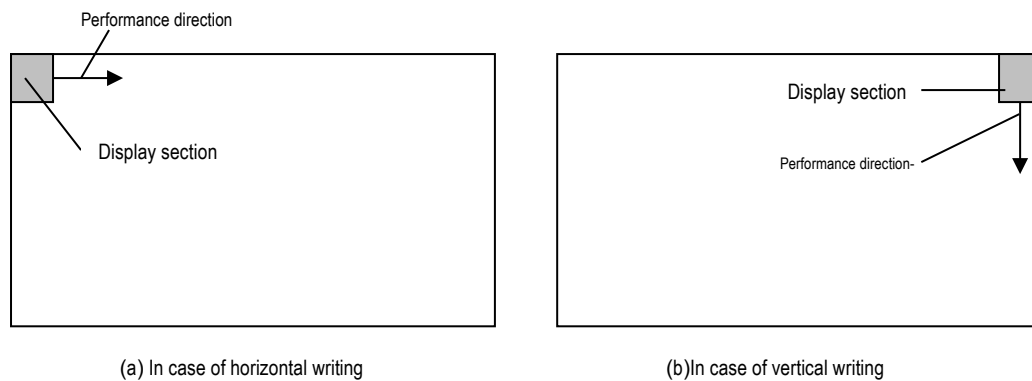


Figure 4-2 Initial operation locations and performance directions

## 4.4 Characters used in caption/superimpose

### 4.4.1 Character codes

The character encoding method used for caption/superimpose is 8-bit character codes. Refer to Chapter 3 mono-media for operating the code set.

### 4.4.2 Character fonts

Round gothic is preferred as the character font used for caption and superimpose.

#### 4.4.3 Character size controls

There are 5 sizes of characters that can be displayed as caption/superimpose, which are 16 dots, 20 dots, 24 dots, 30 dots and 36 dots. For the specification of character size controls at the time of sending, the above mentioned sizes should be specified.

Also, standard, medium and small sizes can be used for each size. The definitions of standard, medium and small are as follows.

Normal size: Specified by the control code SSM.

Medium size: Only the size in lateral direction from standard is half the size.

Small size: Each size in the lateral direction and line direction from standard is half the size.

Restrictions related to character display are stipulated in Table 4-14

**Table 4-14 Area of coding group that can be used for specification of display format and specification of character size controls**

Characters, etc	2 byte code Row (Cell)	Horizontal/vertical writing		
		Standard Doubled vertical Doubled Horizontal Doubled Vertical/ horizontal	Medium	Small
Symbol	1, 2	○	○ <sup>*2,*4</sup>	○
Alphanumeric	3 <sup>*1</sup>	○	○ <sup>*4</sup>	○
Hiragana	4 <sup>*1</sup>	○	○ <sup>*4</sup>	○
Katakana	5 <sup>*1</sup>	○	○ <sup>*4</sup>	○
Greek letters	6	○	○ <sup>*4</sup>	
Russian letters	7	○		
Rule lines	8	○ <sup>*3,*4</sup>		○ <sup>*4</sup>
Kanji	16~84	○		
Additional symbol	90 ( 1~ 6)	○		
	( 8~11)	○		
	(16~17)	○		
	(20~40)	○		
	(64~65)	○		
	91 (1~49)	○		
	92 ( 1~ 4)	○	○ <sup>*4</sup>	
	( 5~12)		○ <sup>*4</sup>	
	(13~15)	○		
	(16~25)	○	○ <sup>*4</sup>	○
	(26~31)	○		
	(32~41)	○	○ <sup>*4</sup>	○
	(42~47)	○		
(48~52)	○	○ <sup>*4</sup>	○	
(53~54)	○	○ <sup>*4</sup>	○	
(55~91)	○			
93 ( 1~45)		○		
	(48~91)	○		
94 ( 1~93)	○			
DRCS	Normal size picture element component	○		
	Medium size picture element component		○	
	Small size picture element component			○
Special codes (space/delete)		○	○	○

\*1 Symbols have been added for 1 byte code sets.

\*2 Excluding Row 2, Cell 94

\*3 Standard only

\*4 Not operated in vertical writing.

(1) If the display format is in vertical writing, the receiver unit will display the following characters in a different shape than in horizontal writing.

Row 1 : Cell 2, 3, 17, 18, 28~30, 33~37, 42~59, 65

Row 4 : Cell 1, 3, 5, 7, 9, 35, 67, 69, 71, 78

Row 5 : Cell 1, 3, 5, 7, 9, 35, 67, 69, 71, 78, 85, 86

Row 92 : Cell 48, 49, 50, 51

And, the *katakana* set and *hiragana* set correspond to the following codes.

*katakana* : 2/1, 2/3, 2/5, 2/7, 2/9, 4/3, 6/3, 6/5, 6/7, 6/14, 7/5, 7/6, 7/9~7/13

*hiragana* : 2/1, 2/3, 2/5, 2/7, 2/9, 4/3, 6/3, 6/5, 6/7, 6/14, 7/9~7/13

(2) Characters in Row 1, Cell 13-18 and Row 2 Cell 94 are non-spacing characters (refer to 4.4.5)

#### 4.4.4 Display zone

Definitions of the display zone are as follows.

Size of lateral direction of the display zone = horizontal spacing/2 + font size + horizontal spacing/2

Size of line direction of the display zone = vertical spacing/2 + font size + vertical spacing/2

\* In case the horizontal(vertical) spacing is an odd number, then the value of the interval between characters on the front side of font/2 (interval between lines/2) shall be rounded down and the value of the interval between characters on the backside of font/2 (interval between lines/2) shall be rounded up. The relationship between the display section, font size and interval between characters and lines in case of horizontal writing is indicated from Figure 4-3 to Figure 4-8, in case of vertical writing is indicated from Figure 4-9 to Figure 4-14.

In case those intervals between characters are not multiples of 4, then the values of the interval between characters/4 (interval between lines/4) are rounded down.

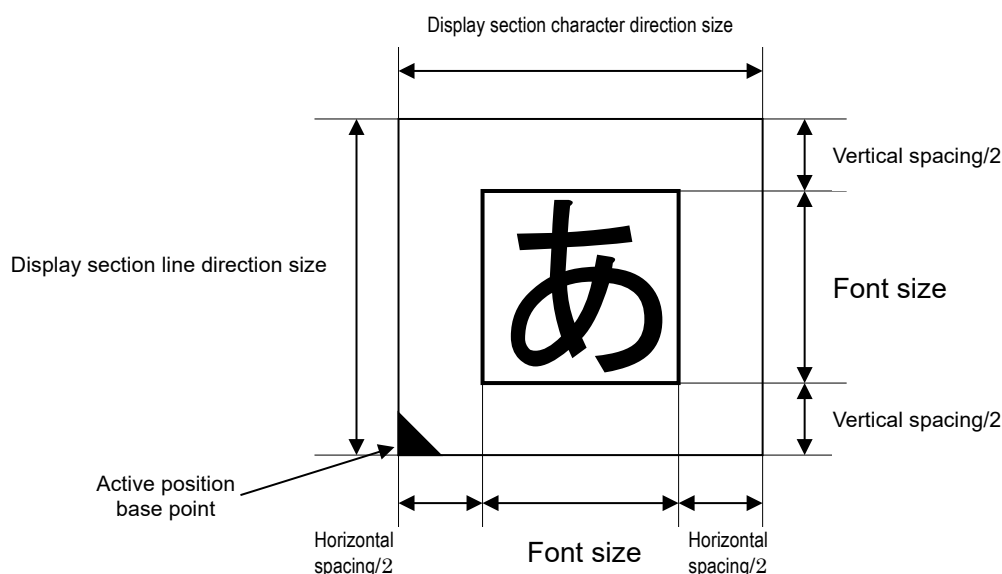
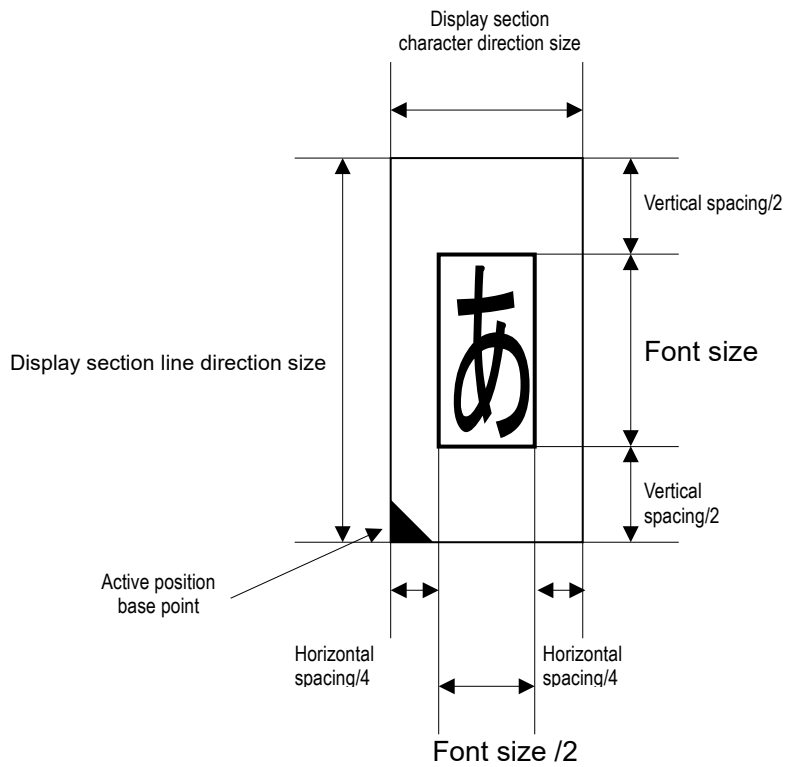
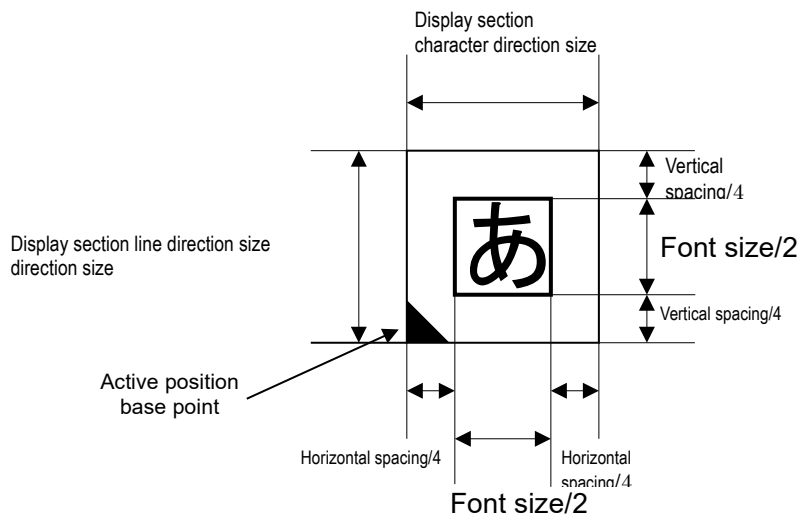


Figure 4-3 Display section and font size (standard) in horizontal writing and the relationship of the horizontal(vertical) spacing.

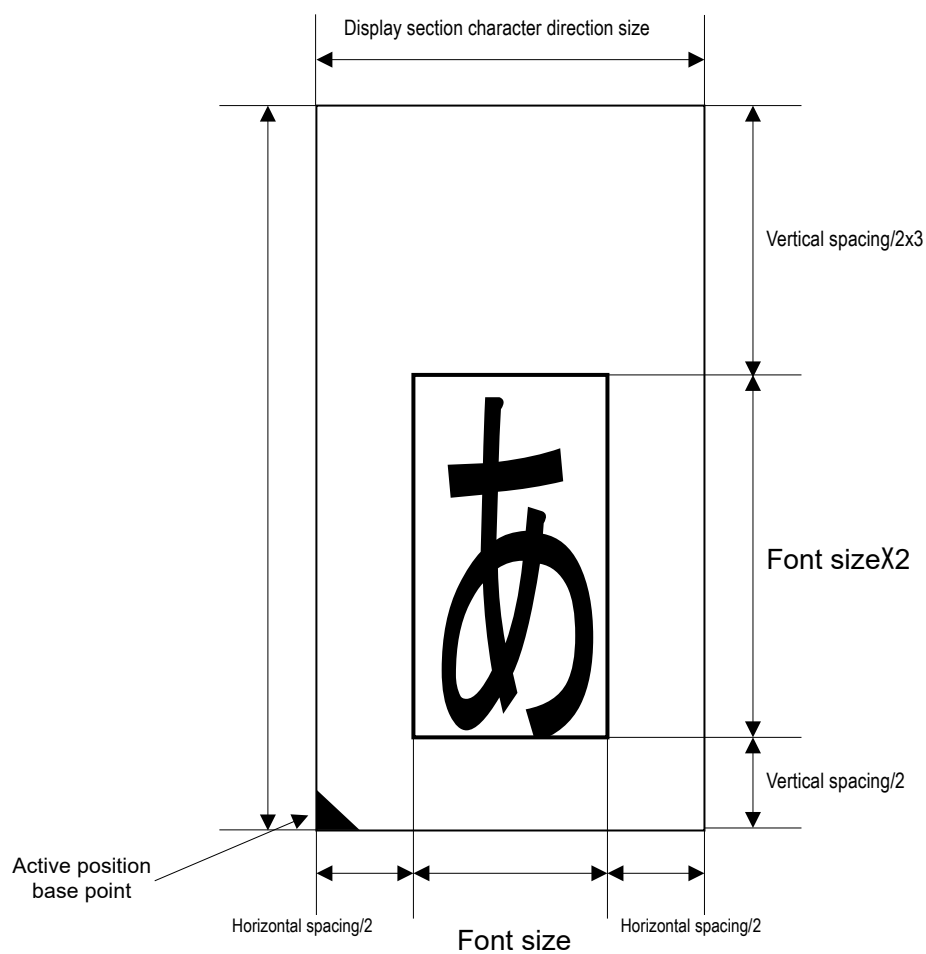


**Figure 4-4 Display section and font size (medium size) in horizontal writing and the relationship of the horizontal(vertical) spacing.**

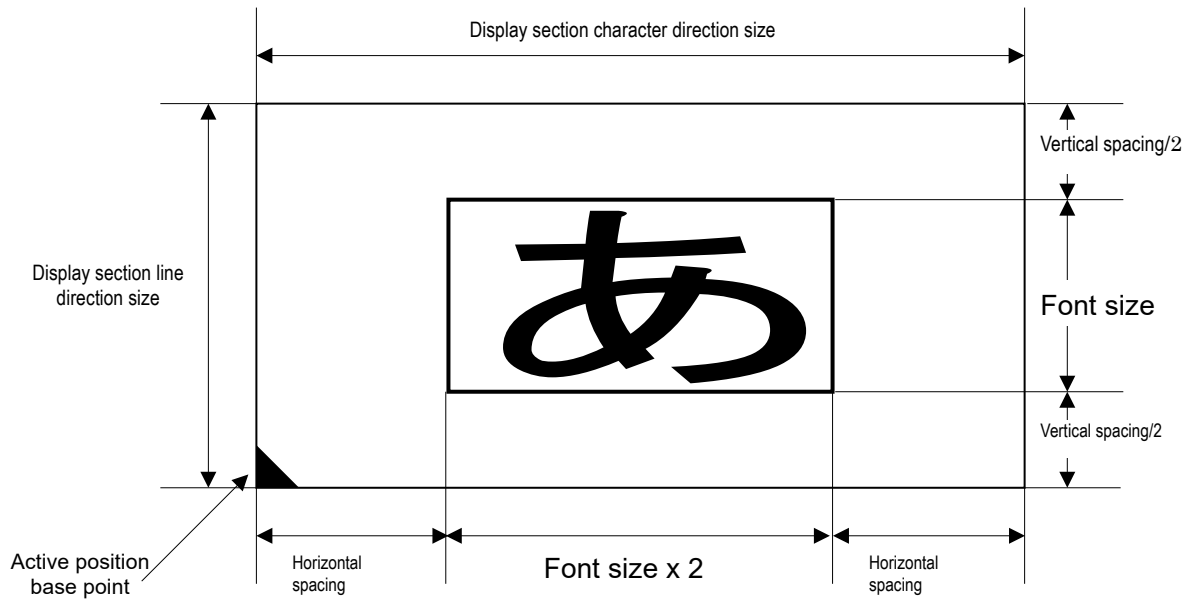


**Figure 4-5 Display section and font size (small size) in horizontal writing and the relationship of the horizontal(vertical) spacing.**

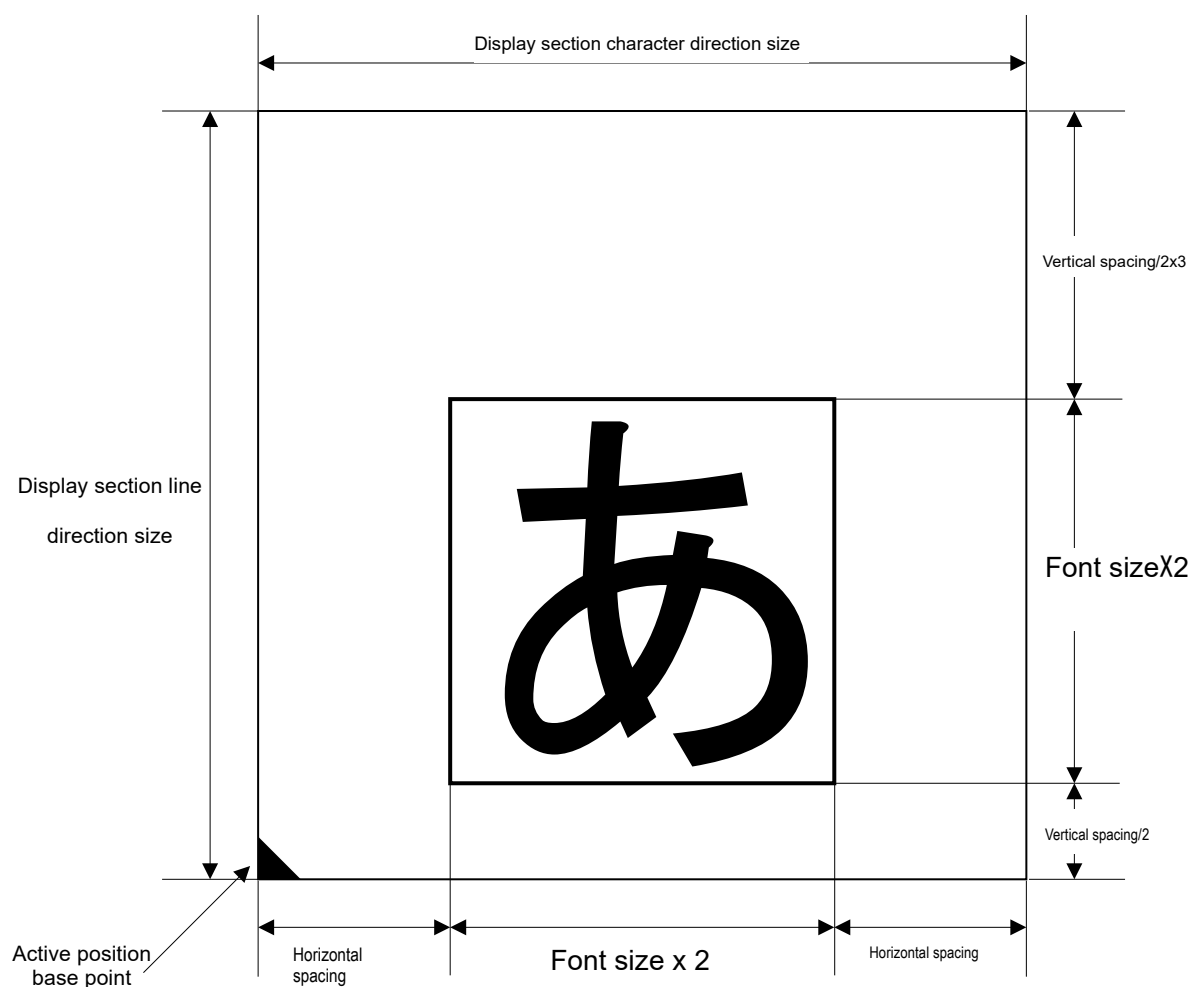




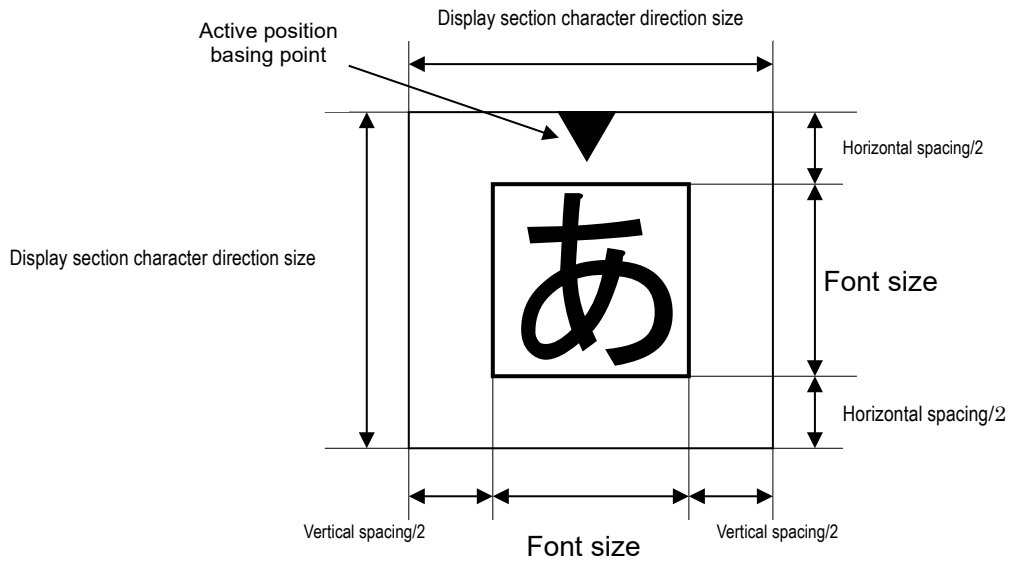
**Figure 4-6 Display section and font size (doubled vertical) in horizontal writing and the relationship of the horizontal(vertical) spacing**



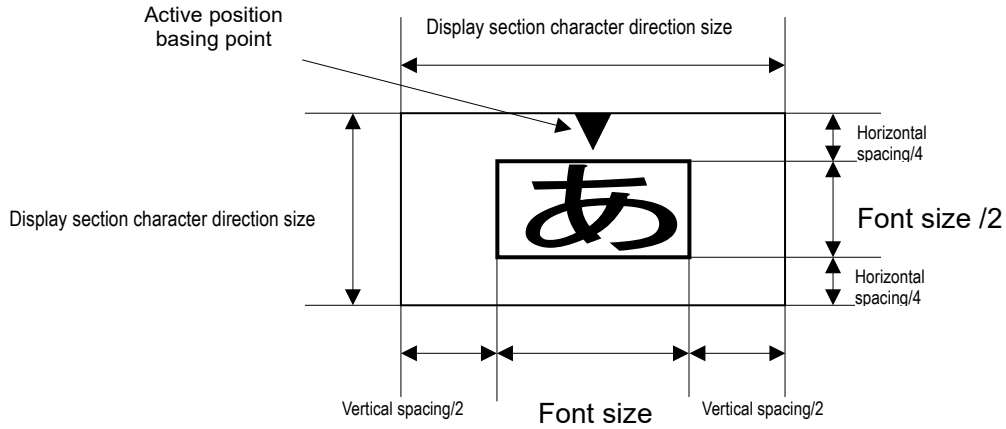
**Figure 4-7 Display section and font size (doubled horizontal) in horizontal writing and the relationship of the horizontal(vertical) spacing.**



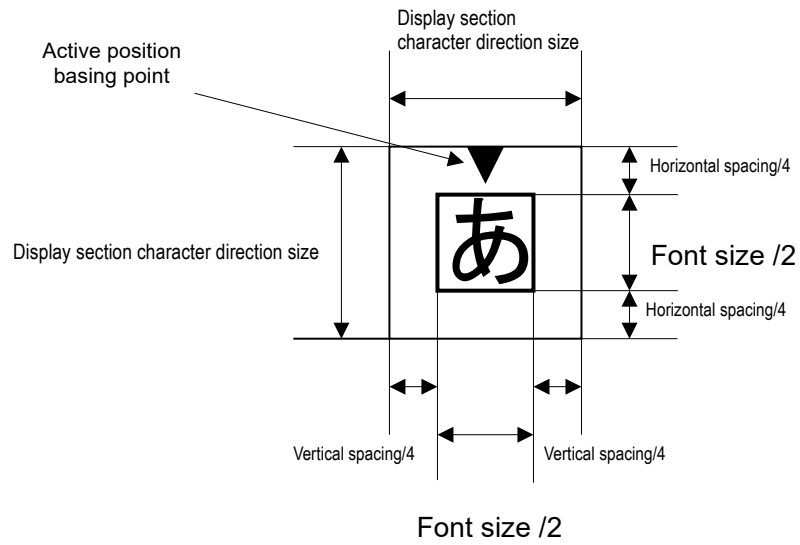
**Figure 4-8 Display section and font size (doubled horizontal and doubled vertical) in horizontal writing and the relationship of the horizontal(vertical) spacing.**



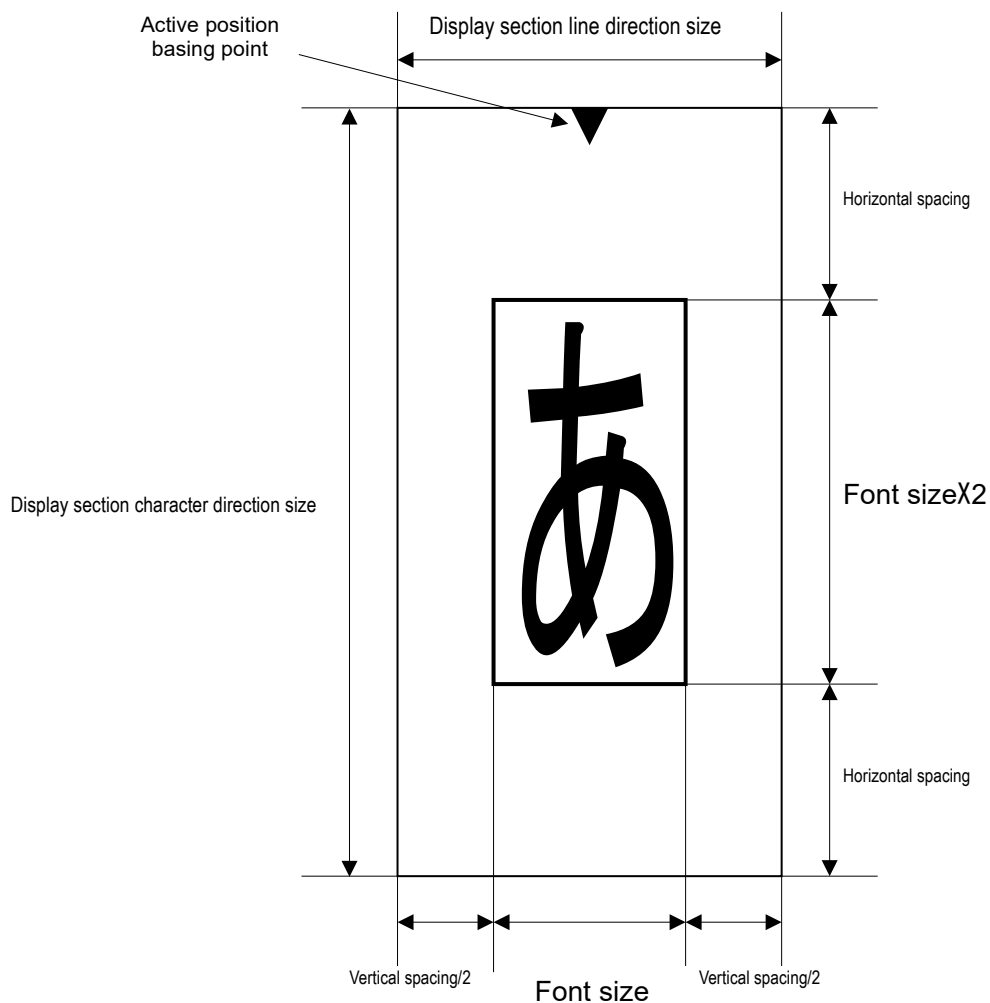
**Figure 4-9** Display section and font size (standard) in vertical writing and the relationship of the vertical(horizontal) spacing.



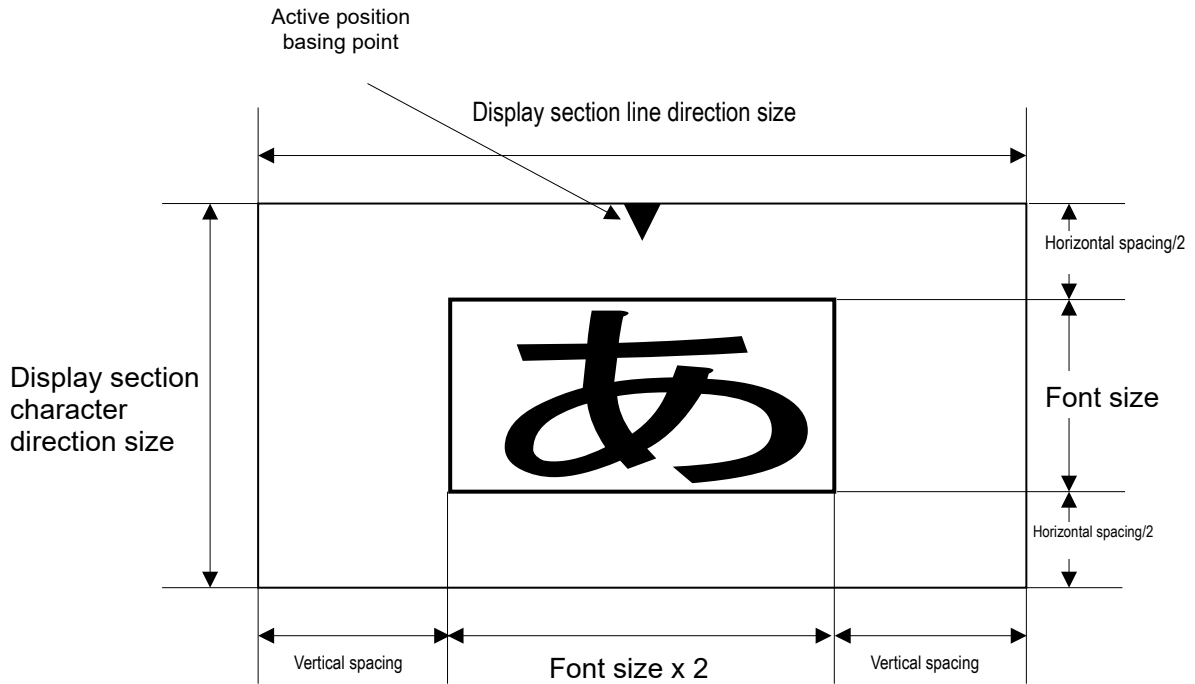
**Figure 4-10** Display section and font size (medium size) in vertical writing and the relationship of the vertical(horizontal) spacing.



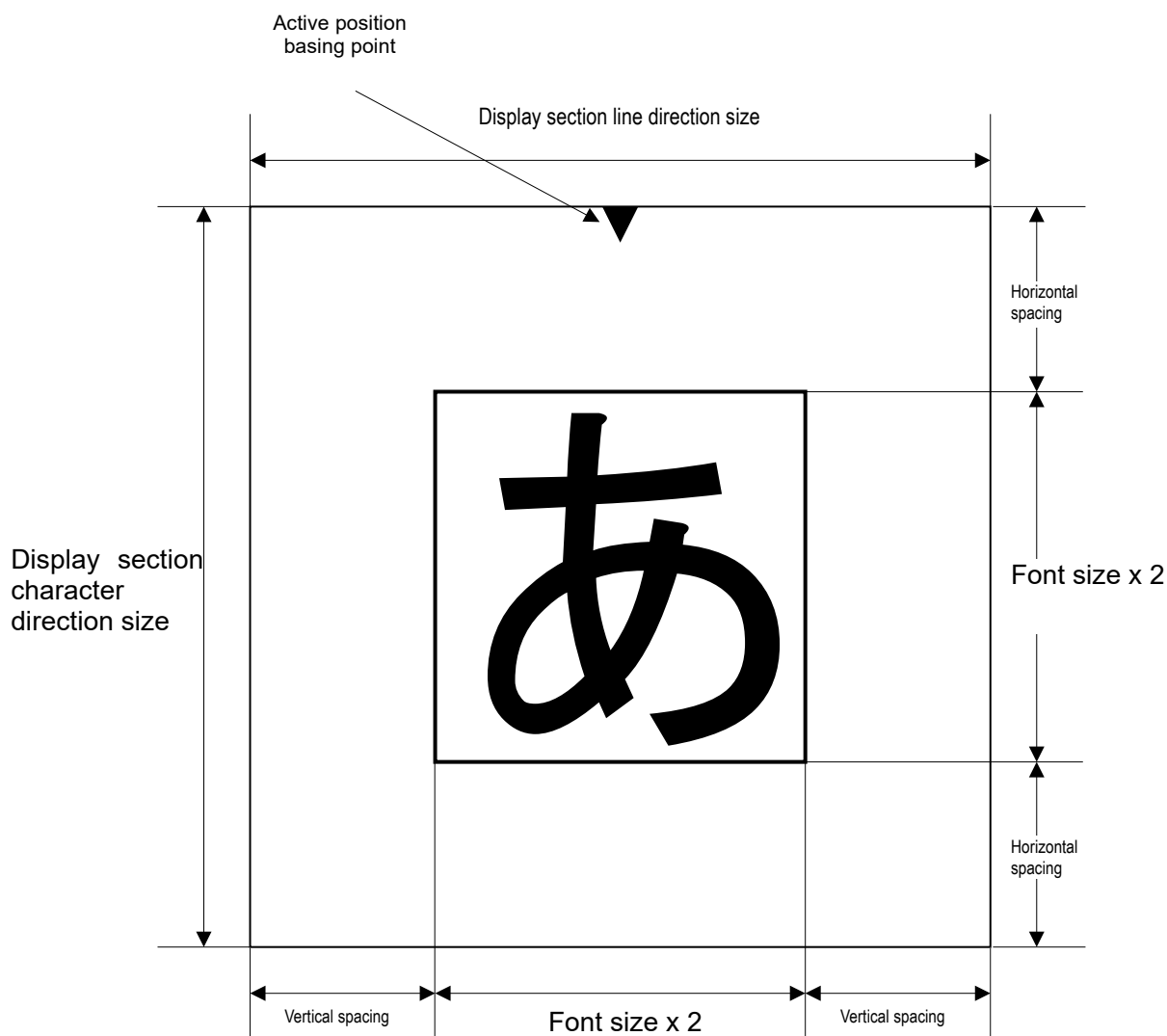
**Figure 4-11 Display section and font size (small size) in vertical writing and the relationship of the vertical(horizontal) spacing.**



**Figure 4-12 Display section and font size (doubled vertical) in vertical writing and the relationship of the vertical(horizontal) spacing.**



**Figure 4-13** Display section and font size (doubled horizontal) in vertical writing and the relationship of the vertical(horizontal) spacing.



**Figure 4-14 Display section and font size (doubled vertical and doubled horizontal) in vertical writing and the relationship of the vertical(horizontal) spacing.**

The initial values of character size controls, horizontal and vertical spacing are indicated in Table 4-15.

**Table 4-15 Initial values of character size controls, horizontal and vertical spacing**

	960X540		720X480	
	Horizontal writing	Vertical writing	Horizontal writing	Vertical writing
Character size controls	36	36	36	36
Horizontal spacing	4	12	4	8
Vertical spacing	24	24	16	24
Display zone(W x H)	40x60	60x48	40x52	60x44

However, ruled line characters(Row 8)are displayed in all areas of the display zone only when the horizontal spacing and the vertical spacing are set in accordance with the horizontal writing of Table 4-16. (The function to display ruled lines in all areas of the display zone when the values other than the ones in Table 4-16 are set for the horizontal spacing and the vertical spacing is optional.)

**Table 4-16 Display zone that can display ruled line characters\*1/box /underline properly**

	Character size controls	Horizontal spacing	Vertical spacing
960X540 horizontal writing	16	0, 1, 2	0, 5, 11
	20	0, 1, 2	0, 7, 13
	24	0, 1, 3	0, 8, 16
	30	0, 2, 3	0, 10, 20
	36	0, 2, 4	0, 12, 24
960X540 vertical writing	16	0, 3, 5	0, 5, 11
	20	0, 3, 7	0, 7, 13
	24	0, 4, 8	0, 8, 16
	30	0, 5, 10	0, 10, 20
	36	0, 6, 12	0, 12, 24
720X480 horizontal writing	16	0, 1, 2	0, 4, 7
	20	0, 1, 2	0, 4, 9
	24	0, 1, 3	0, 5, 11
	30	0, 2, 3	0, 7, 13
	36	0, 2, 4	0, 8, 16
720X480 vertical writing	16	0, 2, 4	0, 5, 11
	20	0, 2, 4	0, 7, 13
	24	0, 3, 5	0, 8, 16
	30	0, 3, 7	0, 10, 20
	36	0, 4, 8	0, 12, 24

\*1 However, ruled line characters are applied only for horizontal writing.

(Reference) The above horizontal spacing value and vertical spacing value are calculated from the following calculation formulas.

Horizontal spacing ratio=initial horizontal spacing value/initial font size

Horizontal spacing value 1=Horizontal spacing ratioXfont size

Horizontal spacing value 2=(Horizontal spacing/2)Xfont size

Vertical spacing ratio =initial vertical spacing value/initial font size

Vertical spacing value 1=Vertical spacing ratio X font size

Vertical spacing value 2=(Vertical spacing ratio /2)Xfont size

The active position base point of the display zone is in the lower left corner of the display zone for horizontal writing and upper middle of the display zone for vertical writing. However, when the display zone size for the line direction in case of vertical writing is an even number, then the active position base point for the line direction is moved by 1 dot towards the line performance direction from the middle value. (For example, when the line direction display zone size = 68 dot then the line direction active position base point is the 35th dot.)



Regarding the active position, it is as follows;

- (1)Active position is not moved by changing the size of the display zone.
- (2)After executing display, the active position is moved forward automatically. However, non-spacing characters are not limited to this.
- (3)When the display zone goes beyond the edges of the display area, display characters after changing lines. In such case, the active position is moved forward by the number of active position lines only by the display zone line direction side for the display zone of characters at the end of the line before changing the line.

#### **4.4.5 Non-spacing characters**

Non-spacing characters are displayed after being combined with characters or the space specified by succeeding codes. The characters and codes that can be used between characters and codes that are combined with non-spacing characters are as follows.

Blank space	: NULL
Expansion control	: Control codes of instruction or calling
Special function	: SP and DEL(Usable as the terminal-end)
Character codes group	: Spacing characters and external characters(Usable as the terminal-end), non-spacing characters

### **4.5 Control codes used in caption/superimpose**

#### **4.5.1 Control codes**

Control codes used in caption are in compliance with ARIB STD-B24 Vol. 1 Part 2, 7.1.2. However, operating regulations for Table 4-17, Table 4-18, Table 4-20 are to be stipulated.

Table 4-17 C0 control set

C0 control codes	Control function	Possible to use or not	Restricted matters and supplements
NUL	Blank	O	
BEL	Bell	X	
APB	Active position backward	O	
APF	Active position forward	O	
APD	Active position down	O	
APU	Active position up	O	
APR	Active position return	O	
PAPF	Parameterized active position forward	O	
APS	Active position set	O	
CS	Clear screen	O	
CAN	Cancel	X	
ESC	Escape	O	
LS1	Locking shift 1	O	
LS0	Locking shift 0	O	
SS2	Single shift 2	O	
SS3	Single shift 3	O	
RS	Record separator	X	
US	Unit separator	O	Used only to identify data units and is not used in 8-bit character codes character strings.

O: possible to use Δ: possible to use with restrictions X: impossible to use

Table 4-18 C1 Control set

C1 control codes	Control function	Possible to use or not	Restricted matters and supplements
BKF (CFLA0: TBD)	Foreground color Color map Lower address specifications	○	CMLA of foreground color is specified as 0.
RDF (CFLA1: TBD)	Foreground color Color map Lower address specifications	○	CMLA of foreground color is specified as 1.
GRF (CFLA2: TBD)	Foreground color Color map Lower address specifications	○	CMLA of foreground color is specified as 2.
YLF (CFLA3: TBD)	Foreground color Color map Lower address specifications	○	CMLA of foreground color is specified as 3.
BLF (CFLA4: TBD)	Foreground color Color map Lower address specifications	○	CMLA of foreground color is specified as 4.
MGF (CFLA5: TBD)	Foreground color Color map Lower address specifications	○	CMLA of foreground color is specified as 5.
CNF (CFLA6: TBD)	Foreground color Color map Lower address specifications	○	CMLA of foreground color is specified as 6.
WHF (CFLA7: TBD)	Foreground color Color map Lower address specifications	○	CMLA of foreground color is specified as 7.
COL	Color Controls	Δ	Color control COL P1(1byte) COL 04/8: CMLA of foreground color is specified as 8. COL 04/9: CMLA of foreground color is specified as 9. COL 04/10: CMLA of foreground color is specified as 10. COL 04/11: CMLA of foreground color is specified as 11. COL 04/12: CMLA of foreground color is specified as 12. COL 04/13: CMLA of foreground color is specified as 13. COL 04/14: CMLA of foreground color is specified as 14. COL 04/15: CMLA of foreground color is specified as 15.

C1 control codes	Control function	Possible to use or not	Restricted matters and supplements
			COL 05/0: CMLA of background color is specified as 0. COL 05/1: CMLA of background color is specified as 1. COL 05/2: CMLA of background color is specified as 2. COL 05/3: CMLA of background color is specified as 3. COL 05/4: CMLA of background color is specified as 4. COL 05/5: CMLA of background color is specified as 5. COL 05/6: CMLA of background color is specified as 6. COL 05/7: CMLA of background color is specified as 7. COL 05/8: CMLA of background color is specified as 8. COL 05/9: CMLA of background color is specified as 9. COL 05/10: CMLA of background color is specified as 10. COL 05/11: CMLA of background color is specified as 11. COL 05/12: CMLA of background color is specified as 12. COL 05/13: CMLA of background color is specified as 13. COL 05/14: CMLA of background color is specified as 14. COL 05/15: CMLA of background color is specified as 15. COL 06/0: CMLA of half foreground color is specified as 0. COL 06/1: CMLA of half foreground color is specified as 1. COL 06/2: CMLA of half foreground color is specified as 2. COL 06/3: CMLA of half foreground color is specified as 3. COL 06/4: CMLA of half foreground color is specified as 4. COL 06/5: CMLA of half foreground color is specified as 5. COL 06/6: CMLA of half foreground color is specified as 6. COL 06/7: CMLA of half foreground color is specified as 7. COL 06/8: CMLA of half foreground color is specified as 8. COL 06/9: CMLA of half foreground color is specified as 9.

C1 control codes	Control function	Possible to use or not	Restricted matters and supplements
			<p>COL 06/10: CMLA of half foreground color is specified as 10.            COL 06/11: CMLA of half foreground color is specified as 11.            COL 06/12: CMLA of half foreground color is specified as 12.            COL 06/13: CMLA of half foreground color is specified as 13.            COL 06/14: CMLA of half foreground color is specified as 14.            COL 06/15: CMLA of half foreground color is specified as 15.            COL 07/0: CMLA of background neutral color is specified as 0.            COL 07/1: CMLA of background neutral color is specified as 1.            COL 07/2: CMLA of background neutral color is specified as 2.            COL 07/3: CMLA of background neutral color is specified as 3.            COL 07/4: CMLA of background neutral color is specified as 4.            COL 07/5: CMLA of background neutral color is specified as 5.            COL 07/6: CMLA of background neutral color is specified as 6.            COL 07/7: CMLA of background neutral color is specified as 7.            COL 07/8: CMLA of background neutral color is specified as 8.            COL 07/9: CMLA of background neutral color is specified as 9.            COL 07/10: CMLA of background neutral color is specified as 10.            COL 07/11: CMLA of background neutral color is specified as 11.            COL 07/12: CMLA of background neutral color is specified as 12.            COL 07/13: CMLA of background neutral color is specified as 13.            COL 07/14: CMLA of background neutral color is specified as 14.            COL 07/15: CMLA of background neutral color is specified as 15.            palette specification COL P1(1byte) P2(1byte)            COL 02/0 04/0: palette number 0 is specified.            COL 02/0 04/1: palette number 1 is specified.            COL 02/0 04/2: palette number 2 is specified.            COL 02/0 04/3: palette number 3 is specified.            COL 02/0 04/4: palette number 4 is specified.            COL 02/0 04/5: palette number 5 is specified.            COL 02/0 04/6: palette number 6 is specified.</p>

C1 control codes	Control function	Possible to use or not	Restricted matters and supplements
			COL 02/0 04/7: palette number 7 is specified.
POL	Pattern polarity Controls	Δ	Only POL 04/0(normal polarity) and POL 04/1(inverse polarity1) are used. The process in case the inverse polarity is set as normal polarity is; background color → foreground color, half foreground color → half background color, half background color → half foreground color, foreground color → background color.
SSZ	Small Size	O	Not specified for DRCS of standard, middle size.
MSZ	Middle Size	O	Not specified for DRCS of standard, small size.
NSZ	Normal Size	O	Not specified for DRCS of middle, small size.
SZX	Character Size Controls	Δ	Doubled vertical, doubled horizontal, doubled vertical and horizontal can be used.
FLC	Flashing control	Δ	Cycling period of flashing is 1 second, and the standard time ratio of On and Off is 1:1. Refer to 4.5.2 for operation of flashing.
CDC	Conceal Display Controls	X	
WMM	Writing Mode Modification	X	
TIME	Time Controls	Δ	Can be used only for process waiting
MACRO	Macro Command	X	Macro definition of control codes is not used, and only the default macro can be used.
RPC	Repeat Character	Δ	In the scroll character string, parameter P1 with "0" is not operated.
STL	Start Lining and mosaic separation	Δ	Only underline is operated. Refer to 4.5.5 for operation of underline.
SPL	Stop lining and mosaic separation.	O	Only underline is operated.
HLC	Highlighting Character Block	Δ	Refer to 4.5.4 for operation of highlighting.
CSI	Control Sequence Introducer	O	

O: possible to use Δ: possible to use with restrictions X: impossible to use

#### 4.5.1.1 Operation of color specifications

Only 128 colors for receiver unit common fixed colors in Appendix-1 can be used for the color specification.

The specification of colors is specified using the following codes to indicate palette number, CMLA value in the palette and to which one of foreground, half foreground, half background that the colors are applied to.

- BKF-WHF is a name assuming the colors under palette0, therefore, it is referred to as CFLA0-CFLA7, which is another name for it in this specification in this volume.

- Palette specification COL 02/0 04/N (N is palette number :0-7)  
However, after the initialisation of caption, if palette0 continues to be used, then the palette specification can be omitted.
- In case the foreground color specification CMLA is 0-7, then CFLA0-CFLA7  
In case CMLA is 8 – 15, then COL 04/M (M is the CMLA number)  
After the initialisation of caption, if index value =7(0x07) continues to be used then both the palette0 specification and CMLA7 specification can be omitted.
- Background color specification COL 05/M (M is the CMLA number)  
After the initialisation of caption, if index value =8(0x08) continues to be used then, both the palette0 specification and CMLA8 specification can be omitted.
- Half foreground color specification COL 06/M (M is CMLA number)  
After the initialisation of caption, if index value =15(0x0F) continues to be used then, both the palette0 specification and CMLA15 specification can be omitted.
- Half background color specification COL 07/M (M is CMLA number)  
After the initialisation of caption, if index value =30(0x1E) continues to be used then, both the palette0 specification and CMLA14 specification can be omitted.

In case the index value of the 128 common colors indicated in appendix-1 are in hexadecimal numeral, then the upper 4 bits are handled by the palette number and the lower 4 bits are handled by the CMLA.

Example 1) In case of specifying the color of index value 0 to the background color.

Palette number =0 and CMLA=0, therefore COL 02/0 04/0 COL 05/0

After initializing, if anything other than palette0 is not specified, then it is only COL 05/0.

Example 2) In case of specifying the color of index value 0 as the foreground color.

Palette number =0, CMLA=0 , therefore COL 02/0 04/0 CFLA0

After initializing, if anything other than palette0 is not specified, then it is only CFLA0

Example 3) In case of specifying the color of index value 47(0x2F) as the foreground color.

Palette number =2 and CMLA=15, therefore COL 02/0 04/2 COL 04/15

Example 4) In case of specifying the color of index value 47(0x2F) as the half foreground color.

Palette number =2 and CMLA=15, therefore COL 02/0 04/2 COL 06/15

The meaning when the palette number is fixed as 0 and the half foreground colors and half backgrounds color are not specified is shown in Table 4-19 and thereafter.

**Table 4-19 The meaning of color specifications when the palette number is 0**

C1 control codes	Control function	Possible to use or not	Restricted matters and supplements
BKF (CFLA0: TBD)	Black Foreground	○	Foreground color is specified as black.
RDF (CFLA1: TBD)	Red Foreground	○	Foreground color is specified as red.
GRF (CFLA2: TBD)	Green Foreground	○	Foreground color is specified as green.
YLF (CFLA3: TBD)	Yellow Foreground	○	Foreground color is specified as yellow.
BLF (CFLA4: TBD)	Blue Foreground	○	Foreground color is specified as blue.
MGF (CFLA5: TBD)	Magenta Foreground	○	Foreground color is specified as magenta.
CNF (CFLA6: TBD)	Cyan Foreground	○	Foreground color is specified as cyan.
WHF (CFLA7: TBD)	White Foreground	○	Foreground color is specified as white.
COL	Color Controls	Δ	<p>Color specification COL P1(1byte)</p> <p>COL 04/8: foreground color is specified as transparent.</p> <p>COL 04/9: foreground color is specified as half-brightness red.</p> <p>COL 04/10: foreground color is specified as half-brightness green.</p> <p>COL 04/11: foreground color is specified as half-brightness yellow.</p> <p>COL 04/12: foreground color is specified as half-brightness blue.</p> <p>COL 04/13: foreground color is specified as half-brightness magenta.</p> <p>COL 04/14: foreground color is specified as half-brightness cyan.</p> <p>COL 04/15: foreground color is specified as half-brightness white.</p> <p>COL 05/0: Background color is specified as black.</p> <p>COL 05/1: Background color is specified as red.</p> <p>COL 05/2: Background color is specified as green.</p> <p>COL 05/3: Background color is specified as yellow.</p> <p>COL 05/4: Background color is specified as blue.</p> <p>COL 05/5: Background color is specified as magenta.</p> <p>COL 05/6: Background color is specified as cyan.</p> <p>COL 05/7: Background color is specified as white.</p> <p>COL 05/8: Background color is specified as</p>



C1 control codes	Control function	Possible to use or not	Restricted matters and supplements
			transparent. COL 05/9: Background color is specified as half-brightness red. COL 05/10: Background color is specified as half-brightness green. COL 05/11: Background color is specified as half-brightness yellow. COL 05/12: Background color is specified as half-brightness blue. COL 05/13: Background color is specified as half-brightness magenta. COL 05/14: Background color is specified as half-brightness cyan. COL 05/15: Background color is specified as half-brightness white. *1

\*1 reference: In analog caption, there is an exceptional provision for background color, half brightness white to use half transparent half brightness, therefore, COL 05/15 is different from the analog provision. In order to create the same condition as analog COL 05/15, the index value of half transparent color in broadcasting station equipment should be determined and it should be specified in accordance with this provision.

Table 4-20 Expansion control codes(CSI)

Character	Control function	Possible to use or not	Restricted matters and supplements
SWF	Set Writing Format	Δ	After initialization of the display screen, characters that involve display of bit maps and display performance, only before the appearance of the control code can be specified. Only 1 code parameter is used and only parameters that can be specified to P11...P1i is 7 (horizontal writing in 960x540) or 8(vertical writing in 960x540) when pixel size of caption plane is 960x540, 9 (horizontal writing in 720x480) and 10(vertical writing in 720x480) when pixel size of caption plane is 720x480. When this control code is specified, it is prioritized from the format specification of caption management data.
CCC	Composit Character Control	X	
RCS	Raster Color Command	O	After initialization of the display screen, characters that involve the display of bit maps and display performance, only before appearance of the control code can be specified. As for types of colors, from 0 (black) to 15 (half brightness white) defined in ARIB STD-B24 on top of common fixed colors from 16 to 127 can be specified. Refer to 4.5.3 for the operation of raster color control.
ACPS	Active Coordinate Position Set	O	
SDF	Set Display Format	O	After initialization of the display screen, characters that involve the display of bit maps and display performance, only before appearance of the control code can be specified.
SDP	Set Display Position	O	After initialization of the display screen, characters that involve the display of bit maps and display performance, only before appearance of the control code can be specified.
SSM	Character composition dot designation	Δ	Only 16x16, 20x20, 24x24, 30x30, 36x36 dots can be specified.
PLD	Partial Line Down	X	
PLU	Partial Line Up	X	
SHS	Set Horizontal Spacing	O	
SVS	Set Vertical Spacing	O	

Character	Control function	Possible to use or not	Restricted matters and supplements
GSM	Character deformation	X	
GAA	Coloring block	X	Colored block is full display zone only.
SRC	Raster Color Designation	X	
TCC	Switch control	X	
CFS	Character Font Set	X	
ORN	Ornament Control	Δ	No character ornament, only outlines can be used. Refer to 4.5.6 for operation of outlines.
MDF	Font	X	
PRA	Built-in sound replay	O	
XCS	External Character Set alternative code string definition	X	
SCR	Scroll definition	Δ	It can be used only in case of horizontal writing. Regarding one line lateral direction scrolling(with or without rollout), only 1 place 1 line can be used.

O: possible to use Δ: possible to use with restrictions X: impossible to use

#### 4.5.2 Operation of flashing

Flashing of the 8-bit character codes character string does the character flashing(foreground color 0.5 seconds, background color 0.5 seconds)to the foreground color (including half colors in case of 4 grayscale fonts)and background color, and flashing in the bit map data does bit map flashing (flashing color 0.5 seconds, raster color 0.5 seconds ) to flashing color and raster color(transparent color if the raster color is not specified)specified by the FLC header defined in the bit map figure encoding of ARIB STD-B24 with a 1 second cycle as the standard. The timing to begin the flashing is when the character specified by the flashing is rendered first or when figure is rendered after the flashing color is specified in the FLC header in the bit map data. When the initialization operation for caption management at the time of update, or instruction to delete the display screen is sent by the CS, the flashing of corresponding characters or picture element ends.

##### 4.5.2.1 Restricted matters

- Characters and bit map of caption and superimpose can be specified as desired within the common 128 colors, however, the color that does the flashing is assumed to be handled

separately from those 128 colors for implementation of the CLUT of receiver units, therefore the number of colors is limited as below.

- Flashing target colors of characters are valid foreground colors, half foreground colors and half background colors between [FLC 04/0] to [FLC 04/15], and there are less than 24 colors for one caption text data. Caption and superimpose can use 24 colors independently.
- Flashing target colors of bit map figures are the colors specified in the `flc_header()` and this is less than 16 colors for 1 caption statement data.
- Flashing is a positive phase only.
- Mixture with the border specification is prohibited.
- Mixture with scrolling specification is prohibited.

#### 4.5.3 Raster color control

Boxes in the entire display area. The raster color controlled area is the rectangular area instructed by the SDF (Set Display Format) and SDP (Set Display Position). Specifications for the raster color control and index values of CLUT (refer to Appendix-1) are specified in P11 ... P1i by RCS (raster color command).

#### 4.5.4 Operation of highlighting

Highlighting perform the control to add frames made of 4 outer sides within a display zone. Only when the display zone is in accordance with Table 4-16 is the correct box rendered across the entire display zone. (Even when the horizontal spacing and the vertical spacing are changed freely, is the function to render boxes across the entire display zone optional). When the box specification is done for lateral directions or line directions, and if the extra interval between characters/lines is less than 1 dot, then rendering a correct box is not guaranteed. Gradation of highlighting is implementation dependent. (Receiver unit implementation to make the box gradation a 4 level gradation is not required.)

#### 4.5.5 Operation of underlines

Underlines are rendered in a 1 dot width on the outer side in the display zone. In case of horizontal writing, it is attached to the side of the next line and in case of vertical writing, it is attached to the side of the previous line. Only when the display zone is in accordance with Table 4-16, is the underline across the entire display zone rendered correctly. (Even when the horizontal spacing and the vertical spacing are changed freely, is the function to render underlines across the entire display zone optional). When a value less than 2 dots are specified between lines, is the rendering of correct underlines not guaranteed. Gradation of underlines including implementation of 1dot width in 720x480 is implementation dependent. (Receiver unit implementation to make underlines a 4 level gradation is not required.)

#### 4.5.6 Operation of outlines

In case of outlining, specify values bigger than 2 dots between characters and lines. When a value less than 2 dots is specified between characters and lines, is the correct outlining display not guaranteed. Since there is no half color specification for the gradation of outlining, it is implementation dependent. (Receiver unit implementation to make bordering a 4 level gradation is not required.)

#### 4.5.7 Operation of scrolling

##### 4.5.7.1 One line lateral direction scroll

It is prohibited to instruct the SCR multiple times in the same text. In case of scrolling, it should be transmitted as a different data unit (this statement) that specified one line worth of display area by SDF.

Receiver unit performance when specifying scrolling is as follows.

- Scrolling is performed in a rectangular area specified by SDF and SDP, and it is not rendered outside the rectangular area.
- Virtual area of 1 character worth on the right side of the first line of the display area is assumed to exist and once the scrolling specification (SCR) is specified, the active position is reset in the virtual writing area.
- Characters that were written in the display area before the scrolling specification are deleted after the scrolling specification.
- Display from the right side of the display from the first character.
- Scrolling is started by writing characters to the virtual writing area.
- If without rollout, after displaying the last character, scrolling is stopped.
- If with rollout, the scrolling is continued until there are no characters on the screen.
- In case data that should be displayed next is received during scrolling, wait until the scrolling is finished.
- In case the values between characters /between lines specified from the beginning of the scrolling instruction to the end of scrolling exceed the maximum values defined in Table 4-16, then the scrolling display is implementation dependent.



Figure 4-15 Display area of one line lateral direction scrolling and the virtual writing area

#### 4.5.7.2 Control codes

The following control codes are not used from the beginning of instruction of scrolling and the end of scrolling.

Format specification	SWF, SDF, SDP
Size specification	SZX, SSM
Change of active position	APB, APR, APF, PAPF, APU, APD, APS, ACPS, SHS, SVS
Display effect	FLC, STL, SPL, HLC
Time control	TIME
Raster specification	RCS

APB, APR, APF, PAPF, APU, APD, APS, ACPS, FLC, STL, SPL, HLC and RCS specified before the scrolling instruction will be released at the start of the scrolling instruction.

#### 4.5.7.3 Scrolling speed

Receiver units will start scrolling as quickly as possible after the scrolling specification, however, there may be some delay between the scrolling specification and the start of scrolling. As for the specification of scrolling speed, less than 3 characters per 1 second is recommended regarding one line lateral direction scroll. Dot unit movement at one time is implementation dependent.

#### 4.5.8 Priorities of the display function

Display function by control codes is prioritized by the following order.

- 1) Size of characters
- 2) Outlining
- 3) Underline
- 4) Highlighting
- 5) Polarity Controls
- 6) Color specification, flashing

Note)Order for the ones with the same priority is as desired.

#### 4.6 Operation of the DRCS

For DRCS calling, DRCS set calling of 1 byte or 2 bytes is used as defined in ARIB STD-B24 Vol. 1 Part 2 7.1.1.5. Also, for the encoding of DRCS patterns, only pattern transmission is operated, and operation by geometric is not executed. Also, specify 0 for the "fontId". If anything other than 0 is specified, then the receiver unit is recognized as 0. Buffers secured by receiver units for the DRCS are 16KB in the DRCS for caption and 16KB in the DRCS for superimpose. The maximum number of DRCS used simultaneously for each caption and superimpose respectively is 188. (In case of placing the "patternData" of multiple sizes in 1 Character code, each one is counted as 1). "patternData" should send out 4 level gradation data for the design frame.

**Table 4-21 Setup parameters of DRCS figure encoding**

Field	Operation
NumberOfCode	Operated as defined.
CharacterCode	Operated as defined.
NumberOfFont	Operated as defined.
FontId	Only 0 can be specified. (Recognized as 0 even if anything other than 0 is specified.)
Mode	Only 0001 can be specified.
Depth	Only 2 can be specified.
Width	Operated as defined.*1
Height	Operated as defined.*1
patternData	Operated as defined.

\*1 The size that is defined in 4.4.3.Character size controls should be specified.

## 4.7 Operation of initialization

Receiver units should perform in accordance with the timing and performance items of the initialization operation defined below and the performance contents should perform in accordance with ARIB STD-B24 Vol. 1 Part 3 Chapter 8 Initializing operation. However, regarding the initialization timing of character size controls, it should be the same as “status instruction”. Also, when the selection of stations is performed, all initialization performance items relating to caption will be initialized. For the overall timing of the initialization of caption, refer to Appendix-13.

### 4.7.1 Initialization by caption management

Receiver units conduct the initialization performance for caption management at the time of update defined in ARIB STD-B24, when data groups of received caption management data is switched from group A to group B, or group B to group A.

At this time, the display area and display location are set in the values of Table 4-13. Also, the values for character size controls, horizontal spacing and vertical spacing are set in the values of Table 4-15.

### 4.7.2 Initialization by caption statements

Receiver units perform the initialization operation defined in ARIB STD-B24 when receiving the same caption statement data as data group and language during the presentation process.

### 4.7.3 Initialization by data units in this document

If data units of this text are included in the same caption text data as language and data group during the presentation process, the receiver unit performs the initialization operation defined in ARIB STD-B24 immediately before the presentation process of the receiver unit of data units of this text. Also, the default value at the time of initialization for the half foreground color is index value = 15, and for the half background color is index value = 30.

#### **4.7.4 Initialization by character control codes**

Receiver units perform the corresponding initialization operation defined in ARIB STD-B24 immediately before receiver unit execution of screen deletion (CS) and format selection (SWF). Also, the default value at the time of initialization for the half foreground color is index value = 15, and for the half background color is index value = 30.

### **4.8 Mono-media used in caption and superimpose**

#### **4.8.1 Operation of geometric**

Geometric is not used for caption and superimpose.

#### **4.8.2 Operation of bit map data**

Bit map data can be operated in superimpose; however, the number of usable colors is 128 colors of receiver unit common fixed colors.

#### **4.8.3 Operation of alert sounds**

Can be operated by both caption and superimpose. However, it is limited to built-in sound.

#### **4.8.4 Operation of additional sound**

Neither caption nor superimpose will be operated.

### **4.9 Expected performance of the receiver units**

- The number of caption and superimpose that can be displayed simultaneously is 1 caption and 1 superimpose for a total of 2.
- For receiver units, the presentation control of caption and superimpose should be controlled independently.
- As a basic rule, caption and superimpose are operated so that their display areas do not overlap with one another, however, when they do overlap, superimpose are prioritized and displayed before caption.
- When bit map data and text, and bit map data and another bit map data are overlapping each other, then what is written later has priority.
- The display size and position of caption and superimpose in data broadcasting programs are displayed with full screen area as the standard.
- Receiver units judge the existence of sending caption data by the existence of receiving caption management data. Displaying marks to notify the receiving of caption to viewers, display of caption, deletion are mainly executed using corresponding data as the standard. Considering the pauses for sending corresponding data between commercials, the timeout process should be



executed after caption management data has not been received for more than 3 minutes. Display control that is linked to other data such as EIT data is implementation dependent.

#### 4.9.1 Start/end of caption display

The performance of receiver units at the time of starting/ending of caption is as written in Table 4-22. However, start means the “start of caption display specified by caption text” and ending means “deletion of caption text”.

**Table 4-22 Operation of DMF in caption management data**

DMF	When starting	When ending
Automatic display at the time of reception	After reception is completed, the corresponding text is displayed immediately. Cannot be deleted until reception is finished.	After reception is completed, the corresponding caption text is deleted immediately.
Automatic non-display at the time of reception	Nothing is done.	Nothing is done.
Select display at the time of reception	After reception is completed, some kind of information to indicate that there are caption and superimpose should be presented. Possible to display, or to delete by selection of viewer.	After reception is completed, if the corresponding caption text is displayed, then delete immediately. If it is not displayed, then nothing is done.

When playing the recordings, it is in compliance with the time of reception.

#### 4.9.2 Start/end of superimpose display

The same as the start/end caption display.

#### 4.9.3 Set-up fields etc. in the receiver units

- Receiver units display the caption and superimpose of the language selected immediately before. For example, when the second language caption is selected during viewing of a program and when another program with caption is started, then the second language is displayed.
- At the time of shipping of the receiver unit, the default set up of receiver unit displays the first language.
- Receiver units that can set up language codes such as Japanese/English display caption and superimpose of the set language code.
- When the language set in the receiver unit or the caption/superimpose of the language code is not being transmitted, then the receiver unit displays the caption/superimpose of the first language.

#### 4.10 Roll-up mode (Optional)

The roll-up mode additionally displays caption data in units of lines, and it is a function to display while performing the roll-up towards the line direction at the time of line break. Implementation of receiver units is optional.

The roll-up mode is classified as a different display mode than normal caption in caption management data. Superimpose do not have the roll-up mode. Operation restrictions on the roll mode for normal caption are defined as below.

##### 4.10.1 Declaration of the roll-up mode

If the "rollup\_mode" field of caption management data is '01' then the language is recognized as roll-up mode. Other than that, even codes that are yet to be defined are recognized as non-roll up.

##### 4.10.2 Operation of caption management data in the roll-up mode

- Format field should be in horizontal writing.
- Only one text data unit can be placed within the same caption management data. DRCS is not operated.
- Data that can be scripted in text is limited to the control codes of SWF,SDF,SDP,SSM,SHS,SVS. In such cases, the dot number of the vertical direction of SDF (display component dot specification) is set as(font size + vertical spacing) times integer with considerations to smooth scrolling. Other control codes are the same as normal caption.

##### 4.10.3 Operation of caption statement data in the roll-up mode

- Only one text data unit can be placed within the same caption management data. DRCS is not operated.

###### 4.10.3.1 Operation of the text data unit of caption statement data

- 1 text data displays 1 line, but line break codes (APD, APR) are not placed.
- Considering the compatibility with the normal mode, CS should be placed at the start of the data unit.
- Only standard character and middle size characters of Table 4-14 can be placed for characters. DRCS is not operated.
- C0control codes: Only CS can be placed.
- C1control codes: Only CFLA0-CFLA7, COLORN, NSZ, MSZ can be placed.

##### 4.10.4 Initialization of the roll-up mode

- Initialization of caption management data is the same as normal caption.
- Receiver units do not execute clear screen or initialization of active positions, character size controls, horizontal spacing or vertical spacing when receiving text for the same caption text as

the language and data group during the presentation process, and only the instruction and calling of codes, color specifications, outlining, size of characters are initialized.

- Initialization by CS is not executed.

#### **4.10.5 Presentation of the roll-up mode**

- Receiver units perform line break execution automatically before displaying and after data unit for new caption text are received. However, when receiving for the first time after roll up declaration, line break should not be performed.
- Screen deletion by CS is not performed. If the text display goes beyond the right side of the display zone set by SDF, SDP, then characters are displayed after the execution of line break.
- Along with line break, if it goes beyond the display area in the vertical direction, then roll-up of the entire display area is by one line, and roll-up execution time is generally one second and it should be done in smooth manner.
- When displayed characters are in more than 2 lines in the display area, roll-up execution should be done for each line break.
- When the PTS time of PES packet of  $n+1$  is exceeded at the time of completing the  $n$ th PES packet presentation by receiving continuous superimpose, the receiver units will start presentation execution (roll-up) of  $n+1$  immediately.

#### **4.10.6 Restrictions of PES packet transmission of roll-up mode**

Changes in transmission methods of 4.2.2 are defined as below.

- In roll-up mode, the PTS standard is not the starting point of packet presentation, but it is at the time of the completion point of packet presentation, which is when the roll-up is started, and the Td is generally 0 seconds. At the time of completion of the  $n$ th PES packet presentation, the total information Vol. of the PES packets, which started sending after  $n$ th should not exceed the receiving buffer Vol. (64KB).
- For the interval of PES packet sending of the caption statement data group, the interval between the PTW time of the  $n$ th PES packet of sending sequence and the PTS time of the  $n+1$ -th PES packet should be bigger than the display lines of  $n$ th data x 1 second.

#### **4.10.7 Compatibility with receiver units not equipped the roll-up mode**

Implementation of the roll-up mode is optional, however receiver units without implementation should display data units of caption text as the normal caption. Because of this, each caption text of the roll-up mode has placed CS in the beginning and this CS is ignored in the roll-up mode.

#### 4.11 Caption outscreen display function (Optional)

The function to display video in a reduced size for the purpose of viewing character telop in television broadcasting screens without overlapping is called the caption outscreen display function. Broadcasting stations will not transmit the control signal to announce that caption are assuming this kind of display. Therefore, receiver units will not automatically recognize the caption broadcasting assuming outscreen display. Receiver units should be in the following display setup by the pressing of one remote control key by the viewer.

In the outscreen mode, video is scaled and positioned in the value preset in the receiver unit (Table 4-23). The video location position can be selected from 2 locations, which are the top of screen or the bottom of the screen. The selection method is implementation dependent.

**Table 4-23 Display method of video in the outscreen display function**

Display format of caption	Video plane pixel size	Display method
960X540	1920X1080	Scale to 96/128 and display in (120X2, 0X2)-(840X2, 405X2) or (120X2, 135X2)-(840X2, 540X2)
720X480 (16:9)	720X480 (16:9)	Scale to 96/128 and display in (90,0)-(630,360) or (90,120)-(630,480)
720X480 (4:3)	720X480 (4:3)	Scale to 96/128 and display in (90,0)-(630,360) or (90,120)-(630,480)

- Any performances other than assumed above will be implementation dependent.

## **5 Operation of multimedia encoding**

### **5.1 Introduction**

Regarding the operation of multimedia encoding, as a basic rule, it should be in accordance with

- ” Appendix 1 Operational guidelines”
  - ” Appendix 2 Operational guidelines for implementing basic services”
- in ARIB STD-B24 Vol. 2 “XML-based multimedia coding scheme”

In this chapter, the items not specified in the above documents will be specified.

### **5.2 Operation of NVRAM used commonly in Multimedia services in Digital Terrestrial Television Broadcasting**

The A-profile memory area for the all broadcasters and A-profile memory area for the affiliation and A-profile memory area for the specified broadcaster and A-profile memory area for the communication of the specified broadcaster and the memory area for bookmark service shown in Table 5-1 are established as allocation of NVRAM used for saving non-volatile information in Digital Terrestrial Television Broadcasting. Contents of data stored in the A-profile memory area for the all broadcasters, the A-profile memory area for the affiliation, the A-profile memory area of communication purpose for the specified broadcaster, the memory area for bookmark service are dependent on contents specified by all broadcasters or broadcaster independently of Digital Terrestrial Television Broadcasting.

**Table 5-1 NVRAM used in Digital Terrestrial Television Broadcasting**

Type	Meaning	NVRAM amount
A-profile memory area for the all broadcasters	Common area available for use by all digital terrestrial broadcasters.	2KB (Fixed length block of 64bytes * 32)
A-profile memory area for the affiliation	Common area available for use by broadcasters that belong to the same affiliation.	·4KB for 1 affiliation (Fixed length block of 64bytes * 64) ·Number of affiliations: more than 8
A-profile memory area for the specified broadcaster	Area occupied by each broadcaster	·4KB for one broadcaster (Fixed length block of 64bytes * 64) ·Number of broadcasters that should be secured by receiver units simultaneously: more than 12
A-profile memory area of communication purpose for the specified broadcaster	Area to share information with broadcasting contents and communication contents	·2KB for 1 broadcaster. (Fixed length block of 64bytes *32) ·Number of broadcasters that should be secured by receiver simultaneously: more than 12
Memory area for bookmark service	Area available to use for the bookmark service	·Total of more than 50 blocks of variable length block with maximum of 320 bytes.
Memory area for root CA certificates	Area to store root CA certificate of general purpose transmitted by carousels in memory.	·3KB for one certificate ·Quantity : 8
Memory area for registration transmission	Area to store messages that carry out registration transmission	·More than 3 blocks of variable length block with maximum of 1.5 KB.

NVRAM equipped in receiver units is a device with a limited number of writing. When writing times are exceeded, then a failure will occur and the lifespan of the receiver unit is shortened as a result. Therefore, it is recommended to pay attention so that exceeding the amount of writing is not done to NVRAM. This is explained in Appendix-6.

### 5.2.1 Allocation of NVRAM area

Numbers of each area for the A-profile memory area for the affiliation, the A-profile memory area for the specified broadcaster and the A-profile memory area of communication purpose for the specified broadcaster are specified by assuming receivable services in the receiving environment are able to be used as the standard. Therefore, in fringe areas, there are cases where it is possible to receive more services than this number of broadcasters and broadcaster affiliations by 1 receiver unit. Also, by changing the receiver setup location, broadcasters providing receivable services may change. Considering such situations, receiver units need to provide a confirmation method for the existence/non-existence of the NVRAM area allocation for broadcasters and broadcaster affiliations and methods to change the existence/non-existence that should be specified to users. Also, in the access from contents before/after allocation of the NVRAM area, a process in accordance with the rules of access limitations needs to be specified in this document.

### 5.2.1.1 Allocation of the A-profile memory area for the affiliation

The minimum number of affiliations that the A-profile memory area for the affiliation can be allocated to is 8. Allocation to affiliations where the affiliation identification (affiliation \_id) specified in Vol. 7 with the following 8 values, is required.

Values of affiliation identification that require allocation of the A-profile memory area for the affiliation :

0, 1, 2, 3, 4, 5, 6, 7

Therefore, since the area allocation for the above 8 affiliations does not need to be changed, methods to change or initialize do not have to be equipped. Also, allocation to affiliations other than the area 8 affiliations above in case the receiver unit has an area that exceeds that of the 8 affiliations depends on a model. However, when an area that is already allocated is being allocated to another affiliation, initializing should be executed. Initialization here refers to the status where an empty string (it is 0 when it is read as a numerical value) is returned at the time of reading.

### 5.2.1.2 Allocation of the A-profile memory area for the specified broadcaster and the A-profile memory area of communication purpose for the specified broadcaster

- In case there are 12 one-touch buttons placed in the remote control.

The A-profile memory area for the specified broadcaster and the A-profile memory area of communication purpose for the specified broadcaster should be allocated to broadcasters providing service that have one-touch remote control buttons registered. Even when there are unregistered one-touch buttons, secure the area for registration in the future. However, keep in mind that one area is allocated to one broadcaster even when the same broadcaster is allocated to multiple one-touch buttons. The NVRAM area that is allocated once will continue to be allocated to the same broadcaster as long as the broadcaster's service is registered to any of the one-touch buttons. That is, even when the one touch button for services specified by one broadcaster which is registered changes from button number "1" to button number "2", allocation of NVRAM will continue and initialization will not be performed.

- In case 12 one-touch buttons are not placed in the remote control.

The A-profile memory area for the specified broadcaster and the A-profile memory area of communication purpose for the specified broadcaster should be allocated to broadcasters providing services where the remote\_control\_key\_id is "1"- "12" and the branch number is "0". Even when any of the remote\_control\_key\_id's between "1"- "12" have not been received, the area should be secured for reception in the future. However, in case the remote\_control\_key\_id

is duplicated by different broadcasters, the broadcaster with the branch code “0” has the priority for allocation as a basic rule, but it is recommended to provide a selection method with another branch number service to the users.

- As the common element.

In case of newly allocating a desired NVRAM area to broadcasters, initialization should take place along with allocation. As a basic rule, release of area allocation for broadcasters who could continue to be received may confuse the users, therefore, it is not preferred. In case of releasing allocation, an adequate announcement to the users is necessary. The allocation of the NVRAM area to broadcasters other than broadcasters allocated as required above, and the continuation or discontinuation of NVRAM area allocation to broadcasters providing services that stray from the required conditions, depend on models, since the different processes are assumed depending on the number of NVRAM areas that the receiver units are equipped with. The timing of the process execution for the release of allocations and changes also depends on a model in the receiver units, but some attention should be paid so that the NVRAM access limit function to the original broadcaster will function without contradictions before and after registration.

In desired services, both the A-profile memory area for the specified broadcaster and the A-profile memory area of communication purpose for the specified broadcaster should be accessible or inaccessible together.

In case a re-scan (refer to Vol. 2) of receivable channels is generated, if a broadcaster who had allocated the NVRAM area before the re-scan is still receivable even after the re-scan, then the same area should be continued to be allocated to the broadcaster

### 5.2.2 Identification of the A-profile memory area for the all broadcasters

In case of reading and writing information for the A-profile memory area for the all broadcasters from the Multimedia service, one fixed length block is recognized as one file and the `readPersistentArray( )/writePersistentArray( )` is executed. The reading and writing of information for the A-profile memory area for the all broadcasters from the Multimedia service is done in fixed length block units. To identify the fixed length block, use the URI shown below.

`nvr://tr_common/<block number >`

`<block number> : 0~N`

Refer to 5.2.8 for details on information stored in each block of the A-profile memory area for the all broadcasters.

### 5.2.3 Identification of the broadcaster affiliates area

In case of reading/writing information for the A-profile memory area for the affiliation from the Multimedia service, 1 fixed length block is recognized as 1 file and the `readPersistentArray( )/writePersistentArray( )` is executed. The reading and writing of



information for the A-profile memory area for the affiliation from the Multimedia service is done in fixed length block units. To identify the fixed length block, use the URI shown below.

`nvrkam://<affiliation_id>;group/<block number>`

<affiliation\_id> : affiliation ID. Written in 2 digit hexadecimal

(when it is less than 2 digits then put 0 on the left side to make it 2 digits)

<block number> : 0~N

<affiliation\_id> cannot be omitted.

In case the broadcaster that is broadcasting the contents that are currently playing does not belong to the affiliation for the affiliation ID specified, then access to this area will fail. Refer to Vol. 4 Chapter 9 for the affiliation ID.

#### 5.2.4 Identification of the A-profile memory area for the specified broadcaster

In case of reading/writing information for the A-profile memory area for the specified broadcaster from the Multimedia service, 1 fixed length block is recognized as 1 file and the `readPersistentArray( )/writePersistentArray( )` is executed. The reading and writing of information for the A-profile memory area for the specified broadcaster from the Multimedia service is done in fixed length block units. To identify the fixed length block, use the URI shown below.

`nvrkam://[<original_network_id>;]local/<block number>`

<block number> : 0~N

<original\_network\_id> is always omitted, and it is regarded as the original\_network\_id of streams where the contents that are currently playing are transmitted is specified.

#### 5.2.5 Identification of the A-profile memory area of communication purpose for the specified broadcaster

In case of reading/writing information for the broadcaster exclusive area for communications from the Multimedia service, 1 fixed length block is recognized as 1 file and the `readPersistentArray( )/writePersistentArray( )` is executed. The reading and writing of information for the A-profile memory area of communication purpose for the specified broadcaster from the Multimedia service is done in fixed length block units. To identify the fixed length block, use the URI shown below.

`nvrkam://[<original_network_id>;]local_web/<block number>`

<original\_network\_id> is always omitted, and it is regarded as the original\_network\_id of streams where the contents that are currently playing is transmitted is specified.

#### 5.2.6 Identification of the memory area for bookmark service

In case of reading/writing information for the memory area for bookmark service from the

Multimedia service, 1 variable length block is recognized as 1 file and the “readBookmarkArray( )” / ”writeBookmarkArray( )” is executed. The reading and writing of information for the memory area for bookmark service from the Multimedia service is done in variable length block units. To identify the variable length block, use the URI shown below.

nvrाम://bookmark/<block number>

<block number> : 0~N(N should be greater than 49)

In case the names other than the above specification methods and the procedure function for the memory area for bookmark service (readBookmarkArray( ),writeBookmarkArray( ))are executed, the reading/writing of NVRAM is not performed and the readBookmarkArray( ) will return null (failure) and writeBookmarkArray( ) will return NaN (failure) as return values.

Regarding the operation of the bookmark service, refer to section 5.15.

### 5.2.7 Use of the viewer residential area information from the Multimedia service

In case of reading/writing from the Multimedia service for the “viewer residential area information “, which is assumed to be recorded in NVRAM by the initial setup function of the receiver unit, use the URI below.

nvrाम://receiverinfo/<regiontype>

Specifiable character strings as <regiontype> are shown in Table 5-2. For details of the “viewer residential area information”, refer to Vol. 2.

**Table 5-2 Types of viewer residential area information**

<regiontype>	Type	Possible/impossible to Read/Write from the Multimedia service	Field type
Prefecture	Prefecture area code (bit location in the prefecture area specification bit map (ARIB STD-B10 Annex G Table G-2))	Only reading is possible as a numerical value.	U:1B
Regioncode	Region code corresponding to the emergency information signal(wireless facility regulation Article 9-3-5, regulations for wireless station operation article 138, region code corresponding to Ministry of Posts and Telecommunications announcement Sho 60 no. 405)	Only reading is possible as a numerical value.	U:2B
Zipcode	Zip code(7 digits)	Read/Write is possible as a fixed length character string with a length of 7. (ex: 5001234)	S:7B

- In case of specifying URI's other than the identification method indicated in 5.2.2 – 5.2.7 to execute the readPersistentArray( ) and the writePersistentArray( ), then read/write of NVRAM is not performed, and the readPersistentArray( ) will return null(failure) and the writePersistentArray( ) will return NaN(failure) as return values.
- In case the viewer residential area information is not setup, and if the regiontype is prefecture then 255, if the regiontype is region code then 0, and if the regiontype is zip code then an empty string will be returned. In case values of other areas are not setup, then empty strings (0 if read as a numerical value) will be returned.

## **5.2.8 Operation of the A-profile memory area for the all broadcasters of receiver unit NVRAM**

### **5.2.8.1 Purpose of the A-profile memory area for the all broadcasters**

The information written in the A-profile memory area for the all broadcasters of the receiver unit NVRAM is used in order to improve the usability of viewers. To be more precise, it is used to supplement input information for viewers to avoid re-entering the same information that has been entered in other contents.

The corresponding information is only the input supplement information and it needs to be used under permission and confirmation of the viewer.

### **5.2.8.2 Format of the A-profile memory area for the all broadcasters**

Allocation within the A-profile memory area for the all broadcasters is as in Table 5-3.

**Table 5-3 Allocation of A-profile memory area for the all broadcasters blocks and the field structure.**

Block number	Viewer information element	Example	Number of characters	Character type <sup>(note 8)</sup>	Field type
0	Date/time of the last update			EUC- JP	(Date/time of the last update) U:6b minute U:5b hour U:5b day U:4b month U:12b year S:4B, network identification
1	Name in hiragana(1)	やまだ__たろう	2 byte character 15 characters (note 1)	EUC-JP Hiragana and symbols 2 byte code	S:30B, Name in hiragana S:24B, Name in kanji (Registration date) U:6b minute U:5b hour U:5b day U:4b month U:12b year S:4B, network identification
	Name in kanji(1)	山田__太郎	2 byte character 12 characters (note 2)	EUC-JP 2 byte code	
2	Name in hiragana and name in kanji(2)		Same as above	Same as above	Same as above
3	Name in hiragana and name in kanji(3)				
4	Name in hiragana and name in kanji(4)				
5	Name in hiragana and name in kanji(5)				
6	Name in hiragana and name in kanji(6)				
7	Name in hiragana and name in kanji(7)				
8	Name in hiragana and name in kanji(8)				

Block number	Viewer information element	Example	Number of characters	Character type <sup>(note 8)</sup>	Field type
9	Hiragana address(1)	みなとくだいば 2-4-8 ふじてれび ほんしゃびる	2 byte character 28 characters (note 3)	EUC-JP  Hiragana, alphanumeric and symbols 2 byte code	S:56B, Hiragana address(1) (registration date) U:6b minute U:5b hour U:5b day U:4b month U:12b year S:4B, network identification
10	Hiragana address(2)	かいはずきよく	2 byte character 24 characters (note 4)		S:48B, Hiragana address(2) (registration date) U:6b minute U:5b hour U:5b day U:4b month U:12b year S:4B, network identification
11	Kanji address(1)	港区台場 2-4-8 フジテレビ 本社ビル	2 byte character 28 characters (note 3)	EUC-JP 2 byte code	S:56B, kanji address(1) (registration date) U:6b minute U:5b hour U:5b day U:4b month U:12b year S:4B, network identification
12	Kanji address(2)	開発局	2 byte character 24 characters (note 4)		S:48B, kanji address(2) (registration date) U:6b minute U:5b hour U:5b day U:4b month U:12b year S:4B, network identification
13	Zip code	1078006	Single byte 7 characters (note 5)	Number of EUC-JP0-9 1 Byte code	S:7B, address zip code(registration date) U:6b minute U:5b hour U:5b day U:4b month U:12b year S:4B, network identification

Block number	Viewer information element	Example	Number of characters	Character type <sup>(note 8)</sup>	Field type
14	Phone number	03 1234 xxxx	Single byte 15 characters (note 6)	EUC-JP Numbers from 0-9 1 Byte code	S:15B, phone number(registration date) U:6b minute U:5b hour U:5b day U:4b month U:12b year S:4B, network identification
15	FAX NUMBER	03 1230 xxxx	Single byte 15 characters (note 6)		S:15B, FAX NUMBER(registration date) U:6b minute U:5b hour U:5b day U:4b month U:12b year S:4B, network identification
16	Mail address (1)	yama@ARIB.or.jp	Single byte 45 characters (note 7)	EUC-JP 1 Byte code	S:45B URI S:11B mobile phone number (registration date)
	Mobile number(1) phone	0901234xxxx	Single byte 11characters	EUC-JP Numbers from 0-9 1 Byte code	U:6b minute U:5b hour U:5b day U:4b month U:12b year S:4B, network identification
17	Mail URI and mobile phone number(2)	Same as above	Same as above	Same as above	Same as above
18	Mail URI and mobile phone number(3)				
19	Mail URI and mobile phone number(4)				
20	Mail URI and mobile phone number(5)				
21	Mail URI and mobile phone number(6)				
22	Mail URI and mobile phone number(7)				
23	Mail URI and mobile phone number(8)				

Block number	Viewer information element	Example	Number of characters	Character type <sup>(note 8)</sup>	Field type
24~31	RESERVED	-	-	-	-

(note 1) Separate the family name and first name with a 2 byte character space and store using hiragana or symbols.

(note 2) Separate the family name and first name with a 2 byte character space and store. Character type does not matter.

(note 3) Omit prefectures and store starting from city and ward.

(note 4) In case it cannot be stored in address (1), store the characters that cannot fit in (1) in address (2). When writing or updating the address in address (1), regardless of whether or not it fits within 28 characters, the block of address (2) should be re-written at the same time.

(note 5) When a hiragana address is input, then the input is required. When the hiragana address is updated, then the zip code should be updated along with it.

(note 6) Separate the area code, local office number, number with a single space and store.

(note 7) Do not record addresses with more than 45 characters.

(note 8) The definition of character types are as follows.

•Hiragana(2 byte code) :

Row 4 specified in ARIB STD B-24(Refer to Vol.1 Part 2 Table 7-4(1)(2))

•Alphanumerical(2 byte code) :

Row 3 specified in ARIB STD B-24(Refer to Vol.1 Part 2 Table 7-4(1)(2))

•Symbols(2 byte code) :

Row 1 and Row 2 specified in ARIB STD B-24(Refer to Vol.1 Part 2 Table 7-4(1)(2))

Except Row 1, Cell 13 to 18, and Row 2, Cell 94.

•Alphanumerical(1 byte code) :

alphanumerical group specified in ARIB STD B-24(Refer to Vol.1 Part 2 Table 7-5)

(1) The record length of each block of the A-profile memory area for the all broadcasters of NVRAM is a variable length.

- In case of executing the “readPersistentArray( )” / “writePersinstentArray( )”, specify the field type shown in Table 5-3 to the argument structure.

(2) Regarding names in hiragana and names in kanji.

- Writing is possible only when the input of kanji is possible in the kanji name field.
- In case of performing registration tasks from registration/change contents, the writing of names in hiragana is required and the writing of names in kanji is optional.
- As a basic rule, names in hiragana and names in kanji are written at the same time. In case of writing names in kanji later on for unavoidable reasons, names in hiragana need to be updated at the same time to avoid inconsistency of names in hiragana and names in kanji.
- Insert a 2 byte character space between the family name and first name for both names in the hiragana field and names in the kanji field.

(3) Regarding hiragana addresses and kanji addresses

- Input of kanji in the kanji address field is possible only when writing is possible.

- In case of performing registration tasks from registration/change contents, writing in the hiragana address is required and writing in the kanji address is optional.
  - As a basic rule, hiragana addresses and kanji addresses are written at the same time. In case of writing kanji addresses later on for unavoidable reasons, hiragana addresses need to be updated at the same time to avoid inconsistency of hiragana addresses and kanji addresses.
- (4) The relationship between zip codes and addresses
- Since prefecture names can be omitted in the address field, when registration/correction contents are writing in address field, make sure that the zip code is input at all times before execution.
  - Writing of zip codes and addresses should be done at the same time.
- (5) Regarding mail addresses
- The area to record mail addresses of individuals corresponds with the number of names in block 1-8. Contents should not record addresses that do not have corresponding names.
  - Writing of mail addresses should be done with the consent of the viewer.
- (6) Regarding mobile phone numbers
- The area to record mobile phone numbers of individuals corresponds with the number of names in block 1-8. Contents should not record phone numbers that do not have corresponding names.
  - Writing of mobile phone numbers should be done with the consent of the viewer.
- (7) Regarding character types of each block of the A-profile memory area for the all broadcasters
- Do not use “,” (single byte commas), and “:” (single byte colons).
  - Names in hiragana
    - Use hiragana and symbols.
  - Names in kanji
    - No restrictions on character types.
    - Use 2 byte code.
  - Hiragana addresses
    - Use hiragana, alphanumerical and symbols.
    - Use 2 byte code.
  - Kanji addresses
    - No restrictions on character types.
    - Use 2 byte code.



- Zip codes
  - Use alphanumerical 1 byte code.
  - Use only numbers from 0-9
- Phone numbers, Fax numbers
  - Use alphanumerical 1 byte code.
  - Insert a single byte space between the area code /local code/numbers.
  - Use only numbers from 0-9.
- Mobile phone numbers
  - Use alphanumerical 1 byte code.
  - Do not insert single spaces between numbers.
  - Use only numbers from 0-9.
- URI
  - Use alphanumerical 1 byte code.

Definitions of hiragana, alphanumerical characters, and symbols in this section are in accordance with the definitions in Table 5-3 (note 8).

(8) Operation of writing history

In case of writing in each block of the A-profile memory area for the all broadcasters, the date of update and network identification of the broadcaster who executed the update should be written to each block in accordance with the format in Table 5-4.

Also, when writing/updating of even just 1 block was executed, then record and update the date/time of the last update and network identification of the broadcaster.

**Table 5-4 Data structure of registration date**

	Data format	Data type	Data length
Registration date, Date/time of the last update	(Year/month/day/hour/minute) U:12b,U:4b,U:5b:U:5b:U:6b <sup>(note 1)</sup>	All UnsignedInteger	Total 4Bytes
Network identification	From the display in hexadecimal character strings of "0XXXXX" format to original_network_id <sup>(note2)</sup> format without 0x.	Text	4Bytes

(note 1) In case of December 1<sup>st</sup>, 2000, 23:59, then 2000,12,1,23,59 is written

(note 2) For "original\_network\_id", data obtained in "getProgramID( )" is written.

### 5.2.9 Specification relating to writing in the A-profile memory area for the all broadcasters

(1) The conditions that writable BML contents should satisfy.

The following conditions are required for BML contents that execute the writing.

- A Customer registration and changed contents of each broadcaster
- Broadcasters that writes/re-writes in the A-profile memory area for the all broadcasters should prepare customer registration, changed contents that satisfy this operational specification.
- B Customer registration specified by each broadcaster and general contents with built-in documents for changes.
- In case there is the need to write or re-write in this area from general contents, it should be executed after embedding the customer registration, documents for changes specified by each broadcaster in the contents.
  - Broadcasters need to prepare customer registrations and documents for changes that satisfy this operational specification and provide them to contents production companies.

(2) The conditions that BML contents that execute writing (re-writing) should satisfy.

When the BML contents specified above (1) is writing (overwriting) blocks, the following conditions are required to be satisfied.

- When information was updated by instruction of viewers from registration/changed contents (document) specified in (1).
- When client DB information is updated by instruction of viewers using methods other than the Internet on the back channel side of each broadcaster, and when information is registered or updated by instruction of viewers by pulling out the corresponding DB information from BML registration/changed contents (documents) by using some sort of method.

Other than when information is updated by the viewer, registration, changed contents should not overwrite fields in the A-profile memory area for the all broadcasters. (Prohibition to change, update without viewer operation)

…For example, when kanji is pulled out in the back channel side after the line is connected, receiver units(A-profile memory area for the all broadcasters of NVRAM)are not overwritten.

Personal information written is used to improve the usability of input supplements of viewers, and it is recommended to explicitly indicate to viewers that this information will not be diverted for other uses without permission from the viewer. Refer to the section “Specification when registering customer information to the center server” for details.

#### **5.2.10 Specification relating to reading from the A-profile memory area for the all broadcasters**

- Contents shall not leak information of the A-profile memory area for the all broadcasters using communication methods such as phone lines or LAN, etc. without permission from the viewer.
- Contents shall not copy information of the A-profile memory area for the all broadcasters from receiver units to external devices without permission from the viewer.

#### **5.2.11 Specification regarding customer registration, changed contents**

- Customer registration and changed contents (document) of each broadcaster should update the date/time of the last update, registration date and network identification only under the following circumstances.
  - 1) When the viewer updates information by using functions to change customer registration and changed contents information.
  - 2) In case incorrect information is recorded such as the registration date, and network identification of each field does not exist, and when the viewer is made to re-enter information in the field.
- When registering new client information, if there is no data written in the A-profile memory area for the all broadcasters, it is recommended that fields prepared in the A-profile memory area for the all broadcasters (names in hiragana, hiragana addresses, zip codes, phone numbers) be written in the fields.
- Customer registration and changed contents should have operations and functions to switch the order of names.
- When a part of a name is deleted, the field is regarded as an empty field and even when names are registered in the numbers after the deleted number, the order will not be rolled forward.
- It is recommended to avoid registering the same person twice.
- If each broadcaster makes additional registrations on its own in the 8 name fields of the common area, it will cause a shortage of number of fields and confusion for the operation, therefore, when newly registering names, methods to avoid duplicating registration such as in the below example should be executed.
  - 1) Display the names (maximum of 8 people) of the common area in a list in the first step of new registration, and among them, only the names that are not registered to each center will be selectable, and the viewer needs to select the name of the person that the viewer is trying to register. In case the name of person who is not registered to the center is not in the name field, then display the direct input field. (Only when there are spaces in the name fields of the common area)

- 2) In case the name of person to be registered is not in the list, the viewer needs to press the new name registration button to display the input field. (Only when there are spaces in the name fields of the common area)
  - 3) Normal registration sequence is as follows.
- To perform customer registration in the center side (by methods other than registration by broadcasting and updated contents), implementation of equipment to pull out center information from the registration and updated contents is desired.

#### **5.2.12 Specification when registering customer information to the center server**

Customer registration and changed contents (document) of broadcasters can be used as registration information to the client database of each broadcaster by reading personal information of the A-profile memory area for the all broadcasters when performing customer registration to the client data base of each broadcaster.

When registering personal information of the broadcaster common NVRAM, it is necessary to indicate the usage and purpose, and to obtain permission from the viewer before registration.

When registering, overwriting data written in the A-profile memory area for the all broadcasters is prohibited unless the information is updated by the viewer.

When registered information is irregular such as when the registration date or service identification of each field does not exist, information of the corresponding blocks is regarded as invalid and contents shall not be used for registration to the center's client DB.

When customer registration (no relationship with the receiver unit) is done in center of each broadcaster (client database side), then pulling out the corresponding information of the center and the writing of personal information to the A-profile memory area for the all broadcasters of NVRAM are permitted only when the viewer opens the registration/update contents of each broadcaster from the receiver unit to instruct registration within contents.

Registration and update contents link up multiple names of the A-profile memory area for the all broadcasters and information registered in the center server side in a ratio of 1:1, therefore, it is recommended to present multiple names to viewers and a screen (function) to prompt the linking operation should be specified.

Pulling out registration information of the center side should be done with an ID for the pull out presented to the registrant at the time as registration on the Internet, etc. and a password entered by the registrant.

### 5.2.13 Contents description guidelines of NVRAM access

- As written in section 5.2.1.2, if services by some broadcasters are not registered in the one-touch remote control buttons, then the A-profile memory area for the specified broadcaster and A-profile memory area of communication purpose for the specified broadcaster may not be used by the service of the broadcaster. Also, when the receiver unit cannot receive a BIT, (i.e.: immediately after power is turned on), whether or not the A-profile memory area for the affiliation is accessible cannot be judged and therefore access to the memory area may fail. Because of that, when accessing to the A-profile memory area for the specified broadcaster and A-profile memory area of communication purpose for the specified broadcaster, or the A-profile memory area for the affiliation, it is recommended that an error process in case of access failure be written each time and any unnatural performance be avoided.
- In case the broadcaster belongs to multiple affiliations, a multiple affiliation\_id is written in the BIT Extended Broadcaster Descriptor. In such services by broadcasters, even if the contents created by some affiliation access the A-profile memory area for the affiliation of a different affiliation by mistake, the receiver unit will not judge this as an error, so this may destroy the information recorded by broadcasters of other affiliations. Therefore, when accessing the A-profile memory area for the affiliation, special attention needs to be paid so that incorrect values are not written in the affiliation\_id written in the URI.
- The receivers supporting the AIT-controlled application linkage function can access the NVRAM's "A-profile memory area for the affiliation", "A-profile memory area for the specified broadcaster", and "A-profile memory area for communication purposes for the specified broadcaster" using the AIT-controlled application. Therefore, the broadcasters shall appropriately manage the contents to be written.

## 5.3 Use of remote control keys from Multimedia services

### 5.3.1 Values that used-key-list specification could take

Value of <key-group> specifiable to style characteristics "used-key-list" and the handling of remote control keys to control exclusive processes of remote control keys in BML browsers and the station selection function are shown in the Table below.

**Table 5-5 Values that key-group could take**

<key-group>	Meanings
basic	Arrow keys (↑, ↓, →, ←), decide key, back key
data-button	Color key (blue, red, green, yellow), bookmark key
numeric-tuning	Numbers key for selecting stations (0 to 9, 10, 11, 12, etc)
other-tuning	Keys relating to selecting stations (one touch station selection button, channel up down key, video key, one touch selection of stations of other media, media selection button)

- Control of the “d” button by used-key-list characteristics is not possible and if the “d” button is pressed when the BML browser is active, then it is always processed by the BML browser.
- When there are keys with a selection station function other than the ones above, whether or not they should be included in “other-tuning” depends on a model.

### 5.3.2 Handling of remote control keys, key codes, and access keys

Remote control keys that are usable in the Multimedia service, and the mapping of characters specified as key codes and access keys are shown in Table 5-6.

**Table 5-6 Handling of remote control keys, key codes, and access keys table**

Remote control keys	Key codes	Access key characters
↑	1	N/A
↓	2	N/A
←	3	N/A
→	4	N/A
0~9,10,11,12	5-17	N/A
“Decide”	18	N/A
“Back”	19	‘X’
“d”	20	N/A
Color key (blue)	21	‘B’
Color key(red)	22	‘R’
Color key(green)	23	‘G’
Color key(yellow)	24	‘Y’
Bookmark key	100	N/A
Station selection related keys that belong to “other-tuning”.	150-	N/A

- When the “d” button is pressed, only the event “DataButtonPressed” will occur and the event “keydown” and “keyup” will not occur.
- Number selection station keys and station selection related keys other than 0 to 9 are not used for purposes other than to detect station selection operations in contents executing online communication. Also, key codes to allocate station selection related to keys depend on models.

### 5.3.3 Guidelines of contents using selection by color keys

In case of creating contents that do not have any other methods for selection except color keys, the color key that should be selected by the viewer should be identifiable by characters, etc. which indicate the corresponding color other than color distinction.

### 5.4 Operation of the BML version

It is set as "major\_version=3", "minor\_version=0."

### 5.5 Operation of character codes

Refer to ARIB STD-B24 Vol. 2 Appendix 2 "4.1. Character codes".

#### 5.5.1 Transmission of DRCS pattern data

- Setup of pattern data of DRCS used in BML documents and 8-bit character codes that are externally referenced to are both done by "loadDRCS( )". However, do not externally refer to 8-bit character codes included in BML documents.
- For 1 CharacterCode, a DRCS pattern of multiple fonts and multiple sizes may be transmitted in some cases.
- The maximum total size of the DRCS pattern used in a BML document is 64KB.
- The relationship between fontID and tpestyle specified in the DRCS pattern data are as shown in Table 5-7. "fontID=0" is not operated.

**Table 5-7 Handling of fontID and tpestyle of DRCS**

fontID	Typestyle
0	(Not operated)
1	Round gothic
2	Angle gothic
3	Bold round gothic

### 5.6 Operation area of media type and mono-media

Should be in accordance with ARIB STD-B24 Vol. 2 Appendix 2 "4.2. Operational range of media types and monomedia" except for the following points.

- CSS data (media type "text/css") may appear in BML documents in some cases and may be transmitted as independent resources in other cases. CSS data transmitted as mono-media should be CSS on its own with a concluded description.
- ECMAScript data (media type "text/X-arib-ecmascript; charset='euc-jp'") may appear in BML documents in some cases and may be transmitted as independent resources in other cases. ECMAScript data transmitted as mono-media should be ECMAScript data on its own with concluded expression.

## 5.7 Operation of BML elements

### 5.7.1 Entity

Refer to ARIB STD-B24 Vol. 2 Appendix 2 "4.3.1.Entity"

### 5.7.2 Operational guidelines of BML elements

Operational guidelines for basic service of elements used in BML are shown in Table 5-8.

The meaning of "O", "Δ", "-" are in accordance with ARIB STD-B24 Vol. 2 Appendix 2 "Chapter 3 Legend"

**Table 5-8 Operational guidelines of elements**

Module		BML(Operation)	Elements
Core	Structure (required)	O	body
		O	head
		-	html
		O	title
	Text (required)	-	abbr
		-	acronym
		-	address
		-	blockquote
		O	br
		-	cite
		-	code
		-	dfn
		O	div
		-	em
		-	h1~h6
		-	kbd
		O	p
		-	pre
		-	q
		-	samp
		O	span
		-	strong
	-	var	
	Hypertext (required)	O	a
		-	
	List (required)	-	dl
		-	dt
-		dd	
-		ol	
-		ul	
-		li	
Applet (deprecated)	-	applet	
	-	param	
Text Extension	Presentation	-	b
		-	big
		-	hr
		-	i
		-	small
		-	sub
		-	sup
		-	tt
		Edit	-
	-		ins



Module		BML(Operation)	Elements
	Interaction channel	-	bdo
Form	Basic Forms	-	form
		-	input
		-	label
		-	select
		-	option
		-	textarea
	Forms	-	form
		O	input
		-	select
		-	option
		-	textarea
		-	button
		-	fieldset
		-	label
-	legend		
-	optgroup		
Table	Basic Tables	-	superimpose
		-	table
		-	td
		-	th
		-	tr
	Tables	-	superimpose
		-	table
		-	td
		-	th
		-	tr
		-	col
		-	colgroup
		-	tbody
		-	thead
-	tfoot		
Image		-	img
CS Image Map		-	a&
		-	area
		-	img&
		-	map
		-	object&
SS Image Map		-	img&
Object		O	object
		-	param
Frames		-	frameset
		-	frame
		-	noframes
Target		-	a&
		-	area&
		-	base&
		-	link&
		-	form&
Iframe		-	iframe
Intrinsic Events		-	a&
		-	area&
		-	form&
		-	body&
		-	label&
		-	input&
		-	select&
		-	textarea&
	-	button&	
Metainformation		O	meta

Module		BML(Operation)	Elements
Scripting		-	noscript
		O	script
Stylesheet		O	style
Style Attribute		O	
Link		O	link
Base		-	base
Name Identification (deprecated)		-	a&
			applet&
			form&
			frame&
			iframe&
			img&
Legacy (deprecated)		-	map&
			basefont
			center
			font
			s
			strike
			u
			body&
			br&
			superimpose&
			div&
			h1-h6&
			ht&
			img&
			input&
			legend&
			li&
			ol&
			p&
			pre&
			script&
			table&
			tr&
th&			
td&			
ul&			
BML	BML	O	bml
		O	bevent
		O	beitem
		O	body&
		O	div&
		O	p&
		O	span&
		-	a&
		-	bdo&
		O	object&
BML	Basic BML	-	bevent
		-	beitem
		-	body&
		-	div&
		-	p&
		-	span&
-	object&		

### 5.7.2.1 Restrictions on the order of elements within head elements

In a “head” element, there are a “title” element, a “style” element, a “link” element, a “script” element, and a “bevent” element, appearing in this order. A “title” element appears once and a “meta” element, a “style” element, a “link” element, a “bevent” element will appear 0 or 1 time. The “script” element appears 0 or 1 time without the “src” attribute specification, and the “script” element with the “src” attribute specification appears 0-2 times.

### 5.7.3 Attributes

Operational guidelines of attributes relating to elements specified for operation in section 5.7.2 are shown in Table 5-9. The meaning of “O”, “Δ”, “-” are in accordance with ARIB STD-B24 Vol. 2 Appendix 2 “Chapter 3 Legend.”

**Table 5-9 Operational guidelines relating to the attributes of elements**

Elements	Attribute	Operation	Restrictions for operation
Common Attributes			
Core Attributes			
	id	O	Character string with a maximum of 128 bytes
	class	O	
	title	-	
I18N Attributes			
	xml:lang	Δ	Fixed to "ja"
Events Attributes			
	onclick	O	
	ondblclick	-	
	onmousedown	-	
	onmouseup	-	
	onmouseover	-	
	onmousemove	-	
	onmouseout	-	
	onkeypress	-	
	onkeydown	O	
	onkeyup	O	
Style Attributes			
	style	O	
Core Modules			
Structure Module			
Body	%Common.attrib;		
	%Core.attrib;	O	
	%I18n.attrib;	Δ	
	%Events.attrib;	-	
	%Style.attrib;	O	
Head	%I18n.attrib;	Δ	
	profile	-	
Title	%I18n.attrib;	Δ	
Text Module			
Br	%Core.attrib;	O	
	%Style.attrib;	O	
Div	%Common.attrib;	O	
P	%Common.attrib;	O	
Span	%Common.attrib;	O	
Hypertext Module			

Elements	Attribute	Operation	Restrictions for operation
a	%Common.attrib;	O	
	accesskey	O	
	charset	Δ	Fixed to "EUC-JP"
	href	O	
	hreflang	-	
	rel	-	
	rev	-	
	tabindex	-	
	type	-	
Forms Modules			
Forms Module			
input	%Common.attrib;		
	%Core.attrib;	O	
	%l18n.attrib;	Δ	
	%Events.attrib;	O	Cannot be specified when "inputmode" attribute is "direct" or "indirect".
	%Style.attrib;	O	
	accesskey	O	
	checked	-	
	disabled	O	
	readonly	O	
	maxlength	O	From 1 to 40. note 1)
	alt		
	name	-	
	size	-	
	src	-	
	tabindex	-	
	accept	-	
	type	O	Either "text" or "password".
	value	O	
inputmode	O		
charactertype	O		
Client-side Image Map			
a&	corrd	-	
	shape	-	
input&	usemap	-	
object&	usemap	-	
Server-side Image Map			
input&	ismap	-	
Object Module			
object	%Common.attrib;	O	
	archive	-	
	classid	-	
	codebase	-	
	codetype	-	
	data	O	
	declare	-	
	height	-	
	name	-	
	standby	-	
	tabindex	-	
	type	O	
	width	-	
Target Module			

Elements	Attribute	Operation	Restrictions for operation
a&	target	-	
Intrinsic Events Module			
a&	onblur	O	
	onfocus	O	
body&	onload	O	
	onunload	O	Note 3)
input&	onfocus	O	
	onblur	O	
	onselect	-	
	onchange	O	Note 2)
Metainformation Module			
meta	%l18n.attrib;	Δ	
	http-equiv	-	
	name	O	
	content	O	
	scheme	-	
Scripting Module			
script	charset	Δ	Fixed to "EUC-JP".
	type	Δ	Fixed to "text/X-arib-ecmascript; charset=&quot;euc-jp&quot;".
	src	O	
	defer	-	
	xml:space	-	
Style Sheet Module			
style	%l18n.attrib;	Δ	
	type	Δ	Fixed to "text/css".
	media	Δ	Fixed to "tv".
	title	-	
	xml:space	-	
Link Module			
link	%Common.attrib;	-	
	charset	Δ	Fixed to "EUC-JP".
	href	O	
	hreflang	-	
	media	Δ	Fixed to "tv".
	rel	Δ	Fixed to "stylesheet".
	rev	-	
	type	Δ	Fixed to "text/css".
BMLmodule			
bml	%l18n.rrtib;	Δ	
	version	-	
	xmlns	-	
bevent	id	O	

Elements	Attribute	Operation	Restrictions for operation
beitem	id	O	
	type	O	One of the following is taken: "EventMessageFired", "ModuleUpdated", "ModuleLocked", "TimerFired", "DataEventChanged", "CCStatusChanged", "MainAudioStreamChanged", "NPTReferred", "MediaStopped", "DataButtonPressed", "IPConnectionTerminated"
	onoccur	O	
	es_ref	O	
	message_group_id	O	It is "0" or "1". When omitted, specification of "0" is assumed.
	message_id	O	
	message_version	O	
	module_ref	O	
	language_tag	O	
	register_id	-	
	service_id	-	
	event_id	-	
	peripheral_ref	-	
	time_mode	O	One of the following is taken: "absolute", "origAbsolute", "NPT"
time_value	O		
	object_id	O	Only the object element ID that indicates data transmitted by carousel and type attribute is either "audio/X-arib-aiff" or "audio/X-arib-mpeg2-aac".
	subscribe	O	
iframe&	align	-	
body&	invisible	O	
div&	accesskey	O	
	onfocus	O	
	onblur	O	
p&	accesskey	O	
	onfocus	O	
	onblur	O	
span&	accesskey	O	
	onfocus	O	
	onblur	O	
a&	effect	-	
bdo&	orientation	-	

Elements	Attribute	Operation	Restrictions for operation
object&	streamposition	O	The frame number is specified (type="image/X-arib-mng")when the mono-media that refers to the relevant object element is MNG. In case of other media, it is "0".
	streamlooping	Δ	Fixed to "1".
	streampositionnumerator	-	
	streampositiondenominator	-	
	streamstatus	O	Initial value is specified by mono-media that corresponds to the relevant object element. Refer to ARIB-B24 Vol. 2 Appendix 2 "4.8.5.2.Operation of attributes regarding presentation of stream data"
	streamlevel	-	
	remain	O	Possible/impossible is determined by mono-media that refers to the relevant object element. Refer to ARIB-B24 Vol. 2 Appendix 2 "4.8.5.1. Application of remain attribute for each type"
	accesskey	O	
	onfocus	O	
	onblur	O	

Note 1) When the input exceeds the maximum length, then it is rounded down. If it goes beyond the frame, the exceeded part will not be displayed.

Note 2) Generated timing of the change event is when the focus is shifted to a different element.

Note 3) Only extended functions for broadcasting that can be used in the "onunload" event handler are "writePersistentArray()" and "unlockModuleOnMemory()". Process contents should be limited to processes that end in a short time such as set up to Ureg where quick document transition is possible and judgement of simple condition, etc.

#### 5.7.4 Operation relating to beitem elements

Refer to ARIB STD-B24 Vol. 2 Appendix 2 "4.3.4. Operation of beitem element".

#### 5.7.5 Operation of the BML elemnts extension module (interruption event)

- The maximum number of ModuleUpdated events where the subscribe attribute can be set as subscribe simultaneously is 16.
- The maximum number of TimerFired events where the subscribe attribute can be set as subscribe simultaneously is 8. When absolute playback time/at the reception time/NPT time specified by TimerFired is time that is already passed at the time of BML document interpretation, then immediate firing will be executed.
- Performance when multiple event handlers are corresponded to for the same event depends on a model. The following descriptions, for example, fall under this category.
  - Multiple beitem elements whose type attribute is "DateEventChanged" are written.
  - Multiple beitem elements whose type attribute is "ModuleUpdated" are written and the same module is specified in the module\_ref attribute.
- Even when the data event is updated in the ES that is not being presented, the DateEventChanged event will not occur.

- Regarding modules transmitted by the ES that is not being presented, when the ModuleUpdated event is subscribed and the data event is updated by the ES transmitting the module, and if the pullout that comes along with it does not happen, then the ModuleUpdated event will occur. As the status value of the ModuleUpdated event in this case, the following values will be newly operated.

Value of status	Meaning
4	The data event was updated in the ES where the corresponding module is transmitted. The corresponding module was not sent out before the update of the data event, and the corresponding module was sent out after the update of the data event.
5	The data event was updated in the ES where the corresponding module is transmitted. The corresponding module was sent out before the update of the data event, and the corresponding module was sent not out after the update of the data event.
6	The data event was updated in the ES where the corresponding module is transmitted. The corresponding module was sent out before and after the update of the data event.

- The following specifications are stipulated regarding the MainAudioStreamChanged event.
  - Factors for generating events where the status is -1 depend on models.
  - When the data attribute of object elements presenting audio streams in documents that are currently presented is other than "/-1" or the occurrence of the MainAudioStreamChanged event when overwriting of the data attribute of corresponding object elements occurs depends on a model of receiver.
  - When the stream status attribute of object elements presenting audio streams in documents that are currently presented is other than "play", or the occurrence of MainAudioStreamChanged event when overwriting of stream status attribute of corresponding object element occurs depends on a model of receiver units.
  - The performance when "/-1" is specified to es\_ref depends on a model.
  - The performance in case the object element that refers to an audio stream does not exist in the document depends on a model of receiver units.
  - When the channel ID is not specified to es\_refAttribute, an event will not occur at the time of switching the main audio from/to primary/secondary.
  - When es\_ref is omitted in the MainAudioStreamChanged event, since specification for the targeted ES and channel ID is not possible, in which case an event should occur depends on a model of receiver units. Therefore, to operate contents safely, the es\_ref should always be specified.



## 5.8 Operation of CSS

Refer to ARIB STD-B24 Vol. 2 Appendix 2 "4.4. CSS-based style sheet".

## 5.9 Operation guidelines relating to procedure description

### 5.9.1 Operation area of DOM

Refer to ARIB STD-B24 Vol. 2 Appendix 2 "4.5.1. Operational guidelines of DOM"

### 5.9.2 Operation area of embedded objects

Refer to ARIB STD-B24 Vol. 2 Appendix 2 "4.5.2.Operation range of build-in objects"

### 5.9.3 Operation area of extended objects for broadcasting

Refer to ARIB STD-B24 Vol. 2 Appendix 2 "4.5.3.Operation range of extended objects for broadcasting"

### 5.9.4 Operation area of browser pseudo-objects

Operated in accordance with Table 5-10. The meaning of the "operation" fields are as follows.

"O"	Basic function in this specification.
"O(*)"	Optional function in this specification. Therefore, in case of using these functions in contents, inspect the possible/impossible of process function of the corresponding function in the "getBrowserSupport( )" function, and call the corresponding function only when the process is possible.
"O(*1)"	Optional function that was once a required function (see Appendix 15). In these guidelines, it is desirable to inspect whether the receiver is capable of processing this function using the "getBrowserSupport( )" function based on the provision in 5.12.6.9(14), and to call this function only when the process is possible.
.	
"-"	This is neither a basic or optional function in this specification.

**Table 5-10 Operation area of browser pseudo-objects**

	Functions	Operation	Comments
Ureg related functions			
	Ureg[]	O	
Greg related functions			
	Greg[]	O	
EPG functions			
	epgGetEventStartTime( )	O	
	epgGetEventDuration( )	O	
	epgTune( )	O	
	epgTuneToComponent( )	O	
	epgTuneToDocument( )	-	
	epgIsReserved( )	O	
	epgReserve( )	O	
	epgCancelReservation( )	O	
	epgReclsReserved( )	O	
	epgRecReserve( )	O	
	epgRecCancelReservation( )	O	
Event group index functions			
	grplsReserved( )	-	
	grpReserve( )	-	
	grpCancelReservation( )	-	
	grpReclsReserved( )	-	
	grpRecReserve( )	-	
	grpRecCancelReservation( )	-	
	grpGetNodeEventList( )	-	
	grpGetERTNodeName( )	-	
	grpGetERTNodeDescription( )	-	
	epgXTune( )	-	
Series appointment functions			
	seriesIsReserved( )	O(*)	(note 1)
	seriesReserve( )	O(*)	(note 1)
	seriesCancelReservation( )	O(*)	(note 1)
	seriesReclsReserved( )	O(*)	(note 1)
	seriesRecReserve( )	O(*)	(note 1)
	seriesRecCancelReservation( )	O(*)	(note 1)
NVRAM functions			
	readPersistentString( )	-	
	readPersistentNumber( )	-	
	readPersistentArray( )	O	
	writePersistentString( )	-	
	writePersistentNumber( )	-	
	writePersistentArray( )	O	
	copyPersistent( )	-	
	getPersistentInfoList( )	-	
	deletePersistent( )	-	
	getFreeSpace( )	-	
Functions for controlling access-controlled non-volatile memory areas			
	setAccessInfoOfPersistentArray( )	-	
	checkAccessInfoOfPersistentArray( )	-	
	writePersistentArrayWithAccessCheck( )	-	
	readPersistentArrayWithAccessCheck( )	-	
Interaction channel communication			

	Functions	Operation	Comments
Interaction channel communication- delayed calling			
	registerTransmission( )	-	
	registerTransmissionStatus( )	-	
	getTransmissionStatus( )	-	
	setDelayedTransmissionDataOverBasic( )	-	
Interaction channel communication -Basic Procedure			
	connect( )	O(*)	(note 4)
	disconnect( )	O(*)	(note 4)
	sendBinaryData( )	-	
	receiveBinaryData( )	-	
	sendTextData( )	O(*)	(note 4)
	receiveTextData( )	O(*)	(note 4)
Interaction channel communication-TCP/IP			
	setSPPParams( )	O	(note 4) (note 5)
	getSPPParams( )	O	(note 4) (note 5)
	connectPPP( )	O	(note 4) (note 5)
	connectPPPWithISPPParams( )	O	(note 4) (note 5)
	disconnectPPP( )	O	(note 4) (note 5)
	getConnectionType( )	O	
	isIPConnected( )	O	
	confirmIPNetwork( )	O	
	saveHttpServerFileAs( )	-	
	saveHttpServerFile( )	-	
	sendHttpServerFileAs( )	-	
	saveFtpServerFileAs( )	-	
	saveFtpServerFile( )	-	
	sendFtpServerFileAs( )	-	
	sendTextMail( )	O(*)	
	sendMIMEMail( )	O(*)	
	transmitTextDataOverIP( )	O	
	setDelayedTransmissionData( )	-	
	setCacheResourceOverIP( )	O(*)	
Interaction channel communication – Acquisition function of delayed calling status that is common for basic system procedures and IP connections.			
	getDelayedTransmissionStatus( )	-	
	getDelayedTransmissionResult( )	-	
Interaction channel communication-Functions to acquire line connection status			
	getPrefixNumber( )	O	(note 2) (note 4)
Interaction channel communication-Large Vol. call acceptance service			
	vote( )	O(*)	(note 4)
Interaction channel communication-Encrypted communication using CAS			
	startCASEncryption( )	-	
	transmitWithCASEncryption( )	-	
	endCASEncryption( )	-	
Interaction channel communication- Encrypted communication with a secret key not using CAS			
	setEncryptionKey( )	-	
	beginEncryption( )	-	
	endEncryption( )	-	

	Functions	Operation	Comments
Operational control functions			
	reloadActiveDocument( )	O	
	getNPT( )	O	
	getProgramRelativeTime( )	O	
	isBeingBroadcast( )	O	
	lockExecution( )	-	
	unlockExecution( )	-	
	lockModuleOnMemory( )	O	
	unlockModuleOnMemory( )	O	
	setCachePriority( )	O	
	getTuningLinkageSource( )	-	
	getTuningLinkageType( )	-	
	getLinkSourceServiceStr( )	-	
	getLinkSourceEventStr( )	-	
	getIRDID( )	O	
	getBrowserVersion( )	O	
	getProgramID( )	O	
	getActiveDocument( )	O	
	lockScreen( )	O	
	unlockScreen( )	O	
	getBrowserSupport( )	O	
	launchDocument( )	O	
	launchDocumentRestricted( )	O(*1)	
	quitDocument( )	O	
	launchExApp( )	O(*)	(note 3)
	getFreeContentsMemory( )	O	
	isSupportedMedia( )	O	
	detectComponent( )	O	
	lockModuleOnMemoryEx( )	O	
	unlockModuleOnMemoryEx( )	O	
	unlockAllModulesOnMemory( )	O	
	getLockedModuleInfo( )	O	
	getBrowserStatus( )	O	
	getResidentAppVersion( )	O	
	isRootCertificateExisting( )	O	
	getRootCertificateInfo( )	O	
	startResidentApp( )	O(*)	
	startExtraBrowser( )	O(*)	
	transmitDataToSmartDevice( )	O(*)	
Receiver audio control			
	playRomSound( )	O	
Timer functions			
	sleep( )	O	
	setTimeout( )	-	
	setInterval( )	O	
	clearTimer( )	O	
	pauseTimer( )	O	
	resumeTimer( )	O	
	setCurrentDateMode( )	O	
External character functions			
	loadDRCS( )	O	
	unloadDRCS( )	-	
External device control functions			
	enumPeripherals( )	-	
	passXMLDocToPeripheral( )	-	

	Functions	Operation	Comments
Other functions			
	random( )	O	
	subDate( )	O	
	addDate( )	O	
	formatNumber( )	O	
Closed caption display control functions			
	setCCStreamReference( )	-	
	getCCStreamReference( )	-	
	setCCDisplayStatus( )	O	
	getCCDisplayStatus( )	O	
	getCCLanguageStatus( )	O	
Directory operation functions			
	saveDir( )	-	
	saveDirAs( )	-	
	createDir( )	-	
	getParentDirName( )	-	
	getDirNames( )	-	
	isDirExisting( )	-	
File operation functions			
	saveFile( )	-	
	saveFileAs( )	-	
	getFileNames( )	-	
	isFileExisting( )	-	
File input/output functions			
	writeArray( )	-	
	readArray( )	-	
Query functions			
	getDirInfo( )	-	
	getFileInfo( )	-	
	getCarouselInfo( )	-	
	getModuleInfo( )	-	
	getContentSource( )	O(*)	
	getStorageInfo( )	-	
Data carousel accumulation functions			
	saveCarouselAs( )	-	
	saveCarousel( )	-	
	saveModuleAs( )	-	
	saveModule( )	-	
	saveResourceAs( )	-	
	saveResource( )	-	
Bookmark control functions			
	writeBookmarkArray( )	O	
	readBookmarkArray( )	O	
	deleteBookmark( )	O	
	lockBookmark( )	O	
	unlockBookmark( )	O	
	getBookmarkInfo( )	O	
	getBookmarkInfo2( )	O	
	startResidentBookmarkList( )	O(*)	

	Functions	Operation	Comments
Print related function API – basic print functions			
	getPrinterStatus()	O(*)	
	printFile()	O(*)	
	printTemplate()	O(*)	
	printUri()	O(*)	
	printStaticScreen()	O(*)	
Print related function API – Memory card related			
	saveImageToMemoryCard()	O(*)	
	saveHttpServerImageToMemoryCard()	O(*)	
	saveStaticScreenToMemoryCard()	O(*)	
IPTV linkage function			
	startVOD()	O(*)	
	startDlcDownload()	O(*)	
	getDlcDownloadStatus()	O(*)	
AIT-controlled application linkage function			
	startReceivingAIT()	O(*)	(note 6)
	getReceivedAIT()	O(*)	(note 6)
	startAITControlledApp()	O(*)	(note 6)

(note 1) Should be equipped on receiver units with the series appointment function.

(note 2) It is equipped on receiver units with modems.

(note 3) In accordance with ARIB STD-B24 Vol. 2 Appendix 1 even when using independent services.

(note 4) When being called by a receiver that does not support this function, a return value indicating an error shall be returned.

(note 5) This function is not required to be supported by receivers without a modem and receivers that do not have a modem but have another high-speed communication function.

(note 6) This function shall be supported by receivers that implement the AIT-controlled application linkage function.

### 5.9.5 Operation range of navigator pseudo-objects

It is not operated.

### 5.9.6 Extended functions that have been added in the digital terrestrial broadcasting

#### 5.9.6.1 Print function (Optional)

Refer to section 6.2.

Refer to "7.6.17 Functions for Printing" in ARIB STD-B24 Vol.2 for the print-related functions.

#### 5.9.6.2 Receiver application identification function (Optional)

Refer to "7.6.8 Operational control functions" in ARIB STD-B24 Vol.2 for getResidentAppVersion( ), the receiver application identification function.

Refer to "5.12.6.9 Operation of operational control function (11)" for the operation of getResidentAppVersion( ).

#### 5.9.6.3 Function to acquire root CA certificate information

- Confirmation of root CA certificate

Refer to "7.6.7.9 Functions for operating root certificates for encrypted transmission" in ARIB STD-B24 Vol.2 for isRootCertificateExisting(), the confirmation function for the root certificate.

Refer to "5.12.6.9 Operation of operational control function (12)" for the operation of `isRootCertificateExisting()`.

- Information acquisition of general-purpose root CA certificates

Refer to "7.6.7.9 Functions for operating root certificates for encrypted transmission" in ARIB STD-B24 Vol.2 for `getRootCertificateInfo()`, the information acquisition function for the general-purpose root certificate.

#### **5.9.6.4 Cache set-up function of communication contents (Optional)**

Refer to "7.6.7.6 Communication functions assuming TCP/IP" in ARIB STD-B24 Vol.2 for `setCacheResourceOverIP()`, the cache setting function for communication content.

Refer to "5.14.6.6 Interaction channel function-TCP/IP" for the details of the operation of `setCacheResourceOverIP()`.

#### **5.9.6.5 Receiver's native application startup function (Optional)**

Refer to "7.6.8 Operational control functions" in ARIB STD-B24 Vol.2 for `startResidentApp()`, the receiver application activation function.

Refer to "5.12.6.9 Operation of operational control function (13)" for the operation of `startResidentApp()`.

#### **5.9.6.6 Specifiable values in arguments of "getBrowserSupport()"**

Refer to "5.12.6.9 Operation of operational control function (14) Operation of `getBrowserSupport()`".

The receivers that implement the IPTV download/VOD (integrated service type 2) function shall support the arguments of `getBrowserSupport()` as provided in 7.2.3.2.3 (1).

The receivers that implement AIT-controlled applications shall support the arguments of `getBrowserSupport()` as provided in 8.8.1.

### **5.10 Restrictions in the BML document description**

Should be in accordance with ARIB STD-B24 Vol. 2 Appendix 2 "4.7 Restrictions on description of BML documents" except for the following points.

- Script elements without the `src` attribute specification will appear 0 or 1 time, and script elements with the `src` attribute specification will appear 0-2 times. When writing the contents of script elements, put the entire element in `<![CDATA[ and ]]>`, and make it as one CDATA section.

### **5.11 Presentation control of BML document**

Should be in accordance with ARIB STD-B24 Vol. 2 Appendix 2 "4.8 Guidelines for presentation control of BML document" except for the following points.

(1) When the object elements whose type attribute value is "image/jpeg" (JPEG object) satisfy

the following conditions, the remain attribute of the element can be made valid.

- Remain attribute of other JPEG objects of the same BML documents are not valid.
- Other JPEG objects and the display area do not overlap each other in the same BML document.
- The target JPEG object is set before the video image in z-order (including implicit z-order based of order of appearance of elements) and the width characteristics and height attribute of object elements for the video image are the same value as the horizontal pixel size and vertical pixel size indicated by the pixel size characteristics of body elements.
- The target JPEG object should be a sub element of the div element, which is the first sub element of the body element. Also, the sub elements of the above div element should be in the following combinations.
  - JPEG object where at most one remain attribute is valid.
  - Object element where at most the value of the type attribute is one is "video/X-arib-mpeg1" or "video/X-arib-mpeg2", and the remain attribute is valid.
  - Object element where at most the value of type attribute is one is "audio/X-arib-mpeg2-aac" and remain attribute is valid.
- In case the document is in transition while the remain attribute of the target JPEG object is valid, then the resource that is specified by the data attribute of the corresponding JPEG object should be locked in beforehand by "lockModuleOnMemory()" or "lockModuleOnMemoryEx()".
- Background-image characteristics of the body element, which is the ancestor element of the target JPEG object should not be specified.

(2) When the stream status attribute is stopped for the file audio with the type attribute of "audio/X-arib-mpeg2-aac", then playback of audio is stopped.

## 5.12 Behavior of BML browsers

### 5.12.1 Operational guidelines relating to presentation

Refer to ARIB STD-B24 Vol. 2 Appendix 2 "5.1.Operational guidelines of presentation" The following specifications, however, need to be added.

- In the case of the following, execute displays in accordance with TV video pixel size.
  - While receiving empty carousels
  - While acquiring BML documents immediately after selecting stations
- When switched to a non empty carousel by updating the data event while receiving empty carousels, execute displays in accordance with TV video pixel size until the BML document is displayed.



- In case the invisible attribute of body element of BML document is invisible, then execute displays in accordance with TV video pixel size just in the same way as when the BML document is not presented.
- During the performance of BML browser, if presentation of the BML document is needed, then
  - Document transition by “launchDocument()”, etc.
  - Transition to entry component by pull flag
  - In case of transition to the startup document by an update of the data event during presentation, etc., during the period until the presentation of the document is started after completing the acquisition of new the BML document, (For example, the period after update of the date event happens and until the presentation of startup document actually starts) the presentation of the BML document that was presented immediately before is continued.
- Regarding the operation of CLUT, it should be in accordance with ARIB STD-B24 Vol. 2 Appendix 3, Table 5-4.

#### **5.12.2 Operational guidelines relating to the performance of external characters**

Refer to ARIB STD-B24 Vol. 2 Appendix 2 “5.2. Guidelines on behavior of management of external characters”.

#### **5.12.3 Behavior of the DOM**

Refer to ARIB STD-B24 Vol. 2 Appendix 2 “5.3 Behavior of DOM”. However, restrictions on the “size of character string set to NodeValue” written in section 5.3.5 Table 5-13 are not applied to CDATA section in script element and style element.

#### **5.12.4 Operation of script language**

Refer below. In BML documents where multiple script elements are scripted, the following restriction will be applied for the status where all scripts (scripts written in resources indicated by the src attribute of script elements, and the script written inside of script elements without the src attribute) are read.

- ARIB STD-B24 Vol. 2 Appendix 3 “5.4.1. Operation of script working environment”
- ARIB STD-B24 Vol. 2 Appendix 2 “5.4.2. Data type” “5.4.3 Effects on basic objects caused by data type restrictions”, “5.4.4. Operational general rule of implementation-dependent behavior”

#### **5.12.5 Extended objects for broadcasting**

Refer to ARIB STD-B24 Vol. 2, Appendix 3, 5.5. “Operational guidelines on extended objects for broadcasting”.

### 5.12.6 Browser pseudo-objects

Operation of browser pseudo-objects is specified below. For those without particular specifications, they should be in accordance with ARIB STD-B24 Vol. 2 “XML-based multimedia coding scheme”.

- “Appendix 1 Operational guidelines”
- “Appendix 2 Operational guidelines for implementing basic services”

#### 5.12.6.1 Operation of Ureg

Should be in accordance with ARIB STD-B24 Vol. 2 Appendix 2 “5.6.1. Operation of Ureg”. The timing for receiver units to write the contents group identifier (service\_id in hexadecimal character string in “0XXXXX” format) to Ureg[0] is when the initial document of the contents group is acquired, in other words, when power is turned on, and when selecting stations.

Initialization of the Ureg value is done by the receiver unit when the browser is started or contents groups are changed (when selecting stations including when media is changed).

The value read after initialization is an empty strings.

See 7.2.3.2.1 for operation of Ureg in receivers that support the IPTV download/VOD (integrated service type 2) function.

See 8.6.1 for operation of Ureg in receivers that support AIT-controlled applications.

#### 5.12.6.2 Operation of Greg

Should be in accordance with ARIB STD-B24 Vol. 2 “7.6.16 Greg pseudo object properties” The value of the Greg is initialized at the time of turning the power on, and the value is kept at all times while the power is ON. The value read after initialization is an empty string. Even when the communication contents that do not support Greg are presented, the Greg value is maintained. In case of presenting other media that do not support Greg, it is preferred for the Greg value to be maintained.

In case the Greg value cannot be maintained unavoidably while presenting other media, the receiver unit initializes the value of Greg when presenting the first media that is supporting Greg after that.

See 7.2.3.2.2 for operation of Greg in receivers that support the IPTV download/VOD (integrated service type 2) function.

See 8.6.2 for operation of Greg in receivers that support AIT-controlled applications.

#### 5.12.6.3 Operation of EPG functions

Should be in accordance with ARIB STD-B24 Vol. 2 Appendix 2 “5.6.2. Guideline for behavior of EPG functions” except for the additional specifications below.

(1) Operation of epgTune(), epgTuneToComponent()

Operation of epgTune()/epgTuneToComponent() are in accordance with ARIB STD-B24 Vol. 2 Appendix 1 "8.5.1 Operation of operational control functions", however, the performance when "arib://-1.-1.-1" or "arib-dc://-1.-1.-1" is specified as a service at transition destination by argument is specified as below. (Regarding the operation of service specification by the name of "arib://-1.-1.-1", also refer to section 5.13.3)

- Receiver units will finish BML browser once. (note1) And then,
  - In case of epgTune(), execute 4 and after the "Basic operation of receiver when selecting channels" written in section 2.1.10.2.
  - In case of epgTuneToComponent(), execute acquisition and presentation of the startup document of specified components.

(note 1) Regarding the implementation of epgTune()/epgTuneToComponent(), securing broadcasting program and contents uniqueness in particular should be carefully taken care of. (Refer to Vol. 2 9.3) For example, when 2 screens are displayed in the receiver features, and when epgTune()/epgTuneToComponent() is executed from the document presented by one of the 2 screens, the BML browser presenting the document that had called the function will end and presentation of video, audio, contents at the newly selected station specified by the function on the same screen is demanded. Implementation to present services specified by the same function to screens other than the screen presenting the document which executed the same function while not ending the original document and the screen presenting the original document will not be allowed.

- In case of TV programs with additional data, for TV video, TV audio, closed caption, the same ES will be played without pauses even after function execution.
  - Display, etc. of channel banners along with selecting station performance is not executed.
- (2) Operation of "epgReserve()" / "epgCancelReservation()" / "epgRecReserve()" / "epgRecCancelReservation()"

After completing performance of the function, the succeeding script will be executed.

#### 5.12.6.4 Operation of series reservation function

Operation is not required. In case of operation, it should be in accordance with ARIB STD-B24 Vol. 2 "7.6.3 Series reservation functions". Also, the following specifications will be added.

- Whether or not receiver units can handle series reservation related to the API is identified by "getBrowserSupport()".
- <series\_scope\_ref> used in identification of series is

<network\_id>/<terrestrial\_broadcaster\_id>/<media\_type>. In here,

<network\_id> Network ID (4 digits hexadecimal)

<terrestrial\_broadcaster\_id> Broadcaster ID (4 digits hexadecimal)

<media\_type> media type

- 1 TV type media type
- 2 Audio type media type
- 3 Data type media type

For ID's specified in hexadecimal, "0x" in the beginning or "h" at the end will not be attached.

- After completing performance of the function, the succeeding scripts will be executed.

#### **5.12.6.5 Operation of the NVRAM functions**

Should be in compliance with ARIB STD-B24 Vol. 2 Appendix 2 " 5.6.3. Operation of non-volatile memory function"

#### **5.12.6.6 Interaction channel function-Operation of TCP/IP**

Refer to "5.14.6.6 Interaction channel function-TCP/IP"

(Note) If this function is called by a receiver that does not support the function, a return value indicating an error shall be returned and the browser operation shall not be stopped.

#### **5.12.6.7 Interaction channel function-Operation of the function that acquires the line connection status**

Refer to "5.14.6.7 Interaction channel function/Function to acquire line connection status"

#### **5.12.6.8 Interaction channel function-Operation of large volume call acceptance service**

Refer to "5.14.6.8 Interaction channel function-Mass calls reception service".

#### **5.12.6.9 Operation of operational control function**

For use from communication contents, refer to 5.14.6.9 "Operational control function".

(Note) If this function is called by a receiver that does not support the function, a return value indicating an error shall be returned and the browser operation shall not be stopped.

(1) Operation of "lockModuleOnMemory( )" and "unlockModuleOnMemory( )".

- Receiver units should always have 1MB of free space to acquire modules from the carousel by execution of "lockModuleOnMemory( )", and in case a memory area of 1MB cannot be secured, then the fixation process of the module should not be performed by this function.
- Even when the module version fixed to memory is updated, receiver units will not re-obtain the module automatically. Procedures to perform the detection of module update/lock release/re-fixation of modules to memory are written in the BML document.

- After “lockModuleOnMemory()” is executed, if transition to the document within the same ES document as the document that is currently presented is executed before the lock is completed, then lock performance is continued.
- Even if the module locked by “lockModuleOnMemory()” disappears from the carousel that is currently transmitted, the lock on the module continues.
- The module locked by “lockModuleOnMemory()” can be unlocked by “unlockModuleOnMemory()” regardless of carousel that is being transmitted is included in the module or not.

(2) Operation of “setCachePriority( )”

- Modules that can be specified with the argument of “setCachePriority()” are limited to modules transmitted by components included in the service that is currently being presented.
- For the priority of “setCachePriority()”, the value of  $n \times 10$  ( $n =$  integer value between 1-12) and negative values can be specified. When a value between 0-120 is specified, receiver units should perform look-ahead caching of modules in the order starting from the larger priority value. However, depending on the performance status of the receiver unit, the expected cache performance may not be executed in some cases. When negative values are set, receiver units should not perform look-ahead caching of the corresponding module.
- The priority value of modules where priority is not specified by “setCachePriority()” is interpreted as 30.
- The priority specified by “setCachePriority()” is valid until the data event that is currently presented is finished.
- The maximum quantity of modules that can be specified by “setCachePriority()” simultaneously is 64.
- In case priority specification is performed multiple times by “setCachePriority()” for the same module, then the latest setup of the priority is valid.

(3) Operation of “launchDocumentRestricted( )”

Refer to 5.14.6.9. “Operational control function”

(4) Operation of “quitDocument( )”

Refer to “5.14.6.9. Operational control function”

(5) Operation of “getFreeContentsMemory( )”

This function is assumed to be used in combinations with “lockModuleOnMemory” and “lockModuleOnMemoryEx”. Before executing “lockModuleOnMemory”, this function is used in

order to acquire free space of contents memory of receiver units to find out whether or not the module that is about to be locked can be stored.

However, in accordance with the operation of (1) Operation of “lockModuleOnMemory( )” in section 5.12.6.9, receiver units do not include 1MB area reserved for carousel acquisition when calculating “getFreeContentMemory( )”.

The number of resource modules read by “lockModuleOnMemory( )” to be executed is assumed to be written in the “number\_of\_resource” argument. The maximum value that can be set to “number\_of\_resource” is 768 in accordance with “5.13.3 Operation of name spaces”.

“number\_of\_resource” is used as hint information when calculating the remaining of contents memory of receiver units, and depending on the design of the receiver unit, corresponding arguments may be ignored. The maximum value of data size of modules is specified as 1MB in section 2.3.3, however, the return value does not follow this restriction, and it is in accordance with free space for the above mentioned contents memory.

In case a compressed module is transmitted, both the compressed module and the deployed module will be accumulated in contents memory. Therefore, when a compressed module is received, the value that is smaller than the contents memory total subtracted by carousel size is returned as free space of contents memory.

In addition, when transmitting compressed modules, specifications in section 2.3.3 should be followed and the total size of compressed modules and deployed modules should be in the range that does not exceed the maximum module size.

Free space of contents memory, which is a return value, may change depending on the processing status of the BML browser from status of performing “getFreeContentsMemory” to status of performing “lockModuleOnMemory”, “lockModuleOnMemoryEx”. The variation of contents memory at this time depends on a model, therefore, this should be used as a reference value to inspect whether or not the targeted module can be locked by the return value acquired by the corresponding API. Following the same logic, when this function is called before the lock of module is completed, there is no guarantee that the same value as when there is no lock request is returned. Contents should take into consideration that even when there is the required volume of free space in the return value, the lock of module may fail in some cases.

(6) Operation of “lockModuleOnMemoryEx( )”, “unlockModuleOnMemoryEx( )”

Using “lockModuleOnMemoryEx( )”, modules that are transmitted by different components than the document that is currently presented can be locked. This function is operated as follows. (Refer to section 5.14.7 for the meaning of linked status and unlinked status.)

- ES that can be locked by “lockModuleOnMemoryEx()” is limited to ES’s with a component\_tag value of 0x40, 0x50, 0x60. The ES that is currently presented can be locked by “lockModuleOnMemoryEx()” only when component\_tag value of the ES that is currently presented is 0x40, 0x50, or 0x60.
- Modules of 0x40, 0x50, 0x60 and the module of the ES that is currently presented may be instructed to be locked simultaneously in some cases, and receiver units need to be able to acquire modules of 4 ES in parallel. The performance to acquire 4 ES modules in order is not permitted. This is excluding cases where the total for the module size acquired simultaneously exceeds 1MB.
- Lock may be released by ”lockModuleOnMemoryEx( )” in the following cases.
  - When explicitly released by execution of “unlockModuleOnMemoryEx( )”, “unlockAllModulesOnMemory( )”.
  - When selecting stations for the service that is currently presented ends.
  - When the data event that is currently presented ends. (Including the disappearance of component that is currently presented.)
  - When the entry component disappears by PMT update.
  - When transition to unlinked status is occurs.
  - When transition to entry component occurs by executing “quitDocument()”.
- Lock by lockModuleOnMemoryEx( ) will continue in the following cases.
  - When the data event is updated in a component that is not currently presented.
  - When the locked module version is updated.
  - When the ES that was transmitting the locked module becomes an empty carousel.
  - When transition to a document of a different component within the same service as the document that is currently presented is executed.
  - When transition to entry component has occurred by a pull flag.
- When transition to linked status occurs.
- Receiver units should secure free space of 1MB to acquire modules from carousels at all times, and in case a memory area of 1MB cannot be secured by the execution of “lockModuleOnMemoryEx( )”, then this function shall not perform the fixation process of the module.
- Receiver units should secure free space specified by the argument of this function at all times and in case a memory area for the corresponding volume cannot be secured by the execution of “lockModuleOnMemoryEx( )”, then this function shall not perform the fixation process of the module. The volume specified in the argument does not include 1MB area indicated in the previous clause. When the argument is omitted, then it is interpreted as 0MB.

- When the fact that module specified by “lockModuleOnMemoryEx()” does not exist becomes clear from the PMT (i.e.: when the ES transmitting the corresponding module is not included in the PMT), then “lockModuleOnMemoryEx()” will end with return value of -3.
- When the module specified by lockModuleOnMemoryEx() does not actually exist, and in case that it could not be judged from the PMT, (i.e.: ES does exist, but the module does not exist, etc.), then “lockModuleOnMemoryEx()” will finish with a return value of 1 and when the target module is judged as non-existence later, a “ModuleLocked” event is generated with status = -2.
- Even when the module locked by “lockModuleOnMemoryEx( )” is updated, receiver units do not re-acquire the module automatically.
- After “lockModuleOnMemoryEx()” is executed, if transition is executed to the document within the same service as the document that is currently presented before lock is completed, then the lock performance will continue. In this case, the “ModuleLocked” event is generated as a basic rule when the corresponding lock performance is completed, however, depending on the timing of document transition, it may not be generated in some cases.
- After “lockModuleOnMemoryEx()” is executed, if data event of the lock target ES is updated before lock is completed, then the “ModuleLocked” event is generated with a status = -1, and locking is not executed.
- After “lockModuleOnMemoryEx()” is executed, if the lock target module is updated before lock is completed, the receiver unit will lock the updated module.
- Lock by “lockModuleOnMemory( )” and lock by “lockModuleOnMemoryEx( )” can be used in combination, however,
  - The module locked by “lockModuleOnMemory( )” cannot be released by “unlockModuleOnMemoryEx( )” or the module locked by “lockModuleOnMemoryEx( )” cannot be released by “unlockModuleOnMemory( )”. (Function will return error.)
  - The module locked by “lockModuleOnMemory( )” cannot be locked again by “lockModuleOnMemoryEx( )” or the module locked by “lockModuleOnMemoryEx( )” cannot be locked again by “lockModuleOnMemory( )”. (Function will return error.)
- As a method to update presentation at the time of updating the module that the object is referring to, a procedure using “lockModuleOnMemory()” is written in ARIB STD-B24 Vol. 2 Appendix 1 “6.6.2 Relationship between update and module lock”, however, the equivalent effect can be obtained by using “lockModuleOnMemoryEx()”.
- As a method to update presentation at the time of updating the binary table that the BML document is referring to, a procedure using “lockModuleOnMemory()” is written in ARIB



STD-B24 Vol. 2 Appendix 2 " 5.5.2.2 Behavior of binary table", however, the equivalent effect can be obtained by using "lockModuleOnMemoryEx()."

- As a method to share JPEG images, DRCS, binary tables between documents, a procedure using "lockModuleOnMemory()" is written in ARIB STD-B24 Vol. 2 Appendix 2 "4.8.5.1 Application of remain attribute for each type", "5.2.2 Sharing of external characters among documents", "5.5.2.3 Inter-document sharing" respectively, however, the equivalent effect can be obtained by using "lockModuleOnMemoryEx()." It is written that sharing is only valid in the respective document group, however, under this specification, sharing is valid within the same contents group. In case of sharing beyond the ES, the sharing target should be locked by using "lockModuleOnMemoryEx()". Performance in case it is not locked shall be receiver unit dependent.
- Under the status of receiving broadcasting, in case of executing a transition to a BML document of another ES, or in case of executing transition to broadcasting contents under the linked status, the existence of a module can be confirmed by the locking module including documents in the transition destination by using "lockModuleOnMemoryEx()" beforehand.
- In case the ES transmitting module locked by "lockModuleOnMemoryEx()" disappears, whether or not the lock of the module will be continued depends on a model. The lock should be released explicitly from contents for such modules. Transition to such modules is not possible.
- The module locked by "lockModuleOnMemoryEx()" can be unlocked by "unlockModuleOnMemoryEx()" regardless of whether or not the module is included in the carousel that is currently in transmission.
- As mentioned above, lock by "lockModuleOnMemory( )" and lock by "lockModuleOnMemoryEx( )" have different conditions for unlocking. A comparison of the two is shown in Table 5-11.

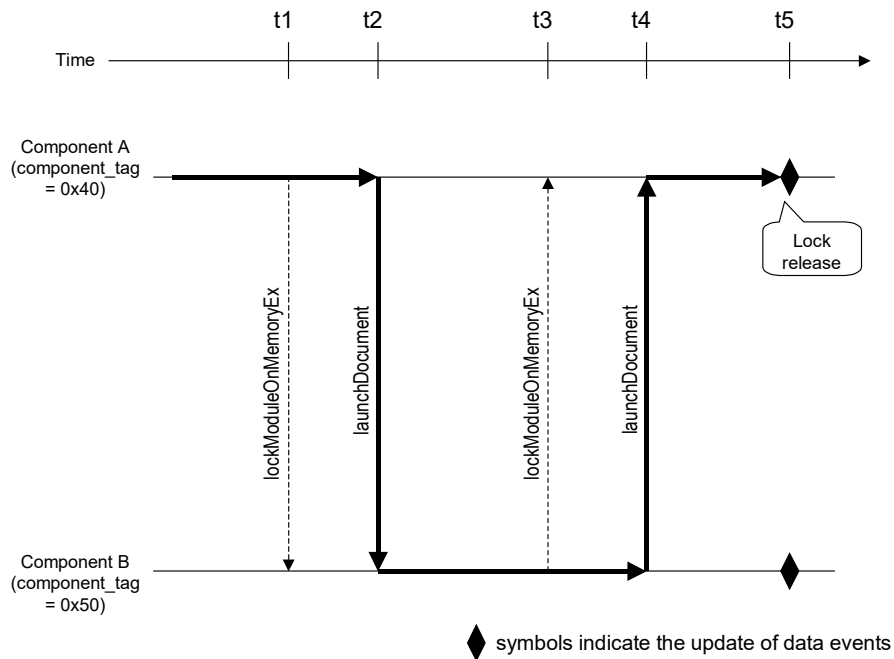
**Table 5-11 Comparison of lockModuleOnMemory and lockModuleOnMemoryEx**

	Module locked by "lockModuleOnMemory"	Module locked by "lockModuleOnMemoryEx"
Update of the data event in the component that is currently presented.	Lock release	Lock release
Update of the data event in the component that is not currently presented.	No changes	No changes
Update of the locked module version	No changes	No changes
Transition to another service than the document currently in presentation.	Lock release	Lock release
Transition to another component within the same service as the document currently in presentation	Lock release	No changes
Transition to entry component by pull flag	Lock release	No changes
Transition to entry component by "quitDocument()" (Including when "quitDocument()" is executed during the viewing of entry components)	Lock release	Lock release
Transition to linked status	Lock release	No changes
Transition to unlinked status	Lock release	Lock release

**Operation Example 1**

Typical operation example of "lockModuleOnMemoryEx( )" is shown in Figure 5-1. In this example, service is comprised of 2 components, which are component A and Component B. In the figure, the diamond shaped symbols rendered on the straight line which indicate the component are indicating the points where data events are updated. The following is explained in accordance with the figure.

- (a) Component A is presented in the beginning. The module of component B is locked to time t1 by "lockModuleOnMemoryEx( )."
- (b) Transition of the module of component B locked in (a) to time t2. Lock of the module continues.
- (c) Module of component A is locked by "lockModuleOnMemoryEx( )" in time t3.
- (d) Transition of module of the component A locked in (c) to time t4. The lock on the two locked modules continues.
- (e) Since the data event of component A that is currently in presentation will be updated in time t5, all the modules locked by "lockModuleOnMemoryEx( )" up to this point will be unlocked.

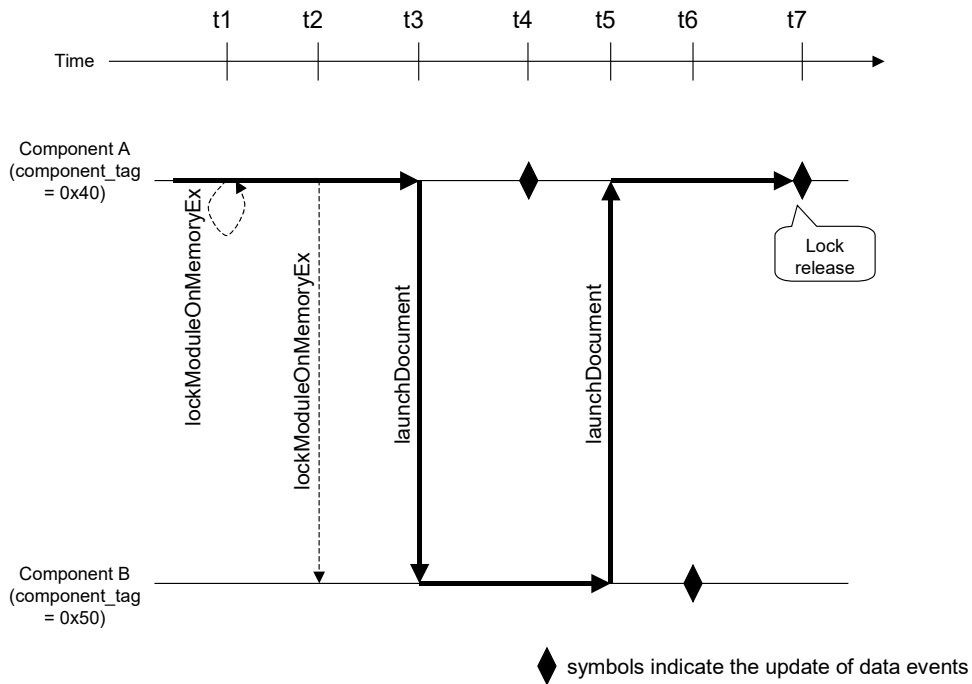


**Figure 5-1 Operation example 1 of lockModuleOnMemoryEx**

**Operation example 2**

A complicated operation example of “lockModuleOnMemoryEx( )” is indicated in Figure 5-2. The service is comprised of two components, which are component A and component B just the same way as Operation Example 1. The following is explained in accordance with the figure.

- (a) Component A is presented in the beginning. The module that’s currently in presentation of component B is locked to time t1 by “lockModuleOnMemoryEx( ).”
- (b) The module of component B is locked to time t2 by lockModuleOnMemoryEx( ).
- (c) Transition to module of component B locked in (b). The locks on the two locked modules continue.
- (d) Data event of component A is updated to time t4, but it does not have impact on modules that are already locked.
- (e) Transition to module of component A that was previously presented to time t5. In component A, new data event is already transmitted, but document presented by this transition is document that was already transmitted at the time of locking in (a).
- (f) Data event of component B is updated in time t6, however, it does not have an impact on modules that are already locked.
- (g) Since the data event of component A that is currently presented is updated in time t7, locks on all modules locked by “lockModuleOnMemoryEx( )” up to this point will be unlocked.



**Figure 5-2 Operation example 2 of lockModuleOnMemoryEx**

(7) Operation of “getIRDID( )”

When acquiring “cardID” as receiver unit unique identifier using “getIRDID()”, as specified in ARIB STD-B24, “CA\_system\_id” should be specified as type. The return value is a character string that is the CardID in hexadecimal (“0x” is not attached to the front) when 0x0005 is specified as type. This function shall be implemented by a receiver that implements the content protection system specified in Part 1, Vol. 5 of these guidelines.

0xF0001 shall be specified as the argument of type in order to allow for a receiver that supports the content protection system specified in Part 2, Vol. 5 to obtain the receiver ID “TVsetID” as hardware that is operated based on these guidelines (Section 2, Vol. 3).

0xF0001 shall be specified as type to obtain a TVsetID using getIRDID(). In this case, the return value is a character string that is the TVsetID in hexadecimal. (“0x” is not attached to the front.) For the operation of the TVsetID, see Appendix 19.

A function to acquire a MakerID and ModelID that are used for download shall not be supported.

An identifier that is unique to each receiver can be identified by specifying an argument provided in the table in (14) “Operation of “getBrowserSupport( )” in 5.12.6.9 “Operation of operational control function” using getBrowserSupport(). (Note 1) (Note 2) (Note 3)

See 7.2.3.2.3 (4) for how to obtain a receiver ID "CorrelatingID" as hardware that supports the IPTV download/VOD (integrated service type 2) function.

Note 1: A receiver that resends an IP as specified in the IPTV Forum standards may not be able to obtain a value even if the value provided in Vol. 7 of these guidelines is specified to the argument of type. See the IPTV Forum Standards (Appendix 16 in IPTVFJ STD-0005 Operational Specifications for IP Retransmission of Digital Terrestrial Television Broadcasting) for the operation of the getIRDID() in a receiver that receives a resent IP.

Note 2: Caution is necessary considering the fact that a receiver that supports the content protection system specified in Part 2, Vol. 5 may not be able to obtain a cardID even if 0x0005 is specified to the argument of type.

Note 3: Caution is necessary considering the fact that a receiver that can obtain a cardID may return 0 when specifying 0x0005 to the argument of getBrowserSupport() as shown in the table in (14) "Operation of "getBrowserSupport( )" in 5.12.6.9 "Operation of operational control function".

(8) Operation of "getLockedModuleInfo( )"

Array[n][0](module name), a return value of the "getLockedModuleInfo" function is a String type and it is always stored in the following format.

```
/<component_tag>/<moduleName>
```

Return values, Array[n][1] and Array[n][2] are number types.

In case modules that are fixed or requesting to be fixed do not exist at all, array with 0 length is returned, but receiver can also return null.

For contents production, the fact that one of the return values is returned should be taken into consideration.

(9) Operation of the closed caption display control function

Should be in accordance with ARIB STD-B24 Vol. 2 Appendix 2 "5.6.4. Operation of subtitle presentation control".

(10) Operation of "detectComponent( )"

Component specified as argument of "detectComponent( )" is limited to data components that are included in the service that is currently in presentation. When it is used in broadcasting contents, it should be in accordance with the specifications of section 5.13.3 and for URI specifying component, contracted forms starting with "/" shall be used. When it is used in communication contents, it should be in accordance with specifications in section 5.14.10.4, and for URI's specifying URI, the absolute URI starting with "arib-dc://-1.-1.-1" shall be used.

(11) Operation of "getResidentAppVersion( )"

- The return value Array[0] shall be as defined in ARIB STD-B24.

- The return values Array[1]-[3] shall be appropriate values for the receiver as defined in ARIB STD-B24.
- The receiver shall return the details of receiver applications defined for each type of receiver application for return value Array[4]. The details for each receiver application are defined as follows. An empty string shall be returned if a receiver application not described below has been specified to argument appName.
  - (i) Operation of the receiver when "ReservedTransmission" has been specified to argument appName from the receiver applications described in additionalinfo when functionname in Table 11-1 "Strings Available to Arguments of getBrowserSupport( )" in ARIB STD-B24 Vol. 2 Appendix 1 is "ResidentApp"
    - For the return value Array[4], the following value shall be returned in String format.
      - “active”: If the receiver has a reservation transmission function and it is enabled
      - “inactive”: If the receiver has a reservation transmission function but it is disabled
      - “none”: If the receiver does not have a reservation transmission function
  - (ii) Operation of the receiver when appName indicating the IPTV browser name as described in Appendix-14 is specified to argument appName

It is not possible to confirm using getBrowserSupport( ) whether the IPTV browser can support any IPTV functions specific to each service operator. However, it can be determined by the string of Array[4] obtained by getResidentAppVersion( ). The argument appName need to be specified appName indicating the IPTV browser name as described in Appendix-14.
- The return value Array[4] is String type and consists two kind of information. One is the item names describing the functions specific to each service operator. Another is the string equivalent to UserAgent given by the IPTV browser. The return value shall be described in the following format:
  - "Item name": "Item name": ...: "Item name" [: "UA=" "Character string equivalent to UserAgent"]
  - A separator ":" (a colon) shall be inserted between each item. ":" shall not be placed at the beginning of the first item.
  - The UserAgent string must always be allocated to the final field. (Allocation is arbitrary)
  - If UserAgent will be inserted, "UA=" shall always be added to the beginning of the character string.
  - Characters useable in item names and the character string equivalent to UserAgent shall be as defined in STD-B24 and TR-B14.

- However, ":" and "UA=" shall be reserved words for item names.
- Refer to Appendix-14 for the item names that can be described in Array[4].

The content of the character string corresponding to UserAgent is implementation dependent and is therefore not defined here. However, it shall be desirable that the IPTV service operator or the vendor implementing the browser disclose the format of the character string to the broadcaster by some means.

It shall be desirable for item names to be names like "Service name\_Function name" that allow the service names and functions provided by each service operator to be clearly identified.

Example: Dpa\_Type1Service:acTvila\_Full:.....:UA=Mozilla/4.0

(iii) Operation when "VOD" or "Download" is specified to the argument of "appName": See 7.2.3.2.3 (2).

(iv) Operation when "AITControlledApp" is specified to the argument of "appName": See 8.8.1 for the operation when "AITControlledApp" is specified to the argument of "appName".

#### (12) Operation of "isRootCertificateExisting ( )"

By confirming whether or not the relevant root certificate exists on the receiver in advance using this function, content can be developed in a way that viewers using a receiver on which the relevant root certificate is not stored will not be led to any content that require encrypted communication. root\_certificate\_id and root\_certificate\_version specified here shall be signed 32 bit values.

If 0 (general-purpose root certificate) has been specified as root\_Certificate\_Type, it shall return whether the specified root certificate exists in the general-purpose root certificate area.

In the event when 1 (operator-exclusive root certificate) has been specified as root\_Certificate\_Type, 1 shall be returned if the receiver has already acquired 1 or more operator-exclusive root certificates; and NaN shall be returned in all other cases.

#### (13) Operation of "startResidentApp( )"

- Combination of values that can be specified to arguments appName, showAV and Ex\_info are as follows. Incorporation of each application shall be implementation dependent.

appName	showAV	Ex_info
HTMLBrowser	No restrictions	URI string to be passed to the HTML browser
ReservedTransmission	1 only	None
MailClient	No restrictions	4th argument: Title of the e-mail sent (corresponds to the Subject: header of the e-mail) 5th argument: Body text of the e-mail sent 6th argument: Destination address of the e-mail sent (corresponds to the To: header of the e-mail) 7th argument: Destination addresses of the copy of the e-mail sent (corresponds to the CC: header of the e-mail)  These arguments shall be empty strings if not specified.
BookmarkList	1 only	None
webBrowserMode1	1 only	4th argument: URI string passed to the HTML browser 5th argument: Supplementary string that is passed to the HTML browser
webBrowserMode2	1 only	4th argument: URI string passed to the HTML browser 5th argument: Supplementary string that is passed to the HTML browser 6th argument: Specifies the broadcast screen size (2/8, 3/8, 4/8, 5/8, or 6/8). 7th argument: Specified the position on the broadcast screen (TR, TL, BR, or BL).

- After the activated receiver application has been terminated, a process equivalent to station reselection for the service that executed this function shall be performed. A function that is called with webBrowserMode1 or webBrowserMode2 shall conform to the following specifications.

● Operation when specifying webBrowserMode1 or webBrowserMode2 to appName

The operation of the argument of “Ex\_info” is as follows.

The maximum length of a URI string that is specified to the 4th argument of Ex\_info shall be 255 bytes.

The maximum length of a string that is specified to the 5th argument of Ex\_info shall be 255 bytes. The interpretation of the 5th argument by a receiver shall be a matter of product planning.

A receiver function to simultaneously display\* a reduced broadcasting screen and the HTML browser that is a resident function of the receiver is called when calling startResidentApp() after specifying webBrowserMode1 and webBrowserMode2 to the argument of “appName”.

Although the operation of a receiver after calling the above two screens with the webBrowserMode1 argument is a matter of product planning, it is desirable to follow the



specifications below. The operation of a receiver after calling the above two screens with webBrowserMode2 must follow the specifications below.

(i) The desirable operation of a receiver when this function is called when specifying webBrowserMode1 to the argument of “appName” is as follows.

- It is desirable to return to a channel selected before displaying two screens after two-screen simultaneous display is closed by the user’s operation, excluding cases in which a channel is selected by the user’s operation. It is also preferable that a document specified by returnUrl is displayed by the data broadcasting browser when returning to the selected channel.
- Whether or not to display data broadcasting on a reduced broadcasting screen is a matter of product planning. However, it is desirable to allow users to set whether or not to display data broadcasting if data broadcasting is displayed on a reduced broadcasting screen.

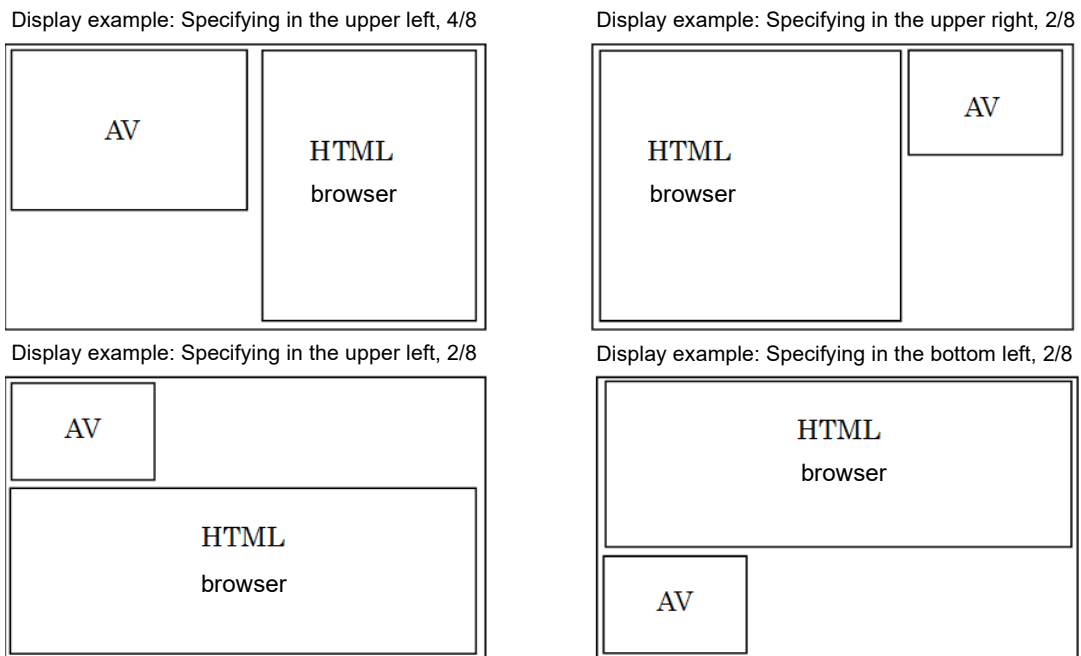
(ii) The operation of the receiver when this function is called by specifying webBrowserMode2 to the argument of “appName” is as follows.

- It is desirable that the size of a broadcasting screen and the display position when simultaneously displaying two screens conforms to the specifications of the 6th and 7th arguments. The display size specified with the 6th argument indicates a reduction rate when full-screen view is regarded as 1. The display position specified with the 7th argument is just a guide.
- Whether or not to display data broadcasting when simultaneously displaying two screens is a matter of product planning. However, users shall be allowed to set whether or not to display data broadcasting if data broadcasting is displayed on a reduced broadcasting screen.
- The “Home” button shall not be displayed by the HTML browser when simultaneously displaying two screens. The “Back” button and “Update” button can be displayed.
- The design of parts other than the broadcasting screen and HTML browser display shall be a matter of product planning (for example, displaying in gray or black).
- It is required to return to a channel selected before simultaneously displaying two screens after the two-screen display is closed by the user’s operation during two-screen display. It is desirable that a document specified by returnUrl is displayed in the data broadcasting when returning to the selected channel.
- The two-screen simultaneous display mode is cancelled and the normal channel selection process is performed when a channel is selected by the user’s operation during two-screen display. Alternatively, channel selection during two two-screen simultaneous display may be disabled.
- Broadcasting is displayed on a full screen if the HTML browser alone is closed during

two-screen simultaneous display.

- The broadcasting screen alone cannot be closed during two-screen simultaneous display.
- If a video content is replayed by the HTML browser during two-screen simultaneous display and if the receiver does not have sufficient processing capacity to replay the HTML browser video at the same time as the broadcasted video (when both the broadcasting screen and the HTML browser cannot be displayed at the same time), then the two screens shall be displayed with `webBrowserMode1`.

\* The figure below shows a conceptual diagram of a two-screen simultaneous display (example of displaying two screens with `webBrowserMode2`).



An optional web browser can be started if the application type is a TV viewing application, etc. for PCs. In such case, the web browser shall be started using the TV viewing application, or the TV viewing application and web browser are started so that a set of two screens is displayed.

If the HTML browser is opened with a separate window using this function, it is desirable to allow viewers to distinctly notice that a separate window is opened by pressing a button on the data broadcasting browser. However, when the HTML browser is opened with a separate window, the viewer can close the broadcasting screen alone; therefore, two screens shall be displayed with `webBrowserMode1` that starts the HTML browser function residing in the receiver.

(14) Operation of “getBrowserSupport( )”

Values that can be specified as argument to getBrowserSupport() shall be added in the following combinations upon adding the values shown in Table 11-1 in ARIB STD-B24 Vol. 2 Appendix 1.

functionname	additionalinfo	Operation of getBrowserSupport()
ResidentApp	“JapaneseInput”	Returns 1 if the receiver implements a character input function (Refer to section 1.6) as one of its applications.
	A character string representing the browser as described in Table in Appendix-14 (2).	Returns 1 if the receiver implements the relevant browser.
	“VOD”	See 7.2.3.2.3 (1).
	“Download”	
	webBrowserMode1 webBrowserMode2	Returns 1 if the receiver supports the application.
Transmission Protocol	“datalink”, “PPP.modem”	Returns 1 if the browser implements a interactive communication function using PPP (i.e. PPP connection using a modem, not PPPoE).
ExtraBrowserFunction	A character string representing the browser function as described in Table in Appendix-14 (2).	Returns 1 if the receiver implements the relevant function.
Unsupported	Misc.Unlink Com.BASIC.Basic Com.BASIC.Vote	Returns 1 if the receiver does not support the function. 0 is returned if a receiver supports the function. 0 is also returned if an undefined string is used.
CASystem	“0x0005” (Note)	Returns 1 if the receiver can obtain a unique identifier by specifying the CA_system_id “0x0005” as the type using the getIRDID() function.
	“0x000D”	Returns 1 if the receiver can obtain a unique identifier by specifying the CA_system_id “0x000D” as the type using the getIRDID() function.
IPTVFunction	“VOD”, “HTTP” “VOD”, “RTSP”	See 7.2.3.2.3 (1).
IRDID	“0xF0001”	Returns 1 if the receiver can obtain a unique identifier by specifying “0xF0001” indicating a TVsetID as the type using the getIRDID() function.
SmartDeviceProfile	SmartDeviceMode1	Returns 1 if the receiver supports the SmartDeviceMode1 operation.
	SmartDeviceMode2	Returns 1 if the receiver supports the SmartDeviceMode2 operation.

(Note) The operation of this argument is a matter of product planning, if the receiver supports only the content protection system that is provided in Part 1, Vol. 5 of these guidelines as a digital terrestrial television broadcasting content protection used in receivers for digital terrestrial television broadcasting.

If the functionname is set to "Unsupported", additionalinfo is operated as follows.

functionname	additionalinfo	Operation of getBrowserSupport()
Unsupported	Misc.Unlink (Note 1)	Returns 1 if the receiver does not have a function to present unlinked contents.
	Com.BASIC.Basic (Note 2) Com.BASIC.Vote (Note 2)	Means the same as the specification of an extended function group in Table 11-2 in Appendix 1 of STD-B24. Returns 1 if the receiver does not support the extended function group function.

(Note 1) If the receiver does not have a function to present unlinked contents, the receiver shall support additionalinfo.

(Note 2) When this additionalinfo value is used for contents, caution is necessary considering the fact that some receivers conform to only the old version of these guidelines and thus do not support the BASIC procedure for use when 0 is returned, whereas the returned value 1 indicates that the receiver does not support the BASIC procedure.

Moreover, the following value shall be added to the extended function group specification (Refer to Table 11-2 in Appendix 1) to be specified as additionalinfo when functionname is "APIGroup".

API	Extended function group specification (additionalinfo argument)
getBookmarkInfo2()	Bookmark.Basic2
getBrowserStatus()	Ctrl.Status
getContentSource( )	Storage.Source

(15) Operation of "startExtraBrowser( )"

Refer to Appendix 14 for values that can be specified to the browserName argument of startExtraBrowser(). Refer to section 6.3 for the activation of the IPTV browser using this function.

(16) Operation of getBrowserStatus( )

Refer to 7.2.3.2.3 (3) for the operation of getBrowserStatus( ) for receivers that support IPTV download/VOD (integrated service type 2).

(17) Operation of transmitDataToSmartDevice( )

Refer to 6.5.1 for the operation of transmitDataToSmartDevice( ) for receivers that support an external device linkage function.

**5.12.6.10 Operation of bookmark function**

Should be in accordance with ARIB STD-B24 Vol. 2 Appendix 1 "8.4 Guidelines on bookmarks" except for the additional following specifications.

(1) Operation of "startResidentBookmarkList()"

In case this function is called by receiver units where "startResidentBookmarkList()" is implemented and the bookmark list display function by receiver's native application is not implemented, a NaN (failure) will be returned.

**5.12.6.11 Operation of IPTV linkage function**

Refer to 7.2.3.2.4 for the operation of the functions related to the IPTV linkage function.

**5.12.6.12 Operation of inquiry function**

(1) Operation of getContentSource()\*

Only "onair" is used for the argument of source. Broadcasting waves are received when the "onair" is specified as the argument. "True" is returned when a streaming content is directly replayed without using a storage device. "False" is returned if a streaming content is not directly replayed. "False" is always returned when any value other than "onair" is specified as the argument. "True" is returned when a streaming content on broadcasting waves is directly replayed using DLNA.

\* Receivers released before May 31, 2013 can use getContentSource() without supporting the function. Receivers without the recording and playback function do not need to support this function.

**5.12.6.13 Operation of the AIT-controlled application linkage function**

See 8.9.1.1 and 8.9.1.2 for the operation of the functions related to the AIT-controlled application linkage function.

**5.12.7 Embedded objects**

- Time handled by data objects of ECMA Script is JST(UTC+9 hours) without adding the time offset of summer time. In case local time with considerations to summer time offset is needed to display the current time, etc., the local time is obtained by adding time offset by using "addDate()" for time obtained by "Date()" function in contents.
- Time handled in the date object during the presentation of communication contents should be in accordance with the time information that broadcasting contents were referring to immediately before the transition to communication contents.

**5.12.8 Other specifications**

Refer to ARIB STD-B24 Vol. 2 Appendix 3 "5.8. Other restrictions"

## 5.13 Transmission of contents and name spaces

### 5.13.1 Mapping of scope to transmission systems

Should be in compliance with ARIB STD-B24 Vol. 2 Appendix 3 "6.1. Scope mapping to transmission system". Regarding the communication contents, refer to section 5.14.7.

### 5.13.2 Restrictions related to reference of mono-media, etc. over media

Should be in compliance with ARIB STD-B24 Vol. 2 Appendix 3 "6.2.2 Restriction on reference of mono-media and others across media types". Regarding the communication contents, refer to section 5.14.9.

### 5.13.3 Operation of name spaces

Should be in compliance with ARIB STD-B24 Vol. 2 Appendix 3 "6.3 Namespace". Regarding the communication contents, refer to section 5.14.10.

Also, the following specifications are added for broadcasting contents.

- Reference to another service is only possible in the following extended functions for broadcasting.  
epgTune, epgIsReserved, epgReserve, epgCancelReservation,  
epgReclsReserved, epgRecReserve, epgRecCancelReservation  
epgTuneToComponent
- Except for the following cases, contracted forms (Refer to ARIB STD-B24 Vol. 2 9.2) are used for the description of name spaces in case broadcasting contents are specified in BML documents.
  - Reference to another service.
  - Argument of the extended function for broadcasting with the event specification as the argument.
  - Reference of broadcasting contents from communication contents.
- Except for arguments to the extended function for broadcasting with the event specification as the argument, "event\_id" is always omitted.
- Resource reference of mono-media, etc. referred from BML documents are operated as follows.
  - Resource transmitted by the same component as the document that is currently presented can be referred to at all times.
  - Regarding resources transmitted by different component than the document that is currently presented, it can be referred under the following regulations.
    - ◇ Resources referred to from BML documents should be locked by "lockModuleOnMemoryEx( )" before the presentation of the BML document. The performance in case of attempting to present BML documents that refer to unlocked resources depends on a model.
    - ◇ Performance in case of explicitly releasing resources that are referred from BML document that is currently presented depends on a model.

- ◇ The resource that is required immediately after the start of BML presentation especially, such as CLUT and JPEG are referred to as the background-image of the body, since the lock by “lockModuleOnMemoryEx( )” should be completed without failure before the presentation of the BML, some attention is required when creating the contents.
- Regarding the reference of ECMAScript and CSS transmitted as independent resources, the following specifications are stipulated in particular.
  - ◇ In case of referring to ECMAScript and CSS from a different BML document component than the component by which they are transmitted, then the ECMAScript and CSS should be locked by “lockModuleOnMemoryEx( )” and transition to the corresponding BML document after the lock is completed should be performed. The performance in case document transition is performed without completing the lock depends on a model.
  - ◇ In case of referring to the ECMAScript and CSS from a different BML document component than the component by which they are transmitted, the locked CSS and ECMAScript cannot be referred to along with the disappearance of the ES, etc. in some cases. The above mentioned discordance can be prevented by transmitting CSS data and ECMAScript data by entry component by the specifications of section 2.1.10.3.
  - ◇ Even in case of referring to the CSS, ECMAScript transmitted by the component that is currently being viewed from the BML document, the ECMAScript and CSS should be locked before the transition of the corresponding BML document and transition should be performed after the completion of the lock.
- For the “href” attribute of “launchDocument( )”, “launchDocumentRestricted( )”, and “a” element, BML documents transmitted by components included in the same service as the document that is currently presented can be specified.
- The maximum number of resources (including both broadcasting contents and communication contents) that can be held in contents memory in receiver units by contents simultaneously is 768. As a method to realize this restriction, the total number of resources (resources with unique name spaces) within one data event period should be less than 768. However, in case the above mentioned restriction can be realized for certain on receiver units at the time of contents production, the total number of resources within a data event period can be more than 768, and this will not cause interference.

However, in case the fixation of resources exceeding the above mentioned number is specified by “lockModuleOnMemory( )”, etc. against the above mentioned restriction, then receiver units may not perform this in some cases.

The resources this section is referring to are the two resources below.

- Resources that are directly mapped by modules.
- Resources stored in modules in entity format of HTTP/1.1.
- The maximum number of modules that can be held in the contents memory of receiver units (including modules that are cached by specifications of “setCachePriority( )”) by contents simultaneously is 256.
- Modules where monitoring of module version update is possible is limited to the ES that is currently presented or ES's with a component\_tag value of 0x40,0x50,0x60.
- As indicated in ARIB STD-B24 Vol. 2 “9.2.5.1 Identification of currently selected broadcasting service on receiver”, in case "arib://-1.-1.-1" is specified as the service name, it is regarded as the service that is currently selected is specified. If during broadcasting reception, the service that is currently selected here refers to the broadcasting service that is currently being received, if during the playback of the partial TS, then it refers to the current service of the partial TS.
- In the BML document that performs transition by setting “remain” in the “remain” attribute of object element, in case the document in the originating point of transition and the document in transition destination are included in different modules, then the specification of module number in the “data” attribute of the object element, which is the remain target, shall not be omitted; it should be specified. Also, in case the document in the transition originating point and the document in the transition destination are included in different components, then the component tag value in the “data” attribute of the object element, which is the remain target shall not be omitted; it should be specified.

#### **5.13.4 Operation relating to communication contents**

Refer to section 5.14.11, section 5.14.12, and section 5.14.13.

### **5.14 Operation of communication contents**

#### **5.14.1 Operation guidelines relating to the presentation of communication contents**

It is the same as section “5.12.1 Operation guidelines relating to presentation”.

#### **5.14.2 Operation guidelines relating to the performance of external characters of communication contents**

It is the same as section 5.12.2 “Operation guidelines relating to the performance of external characters”.

#### **5.14.3 Performance of DOM of communication contents**

It is the same as section 5.12.3 “Performance of the DOM”. However, there are operational restrictions in unlinked status. For details, refer to section 5.14.7.



#### 5.14.4 Operation of ECMA script execution for the script language of communication contents

It is the same as section 5.12.4 "Operation of script language".

#### 5.14.5 Expansion object for the broadcasting of communication contents

It is the same as section 5.12.5 "Expansion object for broadcasting".

#### 5.14.6 Operation of browser pseudo-objects of communication contents

The extended function for broadcasting that is equipped on browser pseudo-objects of communication contents may perform differently depending on the status (data broadcasting reception status, linked status or unlinked status) that the receiver unit is in. Regarding the basic idea of data broadcasting reception status, linked status or unlinked status, refer to section 5.14.7. In case the resource for broadcasting is referred to as an argument in unlinked status, it should be noted that performance of the extended function for broadcasting will fail. Tables 5-12 to 29 shows the operation of substantial functions. The "O/X/Options" in the table has the following meanings, O- "performance will be successful", X- "performance will fail", Options – "performance will be successful if optional functions are implemented." For parts in browsers specified as "performance will be successful", "performance will fail", the receiver unit shall execute the performance specified in 5.14.12.2. The unlinked status can be optionally supported by the receiver. The operation when the unlinked status is supported by a receiver conforms to the specifications provided in Tables 5-12 to 29.

##### 5.14.6.1 Ureg related functions

Operation of Ureg related functions in communication contents is indicated in Table 5-12. For the performance in data broadcasting reception status, refer to section 5.12.6.

**Table 5-12 Performance of Ureg related functions in communication contents**

	Linked status	Unlinked status
Ureg[ ]	O	X
Greg[ ]	O	O

##### 5.14.6.2 EPG functions

Operation in communication contents of EPG functions is indicated in Table 5-13. For the performance of data broadcasting reception status, refer to section 5.12.6.

**Table 5-13 Performance of EPG functions of communication contents**

	Linked status	Unlinked status
epgGetEventStartTime( )	O	X
epgGetEventDuration( )	O	X
epgTune( )	O	O
epgTuneToComponent( )	O	O
epgIsReserved( )	O	O
epgReserve( )	O	O
epgCancelReservation( )	O	O
epgReclsResreved( )	O	O
epgRecReserve( )	O	O
epgRecCancelReservation( )	O	O

#### 5.14.6.3 Series reservation function

Operation of the series reservation function in communication contents is indicated in Table 5-14. For performance in data broadcasting reception status, refer to section 5.12.6. This function is optional. In case of using it, existence of the function should be confirmed by “getBrowserSupport( )” before use.

**Table 5-14 Performance of the series reservation function of communication contents**

	Linked status	Unlinked status
seriesIsReserved( )	Optional	Optional
seriesReserve( )	Optional	Optional
seriesCancelReservation( )	Optional	Optional
seriesReclsReserved( )	Optional	Optional
seriesRecReserve( )	Optional	Optional
seriesRecCancelReservation( )	Optional	Optional

The performance in case it is operated as an option is the same as data broadcasting reception status.

#### 5.14.6.4 NVRAM functions

Operation of the NVRAM functions in communication contents is indicated in Table 5.15. For performance in data broadcasting reception status, refer to section 5.12.6.

**Table 5-15 Performance of the NVRAM functions in communication contents**

	Linked status	Unlinked status
readPersistantArray( )	O	X
wirtePersistantArray( )	O	X

The only NVRAM area that can be used in linked status is the A-profile memory area of communication purpose for the specified broadcaster. Regarding name spaces, refer to section 5.2. In case other name spaces are specified, the performance of receiver units will fail.

#### 5.14.6.5 Interaction channel function-Basic Procedure

Operation of the interaction channel function of basic procedure is shown in Table 5-16. Regarding this function, it is only applied to receiver units with equipped modems. In the case of using it, the existence of the function should be confirmed based on the method specified in Appendix 16. For performance in data broadcasting reception status, refer to section 5.12.6.

**Table 5-16 Performance of the interaction channel function of basic procedure of communication contents**

	Data broadcasting reception status	Linked status	Unlinked status
connect()	Optional (Note)	X	X
disconnect()	Optional (Note)	X	X
sendTextData()	Optional (Note)	X	X
receiveTextData()	Optional (Note)	X	X

(Note) When this function is called by a receiver that does not support it, a return value indicating an error shall be returned and the browser operation shall not be stopped.

#### 5.14.6.6 Interaction channel function-TCP/IP

Operation of the interaction channel function of TCP/IP is indicated in Table 5-17. The details of operation of each function are indicated after Table 5-17.

**Table 5-17 Performance of TCP/IP Interaction channel function of communication contents**

	Data broadcasting reception status	Linked status	Unlinked status
getConnectionType( )	O	O	O
isIPConnected( )	O	O	O
connectPPP( )	O (Note)	X	X
connectPPPWithISPPParams ( )	O (Note)	O (Note)	X
disconnectPPP( )	O (Note)	O (Note)	X
setISPPParams( )	O (Note)	X	X
getISPPParams( )	O (Note)	X	X
sendTextMail( )	Optional	Optional	X
sendMIMEMail( )	Optional	Optional	X
transmitTextDataOverIP( )	O	O	O
setCacheResourceOverIP( )	Optional	Optional	Optional
confirmIPNetwork( )	O	O	O

(Note) This function is not required to be supported by receivers without a modem and receivers that do not have a modem but have a high-speed communication function. When this function is called by a receiver that does not support it, a return value indicating an error shall be returned and the browser operation shall not be stopped.

- Function performance of “getConectionType( )”

This can be performed in all situations.

This function is used to obtain only the hint information in order to estimate the communication speed based on the information of the type of the connection lines.

In ARIB STD-B24 Appendix 1, Informative Explanation 4, it is used as confirmation of different types of priority usage lines, however, the specifications of “getConnectionType( )” do not cover all types of connection lines, therefore, there are cases that do not fall into the sequences in Explanation 4, so keep in mind that this is used as nothing more than hint information.

For details relating to the return value of the function, refer to ARIB STD-B24 Appendix 3 5.6.5.2. In environments where receiver units connect to a TCP/IP network using connection types that are not listed in Appendix 3, this function will return NaN.

- Function performance of “isIPConnected( ) “

This can be performed in all situations.

In this function, receiver units return values after judging whether or not the IP address is acquired or not, therefore, in case, for example, the receiver unit is connected to a dial-up router, then whether or not the router is connected to something else from there does not matter, the judgement is done based on whether or not an IP address is allocated (from DHCP, etc.) for the receiver unit.

For details of operation of return value, refer to Appendix-12.

- Function performance of “connectPPP( )”

It functions only in data broadcasting reception status. A return value indicating an error shall be returned to receivers that do not support this function.

Regarding phone number specification, name server specification specified in “connectPPP( )” should be in accordance with the specifications in ARIB STD-B24 Vol. 2 Appendix 3 5.6.5.4, 5.6.5.5.

Empty strings cannot be specified in the character string specification of phone number of “connectPPP( )”. Significant phone number character strings should be specified.

This function is assuming to provide interaction channel communication to viewers without an environment to connect to ISP, for example, and is to be used as a method to realize free communication services related to broadcasting.

For the usage of this function, advance consent from the viewer in contents is assumed. For the startup of this function, the connection can be secured for certain by executing this function after disconnecting the communication by using “disconnectPPP” and the disconnect function in advance. However, consent from viewers should be obtained for disconnecting communication.

Contents should take the fact that there is already a connection set up in the receiver unit into consideration and confirm whether or not “getISPParams( )” is set up, and when ISP setup is done already in the receiver unit, “connectPPPWithISPParams( )” should be used.

In case connectPPP( ) is specified, even when ISP setup is done, the receiver units will connect parameters specified by “connectPPP( )” in higher priority.

In “connectPPP( )”, timeout can be set using the argument “idleTime” and this value is set in units of 100ms, and the range of value should be more than or equal to 60,000(one minute) and less than or equal to 1,200,000 (20 minutes). When there is no communication (no packets are sent from/to receiver units) during the setup timeout, the receiver units will generate a timeout and disconnect the communication. In such an event, “IPConnectionTerminated” event will be generated.

In case nameServer1 and nameServer2 of connectPPP() are both omitted, then the receiver units will use the name server specified by the dial-up server.

In receiver units with a constant connection environment (including PPPoE) such as Ethernet, and if this function is called by the “connectPPP()” function in the status where the connection is already established, the connecting destination should be switched based on the connection setup of the “connectPPP()” function.

In case a PPP connection is already established by automatic connection by the “connectPPPWithISPParams()” function or PPP, an error will be returned. In case a connection to the “connectPPP()” function is required, the connection should be disconnected once by the “disconnectPPP()” function and should be connected again by the “connectPPP()” function. In case the receiver unit does not have PPP functions, then an error will be returned based on the specifications in ARIB STD-B24 Appendix 3 5.6.5.2.

Connections established by “connectPPP( )” are disconnected by “disconnectPPP( )” or performance of selecting station or occurrence of a timeout.

For details on the operation of return values, refer to Appendix-12.

- Function performance of “connectPPPWithISPParams( )”

It is performed in data broadcasting reception status or linked status. A return value indicating an error shall be returned to receivers that do not support this function.

“connectPPPWithISPParams( )” is used to perform PPP connection using receiver unit setup in case the receiver unit is not connected by PPP. This function is used to connect from contents in case the return value of “isIPConnected( )” is 0. In case a PPP connection is established already based on each priority usage line types, then it will always fail based on the specifications in ARIB STD-B24, 7.6.7.6 and ARIB STD-B24 Appendix 3, 5.6.5.2. Also, when

receiver units are not setup for each priority usage line type or do not have PPP functions, then there will be an error based on the same specifications.

The connections using “connectPPPWithISPPParams( )” will be disconnected at the time of calling “disconnectPPP( )” and performance of selecting stations in normal performance, or in no communication status (status where not a single packet is sent or received) during the timeout time using time setup in receiver units.

At the time of a timeout occurrence, an “IPConnectionTerminated” event is generated.

For operation of return values, refer to Appendix-12.

- Function performance of “disconnectPPP( )”

It is performed in data broadcasting reception status and linked status. A return value indicating an error shall be returned to receivers that do not support this function.

For performance of disconnectPPP(), refer to the specifications in ARIB STD-B24 7.6.7.6, and 5.14.13. For details on the operation of return values, refer to Appendix-12.

- Function performance of setISPPParams( )

It is performed only in data broadcasting reception status. A return value indicating an error shall be returned to receivers that do not support this function.

The following regulations are established for the function argument.

- First argument ispname is a maximum character string length of 64 digits (128 bytes).
- Empty string should be set up in case of omitting the 6<sup>th</sup> argument nameServer1.
- Empty string should be set up in case of omitting the 7<sup>th</sup> argument nameServer2.
- 11<sup>th</sup> argument status value is a broadcaster dependent operation using this function from an unconfigured status[1], and operational regulations are not established.
- Ethernet(401, 402, 403) should not be specified in the 12<sup>th</sup> argument lineType.

Regarding the details of function operations, refer to ARIB STD B24 Vol. 2 Appendix 3 section 5.6.6, and Vol. 6 section 7.3.1 “Transmission conditions of telephone numbers” and section 7.3.2 “Application functions”.

- Function performance of “getISPPParams( )” A return value indicating an error shall be returned to receivers that do not support this function.

It is performed only in data broadcasting reception status.

The following regulations are established for this function.

- From the perspective of protecting personal information, contents should not send acquired information elements to the center.
- In case the return value array[4] is omitted, then empty strings should be returned.

- In case the return value array[5] is omitted, then empty strings should be returned.
- In case the receiver unit sets up the parameter, the return value array[9] should return status=2.
- The definitions of broadcaster identification information acquired in the return value array[10] are indicated in Table 5-18.

**Table 5-18 The definitions of getISPParams( ) return value array[10]**

Value(hexadecimal character string display)			Definitions
00	XXXX	XX	No setup status, or the setup at the time of deletion by receiver features.
8F	XXXX	XX	The setup at the time of setup by receiver features.
Other than the above			When it is set by a broadcaster that has original_network_id, broadcaster_id (note2) in the left.
FF	original_network_id	broadcaster_id	

Note 1 : "X" means "don't care".

Note 2 : In case of digital terrestrial television broadcasting, it is fixed to FF.

- Function performance of "transmitTextDataOverIP( )"

It is performed in all situations.

This function is a function assuming that the sending and receiving of superimpose are performed in TCP/IP networks by BASIC procedure. Mainly it is used from broadcasting contents. The following restrictions are established for the argument of this function.

- There are two types of schemes that can be setup in the URI, which are "http" and "https".
- The maximum length of the character string that can be specified to argument text is 4096Bytes.
- The character code of sending text data is a fixed operation of EUC-JP only. Therefore, "EUC-JP" should be specified for the charset of the 3<sup>rd</sup> argument at all times. However, it should be noted that character codes such as external characters, etc. operated in the DRCS are not included in this sending text data.

When text is sent to servers, receiver units will send it using the POST method to the URI specified by the argument. When there is no text to be sent, then empty strings ("") should be specified.

Receiver units should specify "application/x-www-form-urlencoded" of Content-Type in request message, "Denbun" should be specified as the "name" attribute. The encoding format of the text is in accordance with HTML4.01 "17.13.4 Form content types". In such an event, Japanese character strings should be encoded as the character codes of EUC-JP. In case empty strings ("") are specified, then character strings of "Denbun=" should be considered and specify 7 to the Content-Length.

The request message when sending texts “send denbun” to servers is indicated in Figure 5-3.

```
POST http://localhost/test.cgi HTTP/1.1
...header omitted...
Content-Type: application/x-www-form-urlencoded
Content-Length: 49

Denbun=%73%65%6e%64%20%64%65%6e%62%75%6e
```

**Figure 5-3 An example of a request message issued by “transmitTextDataOverIP( )”**

If there is text data to be passed to the receiver unit, the server, which requested the process, can attach text data on top of the response status. In such case, Content-Type is set as text/plain and EUC-JP should be specified in the Charset.

An example of a response message including text data "Reply texts" in the entity-body is indicated in Figure 5-4.

```
HTTP/1.1 200 OK
...header omitted...
Content-Length: 22
Content-Type: text/plain; Charset=EUC-JP

Reply texts
```

**Figure 5-4 An example of response message received by “transmitTextDataOverIP( )”**

The maximum size of the entity-body received by receiver units is 4096Bytes at this point. The receiving process of text in case text exceeding the maximum size is received by the receiver unit, it depends on a model. Character codes of text data is a fixed operation only in EUC-JP.

In case of unlinked status, a user interface to obtain permission from the viewer is presented by the receiver unit and consent from the viewer should be obtained before sending the text.

Failure performance of the user interface or in case permission could not be obtained from viewers is receiver unit dependent.

- Function performance of “setCacheResourceOverIP()”

“setCacheResourceOverIP” function sets the hint information of cacheable resources on the Internet stored in the Array of argument to receiver units. Cache performance of setup resources and the release performance depend on models. This function is performed in data broadcasting reception status, linked status and unlinked status. Even when cache performance of communication contents is specified by this function from contents in data broadcasting reception status or linked status, situations where the cache of communication contents is prevailed over the cache of broadcasting contents is not assumed by the service. Therefore, it is strongly recommended for receiver units to leave the cache of broadcasting contents by priority.



For receiver units performing communication by PPP connection, if this function is called in the status where a connection is not established, whether or not a PPP connection by automatic connection is used to get connected depends on a model. For using this function, contents should secure a PPP connection with “connectPPP” or “connectPPPWithISPPParams” in advance.

Hint information is valid within the document. Accumulation of and usage of accumulation of hint information by history of document transition is not assumed as a service.

● Function performance of confirmIPNetwork()

This function can be operated in all statuses.\*

An IP address that is obtained through a name resolution shall conform to the protocol stack provided in Chapter 6 in Vol. 6 of these guidelines.

\* Receivers released before May 31, 2013 can use this function without supporting it.

**5.14.6.7 Interaction channel function/Function to acquire line connection status**

The function performance of functions to acquire line connection status is indicated in Table 5-19. Regarding this function, it is only applied to receiver units with equipped modems. In case of using it, the existence of the function should be confirmed before use by “getBrowserSupport( )”.

**Table 5-19 Performance of functions to acquire line connection status in communication contents**

	Data broadcasting reception status	Linked status	Unlinked status
getPrefixNumber( )	O	X	X

The following regulations are established for the return value of getPrefixNumber( )

- Array[2] to Array[4] are operated for the return value.
- From the perspective of protecting personal information, contents should not send the return value as information to center.

For the details of function operation, refer to Vol. 6, section 7.3.1 “Transmission conditions of telephone numbers” and section 7.3.2 “Application functions”.

**5.14.6.8 Interaction channel function-Mass calls reception service**

The function performance of the mass calls reception service is indicated in Table 5-20. Regarding this function, it is only applied to receiver units with equipped modems. This function can be optionally implemented by a receiver. In case of using it, the existence of the function should be confirmed before use by “getBrowserSupport( )”.

**Table 5-20 Performance of mass calls reception service**

	Data broadcasting reception status	Linked status	Unlinked status
vote( )	Optional (Note)	X	X

(Note) When this function is called by a receiver that does not support the function, a return value indicating an error shall be returned and the browser operation shall not be stopped.

#### 5.14.6.9 Operational control function

The performance function for communication contents of operational control functions is indicated in Table 5-21. The details of performance of each function are described in Table 5-21 and then on. Regarding the function performance of data broadcasting reception status, refer to section 5.12.6.

**Table 5-21 Performance of the operational control function of communication contents**

	Linked status	Unlinked status
reloadActiveDocument( )	O	O
getNPT( )	O	X
getProgramRelativeTime( )	O	X
isBeingBroadcast( )	O	X
lockModuleOnMemory( )	X	X
unlockModuleOnMemory( )	X	X
setCachePriority( )	X	X
getIRDID( )	Optional	X
getBrowserVersion( )	O	O
getProgramID( )	O	X
getActiveDocument( )	O	O
lockScreen( )	O	O
unlockScreen( )	O	O
getBrowserSupport( )	O	O
launchDocument( )	O	O
launchDocumentRestricted( )	Optional	Optional
quitDocument( )	O	O
launchExApp( )	Optional	Optional
getFreeContentsMemory( )	O	X
isSupportedMedia( )	O	X
detectComponent( )	O	X
lockModuleOnMemoryEx( )	O	X
unlockModuleOnMemoryEx( )	O	X
unlockAllModulesOnMemory( )	O	X
getLockedModuleInfo( )	O	X
getBrowserStatus( )	O	O
isRootCertificateExisting( )	O	O
getRootCertificateInfo( )	O	O
getResidentAppVersion( )	O	O
startResidentApp( )	Optional	Optional
startExtraBrowser( )	Optional	X
transmitDataToSmartDevice( )	Optional	Optional

- Function performance of reloadActiveDocument( )

It is performed in both linked status and unlinked status. When this function is called, the receiver unit should re-acquire the BML document and mono-media that are currently presented from the server. In case the cache within the receiver unit can be judged as the newest, then the display using the cache can be executed. In case of playing communication contents whose resource name cannot be specified such as when "/" is specified at the end of the URI or ?query is specified, the receiver unit will perform a re-acquisition of the communication contents that are currently presented by re-sending the URI specifying the corresponding BML document to the server.

For example, the performance in case the communication contents that had gone through transition in launchDocument("http://localhost/hoge/", "cut") call "reloadActiveDocument" is the same as when the above mentioned "launchDocument" function is called.

- Function performance of getNPT( )

Performance in linked status is the same as data broadcasting reception status.

- Function performance of getProgramRelativeTime( )

Performance in linked status is the same as data broadcasting reception status.

- Function performance of isBeingBroadcast( )

Performance in linked status is the same as data broadcasting reception status.

- Function performance of getIRDID( ) function

Only for receivers that implement IPTV download/VOD (integrated service type 2), the performance in the linked status is the same as the data broadcasting reception status. See 7.2.3.2.3 (4) for the function performance regarding the receivers that support IPTV download/VOD (integrated service type 2).

- Function performance of getBrowserVersion( )

It is performed in both linked status and unlinked status. Performance is the same as data broadcasting reception status.

- Function performance of getProgramID( )

Performance in linked status is the same as data broadcasting reception status.

- Function performance of getActiveDocument( )

It is performed in both linked status and unlinked status. Return values are returned in the format where the scheme and host name are omitted and starts with / (abs\_path specified in RFC1808).

In case "/" is specified at the end of URI, or "?query" is specified, or in case of playing communication contents whose resource name cannot be specified, the values excluding URI scheme that specified the corresponding BML document and host name should be returned.

For example, the return value when communication contents that had gone through transition in `launchDocument("http://localhost/hoge/foo?query","cut")` called "getActiveDocument" function is `"/hoge/foo?query"`.

- Function performance of `lockScreen( )`

It is performed in both linked status and unlinked status. Performance is the same as data broadcasting reception status.

- Function performance of `unlockScreen( )`

It is performed in both linked status and unlinked status. Performance is the same as data broadcasting reception status.

- Function performance of `getBrowserSupport( )`

It is performed in both linked status and unlinked status. Performance is the same as data broadcasting reception status.

- Function performance of `launchDocument( )`

It is performed upon stipulation of the following restrictions in linked status and unlinked status.

- Restrictions in linked status

- In case of transition to broadcasting contents, an absolute URI starting with "arib-dc:" should be specified. Refer to section 5.14.10 for details.
- In case a BML document outside of base URI directory scope is specified, transition to unlinked status is executed. Refer to section 5.14.12 for details.

- Restrictions in unlinked status

- Transition to a BML document of broadcasting contents cannot be executed. In case broadcasting contents are specified in unlinked status, the receiver unit will execute a failure performance.

- Function performance of `launchDocumentRestricted( )`

This function can be optionally implemented by a receiver. It is used to execute transition from data broadcasting reception status, linked status to unlinked status. For the details of operation, refer to section 5.14.12. In case this function is called in unlinked status, the performance will be equivalent to the performance of the "launchDocument( )" function. In case data broadcasting contents starting with "arib-dc" are specified in this function, then the failure performance, which is the same operation as re-selecting stations shall be performed.

- Function performance of `quitDocument( )`

It is performed in all situations. The "quitDocument( )" function will perform differently in data broadcasting reception status, linked status and unlinked status. Refer to Table 5-22. For details of the status transition, refer to section 5.14.12.

**Table 5-22 Performance of quitDocument( )**

Status	Performance of quitDocument( )
Data broadcasting reception status	The presentation of broadcasting contents that are currently presented ends and after executing 4 of “Basic operation of receiver when selecting channels” described in section 2.1.10.2, 6 and thereafter is executed, and transition to the startup document of the entry component is executed. At that time, all modules that are locked by the “lockModuleOnMemory” function and “lockModuleOnMemoryEx” functions are released. If connection by PPP is done at that time, the connection is disconnected in accordance with Figure 5-7 of 5.14.13.1.
Linked status	The presentation of broadcasting contents that are currently presented ends and after executing 4 of “Basic operation of receiver when selecting channels” described in section 2.1.10.2, 6 and thereafter is executed, and transition to the startup document of the entry component is executed. At that time, all modules that are locked by the “lockModuleOnMemory” function and “lockModuleOnMemoryEx” functions are released. If connection by PPP is done at that time, the connection is disconnected in accordance with Figure 5-7 of 5.14.13.1.
Unlinked status	The presentation of communication contents that are currently presented ends and the service that the communication contents had belonged to immediately before unlinked status is re-selected. At that time, if the connection by PPP has been executed, then the connection is disconnected in accordance with Figure 5-7 of 5.14.13.1.

- Function performance of launchExApp( )

It is performed in all situations. This function is optional. In case of usage, the existence of the function should be confirmed before use by “getBrowserSupport( )”. In receiver units that can use options, performance is the same as data broadcasting reception status.

- Function performance of getFreeContentsMemory( )

Performance in linked status is the same as data broadcasting reception status. Refer to section 5.12.6.

- Function performance of isSupportedMedia( )

Performance in linked status is the same as data broadcasting reception status. Refer to section 5.12.6.

- Function performance of “detectComponent( )”

Performance in linked status is the same as data broadcasting reception status. Refer to section 5.12.6.

- Function performance of “lockModuleOnMemoryEx( )”

Performance in linked status is the same as data broadcasting reception status. Refer to

section 5.12.6. However, the receiver unit should release all modules secured by this function and the “lockModuleOnMemory( )” function at the time of transition to unlinked status.

- Function performance of unlockModuleOnMemoryEx( )

Performance in linked status is the same as data broadcasting reception status. Refer to section 5.12.6, section 5.13.3.

- Function performance of unlockAllModulesOnMemory( )

Performance in linked status is the same as data broadcasting reception status. Refer to section 5.12.6, section 5.13.3.

- Function performance of getLockedModuleInfo( )

Performance in linked status is the same as data broadcasting reception status. Refer to section 5.12.6, section 5.13.3.

- Function performance of getBrowserStatus( )

It is performed in both linked status and unlinked status. Performance is the same as data broadcasting reception status.

- Function performance of isRootCertificateExisting( )

It is performed in both linked status and unlinked status. Performance is the same as data broadcasting reception status.

- Function performance of getRootCertificateInfo ( )

It is performed in both linked status and unlinked status. Performance is the same as data broadcasting reception status.

- Function performance of getResidentAppVersion( )

It is performed in both linked status and unlinked status. Performance is the same as data broadcasting reception status.

- Operation of the startExtraBrowser() function

It shall only operate in linked status. The operation in linked status shall be the same as in the status while receiving data broadcasting. This function will fail if it is called in unlinked status.

- Operation of the transmitDataToSmartDevice() function

The function is operated in both linked and unlinked statuses. The operation is the same as in the status while receiving data broadcasting.

#### **5.14.6.10 Receiver unit audio control**

The function performance of receiver unit audio control for communication contents is

indicated in Table 5-23. Regarding the performance of data broadcasting reception status, refer to section 5.12.6.

**Table 5-23 Performance of receiver unit audio control of communication contents**

	Linked status	Unlinked status
playRomSound( )	O	O

It is performed in both linked status and unlinked status. Performance is the same as data broadcasting reception status.

#### 5.14.6.11 Timer function

The function performance of the timer function in communication contents is indicated in Table 5-24. Regarding the performance of data broadcasting reception status, refer to section 5.12.6.

**Table 5-24 Performance of the timer function in communication contents**

	Linked status	Unlinked status
sleep ( )	O	O
setInterval( )	O	O
clearTimer( )	O	O
pauseTimer( )	O	O
resumeTimer( )	O	O
setCurrentDateMode( )	O	X

Excluding “setCurrentDateMode()”, the timer function is performed both in linked status and unlinked status. “setCurrentDateMode()” is performed only in linked status. Performance is the same as data broadcasting reception status.

#### 5.14.6.12 External character function

The function performance of external characters in communication contents are indicated in Table 5-25. Regarding the operation of data broadcasting reception status, refer to section 5.12.6.

**Table 5-25 Performance of the external character function in communication contents**

	Linked status	Unlinked status
loadDRCS( )	O	O

The external character function is performed both in linked status and unlinked status. Performance is the same as data broadcasting reception status.

#### 5.14.6.13 Other functions

The function performance of other functions in communication contents is indicated in Table 5-26. Regarding the operation of data broadcasting reception status, refer to section 5.12.6.

**Table 5-26 Performance of other functions in communication contents**

	Linked status	Unlinked status
random( )	O	O
subDate( )	O	O
addDate( )	O	O
formatNumber( )	O	O

Other functions are performed both in linked status and unlinked status. Performance is the same as data broadcasting reception status.

#### 5.14.6.14 Closed caption display control function

The function performance of the closed caption control function in communication contents is indicated in Table 5-27. Regarding the performance of data broadcasting reception status, refer to section 5.12.6.

**Table 5-27 Performance of the closed caption control function in communication contents**

	Linked status	Unlinked status
setCCDisplayStatus( )	X	X
getCCDisplayStatus( )	X	X
getCCLanguageStatus( )	X	X

#### 5.14.6.15 Bookmark control function

The function performance of the bookmark control function in communication contents is indicated in Table 5-28. Regarding the performance of data broadcasting reception status, refer to section 5.12.6.

**Table 5-28 Performance of the bookmark control function in communication contents**

	Linked status	Unlinked status
writeBookmarkArray( )	X	X
readBookmarkArray( )	O	X
deleteBookmarkArray( )	X	X
lockBookmark( )	X	X
unlockBookmark( )	X	X
getBookmarkInfo( )	O	X
getBookmarkInfo2( )	O	X
startResidentBookmarkList( )	Optional	Optional



Performances of “readBookmarkArray( )”, “getBookmarkInfo( )”, “getBookmarkInfo2( )” are the same as data broadcasting reception status.

“startResidentBookmarkList()” is performed in both linked and unlinked status. Performance is the same as data broadcasting reception status. In case of using it, the existence of the function should be confirmed before use by “getBrowserSupport()”.

#### 5.14.6.16 Print related functions

The performance function of print related functions in communication contents are indicated in Table 5-29. Regarding the performance of data broadcasting reception status, refer to section 5.12.6.

**Table 5-29 Performance of print related function in communication contents**

	Linked status	Unlinked status
getPrinterStatus()	Optional	Optional
printFile()	Optional	X
printTemplate()	Optional	X
printURI()	Optional	Optional
printStaticScreen()	Optional	Optional
saveImageToMemoryCard()	Optional	X
saveHttpServerImageToMemoryCard()	Optional	Optional
saveStaticScreenToMemoryCard()	Optional	Optional

#### 5.14.6.17 IPTV linkage function

The function performance related to the IPTV linkage function is indicated in Table 5-30. These functions shall be used after confirming that the functions are supported using getBrowserSupport( ).

**Table 5-30 Performance of the IPTV linkage function for communication contents**

	Data broadcasting reception status	Linked status	Unlinked status
startVOD( )	X	O	X
startDlcDownload( )	O	O	X
getDlcDownloadStatus( )	O	O	X

See 7.2.4.2 and 7.2.5.1 for the function performance used by receivers that support IPTV download/VOD (integrated service type 2).

#### 5.14.6.18 Inquiry function

The function performance related to communication contents for the inquiry function is indicated in Table 5-31. See 5.12.6 for the performance of the functions in the data broadcasting reception status.

**Table 5-31 Performance of the inquiry function for communication contents**

	Linked status	Unlinked status
getContentSource()	O	X

#### 5.14.6.19 AIT-controlled application linkage function

The function performance related to the AIT-controlled application linkage function is indicated in Table 5-32. These functions shall be used after confirming that the functions are supported using getBrowserSupport( ).

**Table 5-32 Performance of the AIT-controlled application linkage function**

	Data broadcasting reception status	Linked status	Unlinked status
startReceivingAIT()	O	O	X
getReceivedAIT()	O	O	X
startAITControlledApp()	O	O	X

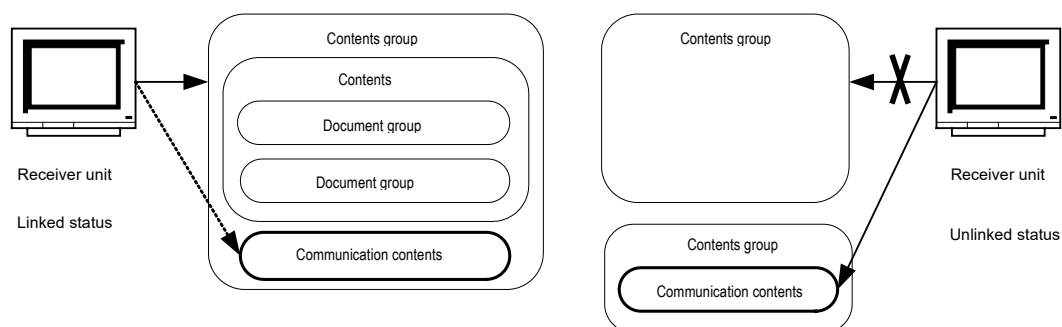
See 8.9.1.1 and 8.9.1.2 for the performance of the functions used by receivers that support AIT-controlled applications.

### 5.14.7 Mapping the scope of communication contents

BML contents can also be located on servers, and receiver units can refer and present BML documents on servers (hereinafter referred to as communication contents).

There are 2 types of statuses in communication contents. One is linked status where broadcasting contents can be referred to and the other is unlinked status where communication contents cannot be referred to. For the reference relationship, refer to section 5.14.9. The unlinked status can be optionally supported by a receiver.

Communication contents in the linked status act as one content of a content group. On the other hand, communication contents in unlinked status act as one content group. Communication contents in this status act as a special content group where data events do not exist.



**Figure 5-5 Linked status and unlinked status**

.The restrictions of communication contents in linked status and unlinked status

- Extended functions for broadcasting that can be used are specified. For operation, refer to section 5.14.6.
- Name spaces are different for BML documents for broadcasting and BML documents for communication. In case of specifying broadcasting contents from communication contents, a scheme with “arib:” or “arib-dc:” should be specified. In case of specifying communication contents from broadcasting contents, a scheme with “http:” or “https:” should be specified. It should be noted that handling of abbreviated forms is different as well. Regarding operation of name spaces in communication contents, refer to section 5.14.10. It is different from broadcasting contents and there is no concept of a startup document. However, it is possible for receiver units to specify the URI in a directory unit without directly specifying the resource in the communication contents. In such an event, communication contents of specifications in accordance with the setup of servers are returned. (i.e.: index.bml, etc.)
  - The restrictions of communication contents that are unique to linked status
- Since data events do not exist, the receivable event messages are limited only to the ones whose “event\_message\_group\_id” is 1. In case the corresponding ID receives an event message with 0, then it is ignored. For operation of “event\_message\_group\_id”, refer to section 2.3.4.
- In case pull back occurs during the playback of communication contents in linked status, the transition to entry component of broadcasting contents will be executed. If the receiver unit is connected to a TCP/IP network by a PPP connection using a modem, in such an event, the communication is disconnected.
- In the case broadcasting contents transition to the communication contents under a URI directory (hereinafter referred to as the base URI directory), a linked status is maintained.

However, the performance if another URI is specified shall be as described in \*1 and \*2. For the concept of base URI directory, refer to “Explanation of a base URI directory” in this section.

- Even in linked status, in case closed caption and superimpose are included in the broadcasting service that it belongs to, the presentation of closed caption and superimpose should be possible.

\*1 If the receiver unit receives a response in 300 s and redirect is specified, and if the redirecting destination is under the base URI directory, a linked status is maintained. However, if the redirecting destination is outside of the URI directory, the operation described in \*2 is performed.

\*2 If the receiver unit supports the unlinked function, the status is changed to the unlinked status. If the receiver unit does not support the unlinked function, an operation that is equivalent to channel reselection is performed. Alternatively, the communication contents in the transition destination shall be prevented from being displayed by maintaining the linked status without transitioning to the unlinked status.

- The restrictions of communication contents that are unique to unlinked status
- Among the extended functions for broadcasting that can use communication contents, since the communication contents in unlinked status is a service without a broadcasting stream, excluding EPG functions, the extended functions for broadcasting that refer to the broadcasting stream cannot be used. For the details of function performance being possible or impossible, refer to section 5.14.6.1.
- There are no restrictions on the URI specification of communication contents in unlinked status. However, the URI that specifies the reference of the broadcasting stream such as video, audio, closed caption, data carousel and event message, etc. cannot be specified. In case the receiver unit in this status is specified in the URI specifying the reference of those broadcasting streams, then the receiver unit will execute a failure performance. For details, refer to section 5.14.12.
- In unlinked status, receiver units should perform presentation, so that viewers can recognize that it is in unlinked status. For details, refer to section 1.8.1 “Display by browser features”.
- In unlinked status, the following regulations are established in addition to 5.7 “Operation of BML elements”.
  - “invisible” attribute of body element cannot be used.
  - “beitem” element cannot set values whose “type” attribute is other than “TimerFired”, “MediaStopped”, “DataButtonPressed” and “IPConnectionTerminated”.In case it is used, receiver performance will depend on a model of receiver.
- In unlinked status, the following regulations are established in addition to 5.12.3 “Performance of the DOM”.
  - In object elements, “setMainAudioStream” method and “getMainAudioStream” method cannot

be used.

In case they are used, receiver unit performance will depend on a model of receiver units.

- In unlinked status, `epgTune()`, `epgTuneToComponent()` with specifications of "arib://-1.-1.-1" or "arib-dc://-1.-1.-1" as a transition destination service cannot be used. In case they are used, receiver unit performance will depend on a model.
  - Explanation of a base URI directory

In linked status, a base URI directory is used as an identifier of document groups of communication contents. A base URI directory refers to the first host name and directory name that are specified by `launchDocument()` from broadcasting contents.

For example, the identifier that indicates the base URI directory of communication contents specified by "launchDocument"("http://localhost/hoge/index.bml", "cut") is "http://localhost/hoge/".

The following specifications are stipulated for the judgement of base URI directories.

- Identifier indicating the base URI directory does not include a port number. For example, the base URI directory of communication contents specified by `launchDocument("http://localhost:10080/hoge/index.bml", "cut")` from the BML document of data broadcasting is "http://localhost/hoge/". Therefore, even in the transition between different port numbers, if the base URI directory is matched, then a linked status will be maintained.
- The URL encoded name space should be handled as the equivalent to name spaces that are not encoded. For example, the base URI directory of "http://localhost/%7Ehoge/index.bml" and `http://localhost/~hoge/test.bml` are handled assuming that they are matched.
- Capital letters and lower case letters are not distinguished in host names. Capital letters and lower case letters are distinguished for directory names.

Resources directly under the base URI directory and resources stored under that directory are all regarded as within the document group of communication contents and a linked status will be maintained. If a URI that does not match the base URI directory is specified, the provisions in "The restrictions of communication contents that are unique to linked status" in this section shall be followed. For details on the status transition, refer to section 5.14.12.

#### 5.14.8 Operation guideline for transmission of communication contents

The transmission of communication contents is in accordance with HTTP/1.1 stipulated in RFC2616. Receiver units and interactive web servers execute communication by HTTP/1.1 on the port specified by the URI if "http:" is specified in the URI, and if "https:" is specified, then after establishing a connection by TLS1.0 and SSL3.0 or TSL1.2 on the port specified by the

URI, encrypted communication by HTTP/1.1 is performed. If the port number is not explicitly presented, then port number 80 in the case of “http:” and port number 443 in the case of “https:” are used as the default port numbers. However, handing ports might be different by firewall setup, etc. depending on the connection status. The connection status should be considered, and a default connection port can be set from the receiver unit. For the operation details of HTTP/1.1 by TLS, refer to Vol. 6.

This specification describes the minimum guidelines for operation and it does not by any means define the implementation range of receiver units.

- Operation guidelines of HTTP/1.1
- Guidelines for the connection method of receiver units
  - Receiver units should satisfy the required conditions for constant connection and message transfer specified in RFC2616.
  - In case of receiving responses from servers, receiver units should be able to receive chunk format transfer encoding receiver specified in RFC2616.
  - In order to acquire the latest status of communication contents, connections based on the “expiration model” and “validation model” specified in RFC2616 should be performed.
- For HTTP version, only HTTP/1.1 is operated. Operation of HTTP/1.0 by servers is not permitted as a basic rule. The performance process in case HTTP/1.0 is received by a receiver unit depends on a model.
- Fixed length subset specified in RFC1123 is operated for the date/time format.
  - For servers, only the date/time format of a fixed length subset specified in RFC1123 should be sent as the date format to the receiver units.
  - The receiver unit should be able to interpret fixed length subsets specified in RFC1123 as the date/time format. In case dates in RFC1036 or ANSI C format are received, then the format of those dates should be interpreted, but they can be ignored as well.

Examples are as follows:

Sun, 06 Nov 1994 08:49:37 GMT ; RFC1123  
Sunday, 06-Nov-94 08:49:37 GMT ; RFC1036  
Sun Nov 6 08:49:37 1994 ; ANSI C

- As a character set, only “EUC-JP” is operated for all of request message/response message/entity. The performance in case the receiver unit receives another character set depends on a model.

- "Identity" is used for contents-coding. "deflate" is operated as an option. The performance in case "deflate" is received by receiver units that do not support "deflate", or other values are received depends on a model.
- In case of specifying transfer-coding, "chunked" should be operated. The performance in case receiver units receive other values depends on a model.
- For product token, in case of specifying name of browser and version, it should be in alignment with the return value of "getBrowserVersion( )".
- Operation of quality value depends on a model.
  - "ja" is operated for the language tag. The performance in case other language tags are received depends on a model.
  - The receiver shall implement "Keep-Alive".
- Operation of general headers
- Operation of Cache-Control
  - Both request messages and response messages are operated.
  - In services where end-to-end connection is required, receiver units should specify "no-cache" in the request message. In such an event, any type of field name attached to the no-cache identifier should not be included in the request message. For request messages, a proxy handling HTTP/1.0 on the transmission path should be considered and a Pragma general header should be located and "no-cache" should be specified.
  - Servers should specify "no-cache" in the value of the Cache-Control header of the response message. In the same way as the request message, a Pragma general header can be located and "no-cache" can be specified, but this is not required. In case the receiver unit receives a "no-store", then the receiver unit should not cache the file specified by the HTTP session. The performance in case receiver units receive values other than "no-cache" or "no-store" depends on a model.
- Operation of Connection
  - Both request messages and response messages are operated. "close" is operated for the connection option. In case of disconnecting constant connections, this header is given and "close" should be specified. The performance in case receiver units receive values other than "close" depends on a model.
- Operation of Date
  - They are operated in the response message. Operation in request messages depends on a model.

- Operation of Pragma
  - Both request messages and response messages are operated as options.
  - Considering the fact that the proxy for HTTP/1.0 lies on the transmission path, receiver units should specify “no-cache” in the request header.
- Operation of Transfer-Encoding
  - Response messages are operated. In case of responding using the chunk format transfer encoding, the server should be added to response header and “chunked” should be specified.
  - It should be noted that Transfer-Encoding cannot be used simultaneously with Content-Length. For details, refer to RFC2616. The performance in case receiver units receive other values depends on a model.
- Other headers
  - Regarding the operation of Upgrade, Trailer, Via and Warning, depends on a model.
  - Operation of request headers
- Accept
  - It is operated. Receiver units can specify the following media types in this header.  
text/plain, text/X-arib-bml, text/X-arib-jis8text,  
text/css, text/X-arib-ecmascript,image/jpeg, image/X-arib-png, image/X-arib-mng,  
audio/X-arib-mpeg2-aac, audio/X-arib-aiff,  
application/X-arib-bmlclut, application/X-arib-btable, application/X-arib-dracs
- Accept-Charset
  - It is operated. In case of attaching this field, receiver units specify “EUC-JP” in this value.
- Accept-Encoding
  - It is operated. Receiver units can only specify "identity" and "deflate" in this value.
- Accept-Language
  - It is operated. Receiver units should specify “ja” in this value.
- Authorization
  - It is operated. The presentation technique and specification of the user interface to input the user ID and password, etc. depends on a model of receiver units.



- Proxy-Authorization
  - It is operated as an option. The presentation technique and specification of the user interface to input the user ID and password, etc. depends on a model of receiver units.
- Host
  - It is operated.
- Operation of “If-Modified-Since”, “If-Unmodified-Since”
  - It is operated as an option. When documents are reloaded by the extended function for broadcasting in communication contents, it can be used to judge whether or not the receiver unit should re-acquire the resources accumulated in Bcontents memory from the server. By using this header, receiver units can minimize the overhead of communication transactions.
- Referer
  - It is operated as an option. By using this header, servers can use the URI information of the caller. The URI of broadcasting contents starting with “arib-dc” can be set in the URI.
- User-Agent
  - It is operated. Receiver units should be in alignment with the return values of “getBrowserVersion( )”
- Other headers
  - The operation of “Expect”, “From”, “If-Match”, “If-None-Match”, “If-Range”, “Max-Forwards”, “Range” and “TE” depends on a model.
  - Operation of response headers
- Location
  - It is operated.
- Proxy-Authenticate
  - It is operated as an option. The presentation technique and specification of the user interface for entering the user ID and password, etc. depend on models of the receiver unit.
- WWW-Authenticate
  - It is operated. The presentation technique and specification of the user interface for entering the user ID and password, etc. depend on models of the receiver unit.

- Other headers

The performance in case receiver units receive “Sever”, “Accept-Range”, “Age”, “ETag”, “Retry-After” and “Vary” depends on a model.

- Operation of entity-headers

- Content-Encoding

It is operated in the response message. For the value, "identity" can be specified. “deflate” is operated as an option. The performance in case receiver units that do not supporting “deflate” receives “deflate”, or in case receiver units receive other values, depends on a model.

- Content-Language

It is operated in the response message. “ja” can be specified as the value. The performance in case other values are received by receiver units depends on a model.

- Content-Length

It is operated both in the request message and response message. However, in case of using chunk format transfer encoding, it should be in accordance with RFC2616. The performance in case receiver units receive response messages that do not have this header attached depends on a model.

- Content-Location

It is operated as an option in the response message.

- Content-Type

Both request messages and response messages are operated.

The media types that can be specified in a request message are as follows.

application/x-www-form-urlencoded

Media types that can be specified in response messages are as follows.

text/plain, text/X-arib-bml, text/X-arib-jis8text,  
text/css, text/X-arib-ecmascript,  
image/jpeg, image/X-arib-png, image/X-arib-mng,  
audio/X-arib-mpeg2-aac, audio/X-arib-aiff,

application/X-arib-bmlclut, application/X-arib-btable, application/X-arib-drcs

The performance in case receiver units receive media types other than the above depends on a model.

- Last-Modified

It is operated as an option. Receiver units can be used as the hint information of “If-Modified-Since” and “If-Unmodified-Since”. Unless there are special reasons, servers should send out this header.

- Other headers

The performance in case receiver units receive “Allow”, “Content-MD5”, “Content-Range”, “Expire” depends on a model.

- Operation of methods

- Operation of “GET”

It is operated.

- Operation of “POST”

It is operated.

- Operation of “CONNECT”

It is operated as an option. It is used to request tunnel performance in order to operate TLS and SSL. In such an event, handling of multistage proxies should be considered. For the details of the migration process from TLS and SSL by HTTP/1.1, refer to RFC2817. Also, for details relating to the operation of TSL and SSL, refer to Vol. 6.

- Other methods

The operation of “OPTIONS”, “HEAD”, “PUT”, “DELETE”, “TRACE” depends on a model.

#### **5.14.9 Restrictions related to mono-media reference from communication contents**

##### **5.14.9.1 Reference from broadcasting contents to communication contents**

Reference of communication contents from broadcasting contents is not executed in accordance with ARIB STD-B24 Appendix 3 6.2 “Guideline on reference across media types”. Only the transition of BML documents from broadcasting contents to communication contents is possible.

#### 5.14.9.2 Reference from communication contents to broadcasting contents

Occurrence of reference from communication contents to broadcasting contents is limited to when it is in linked status specified in section 5.14.7.

- When stream is referred to, the stream of video, audio and events within the same contents group (the same service) can be referred to. The ES in other services cannot be referred to. The restrictions of the ES that can be referred to simultaneously are in accordance with restrictions of broadcasting contents.
- In case of referring to mono-media of JPEG, PNG, binary tables, etc., if the specifications in section 5.14.7 are followed, then communication contents can be interpreted as it is in a different ES than broadcasting, therefore, only the resources within the module of the ES locked by using "lockModuleOnMemoryEx( )" can be referred to. For reference of other ES's locked by "lockModuleOnMemoryEx( )", refer to 5.13.3.
- The CSS style sheet file that can be specified by link elements sent in broadcasting contents and the ECMA Script file that can be specified by script elements are outside of the reference target from communication contents.

#### 5.14.10. Name space of communication contents

##### 5.14.10.1 Restricted matters relating to URI

Operation of name spaces should be in accordance with operation of ARIB STD-B24 ver3.2. Appendix 3 6.3. "Namespace". However, when presenting BML communication contents acquired from servers, if the BML document is in accordance with the specifications of ARIB STD B24 Appendix 2, the following restrictions are established in the name space.

- Multi-byte characters such as Japanese are not used in the UR L.
- "href" attribute of "a" element, and arguments of "launchDocument( )" and the URI that can be specified as arguments of "launchDocumentRestricted( )" should be a BML document.

##### 5.14.10.2 Operation of name spaces for servers

If "/" is specified at the end of the URI at the time of acquiring contents from servers, then the servers should return the BML documents specified in accordance with the setup.

(Example : browser.launchDocument("http://localhost/hoge/", "cut"); )

Receiver unit performance in case BML documents are not returned from servers depends on a model.

- In case an abbreviated URI format (relative URI) is used, then it is the same scheme as displayed contents, and it is handled as the relative format from the directory name where the displayed contents exist.

(Example: In case the URI of the BML document is `http://localhost/test/tmp/index.bml` and if `../hoge.bml` is written within the document, then the URI will be `http://localhost/test/hoge.bml`.)

#### 5.14.10.3 Operation of name spaces when referring to communication contents from broadcasting contents

Reference of communication contents from broadcasting contents is not performed by the specifications of section 5.14.9. When transition from broadcasting contents to communication contents is executed, then the absolute URI starting with “http:” and “https:” are used.

#### 5.14.10.4 Operation of name spaces when referring to broadcasting contents from communication contents

- When referring to stream of broadcasting contents from communication contents, only the absolute URI starting with “arib://-1.-1.-1” is used for specification. The performance in case URI in any other format is specified depends on a model. Also, for the specification method of the “component\_tag” value, section 2.1.2.8 should be followed.

(Example: `<object id="v" type="video/X-arib-mpeg2" data="arib://-1.-1.-1/-1" />`)

- When referring to mono-media of broadcasting contents (not stream) from communication contents, only the absolute URI starting with “arib-dc://-1.-1.-1” should be used at all times.  
(Example: `object id="i" type="image/jpeg" data="arib-dc://-1.-1.-1/40/0000/image.jpg" />`)
- When performing transition from communication contents to BML documents of broadcasting, only the absolute URI starting with “arib-dc://-1.-1.-1” should be used at all times.
- When specifying “es\_ref” or “module\_ref” of the “beitem” element from communication contents, only the absolute URI starting with “arib://-1.-1.-1” or “arib-dc://-1.-1.-1” should be used.  
(Example: `<beitem id="em" type="EventMessageFired" es_ref="arib-dc://-1.-1.-1/54" ..etc../>`)
- In communication contents, the “es\_ref” or “module\_ref” of the “beitem” element should be specified. In case of specification, only the absolute URI starting with “arib://-1.-1.-1” or “arib-dc://-1.-1.-1” should be used.  
(Example: `<beitem id="em" type="EventMessageFired" es_ref="arib-dc://-1.-1.-1/54" ..etc../>`)
- In case “type” attribute of the object element is a character string starting with “video”, then the performance when anything other than “arib:” is specified in the scheme depends on a model.
- When the “remain” attribute of the object element is set as “remain” and transition is executed from broadcasting contents to communication contents, then the absolute URI starting with either “arib://-1.-1.-1” or “arib-dc://-1.-1.-1” should be used for the object element that is the “remain” target of communication contents in the transition destination.
- When the “remain” attribute of the object element is set as “remain” and in case of transition from broadcasting contents to communication contents, if the abbreviated form “/-1” is set for the AV

stream in broadcasting contents, then the value of “data” attribute obtained in the DOM interface of the corresponding object element in communication contents after transition should return "arib://-1.-1.-1/-1".

- When the “remain” attribute of the object element is set as “remain” and in case of transition from broadcasting contents to communication contents, if, for example, the abbreviated form of "/50/0000/a.jpg" is set for still images, then the value of the “data” attribute obtained in the DOM interface of the corresponding object element in communication contents after transition should return "arib-dc://-1.-1.-1/50/0000/a.jpg".
- In case of transition from linked status to unlinked status, or transition from data broadcasting reception status to unlinked status, the “remain” attribute cannot be specified. For details, refer to section 5.14.12.
- JPEG images that exist in servers starting with “http:” or “https” cannot set “remain” in the “remain” attribute of object elements.
- It is possible to limit the reference to services other than the service that is currently referred to by the following extended function for broadcasting. For arguments that can be used for functions at this time, they operate the same way as data broadcasting reception status. (Refer to 5.13.3)

epgTune, epgIsReserved, epgReserve, epgCancelReservation,  
epgReclsReserved, epgRecReserve, epgRecCancelReservation  
epgTuneToComponent

#### **5.14.11 Guidelines of communication contents operation**

##### **5.14.11.1 Precautions during the operation of communication contents**

Since the service provider providing communication contents have the attributes of TCP/IP networks which are employed for the distribution of communication contents, the fact that it is extremely difficult to present BML contents in the same quality as data broadcasting should be taken into consideration. The characteristics of TCP/IP networks are listed below.

- There is no guarantee that the transmission path spectrum is constant.
- Depending on the viewer’s environment, the transmission path spectrum that can be used is different.
- Since the transmission path can take various shapes, a uniformed response to viewers cannot be guaranteed.
- Depending on the characteristics of the en route proxy, data transmission may be blocked.

Furthermore, by putting an excessive load on servers, the response may be lowered greatly. By those factors, the fact that the presentation of communication contents distributed by TCP/IP

networks will not be done in uniformed manner to all viewers, and the presentation of receiver units may not be done in proper way, depending on the status of the network, should be taken into consideration.

Because of those factors, BML contents that fit the following descriptions should be distributed by data broadcasting instead of via communication.

- BML contents whose performance timing is important such as synchronization of video and audio.
- BML contents for which the proper presentation of all elements is important.
- BML contents that use a lot of mono-media such as still images.

#### **5.14.11.2 Guidelines of communication contents descriptions**

Precautions for writing communication contents are as follows.

- Communication contents are written with profiles in accordance with operation of section 5.12.
- It should be noted that name spaces are different for broadcasting contents and communication contents. Refer to section 5.14.10 for details.
- For communication contents assuming operation in linked status, it is strongly recommended that only HD pixel size be operated. In case of operating communication contents of SD pixel size in communication contents in linked status, if it is linked improperly from the data broadcasting contents of HD video, then it will be a combination of data in SD pixel size and video in HD pixel size, which may not be played by receiver units. In case of operating communication contents of SD pixel size in linked status, it should be operated with meticulous care, however, it should not be operated as a basic rule.
- For communication contents assuming operation in linked status, the specification of port numbers should not be performed.
- In case of specifying communication contents from directory specification, "/" should be attached at the end. (Example: "http://localhost" X: incorrect, http://localhost/ O: correct)
- Since the performance in case #fragment is attached to the URL specifying communication contents depends on a model, specification using #fragment should not be performed. (Example: "http://localhost/test.bml#fragment X: incorrect)
- BML documents composing communication contents, binary tables, src attributes of "script" elements that are associated in the "ECMAScript" file, character code of the CSS file associated in the "link" element should be all written in EUC-JP.

- “BML version” written in communication contents should be 3.0, which is the same for broadcasting contents.  
(Example: `<?bml bml-version =”3.0” ?>`)
- In order to distinguish communication contents in linked status and unlinked status, extensions of the BML document composing communication contents should be:
  - For BML documents assuming linked status: `lbml(example: index.lbml)`
  - For BML documents assuming unlinked status: `ibml(example: index.ibml)`Receiver units should not judge linked status or unlinked status by this extension. Performance in accordance with status transition indicated in section 5.14.12.1 should be performed.
- By locating mono-media composing one screen within the same directory, the path searching process of receiver units can be decreased.
- The URI that can be specified in the “href” attribute of “a” element is a BML document only.
- In communication contents, there should be either a link set up to another BML document or writing of a method to finish. This is because when receiver units encounter BML documents without links, the receiver units have no methods for transition to another status other than selecting stations.
- Resources in communication contents (mono-media such as JPEG or PNG) can be specified independently from the scope of the base URI directory. In case mono-media is linked independently from the scope of the base URI directory in linked status, then a linked status will be maintained.
- In case of locking modules with the `lockModuleOnMemoryEx( )` function in linked status, attention needs to be paid to whether or not the specified module exists for certain or the module about to be locked is intentional.
- Since the functions that can be used in linked status and unlinked status are different, the performances are different even for the same contents depending on the differences in status. Therefore, regarding the contents that are written assuming only linked status, it should be noted that the presentation in unlinked status is not guaranteed.
- In case of operating bookmark contents, the cases where it is started from the bookmark list function by receiver features (\*1) should be considered and bookmark contents need be written limited to the extended function for broadcasting, which can be used in unlinked status. In case of writing bookmark contents with functions that can be used in linked status, viewers will not be able to reach the corresponding bookmark contents due to the failure performance of receiver units by calling the function of linked status in unlinked status.
- In case of specifying resources in communication contents in linked status, the names in abbreviated forms specified in ARIB STD-B24 Vol. 2 9.2.1.3 cannot be used. In case of referring



to resources transmitted by broadcasting, the absolute URI starting with either “arib://-1.-1.-1” or “arib-dc://-1.-1.-1” should be used.

- Considering that the area appointed for module acquisition in the module is 1MByte, the total volume of resources in the communication contents that are referred to within the BML documents at one time should not exceed 1MByte.
- If descriptions using overwriting of the data attribute are done to perform switching of mono-media such as button shapes, then the receiver units will access the web server each time and the performance response may greatly be worsened. For the switching of shapes, if it can be realized by overwriting “visibility”, then worsening of response can be avoided.

For within document scope (during presentation of one BML document), the timing for receiver units to follow the update of resources on the web server depends on a model (the newest resource is acquired and implemented in the presentation). Therefore, if the receiver unit wants to guarantee the acquisition of the newest resources, then along with document transition, consider the possibility that the corresponding resource has been referred to before and has been acquired already, Cache-Control: no-cache should be attached and transmitted for the corresponding resources.

- TLS1.0/SSL3.0 or TLS1.2 shall be used when handling personal information in interactive data broadcasting services and as well as communication content in the linked status.

(\*1) Refer to transition H of section 5.14.12.1.

#### **5.14.12 Guidelines of receiver unit performance when receiving communication contents**

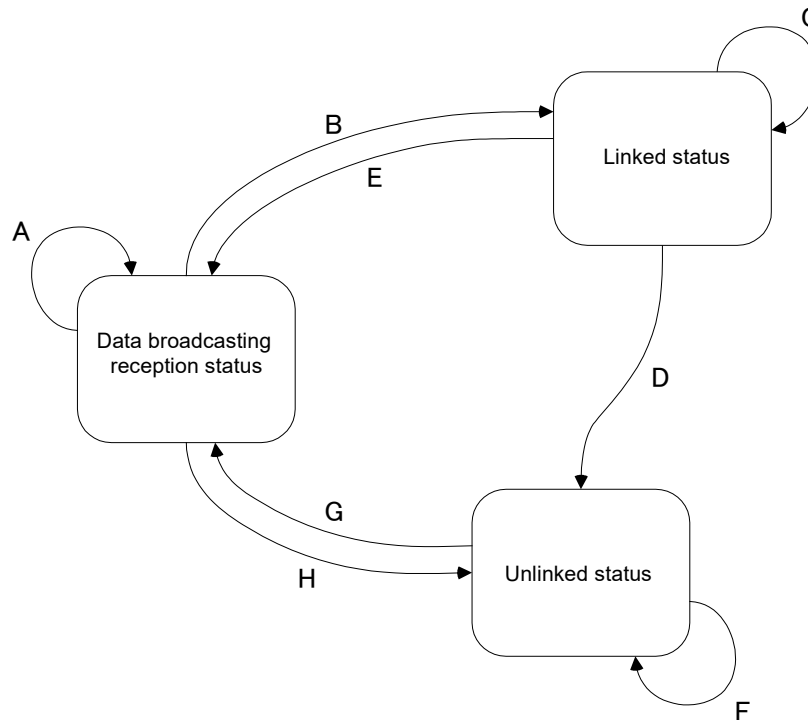
##### **5.14.12.1 Guidelines of the receiver unit status transition performance**

Receiver units can take the following 3 statuses for receiving BML contents.

- (1) “Data broadcasting reception status” which is receiving and playing data broadcasting and video/audio
- (2) “Linked status” which is receiving and playing both communication contents and video/audio
- (3) “Unlinked status” which is receiving and playing only the communication contents

In order for viewers to be able to distinguish linked status/unlinked status, receiver units should perform the display specified in 1.8.1 in unlinked status.

Regarding linked status and unlinked status, refer to section 5.14.7. Receiver units perform transitions between those 3 statuses upon instructions from contents or instructions from users. The structure of status transition is indicated in Figure 5-6. \* The unlinked status function can be optionally supported by a receiver.



**Figure 5-6 Status transition of receiver units when receiving communication contents**

- Explanation of transition A (Data broadcasting reception status→Data broadcasting reception status)

Transition A is a transition from normal data broadcasting to data broadcasting. For the operation details, refer to Chapter 2.

- Explanation of transition B (Data broadcasting reception status→Linked status)

Transition to linked status from data broadcasting reception status is possible by “launchDocument( )” or “a” element. There are two types of schemes of the URI to specify communication contents, which are “http:” and “https:”. For the details of the name spaces of communication contents, refer to section 5.14.10. (Example: `browser.launchDocument("http://localhost/hoge/index.bml", "cut");` )

Communication contents can specify the directory without specifying resources. In case a directory is specified, servers will return specified communication contents in accordance with the setup. (For example, `index.bml`, etc.) In case of specifying directory, “/” at the end should be attached as an indication of directory.

(Example: `browser.launchDocument("http://localhost/hoge/", "cut");` )

- Explanation of transition C (Linked status→Linked status)

The scope of the document group of communication contents in linked status will make the directories under the URI host address and directory (hereinafter referred to as base URI directory) the target area. It does not depend on scheme (protocol) specification. (Example: //localhost/hoge/)

For example, in case of transition from "http://localhost/hoge/index.bml" to "http://localhost/hoge/foo/test.bml", the linked status will be maintained. In case of transition to "https://localhost/hoge/foo/test.bml", the linked status will be maintained as well.

- Explanation of transition D (Linked status→Unlinked status)

If transition of the base URI directory to a BML document of a different URI, for example "//localhost/hoge/", is performed, the operation shall conform to the provision in \*2 of "Restrictions on communication contents that are unique to linked status" in 5.14.7. If the receiver supports a presentation function in the unlinked status, the receiver unit should be in an unlinked status. If the receiver does not support the unlinked status, an operation equivalent to channel reselection shall be performed. Alternatively, the communication contents in the transition destination shall be prevented from being displayed by maintaining the linked status without transitioning to the unlinked status. For example, in the case of transition to "http://otherhost/startup.bml" or "http://localhost/test/startup.bml", the receiver unit that supports the unlinked status presentation function will be in an unlinked status.

In case of transition to an unlinked status, the receiver unit should release all modules locked by "lockModuleOnMemoryEx( )".

In case a transition to an unlinked status is desired even though the transition destination is within the base URI directory scope, "launchDocumentRestricted( )" is used. By using this function explicitly, the reference to broadcasting contents in the directory within the base URI directory scope can be terminated. \*1, \*3

(Example: browser.launchDocumentRestricted("http://localhost/hoge/another.bml", "cut"); )

- Explanation of transition E (Linked status→Data broadcasting reception status)

Transition from a linked status to a data broadcasting reception status is possible by using "launchDocument( )", "a" element or "epgTune()", "epgTuneToComponent()". Before the transition, contents should lock the BML documents in the transition destination by "lockModuleOnMemoryEx( )" to confirm the existence of documents in the transition destination to perform the transition safely. The performance in case BML documents do not exist in the transition destination depends on a model.

"arib-dc:" should be specified for the URI scheme to specify the broadcasting contents. In case of transition using "launchDocument", it should be noted that only the current service such as "arib-dc://-1.-1.-1/50/0000/startup.bml" can be specified. In case of "epgTune( )", "epgTuneToComponent( )", "arib:" or "arib-dc:" should be used in the URI scheme to specify the broadcasting contents, and in case of using those functions, services other than the current service can be specified.

(Example: browser.launchDocument("arib-dc://-1.-1.-1/50/0000/startup.bml","cut"); )

By using "quitDocument( )", transition from a linked status to the data broadcasting reception status is possible. In such an event, transition of the receiver units to the entry component of broadcasting service that the receiver unit currently belongs to is executed.

The transition from linked status to data broadcasting reception status can be generated by switching the performance of the channel (service) by user. In such an event, if the receiver unit is performing PPP connection using a modem, the receiver unit will disconnect the connection in accordance with Figure 5.7 in 5.14.13.1. In such an event, performance to select stations should be performed after presentation of a message to notify disconnection, and the connection should be disconnected upon the approval of the viewer. \*2

- Explanation of transition F (Unlinked status→Unlinked status)

The concept of a base URI directory does not exist in the unlinked status and BML documents that exist in name spaces of "http:" and "https:" can be specified freely with the "launchDocument( )" argument or the "href" attribute of the "a" element.

For details of operational restrictions on the extended function for broadcasting in this status, refer to section 5.12.6.

As it is indicated in the transition figure, a transition of receiver units from an unlinked status to linked status is not possible.

- Explanation of transition G (Unlinked status→Data broadcasting reception status)

The transition from an unlinked status to a data broadcasting reception status can be performed by using "quitDocument( )", "epgTune( )" and "epgTuneToComponent( )". When "quitDocument( )" is specified in the unlinked status, the receiver unit will re-select the station in the broadcasting service that was referred to until immediately before the transition to unlinked status. Designers of receiver units should keep in mind that the performance of "quitDocument( )" is different from transition E.

Transition from an unlinked status to a data broadcasting reception status can be generated

by the switching performance of channels (service) by the user. In such an event, performance to select stations should be performed after presentation of a message to notify disconnection, and connection should be disconnected upon approval of the viewer. \*2

- Explanation of transition H (Data broadcasting reception status→Unlinked status)

If the status of a receiver with the unlinked status presentation function is changed from a data broadcasting reception status to an unlinked status, "launchDocumentRestricted( )" is used. By using this function explicitly, receiver units can be switched to an unlinked status at the time of transition from broadcasting contents to communications and the reference to broadcasting contents can be terminated. \*1, \*3

(Example: browser.launchDocumentRestricted("http://localhost/hoge/index.bml", "cut"); )

Even when receiver units with the function to specify the URI of communication contents directly refers to communication contents, transition to an unlinked status should be executed if the receiver units have the unlinked status function as well. Receivers without the unlinked function shall not transition to an unlinked status.

In case of transition to an unlinked status, receiver units should release all modules locked by "lockModuleOnMemoryEx( )".

- \*1 Contents producers should keep in mind that in case of transition from a data broadcasting reception status or a linked status to an unlinked status, the presentation of video/audio/still images is terminated. In other words, the "remain" attribute cannot be used during this transition. Receiver units should delete the still image plane at the time of transition and should guarantee that the mixed display of data broadcasting contents does not happen to contents in an unlinked status.
- \*2 Contents producers should keep in mind that receiver units may not present the message of disconnection in some cases for cases other than channel change operation. In case of disconnecting, when elements associated with "launchDocument( )" or "quitDocument( )" on the contents are activated, then a message of disconnection should be presented.
- \*3 Contents producers should keep in mind that there are receivers that do not support the unlinked status presentation function. Therefore, it is desirable that content producers identify the receivers that do not support the unlinked status presentation function using the functionname "Unsupported" and additionalinfo "Misc.Unlink" of the getBrowserSupport( ) when the contents' status is changed from the data broadcasting reception status or linked status to the unlinked status. It is also recommended that the producers take measures, such as disabling a button to change the status to the unlinked status for receivers that do not support the unlinked status presentation function. See 5.12.6.9 (14).

For receiver units with the bookmark list function as a receiver feature, when bookmark contents are selected by communication contents, transition of the receiver unit to an unlinked status should be performed if the receiver units support the unlinked function as well. Receivers without the unlinked function shall not transition to an unlinked status. (In order to prevent the status from being changed, it is recommended to take measures, such as notifying viewers that such transition cannot be made or disabling the presentation in the bookmark list as the receiver's default features of the bookmark list).

#### **5.14.12.2 Guidelines for when receiver unit performance fails**

In case functions that cannot be used in status specified in 5.14.12.1 are used or broadcasting contents are referred in an unlinked status, receiver units will fail.

Receiver units will perform the following processes for the parts specified as “performance failure” or “failure performance” of browsers in 5.14.6, etc.

- (1) Browser is finished.
- (2) Along with it, in case the connection to the TCP/IP network was made by a PPP connection using a modem, it should be disconnected.
- (3) In case it is in linked status, 4 and after of “Basic operation of receiver when selecting channels” described in section 2.1.10.2 are executed.
- (4) In case it is in unlinked status, re-selecting of the service that it belonged to until immediately before the transition to unlinked status is executed.

#### **5.14.12.3 Guidelines of receiver unit performance when receiving error responses**

In case the receiver unit requests communication contents and resources composing the communication contents in the server, and if the response result is an error response, the guidelines for performance are as follows.

- Receiver unit performance in case of receiving error responses in BML documents depends on a model.
- The presentation performance in case of receiving error responses in resources composing BML documents depends on a model.

\*1 In case status codes are in 400s or 500s. For details, refer to RFC2616.

#### **5.14.13 Guidelines for connecting/disconnecting**

For the operation of the return value of the following extended functions for broadcasting relating to connection and disconnection, refer to Appendix-12. For details of status to be used in Appendix-12, refer to 5.14.13.1.

isIPConnected, connectPPP, connectPPPWithISPPParams, disconnectPPP

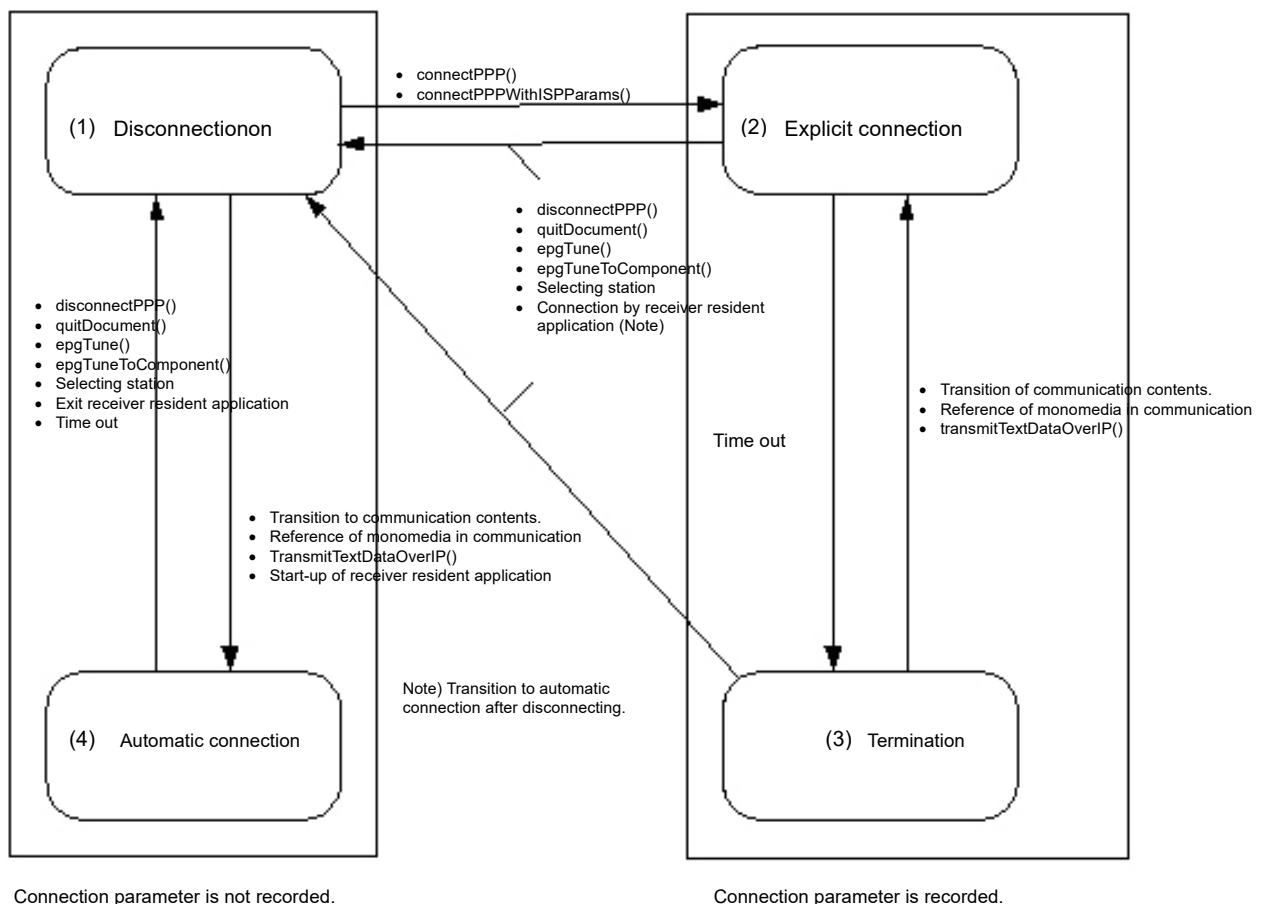
##### **5.14.13.1 Receiver performance guidelines for disconnecting and re-connecting**

Using “connectPPP( )”, for example, when disconnected by reasons such as a timeout occurrence during the presentation of some contents on the server whose access is permitted only by certain providers, if the receiver unit performs an auto-connection, then there is concern that the contents cannot be received from there on.

In order to avoid such a situation, receiver units will temporarily store connection parameters specified by contents in the receiver unit at the time of connection. In this section, the area of temporary memory is called “connection parameter temporary memory area”. The value of the

connection parameter recorded in this area is a value set by the argument in case of “connectPPP( )”, and a value set by the receiver unit in case of “connectPPPWithISPParams( )”. For details, refer to 5.14.13.2 and 5.14.13.3.

The performance of disconnection and re-connection of receiver units is indicated in Figure 5-7. Disconnected status in (1) is the initial status. Status transition performance in Figure 5-7 is a guideline of PPP connection regarding connection and re-connection. Receiver units in constant connection environments such as Ethernet connections are out of target.



**Figure 5-7 Status transition of receiver units in performance of disconnecting and re-connecting**

● Status transition from 1

In case the script uses “connectPPP( )” or “connectPPPWithISPParams( )”, the receiver unit will store the connection parameter to the connection parameter temporary memory area (refer to 5.14.13.2) and connection is performed. If the connection is successful, transition to status 2 is executed.

In case of connecting to TCP/IP networks with factors other than above (refer to 5.14.3.3 for

details), the receiver unit performs connection without recording the connection parameter to the connection parameter temporary memory area. If the connection is successful, transition to status 4 is executed.

- Status transition from 2

In case disconnection occurs for the following reasons, parameters are cleared and transition to status 1 is executed. "IPConnectionTerminated" event is not generated in this disconnection.

- Disconnection by "disconnectPPP( )"
- Transition to broadcasting by "quitDocument( )"
- Selecting station performance by users and selecting station performance by "epgTune( )".

In case disconnection is made by other factors, the connection parameter is not cleared and transition to 3 is executed. In this disconnection, the "IPConnectionTerminated" event is generated.

For the details of disconnection performances, refer to 5.14.13.4.

- Status transition from 3.

In status of 3 where it is terminated already

- Transition to broadcasting by "quitDocument( )"
- In case the selecting station performance by the user or selecting station performance by "epgTune( )" is executed, the receiver unit will clear the connection parameter and transition to status 1 is executed.

For the details of process relating to disconnection and termination, refer to Appendix-12.

In case a connection to a TCP/IP network is generated by the following factors from the status of 3, then the receiver units will re-connect using parameters stored in the connection parameter temporary memory area.

- In case a mono-media reference on a server is generated from the BML document that is currently presented.
- In case "transmitTextDataOverIP( )" is called from the BML document that is currently presented.
- In case transition from the BML document that is currently presented to another BML document has occurred.

In case a connection by receiver resident application is generated, then the connection parameters are cleared and after transition to status 1, transition to automatic connection is done.

- Status transition from 4

While in the status of automatic connection, if disconnected by a station selection operation by a viewer or by calling the "epgTune()" function, "epgTuneToComponent()" function, "quitDocument()" function, "disconnectPPP()" function from BML contents, then a transition to status 1 is executed.



#### 5.14.13.2 Receiver unit performance guidelines when connecting by scripts

It should be noted that connection by the “connectPPP( )” function can only be used when the receiver unit is in the status of viewing data broadcasting. (Refer to 5.14.6.6). When connection is done, the receiver unit will store the “connectPPP( )” argument as a connection parameter in the connection parameter temporary memory area specified in 5.14.13.1.

When connection by script is done using “connectPPPWithISPParams( )”, the receiver unit will connect using the connection parameter that is preset in the receiver unit. For this performance, refer to 5.14.6.6.

When connection performance by the “connectPPP( )” function or the “connectPPPWithISPParams( )” function is executed, the receiver unit will set the connection parameter in the connection parameter temporary memory area and by maintaining the area, it can be distinguished from the automatic connection status in 5.14.13.3. This is transition performance of 1 → 2 in Figure 5-7.

#### 5.14.13.3 Receiver unit performance guidelines for auto-connection

In case the receiver unit has not done an IP connection and the connection parameter is not recorded in the connection parameter temporary memory area specified in 5.14.13.1 (when in the status of 1), if the following events are generated, the receiver unit will automatically connect to the TCP/IP network.

- In case of transition to the BML document on the server is performed (Transition using “launchDocument( )”, “launchDocumentRestricted( )” and “href” attribute of anchor element)
- In case of mono-media reference on the server is generated from the BML document.
- In case transmitTextDataOverIP( ) is called.
- In case of connecting to the TCP/IP network from a receiver’s native application.

In case of automatic connection, the connection parameters that are preset in the receiver unit should be used in accordance with the setup for each priority usage line type. In automatic connections, values will not be set in the connection parameter temporary memory area. Automatic connections can be distinguished from connections by “connectPPPWithISPParams( )” by existence/non-existence of values in the connection parameter temporary memory area.

In case connection parameters to the TCP/IP network are not set in the receiver unit, then automatic connection will fail and transition to status of 1 is executed. Also, when connection parameters to the TCP/IP network are set in the receiver unit, but if they are not a connection by PPP, then automatic connection to the TCP/IP network will succeed and transition to status of 4 is executed.

The performance when the receiver unit fails to connect to the TCP/IP network depends on a model. However, in case the receiver unit has multiple communication methods, automatic connection will be attempted in accordance with the set up for each priority usage line type.

#### **5.14.13.4 Receiver unit performance guidelines when disconnected**

When an explicit disconnection performance (\*) is executed by the viewer or BML contents, then the "IPConnectionTerminated" event is not generated by the receiver unit. In such an event, if the connection parameter temporary area is set, then the contents will be cleared.

In case disconnection occurs by factors other than an explicit disconnection performance, in case of, for example, occurrence of a no communication timeout by the receiver unit, or external factors (disconnected from the ISP, etc.), the receiver unit will generate a "IPConnectionTerminated" event at the time of disconnection. Regarding connection parameters stored in the connection parameter temporary area at this time should follow the transition performance of 5.14.13.1.

(\* Explicit disconnection performance: "disconnectPPP", "quitDocument", "epgTune", "epgTuneToComponent", selection of station, connection by receiver unit resident application)

In case "disconnectPPP( )" is called in the termination status (status 3 in Figure 5-7), value of success is returned and the contents of the connection parameter temporary area are cleared and a transition to the disconnected status in 1 is executed.

In disconnection of communication by "quitDocument()", an "IPConnectionTerminated event" will not be generated. Receiver units should complete disconnection performance before the startup document specified in 5.14.12 is displayed and the timing to disconnect after the calling of "quitDocument( )" to the display of the startup document depends on a model. Also, if the connection parameter temporary memory area is set, then the receiver unit will clear the contents at this timing.

The "IPConnectionTerminated" event is not generated at the disconnection of communication caused by the selecting station performance, "epgTune()" function, "epgTuneToComponent()" function. Receiver units should complete the disconnection performance before completion of station selection and the timing to disconnect from after selecting station or calling of the "epgTune( )" function to completion of selecting station depends on a model. If the connection parameter temporary memory area is set, then the receiver unit will clear the connection parameters at this timing.

In the status of explicit connection, in case the receiver unit resident application with the communication function is started, then clearing of the contents of disconnection of the connection/connection parameter temporary memory area is executed, and after transition to a disconnected status is done, connect from automatic connection.

#### 5.14.13.5 Connection guidelines from receiver's native application

In case the receiver's native application connects to the TCP/IP network, connection should be made using connection parameters that are set in the receiver unit, and connection using connection parameters used in "connectPPP()" stored in the connection parameter temporary memory area should not be executed.

Also, in case the receiver's native application performs communication while the receiver unit is in a connected status using "connectPPP()", then connection by "connectPPP()" is explicitly disconnected once and after transition to 1 is done, communication should be established. While connected by "connectPPPWithISPParams()" or during automatic connection, communication from the receiver's native application can be performed.

Receiver unit performance in case the receiver's native application attempts to perform communication while connected by "connectPPP()" depends on a model.

#### 5.14.13.6 Display guidelines of confirmation messages when connecting or sending data

Whether display of confirmation message to viewer by dialogue, etc. is required or not in case of executing connection performance of communication is as shown in Table 5-33. Also, whether the display of confirmation messages to viewers by dialogue, etc. is required or not in case of executing data sending performance is as shown in Table 5-34.

**Table 5-33 Confirmation message display of receiver units relating to connection**

	Connection factors		Status		
			Broadcasting	Linked	Unlinked
1	connect()		X	-	-
2	vote()		X	-	-
3	connectPPP()		X	-	-
4	connectPPPWithISPParams()		X	X	-
5	Automatic connection		O	O	O
6	Reconnection		O	O	O
7	transmitTextDataOverIP()	Dialup	O(*1)	O(*1)	O(*1) (*2)
8		Constant connection	X	X	X(*2)

O : Should be displayed(optional feature of receiver unit)

X : Not displayed(Obtaining permission from viewer in contents before execution is assumed)

- : Not operation (Execution error)

(\*1) : In case of already connected status (regardless of explicit connection by function or automatic connection format), do not display confirmation messages.(This is because execution of "transmitTextDataOverIP()" is not a connection factor.)

(\*2) : Confirmation message related to the sending of data should be displayed. (Refer to Table 5-34)

**Table 5-34 Display of confirmation message of receiver units relating to sending of data**

	Factor of sending of data		Status		
			Broadcasting	Linked	Unlinked
1	sendTextData( )		X	-	-
2	transmitTextDataOverIP()	Constant connection / Dialup	X	X	O

- O : Should be displayed
- X : Not displayed (Obtaining permission from viewer in contents before execution is assumed)
- : Not operation (Execution error)

**5.14.14 Operation of root CA certificates**

In encrypted communication using TLS and SSL, root certificate to authenticate the broadcaster to operate the server is necessary. There are two types of certificates which are the root CA certificates of general purpose and the root CA certificates of limited purpose by broadcasters. Root CA certificates of general purpose are stored in the receiver units permanently; however, root CA certificates of limited purpose by broadcasters are effective only within the service operated individually by broadcasters. When root CA certificates is written in this chapter, it refers to the general term of root CA certificates, which does not differentiate between root CA certificates of general purpose and root CA certificates of limited purpose by broadcasters, and in case individual operation is necessary, then it is written explicitly as root CA certificates of general purpose and root CA certificates of limited purpose by broadcasters.

For details relating to the transmission of route certification documents, refer to Chapter 2. For details of encrypted communication and for details of the root CA certificate format, refer to Vol. 6.

**5.14.14.1 Operation of the memory area for root CA certificates**

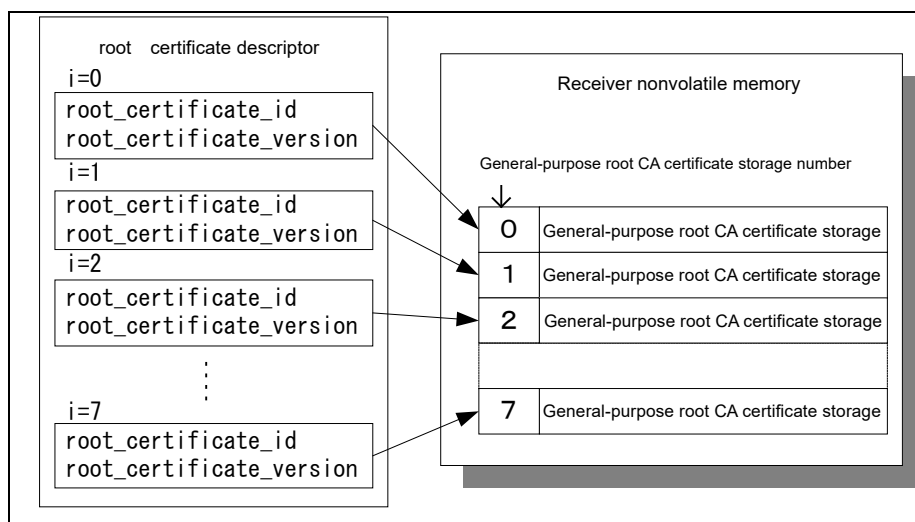
Receiver units establish a memory area to store root CA certificates. If the root CA certificate type (root\_certificate\_type) in the Root Certificate Descriptor is 0, then the receiver unit handles it as a root CA certificate of general purpose.

The root CA certificates of general purpose acquired from the data carousel are stored permanently in the area indicated in Table 5-35. The receiver unit refers to the root CA certificate stored in this memory area when performing encrypted communication regardless of the receiver unit status (data broadcasting reception status/linked status/unlinked status).

**Table 5-35 Memory area for root CA certificates used in Digital Terrestrial Television Broadcasting**

Type	Meaning	Capacity of memory area
Memory area for root CA certificates	Area to store root CA certificates of general purpose transmitted by carousel	<ul style="list-style-type: none"> <li>• 3KB per one certificate</li> <li>• Quantity: 8</li> </ul>

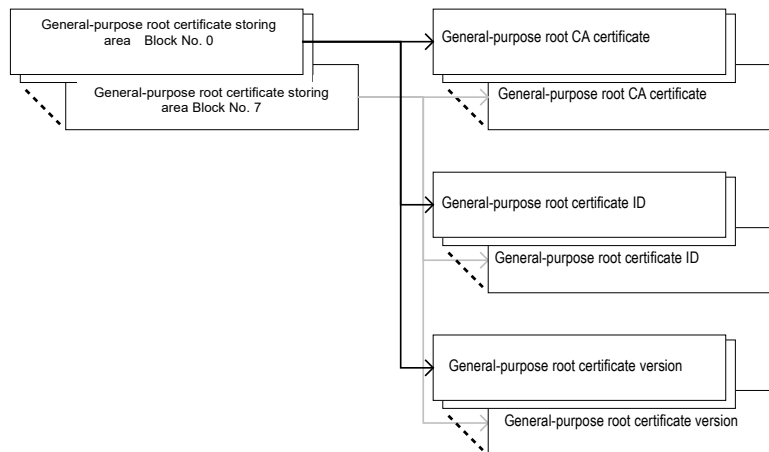
Each block of each memory area for root CA certificates is indexed with a root CA certificate storage number of 0-7 corresponding to the line of “root\_certificate\_id” and “root\_certificate\_version” in the Root Certificate Descriptor. (Refer to Figure 5-8) If the “root\_certificate\_id” is 0xFFFFFFFF in the Root Certificate Descriptor, then it is indicating that root CA certificates of general purpose to be stored do not exist in the transmitted module for storing root CA certificates in transmission.



**Figure 5-8 Corresponding relationship of the Root Certificate Descriptor and the memory area for root CA certificates**

#### 5.14.14.2 Storage of root CA certificates of general purpose to receiver units

When storing a root CA certificate of general purpose, the ID of the root CA certificate for general purpose and the version of the root CA certificate for general purpose (hereinafter referred to as the root CA certificate information) written in the module for root CA certificates correspond to the storage number of memory area for root CA certificates by receiver within the receiver units. (Refer to Figure 5-9) Regarding the format to store root CA certificate storage information, it depends on a model of receiver units. For details of the structure of module for storing root CA certificates, refer to Vol. 6. For details of receiver unit performance of root CA certificate of general purpose, refer to 5.14.14.3.



**Figure 5-9 The relationship between storage numbers of memory area for root CA certificates and related root CA certificates information**

For handling expired root CA certificates in the memory area for root CA certificates, it depends on a model of receiver units.

#### **5.14.14.3 Guidelines of receiver unit behavior for updating the root CA certificate of general purpose**

##### **(1) Confirmation of the Root Certificate Descriptor**

Receiver units check the existence of a Root Certificate Descriptor immediately after the selection of a station. When the Root Certificate Descriptor is received and if the `root_certificate_type` is 0, then it is judged as a root CA certificate of general purpose. After that, the “`root_certificate_id`” within the Root Certificate Descriptor is checked and if all of the “`root_certificate_id`” is 0xFFFFFFFF, then it is judged that there is no root CA certificates of general purpose to be stored, and succeeding processes will not be executed.

In case the receiver unit detects a “`root_certificate_id`” other than 0xFFFFFFFF, then it refers to the corresponding “`root_certificate_version`” to confirm the value is not 0xFFFFFFFF and the succeeding process is executed. If the “`root_certificate_version`” is 0xFFFFFFFF, then it is judged as an incorrect root CA certificates of general purpose, and succeeding processes will not be executed.

For the process in case the “`root_certificate_type`” = 1, refer to 5.14.14.4.

Confirmation of the root CA certificate is done immediately after the selection of a station or in case a process equivalent to re-selecting a station is executed (when the PID of the component that is currently viewed is changed, etc.), however, it should be noted that this is not executed in any other cases (when the data event of the component that is currently viewed is changed or a

pull back occurs). As for the performance of contents, in case a forced reacquisition of the root CA certificate is desired, it can be reacquired by executing “epgTune()” in the service that is currently viewed.

(2) Update confirmation of the root CA certificate of general purpose

The “root\_certificate\_id” of the Root Certificate Descriptor and the value of the root\_certificate\_version and the value of the root CA certificates information corresponding to this line are compared. This process is done for every “root\_certificate\_id” except for the detected 0xFFFFFFFF. \*1

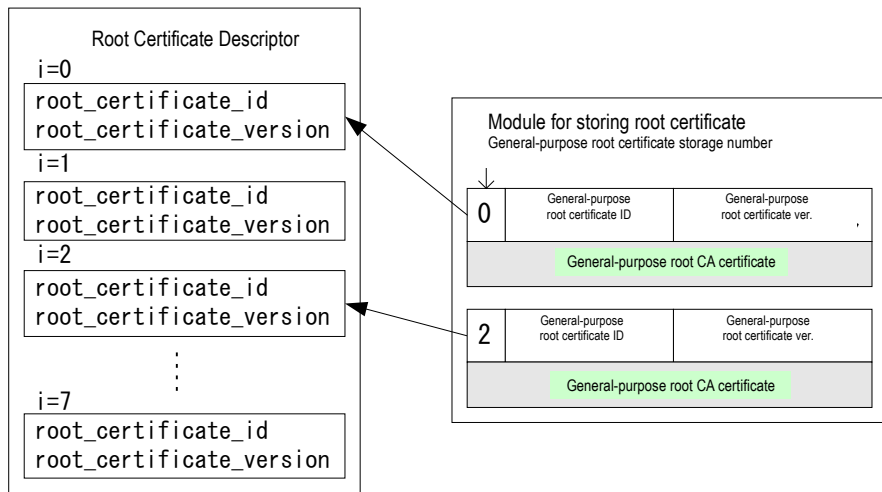
In case the contents of the Root Certificate Descriptor and root CA certificates information have the same values, it is judged as a certificate that is already stored, and the succeeding process is not executed.

In case a root CA certificate of general purpose is not stored in the specified storage area, or in case the Root Certificate Descriptor and root CA certificates information are different, proceed to the next process.

\*1 Up to two root CA certificates can be stored in the module for storing root CA certificates, therefore the maximum number of “root\_certificate\_id” that can be detected other than 0xFFFFFFFF is 2 at most.

(3) Authentication inspection of modules for storing root CA certificates

In case the contents of the Root Certificate Descriptor and root CA certificates information corresponding to the line stored within the receiver unit have different values, then the receiver unit will judge that a new root CA certificate has been transmitted and a module for storing root CA certificate is received. The “root\_certificate\_id “ and “root\_certificate\_version“ in the Root Certificate Descriptor corresponding to the root CA certificate storage number that is written within the module, and the root certificate ID and root certificate version number that are written within module are compared. In case they do not match, it is judged as an incorrect module for storing the root CA certificate and the succeeding processes will not be performed. For the details on the module for storing root CA certificates, refer to Vol. 6.



**Figure 5-10 The relationship between the module for storing root CA certificates and the Root Certificate Descriptor**

(4) Storage of root CA certificates of general purpose

In case the validity of the module for storing the root CA certificate is confirmed, the corresponding root CA certificate of general purpose will be stored in the storage number of the memory area for root CA certificates which is written in the module for root CA certificates. Receiver units will store the root CA certificate of general purpose, and at the same time, the root certificate version written in the module for root CA certificates are stored in the storage number of the memory area of root CA certificates.

**5.14.14.4 Guidelines of receiver unit behavior when receiving root CA certificates of limited purpose by broadcasters**

Broadcasters can operate independent root CA certificates. In case the “root\_certificate\_type” of the Root Certificate Descriptor is 1, then the receiver unit will handle the root CA certificate transmitted by the carousel as a root CA certificates of limited purpose by broadcasters. In case of operating a root CA certificates of limited purpose by broadcasters,, broadcasters can use a root CA certificate of general purpose in addition to the root CA certificate of limited purpose by broadcasters

(1) Confirmation of the Root Certificate Descriptor

Receiver units check the existence of a Root Certificate Descriptor immediately after the selection of a station. In case the “root\_certificate\_type” of the Root Certificate Descriptor is 1, then the receiver unit will judge it as a root CA certificate of limited purpose by broadcasters and the succeeding process is performed.



(2) Confirmation of the validity of the module for storing the root CA certificate

Receiver units receive modules for storing root CA certificates, and if the first character of the storage number of the root CA certificates of limited purpose by broadcasters stored in the module for root CA certificate is '\_', then it is judged as a correct module, and the root CA certificates of limited purpose by broadcasters is stored and it proceeds on to the next process. If it's other than '\_', then the receiver unit will judge it as an incorrect module for storing root CA certificates, and the succeeding process will not be performed.

(3) Storage of root CA certificates of limited purpose by broadcasters

After validity of root CA certificates of limited purpose by broadcasters is confirmed, the receiver unit will store it temporarily. In case of encrypted communication, the root CA certificate of limited purpose by broadcasters and root CA certificate of general purpose are referred to.

Since the attribute of the root CA certificate of limited purpose by broadcasters is determined by the "root\_certificate\_type" in the Root Certificate Descriptor in DII, referring of a carousel is necessary. For this purpose, it can be used only in a data broadcasting reception status or linked status. The root CA certificate of limited purpose by broadcasters is effective only within the service.\*1

(4) Discarding of root CA certificates of limited purpose by broadcasters

As soon as the performance of selecting a station or transition to an unlinked status of the receiver unit occurs, the stored root CA certificate of limited purpose by broadcasters is no longer valid.

## 5.15 Operation of the bookmark service

The bookmark service is operated. Regarding the operation of the bookmark service, refer to ARIB STD-B24 Vol. 2 Appendix 1.

### 5.15.1 Service type of the bookmark list service

As a basic rule, the bookmark list service is broadcast as a service type of the "bookmark list service" (service\_type=0xAA). However, broadcasters who do not have a license for (independent) data services may broadcast the bookmark list contents in other the service type in some cases.

### 5.15.2 Operation of the pre-list

Prelist by multi-media contents is not operated as a basic rule. However, this is not to

interfere with broadcasting of the prelist by the desired broadcaster. In case of operation, the statement regarding the prelist in ARIB STD-B24 Vol. 2 Appendix 1 should be followed.

### 5.15.3 Registration by bookmark type

- Registration of broadcaster individual operation type  
Registration by bookmark types of new broadcaster individual operation type by broadcaster. The operation will be specified separately.

### 5.15.4 Bookmark list display function by receiver's native application (Optional)

The bookmark list display service is assumed to be realized in contents, however, considering the convenience of viewers, the receiver unit should have a bookmark list display function that is not dependent on broadcasting contents as a receiver's native application for the reasons listed below. Also, this function should be used by a simple operation such as by an exclusive key, etc. on a remote control. Regarding the operation of implementation of this function, refer to ARIB STD-B24 Vol. 2 Appendix 1.

The specification for writing time to Greg while selecting stations from the bookmark list by receiver features is written in ARIB STD-B24 Vol. 2 Appendix 1 "8.4.3.5 Operation of bookmark list service", however, if the receiver unit is in situation where it cannot acquire the time, then "000000000000" as a decimal number character string of 12 bytes indicating the time should be written.

### 5.15.5 General address definitions relating to the bookmark service

Address definitions relating to the bookmark service are specified as follows.

**Table 5-36 Address definitions relating to bookmarks**

Item	Address term
Bookmark service	Bookmark service
Bookmark key	Bookmark button

### 5.15.6 Operation of the expansion data section of the memory area for bookmark service

Among the expansion data section in the memory area for bookmark service, in case of omitting items that are specified as omissible, the following values should be written in the corresponding fields.

- In case of omitting variable length character strings, set 0 to the length byte.
- In case of omitting fixed length character strings, then 0x00 is written in all of them.
- In case of omitting numerical values of a fixed length, then 0 is written in all of them.

## 5.16 Operation of registration transmission

Registration transmission is assumed to be performed by instruction from viewers using

contents or receiver's native applications. The operation by contents is written in section 5.16.4, and the operation of registration transmission by receiver's native application is written in section 5.16.5. Registration transmission by receiver's native application is optional.

#### **5.16.1 Operation of the memory area for registration transmissions**

In order to store superimpose that could not be sent, receiver units will establish a memory area for registration transmission in NVRAM. Broadcasters can store superimpose to perform registration transmission in the memory area for registration transmission. For data to be written in the memory area for registration transmission, the specifications in section 5.16.3 should be followed.

#### **5.16.2 Identification of the memory area for registration transmission**

In case of performing reading/writing of information for the memory area for registration transmission from the Multimedia service, 1 variable length block should be regarded as one file and "readBookmarkArray()/writeBookmarkArray()" is executed. Reading/writing of information for the memory area for registration transmission from the Multimedia service is done in units of variable length blocks. The following URI is used to identify variable length blocks.

```
nvr:// denbun/<block number>  
<block number>:0-N (N is bigger than or equal to 2)
```

In case the extended function for broadcasting(readBookmarkArray(),writeBookmarkArray())is executed for the memory area for registration transmission after names other than above mentioned specification methods are specified, then the reading/writing of NVRAM is not performed and "readBookmarkArray()" will return null (failure) and "writeBookmarkArray()" will return NaN (failure) as return values.

#### **5.16.3 Operation of the bookmark function for registration transmission**

##### **5.16.3.1 Operation of the "writeBookmarkArray" function in registration transmission**

The performance of this function is in accordance with specifications of ARIB STD-B24, however, there is a limit for arguments in operation of registration transmission. The restricted matters are specified in Table 5-37. This function has a process to set the time that writing basic data section of registration transmission information is performed, however, if acquisition of the time fails, then writing returns a failure.

**Table 5-37 Argument operation of “writeBookmarkArray function “ in registration transmission**

Argument	Values to be set	Comments
filename	URI to identify the memory area for registration transmission is specified.	Regarding the identification of the memory area for registration transmission, refer to section 5.16.2.
title	Program name that had set the superimpose of the registration transmission is set	It is recorded in S:1V format, however, length is not specified in the argument.
dstURI	Server of the sending destination	It is recorded in S:1V format, however, length is not specified in the argument. In case of BASIC procedure, specification is not necessary.
expire_str	Expiration date of registration transmission superimpose is set.	YYYYMMDDHH 10 digits should be specified for the expiration date. If the digit number of the specification is not 10 digits, then the writing is regarded as a failure.
bmType	199 is specified.	EUC-JP character strings
linkMedia	0 is specified.	
usageFlag	0 is specified.	
extendedStructure	Format of registration transmission information is specified.	Refer to section 5.16.3.9.
extendedData	Registration transmission superimpose data is established.	

### 5.16.3.2 Operation of the “readBookmarkArray” function in registration transmission

The performance of this function is in accordance with the specifications in ARIB STD-B24, however, there is a limit for arguments in operation of registration transmission. The restricted matters are specified in Table 5-38.

**Table 5-38 Argument operation of the “readBookmarkArray” function in registration transmission**

Argument	Values to be set	Comments
filename	URI to identify the memory area for registration transmission is specified.	Regarding the identification of the memory area for registration transmission, refer to section 5.16.2.
bmType	199 is specified.	
extendedStructure	Format of the registration transmission information is specified.	Refer to section 5.16.3.9.

Among the return values of the “readBookmarkArray” function, Array[6] and Array[7] are operated. Expansion data is stored in Array[8].

#### **5.16.3.3 Operation of the “deleteBookmarkArray” function in registration transmission**

Performance of this function is in accordance with specifications in ARIB STD-B24, however, in operation of registration transmission, the URI specified in section 5.16.2 should be specified as the “filename” of the argument.

#### **5.16.3.4 Operation of the “lockBookmarkArray” function in registration transmission**

Performance of this function is in accordance with specifications in ARIB STD-B24, however, in operation of registration transmission, the URI specified in section 5.16.2 should be specified as the “filename” of the argument.

#### **5.16.3.5 Operation of the “unlockBookmarkArray” function in registration transmission**

Performance of this function is in accordance with specifications in ARIB STD-B24, however, in operation of registration transmission, the URI specified in section 5.16.2 should be specified as the “filename” of the argument.

#### **5.16.3.6 Operation of the “getBookmarkInfo2” function in registration transmission**

Performance of this function is in accordance with Appendix-8, however, the meaning of the return value is specified as the following in operation of registration transmission.

Array[0]: Number, all numbers of the implemented registration transmission area

Array[1]: Number, number of the remaining registration transmission superimpose that can be registered.

Array[2]: String, URI indicating the registration transmission area that can be newly registered.

#### **5.16.3.7 Format of registration transmission information**

For the maximum size of calling information stored in the registration transmission area, the total of the basic data section and expansion data section should be less than or equal to 1.5KB.

#### **5.16.3.8 Basic data section of registration transmission information**

The basic data section of registration transmission information should be in accordance with ARIB STD-B24 Appendix 1. Operation of registration transmission on its own is described below.

Argument of writing function (writeBookmarkArray()) and values specified by receiver units are written in the basic data section.

**Table 5-39 Operation of the basic data section of registration transmission information**

	Property	ECMA type	Writing type	comments
Basic data section	title	String	S:1V(* Maximum 41Bytes)	EUC-JP character string * including length
	dstURI	String	S:1V(* Maximum 61Bytes)	EUC-JP character string * including length
	expire	String	S:10B	YYYYMMDDHH
	registerDate	String	S:10B	YYYYMMDDHH (set by the receiver unit)
	bmLock	String	S:1B	1 alphanumerical character (set from exclusive API)
	bmType	String	S:3B	Fixed to 199
	linkMedia	String	S:1B	Fixed to 0
	usageFlag	String	S:1B	Fixed to 0

- Operation of titles (title)

Program name that performed the registration of registration transmission is written. This field should be written by contents at the time of registration of registration transmission information.

For titles, the maximum is 40 bytes. (Not including length) In case it exceeds 40 bytes, the receiver units will not store beyond the 41st byte. In case the 40<sup>th</sup> byte is the first byte of a double byte character, then the character should not be stored.

- Operation of a linked destination URI (dstURI)

For the URI of a linked destination, the maximum is 60 bytes. (Not including length) In case the sending destination is a server, then this field should be written by the contents.

In case it exceeds 60 bytes, the receiver units will not perform writing and NaN will be returned. In case of a Basic procedure, an empty string with a length of 0 should be specified.

- Operation of expiration dates (expire)

In expiration dates, the last date/time that registered registration transmission information will remain effective until is written. This field should be written by contents at the time of registering registration issue information.

- Operation of writing date/times (registerDate)

Absolute time at the time that the writing function of registration transmission information is called is written. In case the acquisition of time fails, then a writing failure will be returned.

- Operation of the deletion prohibition flag(bmLock)

This flag indicates that deletion is prohibited by instructions of the viewer.

0: If immediately after bookmark is recorded or “unlockBookmark()” is called, it is 0.

1: 1 is stored if deletion is prohibited by “lockBookmark()”

- Operation of each bookmark type(bmType)

Fixed value of 199 is written. The performance in case other values are set depends on a model of receiver units.

- Operation of linked media identification(linkMedia)

Fixed value of 0 is written. The performance in case other values are set depends on a model of receiver units.

- Operation of each permission type(usageFlag)

Fixed value of 0 is written. The performance in case other values are set depends on a model of receiver units.

#### **5.16.3.9 Expansion data section of registration transmission information**

The format of information to be stored in the expansion data section of registration transmission information is indicated in Table 5-40.

**Table 5-40 Format of data written in the memory area for registration transmission**

Property	ECMA Type	Writing Type	Comments
RegisterNetwork	Number	U:2B	Network ID of the service that performed the writing
RegisterService	Number	U:2B	Service ID of the service that performed the writing
EventInfo1	String	S:1V	Supplemental information related to the program
EventInfo2	String	S:1V	Supplemental information related to the program
Information	String	S:1V	Announcement to viewers
ConnectRequest	String	S:12B	Desired date and time for the calling YYYYMMDDHHmm
ConnectServiceStartTime	String	S:4B	Time to start accepting the calls HHMM
ConnectServiceEndTime	String	S:4B	Time to finish accepting the calls HHMM
ServiceUtilityArea	String	S:1V	Utility area
Interactive	Number	U:1B	Flag to indicate calls by instruction of the viewer is possible or not outside of reception hours.
ConnectType	Number	U:1B	In case of 0, then connection using "receiverISP" setup In case of 1, then connection using "connectPPP" In case of 2, then connection using "Basic procedure".
Tel	String	S:1V	*1*2*4*6
bProvider	Number	U:1B	*1*4*6
uid/hostNo	String	S:1V	*1*4*6 If "ConnectType" is 1, then the uid is set. If 2, then the "hostNo" is set.
passwd	String	S:1V	*1*5*6
nameServer1	String	S:1V	*3*5*6
nameServer2	String	S:1V	*3*5*6
softCompression	Number	U:1B	*1*5*6
headerCompression/ speed	Number	U:1B	*1*4*6 If the "ConnectType" is 1, then the "headerCompression" is set. If 2, then the speed is set.
timeout/idleTime	Number	U:3B	*1*4*6*7
Denbun	String	S:2V (Maximum of 1024Bytes)	EUC-JP Character strings

\*1 Required if "ConnectType" is 1.

\*2 Even if "ConnectType" is 1, in case of PPPoE, it can be omitted. In case of omission, empty strings should be set.

\*3 Even if "ConnectType" is 1, it can be omitted in case PPP driver can obtain value dynamically. Empty strings should be set in case of omission.

\*4 Required if "ConnectType" is 2.

\*5 If "ConnectType" is 2 and in case of String, then empty strings, in case of Number, then 0 should be set.

\*6 If "ConnectType" is 0, and in case of String then empty strings, in case of Number, then 0 should be set.

\*7 If "ConnectType" is 1, then timeout should be set, If 2, then idleTime should be set.



#### 5.16.3.10 Operation of each field of registration information

- RegisterNetwork  
The network ID of the service that performed the registration is written. Contents should fill in the network ID of their own service at the time of registering the calls.
- RegisterService  
The service ID of service that performed the registration is written. Contents should fill in the service ID of their own service at the time of registering the calls.
- EventInfo1  
Information related to the programs that performed registration. It can be used by contents displaying the list of registration transmissions to present information related to programs that performed registration for viewers. Writing in this field is recommended. In case it is not written in for unavoidable reasons, it should be an empty string with a length of 0. The maximum number of characters is 40 bytes and double byte codes should be used. Single byte code characters are not used.
- EventInfo2  
Separately from “EventInfo1”, information related to programs is written. In case it is not necessary, this field should be an empty string with a length of 0. The maximum number of characters is 40 bytes and double byte codes should be used. Single byte code characters are not used.
- Information  
Field that is used when a registered program wishes to communicate some sort of information to the viewers. Writing in this field is recommended. In case it is not written for unavoidable reasons, it should be an empty string with a length of 0. The maximum number of characters is 120 bytes and double byte codes should be used. Single byte code characters are not used.
- ConnectRequest  
The desired time for contents to call programs is written. In case the writing in this field is not necessary, then a fixed character string “FFFFFFFFFFFF” should be written.
- ConnectServiceStartTime  
Time to start accepting calls is written. It should be written in a 24 hour time format. In case the writing in this field is not necessary, then a fixed character string “FFFF” should be written. In case fixed character strings “FFFF” are set, then fixed character strings “FFFF” should be set in the ConnectServiceEndTime field as well.
- ConnectServiceEndTime  
Time to finish accepting calls is written. It should be written in a 24 hour time format. Time

that straddles over days can be written from ConnectServiceStartTime.

(Example: ConnectServiceStartTime : 23:00 / ConnectServiceEndTime : 05:00)

In case writing in this field is not necessary, then a fixed character string "FFFF" should be written.

- ServiceUtilityArea

The field to be used for passing parameters that should be set in registration transmission contents. In case writing in this field not necessary, this field should be an empty string with a length of 0. For the size of this field, the total of the basic data section of registration transmission information and expansion data section should be in a range not exceeding 1.5KB (1536 bytes).

- Interactive

Flag to indicate whether or not to accept calls by instructions from viewers outside of the reception hours for registration transmission indicated by "ConnectServiceStartTime" and "ConnectServiceEndTime". In case the value of this field is '1', it means that calls by instructions from viewers are accepted even outside of the reception hours for registration transmission. On the other hand, when it is '0', then it means that calls by instructions from viewers cannot be accepted outside of the reception hours for registration transmission. In case the value of this field is '0', and when a call is instructed by viewers outside of the reception hours for registration transmission then performance of calling should be restrained. Also, when fixed character strings "FFFF" are set in "ConnectServiceStartTime" and "ConnectServiceEndTime", then '1' should be set in this field.

- ConnectType

Specifies the type of calling function. In case it is 0, then connection is made by "connectPPPWithISPPParam" or automatic connection. In case it is 1, then connection is made by "connectPPP". In case it is 2, then it means the connection is made by basic procedure. In case of registering calls, this field should be written by contents.

- Tel

In case of using the "connectPPP()" function or "connect()" function, then specifying phone number should be written. In case of the "connectPPPWithISPPParams()" function or automatic calling, then it should be an empty string with a length of 0. The maximum number of characters is 30 bytes.

- bProvider

In case of using the "connectPPP()" function or "connect()" function, network specification identification should be written. In case of connecting by the "connectPPPWithISPPParams()" function or automatic connection, value 0 should be written.

- uid/hostNo

In case of using the “connectPPP()” function, then the user ID should be written, and in case of using the “connect()” function, the host number should be written. In case of connecting by “connectPPPWithISPParams()” or automatic connection, then values do not have to be written. In such case, it should be an empty string with a length of 0. The maximum number of characters is 64 bytes.
- passwd

In case of using the “connectPPP()” function, then the password for user authentication should be written. In case of connecting by the “connect()” function, “connectPPPWithISPParams()” function or automatic connection, then values do not have to be written. In such case, it should be an empty string with a length of 0. The maximum number of characters is 32 bytes.
- nameServer1

It should be written in case of using “connectPPP()”. In case of connecting by the “connect()” function, “connectPPPWithISPParams()” function or automatic connection, then values do not have to be written. In such case, it should be an empty string with a length of 0. The maximum number of characters is 15 bytes.
- nameServer2

It should be written in case of using “connectPPP()”. In case of connecting by the “connect()” function, “connectPPPWithISPParams()” function or automatic connection, then values do not have to be written. In such case, it should be an empty string with a length of 0. The maximum number of characters is 15 bytes.
- softCompression

In case of using the “connectPPP()” function, then a flag to indicate whether or not the compression of software is required or not should be written. In case it is compressed then ‘1’, and in case it is not compressed then ‘0’ should be set. In case of connecting by the “connect()” function, “connectPPPWithISPParams()” function or automatic connection, then values do not have to be written.
- headerCompression/speed

In case of using the “connectPPP()” function and in case it is compressed then ‘1’ should be set, and in case it is not compressed then ‘0’ should be set as a flag to indicate whether header compression is used or not. In case of using the “connectPPP()” function, the specification value of line speed to be connected should be written. In case of connecting by the “connectPPPWithISPParams()” function or automatic connection, then value 0 should be written.

- timeout/idleTime

In case of using the “connectPPP()” function, continuous time of no communication status (idleTime) before disconnecting should be written, and in case of using the “connect()” function, time to recognize it as timeout (timeout) should be written. In case of connecting by the “connectPPPWithISPPParams()” function or automatic connection then value 0 should be written.

- Denbun

Superimpose that should be sent is set. In case of using the registration transmission function, this should be written. The maximum byte number of superimpose is 1024 bytes and the character string should be EUC-JP.

#### **5.16.4 Guidelines of registration transmission operation by contents**

##### **5.16.4.1 Operation when registration transmission is set up**

Contents that set up registration transmission should store registration transmission information in the memory area for registration transmission upon consent from the viewers. Contents that perform registration transmission setup that depend on time elapse should write contents in order not to set up the registration transmission during the playback of recording.

Regarding the writing of registration transmission information, contents should use the “getBookmarkInfo2()” function to obtain the URI for the memory area for registration transmission and the existence of free space should be confirmed. In case free space is confirmed, registration transmission information can be stored in the URI obtained by the above mentioned function.

As a basic rule, new registration transmission information should not be overwritten on stored mamory areas for registration transmission.

In case there is no free space, writing can be done after deleting stored registration transmission information. The registration transmission information that can be deleted is as follows.

- Expired. In such case, the contents to be deleted should present expired registration transmission information to viewers \*1, and explicitly announce deletion due to expiration.
- The oldest among the registration transmission information without deletion prohibition flag. In such case, registration transmission information of deletion candidates is presented \*1, and contents to be deleted should be deleted upon consent for deletion from the viewers.

\*1 For the registration transmission information that should be displayed, refer to Table 5-41.

#### 5.16.4.2 Operation of registration transmission by contents

Contents that perform registration transmission should be equipped with the following functions.

- (1) Function to display a list of registration transmission information that has set stored registration transmission. (hereinafter referred to as, registration transmission information list function)

For the registration transmission information that should be displayed, refer to Table 5-41.

- (2) Function that allows viewers to select desired registration transmission from a registration transmission information list and to send superimpose to desired site of the registration transmission information.

However, in case expired registration transmission information is selected, contents should explicitly inform the viewer of expiration and superimpose is not sent out and the registration transmission information is deleted. The operation of deletion of registration transmission information refers to the following.

- (3) Delete management function for registration transmission information

Deletion management should have the following functions.

- Manual deletion of registration transmission information  
Function to delete desired registration transmission information by operation of the viewer.
- Automatic deletion of registration transmission information  
Registration transmission information can be automatically deleted without instruction from viewers in the following cases.
  - Expired registration transmission information. However, viewers should be explicitly informed of the deletion of corresponding registration transmission information at the time of deletion.
  - In case the communication of registration transmission information is a success.
- Deletion refusal specification of registration transmission information  
Function to turn ON/OFF the flag to delete the desired registration transmission information by operation of the viewer.  
In case the flag to prohibit deletion is ON, then it cannot be deleted even if it was expired. In such case, contents should explicitly inform viewers that even though it is expired, it cannot be deleted because the flag is ON.

**Table 5-41 Registration transmission information that should be displayed**

Fields	Meaning	Comments
title	Program name that has set the superimpose of registration transmission	
expire_date	Expiration date of registration transmission superimpose	
RegisterService	Service ID that has set the superimpose of registration transmission	Presentation of a service logo etc., is possible.
EventInfo1	Supplementary information regarding programs - Part I	
Information	Announcement to viewer	
ConnectRequest	Desired time for calling	
ConnectServiceStartTime	Desired time to start accepting calls	
ConnectServiceEndTime	Desired time to finish accepting calls	

### **5.16.5 Registration transmission using registration transmission information by receiver's native application (Optional)**

Registration transmission information can be realized by receiver's native applications as well. In this section, requirements that registration transmission function should fulfill are explained.

#### **5.16.5.1 Registration of registration transmission information**

Registration transmission information is assumed to be registered by contents. For details, refer to section 5.16.4.1.

#### **5.16.5.2 Operation of registration transmission by receiver's native application**

Registration transmission function by receiver's native application basically depends on a model. An example of a registration transmission function by receiver features is shown in this section.

Registration transmission function by receiver features has the following functions.

- (1) Function to display a list of registration transmission information that has set stored registration transmission. (hereinafter referred to as registration transmission information list function) Registration transmission information that should be displayed is in accordance with Table 5-41.
- (2) Function that allows viewers to select the desired registration transmission from the registration transmission information list and to send superimpose to the specified site of

the registration transmission information.

However, in case expired registration transmission information is selected, the receiver's native application should explicitly inform the viewer of expiration and the superimpose is not sent out and the registration transmission information is deleted. For the operation of deletion of the registration transmission information, refer to the following.

(3) Delete management function of registration transmission information

Deletion management should have the following functions.

- Manual deletion of registration transmission information

Function to delete the desired registration transmission information by operation of the viewer.

- Automatic deletion of registration transmission information

Function to delete registration transmission information automatically in case of the following.

- Expired registration transmission information. However, viewers should be explicitly informed of the deletion of corresponding registration transmission information at the time of deletion.

- In case the communication of registration transmission information is a success.

- Deletion refusal specification of registration transmission information

Function to turn ON/OFF the flag to delete the desired registration transmission information by operation of the viewer.

In case the flag to prohibit deletion is ON, then it cannot be deleted even if it has expired.

(4) Startup of the registration transmission function by the bookmark button

For receiver units equipped with the bookmark list function by receiver's native application, it should be able to be started by the bookmark button. In such case, both the bookmark list function and registration transmission information list function should co-exist by the display switching function by receiver's native application.

(5) Receiver units that have receiver's native applications providing the registration transmission information list function should display the registration transmission information list function application and exclusive process on display of BML in order to avoid competition with the registration transmission information list function by contents.

### 5.16.5.3 Reservation transmission function by receiver features

The calling function of registration information by receiver's native applications can have the reservation calling function, which makes calls automatically within the specified range of date/time by using information of "connectRequest", "connectServiceStartTime", "connectServiceEndTime" fields in registration transmission information.

Performance of reservation calling depends on a model of receiver units, however, the calling should be performed within the time range specified by "connectServiceStartTime" and

“connectServiceEndTime” by the date specified by “connectRequest”. If the call can be made close to the time that was specified by “connectRequest” field, then it is better.

Regarding the process when the sending of superimpose fails, or the number of retry times and time accuracy depend on models.

#### **5.16.5.4 Guidelines for sending messages by receiver’s native application**

Guidelines for receiver unit performance in case of sending superimpose of receiver’s native application as instructed by the viewer is as follows.

(1) Judgement of whether or not sending of superimpose is possible

- Check the “ExpireDate” and “bmType” and check whether or not they are valid registration transmission information. In case they are valid registration transmission information, then proceed on to the next process. In case they are invalid registration transmission information, a message to notify that they cannot be sent because the data has problems is presented to viewers and after deleting invalid registration transmission information, the process should be end.
- Check the “Interactive” field in the expansion data and if it is 0, then proceed on to (2). If it is 1, then proceed on to the following process.
- Check the range of “ConnectServiceStartTime” and “ConnectServiceEndTime” and check if the absolute time of the receiver unit is within the corresponding range. If it is within the range, then proceed on to (2). If it is out of the range, then a message to prompt calls before the expiration date should be presented and finish the process.

(2) Identification of the superimposeending method

- Check “ConnectType” in the expansion data and identify the sending method. In case of receiver units with equipped modems, proceed on to the next process. If “ConnectType” is 1 or 2 in case of receiver units without modems, then a message to notify that sending cannot be performed to the viewers should be presented and finish the process.
- Check whether or not the identified sending method can be used. If it can be used, then proceed on to the next process. If it cannot be used, then a message to notify that sending cannot be performed now to the viewers should be presented and finish the process.
- In case the “ConnectType” is 0, then proceed on to the process of (3). If the “ConnectType” is 1, then Tel, bProvider, uid/hostNo, Passwd, nameServer1, nameServer2, softCompression, headerCompression/speed, timeout/idleTime fields are read and proceed on to the process of (3) as the connection parameter. If the “ConnectType” is 2, then read Tel, bProvider, uid/hostNo, headerCompression/speed fields and proceed on to the process of (3) as the connection parameter.

(3) Sending of superimpose

- Refer to “denbun” field in the expansion data and read the superimpose data and proceed on to the next process.



- In case the “ConnectType” is 0, then refer to the “dstURI” field of the basic data section, and for the specified server, superimpose data is sent via TCP/IP protocol using receiver unit ISP setup. In such an event, the sending format should be a format in accordance with the transmitTextDataOverIP() function in section 5.14.6.6. Performance in case sending fails or performance in case reading of superimpose data fails depend on models.
- In case the “ConnectType” is 1, connect to the TCP/IP network by parameters read in (2) by performance equivalent to the connectPPP() function indicated in section 5.14.6.6. However, the handling of values specified in “timeout/idleTime” field depends on a model of receiver units. Regarding the sending format of superimpose, it is the same as when the “ConnectType” is 0. Performance when the sending fails or performance when the reading of superimpose data fails depends on a model of receiver units.
- In case the “ConnectType” is 2, communication is established by parameters read in (3) by performance equivalent to the “connect()” function of basic procedures in section 5.14.6.5. After that, superimpose data is read from the “denbun” field and superimpose data is sent by performance equivalent to the “sendTextData()” function. The handling of return values from the interaction channel center of receiver units does not have to be considered. In other words, receiver units do not have to guarantee the performance equivalent to “receiveTextData()”. Also, the handling of values specified by “timeout/idleTime” field depends on a model of receiver units. Performance in case sending fails or performance in case reading of superimpose data fails depends on a model of receiver units.

## 6 Operation of extended service (Optional)

Should be in accordance with ARIB STD-B24 Vol. 2 Appendix 3 “Optional guidelines for implementing extended services for fixed receiving system”.

### 6.1 Display of HTML contents (Optional)

To play HTML contents from BML contents by starting the internet browser, “Extended function like startResidentApp()” or “launchExApp()” is used. ( “startResidentApp()” is recommended.)

Internet browsers started by “launchExApp()” shall be in accordance with ARIB STD-B24 Vol. 2 Appendix 1 section 8.5.2., and display of HTML contents transmitted by broadcasting (In other words, HTML documents with URI’s starting with “arib-dc://”) shall be possible. It is implementation dependent if internet browsers started by “startResidentApp()” may support the display of transmitted HTML contents, or not.

details on the method to start internet browsers by “launchExApp()” shall be follow ARIB STD-B24 Vol. 2 Appendix 1 section 8.5.2.

It is implementation dependent if BML browsers might be terminated by starting internet browse,or not

Regarding the presentation of internet browsers, the contents written in Vol. 2 “9.3 Guarantee of uniqueness of broadcasting programs and contents” should be considered.

### 6.2 Print function

Regarding functions and specifications relating to the print function, they should be in compliance with ARIB STD-B24 Vol. 2 “7.6.17 Functions for printing” and Vol. 2 Appendix 1 “Guidelines on functions for printing”.

#### 6.2.1 Expansion API group

Since the print function is an implementation option, before execution of print related functions,, broadcaster should check by “getBrowserSupport()” if related processes in receiver is possible and should call print related functions only when the process is possible.

Print related functions are classified into the following 2 groups.

A) Function group to print with printers:

- getPrinterStatus()
- printFile()
- printTemplate()
- printUri()
- printStaticScreen()

B) Function group to store printing data into memory cards

- saveImageToMemoryCard()
- saveHttpServerImageToMemoryCard()
- saveStaticScreenToMemoryCard()

Either A) or B), or both A) and B) can be supported by receiver. Detection of the function implementation status of A) or B) of receiver units is executed by specifying “APIGroup” to “functionname” of “getBrowserSupport()”. The extended Functions Group specification that specifies “additionalinfo” in case “functionname” is “APIGroup”, refer to section 5.9.6.6.

### 6.2.2 Print data format

Print data (XHTML-Print) should be in compliance with ARIB STD-B24 Vol. 2 Appendix 1 “Guidelines on functions for printing”. However, still image files (JPEG, PNG) referred from XHTML documents for printing, they may be shared by presentation and print in BML. In some cases, they may be exclusively for printing in other cases. Rules below should be applied in each respective case.

	Image file	Operation
In case of sharing with display and print.	JPEG	Refer to 3.2.1.
	PNG	Refer to 3.2.2 and ARIB STD-B24 Vol. 1 Part 2 Annex B However, when the color type is 3, then the PLTE chunk may not be omitted.
In case of exclusive printing	JPEG	ISO/IEC10918-1 baseline and JFIF(Jpeg File Interchange Format) and Exit The maximum picture element size is 2560x1920 Maximum file size 3MB* Maximum picture element count 5Mpix Sampling 4:2:0 or 4:2:2 Picture element aspect ratio 1:1(dimetric picture element)
	PNG	Compliance with W3C PNG Specification Ver1.0. However, file type should be “0” and interlace should be “0”. Maximum picture element size 2560x1920 Maximum file size 3MB* Maximum picture element count 5Mpix Picture element aspect ratio 1:1(dimetric picture element)

\* In case of using printFile(), printTemplate(), the maximum size of the module total is limited to 1MB.

Regarding the size restriction of print data (XHTML-Print)is as follows.

API	Maximum size of XHTML-Print (Including CSS)	Total of XHTML-Print documents and referred still image files(Maximum size of 1 content)
printFile() printTemplate() printStaticScreen()	256KB	1MB(Total module size)
printUri()	256KB	Not specified

### 6.2.3 Supplementary information regarding each print related API

- The number of modules that can be specified in “printFile()”, “printTemplate()”  
The maximum number of modules that can be specified by these functions is 8.
- Image file format that can be specified in “saveHttpServerImageToMemoryCard()”  
The file format of images that can be specified with this function is either JPEG or PNG specified in the table above. The extensions of the file names is either “jpg” or “png” respectively. The maximum size of image files is specified in the table above as well.
- Specifications in this volume for document resolution of data broadcasting at the time of “printStaticScreen()” or “saveStaticScreenToMemoryCard()” execution  
This function can be used when document resolution of data broadcasting is 960x540. α composite between planes is not required. Also, the video plane is not rendered.
- Prescriptions regarding URI in printUri()  
The URI specified in printUri() shall be a resource in the form of http:// or https:// and less than 256 bytes in size. The URI of the image file referenced from XHTML-Print shall be less than 128 bytes in size.  
Contact the governing organization described in [Section 4] Appendix 13 regarding the root certificate for performing https communication in the service for printing the printable content that have been placed on the server using printUri().
- URI's specified by “saveHttpServerImageToMemoryCard()”  
For URI's specified by “saveHttpServerImageToMemoryCard()”, only resources indicated with http:// or https:// can be specified and it should be less than or equal to 256 bytes.
- Regarding the process for the function return values of “printFile”, “printTemplate”, “printStaticScreen”, “printUri”  
When printing is started, the printer might be out of paper,the receiver might not return proper return value indicated out of paper in submission time, because the printer are not implemented paper sensor as an implementation dependent.In this case, the return value of the function shall be 1: success. After that, the printer will notice about the out of paper

status during processing period. Print errors, etc. during processing period may not be reflected by the return value of the function. Because those errors are handled by the receiver or printer. If printing is in pending status due to running out of paper during processing period, the printer that has executed printing API such as printFile may be in the following status.

- a) Printing job is pending in the printer and the buffer is full and out of paper.
- b) Buffer is not full, but out of paper.

The status of a) is equivalent to both the API return values of -3 and -4. In other words, whether return value will be -3 or -4 in such case implementation dependent in the features of the receiver and printer.

In order to correspond to the above cases, as a function process on the calling side for the return values of -3 and -4, it should be handled in the same way as the “printer busy status”.

#### **6.2.4 Regarding presentation by receiver**

Just as in the example indicated below, messages displayed by the receiver and user interface can be displayed over broadcasting screen. The presentation of messages by the receivers is implementation dependent, but the size of the display area on the screen should be as small as possible upon securing of the convenience of the users. Receivers do not present user interface prompting confirmation for the user operation within the function.

- (1) All print related functions are synchronous functions, possibly implementation to connect with printers within function is assumed. In that case it could take much time, therefore, dialog windows to be currently in communication and to abort execution may be displayed. However, the receivers should delete the above mentioned dialog window when function is returned.
- (2) While executing the printing function, if tuning, etc. is done by the user, then the receiver may terminate the execution of the function to tune, or the tuning may be pending and a user interface to confirm whether or not the printing process should continue may be displayed.
- (3) For events that occur asynchronously in printers (paper jam, out of ink, etc.), from the perspective of user convenience, a message should be displayed.

In order to prevent printing by an incorrect operation of the user, presentation to confirm the print execution is done at the time of executing the print functions (printFile(), printTemplate(), printUri(), printStaticScreen()) function. Receivers should provide a user interface that performs setup of printing conditions (changes of paper size, paper type, number of prints, etc.) separately from the behaviour of BML contents.

Also, at the time of executing store functions of print data to memory cards (such as `saveImageToMemoryCard()`, `saveHttpServerImageToMemoryCard()`, `saveStaticScreenToMemoryCard()`), BML contents should provide messages prompting the insertion of a memory card to users and should correspond to the case file names are duplicated (contents that users can change or input file name by themselves). Directory in storage device is implementation dependent.

In case a data event is updated, even if the receiver displays a message, etc. in the print function, current document shall be terminated and re-presentation shall be performed by browser.

In the following cases, the receiver will delete above messages displayed in the print function (excluding the above mentioned (3)) and the user interface and printing is aborted.

- When an update of a data event has occurred in the ES being presented.
- When the tuning is executed.

If `http://`(or `https://`) is specified in “`saveHttpServerImageToMemoryCard`”, the function can take some time from function call to since the receiver will acquire print data via network. If some event occurred while the receiver is acquiring print data, the event is put in an interrupting queue, however, it is not executed until function is returned from the function; this should be taken into consideration.

### **6.3 Broadcasting integrated IPTV service type 1 (optional)**

#### **6.3.1 Activation and display of the IPTV browser**

IPTV service can be launched by activating IPTV browser<sup>1</sup> on the receiver from BML content, using `startExtraBrowser()` with its argument `browserName` set to a character string representing the browser name as shown in Table in Appendix-14 (1). Some constraints may have been set by the IPTV service operator on the value of argument `uri` which indicates the first content to be presented when the IPTV browser is launched. The value of argument `uri` shall therefore be specified according to the guidelines set by each IPTV service operator. Whether or not the BML browser will be terminated upon activation of the IPTV browser shall be implementation dependent. However, if the BML browser will not be terminated, care must be taken to ensure that the presentation will not be mixed up. In presenting programs and contents in the IPTV browser, the content described in Vol.2 “9.3 Guarantee of Uniqueness of Broadcasting Programs and Contents” shall be taken into consideration.

---

<sup>1</sup> A browser for presenting the IPTV service defined by the IPTV Forum.

### 6.3.2 Returning to BML browser from IPTV browser

Argument returnURI of startExtraBrowser() is a hint information for returning to the BML browser from IPTV browser. However, if a return destination has been specified to this argument, it shall be desirable to present the startup document of the specified component (if possible). The period for which the argument information will be stored on the receiver shall be implementation dependent.

### 6.3.3 Confirmation of the IPTV browser function from BML content

The getBrowserSupport() can be used to confirm whether IPTV browser is implemented on the receiver from BML content, by specifying the character string for when functionname is ResidentApp in Table in Appendix-14 (2) as argument.

If the return value of this function is 1, it indicates that the relevant IPTV browser can be launched using startExtraBrowser() from the BML browser. It shall be kept in mind that getBrowserSupport() will return 0 if the IPTV browser is implemented on the receiver but cannot be activated from BML content.

Similarly, functions of the IPTV browser on the receiver can be confirmed using the getBrowserSupport() function, by specifying the character string for when functionname is ExtraBrowserFunction in Table in Appendix-14 (2) as argument.

The getResidentAppVersion() function can be used instead to confirm whether any IPTV functions provided uniquely by the service operator have been implemented. To confirm whether any IPTV functions provided uniquely by the service operator have been implemented, a character string representing the browser of the relevant service operator as shown in Table in Appendix-14 (3) shall be specified to argument appName.

### 6.3.4 Return values of the getResidentAppVersion() function

Refer to Appendix-14 (4) for the character strings to be used as item names in the return value Array[4] of this function. The character string equivalent to UserAgent that can be placed in the final field of Array[4] is not defined in this document. However, it shall desirably be the same as the UserAgent string of the IPTV browser.

### 6.3.5 Positioning of the IPTV activation function startExtraBrowser()

Although the implementation of startExtraBrowser() is optional, it shall be desirable that a receiver implementing an IPTV browser that can be specified to argument browserName implements this function as well.

On a receiver implementing multiple IPTV browsers, which IPTV browser(s) can be launched from this function shall be implementation dependent.

#### **6.4 Broadcasting integrated IPTV service type 2 (optional)**

See Chapter 7 for the broadcasting integrated IPTV service type 2 function handled in BML documents.

#### **6.5 External device linkage function (optional)**

The external device linkage function is used to display information that is associated with broadcasting on an external device screen by connecting to an external device, such as a tablet or smartphone. When linking (the BML document of) a receiver to an external device using this function, the extended function “transmitDataToSmartDevice()” is used. There are multiple service profiles that are used by the external device linkage function depending on the service contents to be provided. Whether a receiver supports each profile can be confirmed by means of getBrowserSupport() designating "SmartDeviceProfile" and one of the strings shown in Table 6.5.1 to functionname and additionalinfo, respectively.

Regarding the connection formed from an external device to a receiver, while these guidelines have not yet established detailed provisions, it is assumed that the receiver receives data with the same operation as that when receiving event messages. (See the text in the box at the end of the section.)

##### **6.5.1 Service profile and the operation**

The table below shows strings for specifying a service profile and a service overview. The service profile can be operated under a registration system. Contact the governing organization shown in Appendix 13, Section 4, Vol. 3 regarding the registration details.



Character string to specify a service profile	Service overview
SmartDeviceMode1	Transfers URI information to a receiver with no terminal specified by the contents. The receiver simultaneously sends the URI information to all external devices that can be connected. The external devices display the information based on the received URI information.
SmartDeviceMode2	Transfers an arbitrary character string to a receiver with no terminal specified by the contents. The receiver simultaneously sends the character string to all external devices that can be connected. It is assumed that the external devices interpret the received string and proceed with operation including starting an application and displaying the information.

#### (1) Operation of SmartDeviceMode1

When SmartDeviceMode1 is used, the following service is expected to be provided: A receiver transmits URI information to (an) external device(s), and the external device(s) obtain the information corresponding to the received URI information via communication, etc., and then displays the information. This service does not include sending information from (an) external device(s) to a receiver.

However, the scope specified in these guidelines for this operation is limited to tasks up to passing a character string from the BML documents to a receiver function. The expected operation after that, including establishing connection to (an) external device(s), shall be determined based on the product planning of receivers and external terminals.

A function to manage the connection between a receiver and (an) external device(s) (pairing) shall be also a matter of product planning. If there are multiple external devices that are paired with a receiver, URI information is assumed to be simultaneously sent to all paired external devices with no terminal specified. However, a receiver can limit external destination devices by being specified by the user, etc.

As URI information, contents that can be displayed on a general browser can be specified. The contents are not limited to html documents.

- Operation of transmitDataToSmartDevice()

- Operation of the argument

The character string specified with the argument of "profile" is "SmartDeviceMode1".

A character string that can be specified by the argument of "data" is a URI character string defined by RFC2396 (or RFC3986). NaN is returned if any character string other than a URI character string is specified.

additionalinfo is not used as the argument.

- Operation of return values

1 is returned as the return value if data is successfully transmitted. However, as the interpretation of "successful data transmission" is a matter of product planning, it can be regarded that data is successfully transmitted at least if data is transferred to the external device linkage function of the receiver. Therefore, it should be noted that it is not guaranteed that an external device has actually received data of the contents even if the return value is 1.

The table below shows the return values when data transmission has failed. If data transmission has failed due to the situations corresponding to -1 or -2 in the table below, NaN can be returned without returning the specified values. If -1 or -2 is returned, the contents can display the appropriate messages to the user.

Return value	Status
-1	A paired terminal does not exist.
-2	Failed to communicate with a terminal.
NaN	Failure due to other reasons

## (2) Operation of SmartDeviceMode2

When SmartDeviceMode2 is used, the following service is expected to be provided: A receiver sends an arbitrary character string to (an) external device(s), and the external device(s) interpret the received string and proceed with operation including starting an application and displaying the information. Whereas these guidelines do not specify the operation of character strings, service providers that provide the service using the profiles in these guidelines based on the service to be realized using external devices shall determine such operation. This service does not include sending information from an external device to a receiver.

However, the scope specified in these guidelines for this operation is limited to tasks up to passing a character string from the BML documents to a receiver function. The expected operation

after that, including establishing connection to (an) external device(s), shall be determined based on the product planning of receivers and external terminals.

A function to manage the connection between a receiver and (an) external device(s) (pairing) shall be also a matter of product planning. If there are multiple external devices that are paired with a receiver, an arbitrary character string is assumed to be simultaneously sent to all paired external devices with no terminal specified. However, a receiver can limit external destination devices by being specified by the user, etc.

The service providers that provide service using the profiles in these guidelines shall determine how an external device receives a character string, how to provide the service, and how to implement the service.

- Operation of `transmitDataToSmartDevice()`

- Operation of the argument

The character string specified to the argument of "profile" is "SmartDeviceMode2".

Any character string can be specified to the argument of "data".

The argument of "additionalinfo" is not used.

- Operation of return values

1 is returned as the return value if data is successfully transmitted. However, as the interpretation of "successful data transmission" is a matter of product planning, it can be regarded that data is successfully transmitted at least if data is transferred to the external device linkage function of the receiver. Therefore, it should be noted that it is not guaranteed that an external device has actually received data of the contents even if the return value is 1.

The table below shows return values when data transmission has failed. If data transmission has failed due to the situations corresponding to -1 or -2 in the table below, NaN can be returned without returning the specified values. If -1 or -2 is returned, the contents can display the appropriate messages to the user.

Return value	Status
-1	A paired terminal does not exist.
-2	Failed to communicate with a terminal.
NaN	Failure due to other reasons

(Reference information) Data reception from an external device using the external device linkage function

For expected future operation (profiles) in which connection with external devices is established more closely than that using SmartDeviceMode1 or SmartDeviceMode2, interactive connection in which external devices transmit data to BML contents is expected to be created.

The following operation is assumed for obtaining data from BML contents when a receiver receives text data from an external device. However, it should be noted that the description in this box is just for reference and needs to be reviewed when it is actually applied.

- When data is associated from an external device with BML contents

When a receiver acquires data from an external device, an interruption by EventMessageFired shall occur. The following values shall be specified to acquire the interruption so as to “subscribe” the interruption in the same way as the operation of an ordinary general event message.

- es\_ref="arib-dc://-1.-1.-1/-1"
- event\_msg\_group\_id=0x001
- message\_id=200(T.B.D.)

The text data sent from an external device is acquired by accessing “private\_data\_byte” when an interruption occurs. “message\_version” shall be updated every time new data is acquired from an external device. However, the initial value is not specified in particular.

It is necessary to employ a method to allow events to occur in the target BML contents when a receiver receives data from an external device in order to prevent the BML contents from accidentally receiving data from a content that is not to be linked. Therefore, when sending data, a service ID, broadcaster ID, or group ID shall be specified in the contents on the external device side. A receiver needs to implement a filtering function to discard data if the received ID is different from the service ID, broadcaster ID, or group ID of the service that is being provided.

A method to realize data transmission from an external device to a receiver and the implementation method shall be a matter of product planning.

## 6.6 AIT-controlled application linkage function (optional)

See Chapter 8 for the AIT-controlled application linkage function used in BML documents.

## 6.7 AIT-controlled application external activation function (optional)

This is the function to activate AIT-controlled application directly by an external device such as a smartphone. A receiver is required to support AIT-controlled application linkage function when implementing the function.

Execution of this function by an external device shall comply with IPTVFJ STD-0010 "IPTV Standard: Integrated Broadcast-Broadband System Specification"

### **6.7.1 Operation at the time of AIT-controlled application external activation**

When AIT-controlled application external activation is accompanied with channel selection, AIT-controlled application with external activation starts before judging activation priority for data broadcasting and AIT-controlled application according to **the** activation priority information in PMT after channel selection.

AIT-controlled application external activation without channel selection terminates data broadcasting and AIT-controlled application, and then starts.

### **6.7.2 Operation at the time of termination of AIT-controlled application with external activation**

When AIT-controlled application with external activation is terminated, activation process is executed in accordance with the activation priority information in PMT

## **7 IPTV download/VOD (integrated service type 2) function**

### **7.1 Overview of the IPTV function used in BML documents**

This chapter provides the specifications to realize the IPTV function in BML documents. The IPTV function refers to a function to realize on-demand communication service (IPTV download/VOD). IPTV download is also a function to process on-demand content download via an IP network communication.

As services that are realized by integrating the data broadcasting and IPTV service (referred to as the "broadcasting integrated IPTV service"), the following two integration types are defined.

In the integrated service type 1, IPTV services are provided by starting the IPTV browser installed in a receiver from the BML using the "startExtraBrowser( )" function, as specified in 6.3 "Broadcasting integrated IPTV service type 1 (optional)". Generally, the data broadcasting browser and IPTV browser are exclusively displayed.

In the integrated service type 2, the BML directly executes the IPTV function (IPTV download/VOD), and realization of the IPTV services is allowed as part of a data broadcasting function. This chapter specifies the IPTV download/VOD function that is used to provide the broadcasting integrated IPTV services of this integrated service type 2.

The IPTV download/VOD function (integrated service type 2) can be optionally implemented in a receiver.

#### **7.1.1 Overview of the IPTV download/VOD (integrated service type 2) function**

##### **7.1.1.1 IPTV download function (integrated service type 2)**

The IPTV download is a function to temporarily store the contents forwarded from a server in the storage media in a receiver in order to allow for operation (such as replaying) of the stored contents using the receiver. The role of BML documents in the IPTV download function is to make users select a content to download by presenting downloadable contents and to instruct the receiver to download the selected contents. The IPTV download function can be used in the broadcasting status and linked status.

The content download is processed separately from the broadcasting reception and data broadcasting reception. The downloaded contents are managed by the ECG of the receiver and replayed, etc., according to the user's instructions.

### **7.1.1.2 VOD function (integrated service type 2)**

The VOD function is used for full-screen display of VOD contents provided by the VOD service. The VOD function controls the replay of the VOD contents by directly controlling the AV player using a remote controller.

The VOD contents display can be started only in the linked status. The display of VOD contents can finish when the contents end or with the stop operation of a remote controller. After finishing the display of the VOD contents, the BML documents that are specified when starting the VOD contents are acquired, and the VOD replay stop position and the stop status are returned to the server in order to resume the normal linked status.

Whether the BML browser is operated or stopped while replaying the VOD is determined by the product planning policy of receivers. However, in either case, the replay of VOD contents will not be affected by the BML contents.

## **7.1.2 Internet scope and CDN scope**

Two main scopes are assumed for IPTV services in these guidelines. One is called the Internet scope, which is a type of IPTV service provided based on "IPTV Standard Internet Scope Service Approach Specifications". In 7.2, the operation when providing this type of IPTV service through broadcasting service is specified. The other is called the CDN scope, which is provided based on "IPTV Standard CDN Scope Service Approach Specifications". In 7.3, the operation when providing this type of IPTV service through broadcasting service is specified.

## **7.2 IPTV download/VOD (integrated service type 2) function in the Internet scope**

### **7.2.1 Specification range**

In 7.2, the BML specifications to use the IPTV download/VOD (integrated service type 2) function in the Internet scope. Matters related to IPTV services are specified as IPTV Forum technical specifications. (See Chapter 2, Section 1 of this volume.)

In the service realized using the integrated service type 2, broadcasters may directly provide IPTV services or they may realize services using equipment and service models of IPTV service providers.

### **7.2.2 Functions to be implemented in receivers that support the IPTV download/VOD (integrated service type 2)**

This section describes the functions that should be implemented in receivers that support the IPTV download/VOD (integrated service type 2).

It is desirable that receivers that support the IPTV download or VOD (integrated service type 2) also support the provisions in 6.3 "Broadcasting integrated IPTV service type 1 (optional)".

### **7.2.2.1 Classification of the functions of receivers that support the IPTV download/VOD (integrated service type 2)**

The functions of receivers that support the IPTV download or VOD (integrated service type 2) can be classified into the following three types.

- (1) IPTV download/VOD common function (See 7.2.2.2.)
- (2) IPTV download function (See 7.2.2.3.)
- (3) VOD function (See 7.2.2.4.)

The receivers that support the IPTV download (integrated service type 2) shall implement the functions described in (1) and (2). The receivers that support the VOD (integrated service type 2) shall implement the functions described in (1) and (3).

The IPTV download function and VOD function are handled as individual options. Therefore, only one of these can be implemented.

### **7.2.2.2 IPTV download/VOD common functions**

#### **7.2.2.2.1 Remote controller**

It is desirable that the receivers that implement the IPTV download/VOD (integrated service type 2) can control the replay of VOD contents or IPTV download contents by the operation of a remote controller. Therefore, it is recommended that the following operations can be performed in addition to the operations specified in 6.5 "Remote Controller" in Vol. 2 of ARIB TR-B14.

- Playback  
Replays the selected IPTV download/VOD contents that have paused.
- Stop  
Stops the IPTV download/VOD contents that are being replayed.
- Pause  
Temporarily stops the stream distribution of the IPTV download/VOD contents that are being replayed, assuming that contents replay will be resumed.
- Fast-forward/rewind  
Performs the fast-forward or rewind operation of the IPTV download/VOD contents that are being replayed.
- Forward/backward skip  
Jumps from the replay position of the IPTV download/VOD contents that are being replayed to a predetermined forward or backward replay position from which replaying is continued.



### 7.2.2.2.2 Video decoding

Refer to the coding systems in the description of video decoding in 7.3 "Video Decoding Process and Output" in Vol. 2 of ARIB TR-B14 and video coding systems described in 6.1.1.2 "Details of Operation of MPEG-2 (Video)" and 6.1.1.3 "Details of Operation of H.264/MPEG-4 AVC Video" in the IPTV Standard VOD Specifications as well.

### 7.2.2.2.3 Audio decoding

Refer to the coding systems in the description of audio decoding in 7.4 "Audio Decoding Process and Output" in Vol. 2 of ARIB TR-B14 and audio coding systems described in 6.1.2.2 "MPEG2 (Audio)" in the IPTV Standard VOD Specifications as well.

## 7.2.2.3 IPTV download (integrated service type 2) function

### 7.2.2.3.1 Reference model for receivers

Figure 7-1 shows a reference model for receivers with the IPTV download (integrated service type 2) function. For details of new IPTV download functions, see 3.1 "Receiver Model" in the IPTV Standard Download Specifications. For details of other functions, follow the specifications provided in 1.1 "Components of receiver units" in Section 2 of this volume.

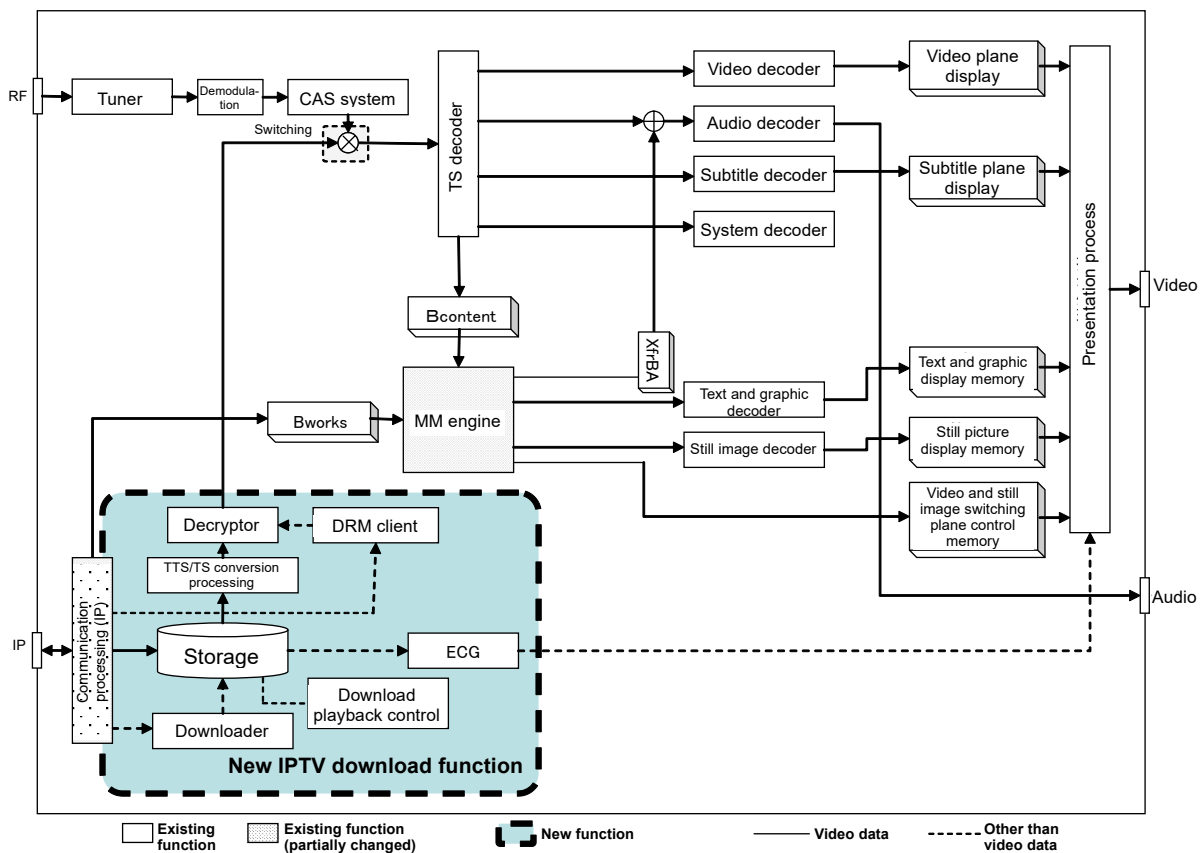


Figure 7-1 Receiver reference model with the IPTV download (integrated service type 2) function

#### **7.2.2.3.2 Communication function**

Refer to communication protocols described in 6.2.2 (3) "Communication protocol for Ethernet" in Vol. 6 of ARIB TR-B14 and 2.3.3 "Protocol Stack" in the IPTV Standard Download Specifications (except for the specifications regarding HTML documents and IP v6). Implementation of IP v6 shall be a matter of product planning.

#### **7.2.2.3.3 IPTV download function of receivers**

- Receivers shall have a storage device for storing IPTV download contents.
- Receivers shall have the functions described in the following sections and chapter in the IPTV Standard Download Specifications.
  - 3.4 Downloader
  - 3.5 Download Status Display
  - 3.6 Local Navigation
  - 3.7 Content Replay
  - 3.8 External Output Interface (Optional)
  - 3.9 Parental Control
  - 8. DRM Specifications

#### **7.2.2.3.4 BML browser function implemented in receivers with the IPTV download (integrated service type 2) function**

See the column regarding receivers that support the IPTV download (integrated service type 2) function in Table 7-1 in 7.2.3.1.

#### **7.2.2.3.5 Remote controller**

It is desirable that an IPTV download application of receivers can be started by the operation of a remote controller. However, a button is not physically required for the operation. "download status display", "local navigation", and "content replay" functions are assumed as IPTV download application functions. See the following sections in the IPTV Standard Download Specifications for each function.

- 3.5 Download Status Display
- 3.6 Local Navigation
- 3.7 Content Replay

### 7.2.2.4 VOD (integrated service type 2) function

#### 7.2.2.4.1 Reference model for receivers

Figure 7-2 shows a reference model for receivers with the VOD (integrated service type 2) function. For details of new VOD functions, see 3.1 "Receiver Model" in the IPTV Standard Download Specifications. For details of other functions, follow the specifications provided in 1.1 "Components of receiver units" in Section 2 of this volume.

The reference model in Figure 7-2 shows a case of a receiver that has a resource for exclusive use for the VOD function. However, implementation of actual receivers is not limited to such. For details, see Appendix 17.

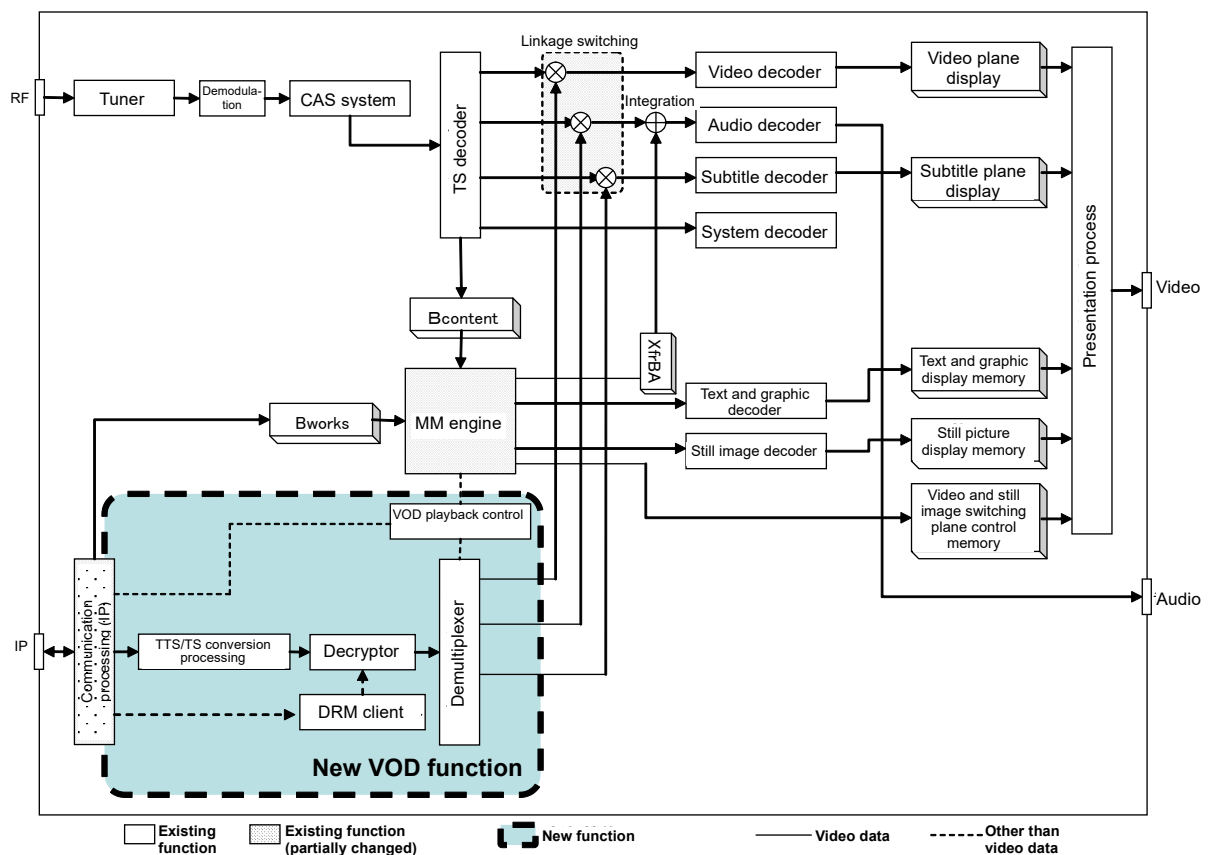


Figure 7-2 Receiver reference model with the VOD (integrated service type 2) function

#### 7.2.2.4.2 Communication function

Refer to communication protocols described in 6.2.2 (3) "Communication protocol for Ethernet" in Vol. 6 of ARIB TR-B14 and 3.2.1 "Communication Processing" in the IPTV Standard VOD Specifications (except for the specifications regarding HTML documents and IP v6). Implementation of IP v6 shall be a matter of product planning.

As a streaming protocol, the HTTP streaming protocol defined in 4.2 "Video Transmission

Protocol based on HTTP" in Chapter 4 of the IPTV Standard VOD Specifications or the RTP/RTSP streaming protocol defined in 4.1 "Video Transmission Protocol based on RTP/RTSP" in the IPTV Standard VOD Specifications and 5.2.1 "Video transmission protocol based on RTP/RTSP" in the IPTV Standard Internet Scope Service Approach Specifications shall be supported. Both HTTP streaming and RTP/RTSP streaming protocols can be supported.

#### **7.2.2.4.3 VOD functions of receivers**

- Receivers shall have the functions described in following sections in the IPTV Standard VOD Specifications.
  - 3.2.3 DRM Processor/Decrypter
  - 3.2.6 VOD Playback Control
- As video content processing, the process provided in 5.3 "Operation based on Video Content Specifications" in the IPTV Standard Internet Scope Service Approach Specifications should be supported.

#### **7.2.2.4.4 BML browser function implemented in receivers with the VOD (integrated service type 2) function**

See the column regarding receivers that support the VOD (integrated service type 2) function in Table 7-1 in 7.2.3.1.

### **7.2.3 Operation of the BML browser in receivers with the IPTV download/VOD (integrated service type 2) function**

#### **7.2.3.1 Implementation of the IPTV download/VOD (integrated service type 2) function**

The table below shows the correspondence relationships between extended functions and receivers with the IPTV download/VOD (integrated service type 2) function that require the extended functions.

Table 7-1 Correspondence relationships between required extended functions and receivers with the IPTV download/VOD (integrated service type 2) function

	Receivers with IPTV download (integrated service type 2) function	Receivers with VOD (integrated service type 2) function
getBrowserSupport( )	O (Note 1)	O (Note 1)
getResidentAppVersion( )	O (Note 2)	O (Note 2)
getBrowserStatus( )	O	O (Note 3)
startDlcDownload( )	O (Note 4)	—
getDlcDownloadStatus( )	O (Note 4)	—
startVOD( )	—	O (Note 4)

○: Support required

—: No need to support

(Note 1) See 7.2.3.2.3 (1).

(Note 2) See 7.2.3.2.3 (2).

(Note 3) See 7.2.3.2.3 (3).

(Note 4) See 7.6.19 in Vol. 2 of ARIB STD-B24.

## 7.2.3.2 Operation of browser pseudo-objects related to the IPTV download/VOD (integrated service type 2) function

### 7.2.3.2.1 Operation of Ureg

A Ureg value is not guaranteed after calling startVOD( ).

### 7.2.3.2.2 Operation of Greg

A Greg value shall be retained when calling startVOD( ) and during the display of VOD contents.

### 7.2.3.2.3 Operation of operation control functions

(1) Operation of getBrowserSupport( )

Table 7-2 Character strings that can be specified to the `getBrowserSupport()` argument and the definitions

functionname	additionalinfo	Performance of <code>getBrowserSupport()</code>
ResidentApp	"VOD"	Returns 1 if the receiver supports the VOD function provided in 5. "VOD Operation" in the IPTV Standard Internet Scope Service Approach Specifications and if the VOD function can be used with <code>startVOD()</code> .
	"Download"	Returns 1 if the receiver supports the IPTV download function provided in 6. "Download Operation" in the IPTV Standard Internet Scope Service Approach Specifications, and if the IPTV download function can be used with <code>startDlcDownload()</code> and <code>getDlcDownloadStatus()</code> .
IPTVFunction	"VOD","HTTP"	Returns 1 if the receiver supports the VOD function provided in 5.2.2 "Video Transmission Protocol based on HTTP" in the IPTV Standard Internet Scope Service Approach Specifications, and if the VOD function can be used with <code>startVOD()</code> .
	"VOD","RTSP"	Returns 1 if the receiver supports the VOD function provided in 4.1 "Video Transmission Protocol based on RTP/RTSP" in the IPTV Standard VOD Specifications and 5.2.1 "Video transmission protocol based on RTP/RTSP" in the IPTV Standard Internet Scope Service Approach Specifications, and if the VOD function can be used with <code>startVOD()</code> .

(2) Operation of `getResidentAppVersion()`

When "VOD" or "Download" is specified as the argument of `appName` of `getResidentAppVersion()`, a character string that indicates the receiver profile allocated by each individual IPTV service provider and that is supported by the receiver as an IPTV function is returned to the `Array[4]` return value. The array rule is as follows.

- An item name that is used to identify each service provider and indicates a receiver-specific function that is specified by the provider is described as string-type data.
- A separator, ":" (colon), is placed between each item, and ":" is not placed at the beginning.
- For a character string to identify a service provider and to indicate a receiver-specific function that is specified by the provider, see the item name included in the `Array[4]` return value provided in 4.2.1 in the IPTV Standard Specifications of IPTV Service Approach by Integration with Broadcasting.

[Format]

"Item name": "Item name": . . . . "Item name"

Example: AB001:AC002

(The receiver profiles, AB001 and AC002, specified by an IPTV service provider can be used.)

### (3) Operation of getBrowserStatus( )

When VOD contents finish and the browser status is returned to the linked status, the receiver is synchronized with the status changes and changes the return value of getBrowserStatus( ) (extended function for broadcasting). Specifically, 1 is returned when the function is called under the condition in which "TerrP", "IRDState", and "Link" specified respectively as sProvider, statusname, and additionalinfo are used for the getBrowserStatus( ) arguments.

After VOD contents finish, NaN shall be returned as a return value when getBrowserStatus( ) is called with the combination of the arguments above specified until a return value is changed using this procedure.

### (4) Operation of getIRDID( )

Receivers that support the IPTV download/VOD (integrated service type 2) function use an ID (CorrelatingID) to associate registration information for the IPTV service with individual receivers.

Only receivers that return 1 for the getBrowserSupport( ) function under the condition of (i) or (ii) as follows can obtain a CorrelatingID using getIRDID( ).

(i) "ResidentApp" specified to functionname and "VOD" specified to additionalinfo

(ii) "ResidentApp" specified to functionname and "Download" specified to the additionalinfo

When a CorrelatingID is obtained using getIRDID( ), a value from 0xE0001 to 0xE0FFF that corresponds to the IPTV service is specified as type. In this case, a return value shall be a character string with a CorrelatingID. For details of the operation of a CorrelatingID, see Appendix 18.

It shall be noted that the function concerned can be operated in both data broadcasting reception status and linked status if the receiver supports the IPTV download/VOD (integrated service type 2) function.

#### **7.2.3.2.4 Operation of the IPTV linkage function**

##### (1) startDlcDownload( )

See 7.2.4.2 "Operation of browser pseudo-objects involving IPTV download".

##### (2) startVOD( )

See 7.2.5.1 "Operation of browser pseudo-objects involving VOD".

### **7.2.3.3 Disabling the return flag**

Normally, when communication contents are displayed on a BML browser, the main stream is obtained and monitored. If a return flag that is set in the DII of an entry component is 1, return to the entry component is performed. However, such operation is not performed in receivers with the VOD (integrated service type 2) function while VOD contents are being replayed.

### **7.2.3.4 Identification of IPTV download/VOD (integrated service type 2) function (contents guidelines)**

As described in 7.2.2.1, the IPTV download and VOD functions are separate options. The VOD function can be implemented by selecting either HTTP or RTSP or by selecting both protocols.

Contents determine the available IPTV function and protocols using the `getBrowserSupport()` by specifying an appropriate value to `functionname` and `additionalinfo` as provided in 7.2.3.2.3 (1) "Operation of `getBrowserSupport()`".

If broadcasters provide a service in which functions specified by individual IPTV service providers are included as a premise, then it is necessary for contents to assess whether or not the IPTV download/VOD function concerned is implemented using `getResidentAppVersion()`.

## **7.2.4 Operation of IPTV download (integrated service type 2) function**

In these guidelines, the basic operation of the IPTV download processes executed from BML documents conforms to the "IPTV Standard Download Specifications" and 6. "Download Operation" in the IPTV Standard Internet Scope Service Approach Specifications.

### **7.2.4.1 Execution sequence of IPTV download**

The IPTV content download service is assumed to be provided using the following steps.

1. Searches for a content to be downloaded.
2. Obtains download control information.
3. Obtains contents and supplementary information of the contents.
4. Searches for the downloaded contents and views the information.
5. Views or copies the contents.

Among the steps above, 1 and 2 are realized using BML documents operated on the BML browser.



Step 2 is executed by describing startDlcDownload( ) in BML documents.

Steps 3, 4, and 5 are IPTV download processes that are executed from BML documents and that are assumed to be realized using a receiver's resident application.

Step 3 is automatically executed based on the description content of the download control information obtained in step 2. Usually, acquisition of contents are executed separately from broadcasting reception and data broadcasting. It is assumed that the obtained contents are stored in an internal storage while a broadcasting program is being viewed or a receiver is in the standby status.

Step 4 is realized using a resident application that is activated through the operation of a remote controller. The contents stored in the internal storage are searched based on the supplementary content information obtained in step 3, and a means to view detailed information is provided.

Step 5 is executed based on the user's instruction using the content search means specified in step 4. The contents shall be replayed using the remote controller provided in 7.2.2.3. The contents are copied from the internal storage to a removable medium or an external device with network connection.

#### 7.2.4.2 Operation of browser pseudo-objects involving IPTV download

- Operation of startDlcDownload( )

The format of src\_path specified by the arguments of startDlcDownload( ) is as follows.

SCHEME `“://”` HOST [ `“.”` PORT ] `“/”` PATH [ `“?”` QUERY ]

SCHEME = `“http”` | `“https”`

HOST Conforms to RFC3986.

PORT Conforms to RFC3986.

PATH Conforms to RFC3986.

QUERY Conforms to RFC3986.

\* PORT or QUERY that is enclosed with [ ] in this format can be omitted.

- The maximum length of the URI specified by src\_path shall be 1,023 bytes.
- TLS1.0/SSL3.0 or TLS1.2 is used if a secure communication protocol is used to obtain download control information.
- See 4.2.1 (4) in the IPTV Standard Specifications of IPTV Service Approach by Integration with Broadcasting for a certificate used by a receiver when using a secure communication protocol to obtain download control information.

#### 7.2.4.3 About download control information

The download control information to be obtained in step 2 in 7.2.4.1 describes the attributes and access information of a content file obtained via a network and information to access the following content's supplementary information.

- ECG metadata necessary for searching for the stored content and for also displaying the details
- Replay control information necessary for replaying and copying a content
- License information that manages conditions for replaying and copying a content

For details of specifications for download control information, see 5 "Download Control Information" in the IPTV Standard Download Specifications.

#### 7.2.4.4 Operation of BML contents for the IPTV download function

- Identification of receivers with the IPTV download function

When IPTV download is performed from a BML document, it is first of all necessary for the BML document to distinguish whether or not a receiver supports the IPTV download function using the following methods.

##### (1) Distinguishing the implementation of the basic IPTV download functions

Whether or not a receiver supports the basic IPTV download functions is distinguished from a BML document using the `getBrowserSupport( )` function with "Download" specified to the argument. "Download" is the character string used when `functionname` is `ResidentApp` as shown in the table in 7.2.3.2.3 (1). If 1 is returned as a function return value, it is indicated that the receiver supports the IPTV download (integrated service type 2) function provided in these guidelines.

##### (2) Distinguishing the implementation of IPTV download-related functions that are uniquely implemented in individual receivers

Whether or not a receiver supports functions that individual IPTV service providers uniquely specify (e.g., authentication and DRM) and the functions for the integrated service type 1 shall be distinguished using the `getResidentAppVersion( )` function with the character string "Download" specified to the argument of `appName`.

For the character string of an item name arranged in the return value `Array[4]` of the same function, see 4.2.1 (4) in the IPTV Standard Specifications of IPTV Service Approach by Integration with Broadcasting.

- Browser status where an IPTV download-related function can be called

The `getDlcDownloadStatus( )` and `startDlcDownload( )` functions that are related to the IPTV download function can be called from a BML document both in the broadcasting status and linked status.

- Browser status where a browser can receive an IPTV download instruction

It is desirable for a BML document to confirm that a browser is ready to receive an IPTV download instruction using `getDlcDownloadStatus( )` before the BML document instructs IPTV download using `startDlcDownload( )`.

The following shall be taken into account: If the instruction is sent soon after `startDlcDownload( )` is called, a receiver may not be able to receive an IPTV download instruction depending on the functions implemented in the receiver.

- About encouraging the using receiver's functions to display the IPTV download status and to replay downloaded contents

A BML document cannot be used to enable a receiver to obtain the execution status of the IPTV download or to enable the receiver to replay video obtained using the IPTV download function. It is desirable that BML contents are written so as to prompt users to use the receiver's functions, such as ECG, by providing appropriate information.

### 7.2.5 Operation of VOD (integrated service type 2)

In these guidelines, the basic operation of the VOD function that is activated from a BML document conforms to the IPTV Standard VOD Specifications and 5. "Operation of VOD" in the IPTV Standard Internet Scope Service Approach Specifications.

#### 7.2.5.1 Operation of browser pseudo-objects involving VOD (integrated service type 2)

##### (1) Operation of `startVOD( )`

`startVOD( )` is an asynchronous function and returns a result of accepting a VOD content replay request as a return value. Specifically, after the statuses of specified parameter errors and the AV player are confirmed, NaN is returned if a request cannot be accepted.

However, even if the AV player is successfully activated, a return value is not returned in some cases by a receiver that stops the BML browser when the AV player is started.

This function will return NaN, if a BML document that is presented after completion of the VOD function specified by `metafile_uri` as the first argument does not exist under the base URI directory of a BML document that is used to activate the function.

`startVOD( )` can be used only when a receiver is in the linked status.

- metafile\_uri (first argument) is defined as follows.
- A character string has a URI form that consists of the following.
  - ◇ The URI of a replay control metafile corresponding to VOD contents
  - ◇ The license ID of a license that is used when replaying
  - ◇ The URI of a BML document presented after the VOD function is completed
  - ◇ A position to start replaying VOD contents (omittable)
- The metafile\_uri shall be in a query format with parameters other than those of a replay control metafile added to the URI of the replay control metafile.
- For the metafile\_uri format, see 4.2.1 in the IPTV Standard Specifications of IPTV Service Approach by Integration with Broadcasting.
- The maximum character string length is 1,024 characters (1,024 bytes).
  
- The format of option (second argument) is as follows.
  - Array[0] : String service\_ref      Service identification  
    service\_ref is used to enable the BML browser to reestablish the linked status after the VOD function is completed. When this function is called using a BML document, service\_ref that indicates a selected service shall be specified. The operation of a receiver when a service indicated by specified service\_ref is different from the selected service is a matter of product planning.  
    The description of the service\_ref conforms to the name space specifications provided in 9.2.5 in Vol. 2 of ARIB STD-B24. However, an abbreviated form specifying -1 to <original\_network\_id>, <transport\_stream\_id>, and <service\_id> provided in 9.2.5.1 in Vol. 2 of ARIB STD-B24 is not used.
  
  - Array[1] : String vod\_service\_type      Identification of VOD content providing service  
    vod\_service\_type is a character string to identify VOD content providing service. A receiver distinguishes an error code system that is specified for each VOD content service provider and that is returned to the server using this argument after the VOD function is completed. A value of vod\_service\_type shall be one of the item names included in a return value Array[4] that is returned when getResidentAppVersion( ) is called with "VOD" specified to the argument of appName of getResidentAppVersion( ). startVOD( ) returns NaN if vod\_service\_type that is not supported by the receiver is specified.

#### 7.2.5.2 Execution sequence to activate the VOD function

The following procedure is used to activate the VOD function.

1. Activates an AV player with startVOD( ), which is an extended function for broadcasting, using a BML document for a receiver that is in the linked status.
2. Obtains a replay control metafile based on metafile\_uri as an argument of the startVOD( ) function.
3. Analyzes the replay control metafile and obtains information regarding the streaming control of the VOD contents, license information, and URI of the video contents.
4. Specifies a license ID to identify the license necessary to replay the target VOD contents and acquires the license via a DRM server as provided in the IPTV Standard VOD Specifications.
5. Enters the license to the DRM client to obtain content usage conditions (RMPI) and a content key.
6. Accesses the video content server to receive and replay the streaming data via HTTP or RTP.

#### 7.2.5.3 Specifications for the operation when starting the VOD function

For a certificate that is used by a receiver when the secure communication protocol is used to obtain a replay control metafile, see 4.2.1 (4) in the IPTV Standard Specifications of IPTV Service Approach by Integration with Broadcasting.

Product planning shall determine whether or not an unload interruption event occurs to a BML document when startVOD( ) is called from the BML document.

#### 7.2.5.4 Specifications for the operation while replaying VOD contents

The operation of a receiver while VOD contents are being replayed is provided as follows.

- The sequence of replaying and viewing VOD contents conforms to 2.3 "System Flow" in the IPTV Standard VOD Specifications.
- The channel selection status and the BML browser status while replaying VOD contents shall be determined based on the product planning policy. However, such a status shall not affect the display of the VOD contents regardless of the description contents of the BML document that is used to call the VOD function.
- If a pullback or an interruption event that needs to be handled by the BML browser occurs while replaying VOD contents, the BML browser shall ignore it.

### 7.2.5.5 Sequence to terminate the VOD function

- (1) A receiver terminates the VOD function when VOD contents end or when stop is selected by user's remote controller operation. When the VOD function is completed, the linked status in data broadcasting shall be reestablished based on the following procedure: The AV player stops retaining the replay stop position and replay stop status.

A BML document that is used to call the VOD function shall not be presented during the period from completion of the VOD function to start of presentation of the BML document as described in (3) below. It is desirable that broadcasting is not displayed on a full screen during this period. However, if broadcasting needs to be displayed on a full screen, the display shall be in a manner that does not cause discomfort to users. (For instance, it is expected that users may experience discomfort if the display is switched to that of the data broadcasting screen soon after broadcasting is displayed on a full screen for a brief second).

- (2) The receiver accesses the URI of the BML document to be presented after completion of the VOD function that is specified by the query parameter of the argument of `metafile_uri` when activating the AV player using `startVOD()` (extended function for broadcasting). The receiver then returns the replay stop position and replay stop status that are saved in (1) above to the server based on the procedure described below. At the same time, the receiver obtains the BML document.
- (3) The BML browser presents the obtained BML document.
- (4) While displaying the VOD contents, if the receiver cannot maintain the linked status, the receiver selects a channel while presenting the BML documents as described in (3), and starts monitoring DII that is specified based on the receiver's behavior provided in 2.3.1.7 "Basic behavior of the receiver during data broadcasting program presentation".
- (5) After reestablishing the linked status for data broadcasting, the BML browser returns 1 as a return value when it is called with "TerrP", "IRDState", and "Link" specified to `sProvider`, `statusname`, and `additionalinfo`, respectively, to the `getBrowserStatus()` argument that is an extended function for broadcasting.

After the VOD function is completed, the receiver adds the following query parameter to the URI of the BML document to be presented after completion of the VOD function that is specified using the query parameter for the first argument of `startVOD()`.

- Adding a VOD completion status query parameter

A message indicating how the replay of the VOD contents has been finished is added using a parameter. The parameter format is as follows.

Status=<status\_code>

The Status parameter using the format above shall be added after "?" of the query (the <query> component as shown in 4.1.5.1 in the IPTV Standard CDN Scope Service Approach Specifications) of the URI of a BML document that is presented after the VOD function is completed.

If a query parameter already exists in the above-mentioned URI that is specified as the argument of startVOD( ), the Status parameter in the above format shall be placed after "&", which is an additional parameter.

<status\_code> Normal termination: 0 Other : 1

The <status\_code> above indicates operation when Array[1] vod\_service\_type is not specified for option (second argument) of startVOD( ). When the vod\_service\_type is specified, <status\_code> is determined based on the operation of VOD contents provision service determined by vod\_service\_type.

- Adding a VOD replay stop position query parameter

When the replay of VOD contents is terminated, the stop position is added using a parameter.

The parameter format is as follows.

StopPos=<stop\_time>

The description format of <stop\_time> shall follow the NPT (normal play time) format provided in RFC2326[12.29]. The integer part can include up to five digits and the decimal part, up to one digit.

This parameter is written by adding "&" after the Status parameter described in the previous section. The parameter is not added if the replay of the VOD contents has not been terminated normally.

- Setting example

When the replay of VOD contents has been terminated normally (when an existing query does not exist in a specified return URI):

http://<server\_name>[:<port>]/<path>/return.cgi?Status=xxxx&StopPos=1024.8

When the replay of VOD contents has been terminated abnormally (when an existing query does not exist in a specified return URI):

http://<server\_name>[:<port>]/<path>/return.cgi?Status=xxxx

When the replay of VOD contents has been terminated abnormally (when an existing query exists in a specified return URI):

http://<server\_name>[:<port>]/<path>/return.cgi?unknown\_query=zzz&Status=xxxx

### 7.2.5.6 Specifications for the operation when the VOD function is completed

#### 7.2.5.6.1 Reestablishment of the linked status

Unlike the reference model for receivers defined in 7.2.2.4.1, if a receiver cannot maintain the linked status while replaying VOD contents, the linked status needs to be reestablished when presenting a BML document that is obtained after the VOD function is completed. Appendix 17 provides examples.

If a receiver cannot maintain the linked status, the receiver shall retain the base URI directory before executing startVOD( ) while replaying the VOD contents and shall use the base URI directory value when reestablishing the linked status.

If a function other than the extended function for broadcasting as shown in Table 7-3 is called, the operation of a receiver until the linked status is established shall be a matter of product planning. The receiver operation when a document transition occurs while reestablishing the linked status shall be also a matter of product planning.

The operation when a receiver cannot reestablish the linked status shall be determined based on the product planning policy. However, it is desirable that the receiver close the browser and then be in a broadcasting viewing status.

Table 7-1 Extended functions for broadcasting that can be used before reestablishing the linked status

Extended functions for broadcasting that can be used before reestablishing the linked status
getBrowserStatus( ) setInterval( ) clearTimer( )

#### 7.2.5.6.2 Receiver information when the VOD function is completed

Retention of the following information after replaying VOD contents is a matter of product planning.

- Value stored in Ureg
- Module locked by lockModuleOnMemory( ) and lockModuleOnMemoryEx( )

#### 7.2.5.7 Operation of BML contents for VOD (integrated service type 2) services

In these guidelines, the sections shown below include specifications for absorbing difference in operation due to difference in functions implemented in individual receivers while VOD contents are being replayed or after the replay of the VOD contents is completed. These specifications need to be noted when creating BML contents.



- Specifications to prevent the VOD function from being suspended while VOD contents are being replayed
  - 7.2.3.3 Disabling the return flag
- Sections that include specifications for making the operation after completion of the VOD function the same
  - 7.2.3.2.1 Operation of Ureg
  - 7.2.3.2.3 Operation of operation control functions, (3) Operation of `getBrowserStatus( )`
  - 7.2.5.3 Specifications for the operation when starting the VOD function
  - 7.2.5.4 Specifications for the operation while replaying VOD contents
  - 7.2.5.6 Specifications for the operation when the VOD function is completed

- Distinguishing receivers that have the VOD (integrated service type 2) function

When the VOD (integrated service type 2) function is used via a BML document, the BML document shall first of all distinguish whether or not the receiver supports the VOD (integrated service type 2), using the methods below.

(1) Distinguishing the implementation of the basic VOD (integrated service type 2) functions

Via a BML document, whether or not a receiver supports the basic VOD (integrated service type 2) functions is distinguished using the `getBrowserSupport( )` function by specifying "VOD", "HTTP" or "VOD", "RTSP", which is the character string of `additionalinfo` when `functionname` as shown in the table in 7.2.3.2.3 is `IPTVFunction`, to the arguments. If 1 is returned as a function return value, it indicates that the BML browser can use the functions as shown in the table in 7.2.3.2.3 among the specifications provided in the IPTV Standard VOD Specifications.

(2) Distinguishing the implementation of VOD-related functions that are uniquely implemented in individual receivers

When a broadcaster confirms whether or not a receiver supports functions that are uniquely specified by individual IPTV service providers (such as authentication and DRM), the `getResidentAppVersion( )` function is used with the character string "VOD" specified to the argument `appName`.

For the character string of an item name arranged in the return value `Array[4]` of the same function, see 4.2.1 in the IPTV Standard Specifications of IPTV Service Approach by Integration with Broadcasting.

- Status of a BML document that can present VOD contents

The VOD function that is activated from the BML browser can only be presented using a BML document of a receiver that is in the linked status. Before providing VOD contents with `startVOD( )`, it is desirable for the BML document to confirm that the browser is in the linked status using `getBrowserStatus( )`.

- Arguments of the function to present VOD contents

For a parameter that is specified to the argument of `metafile_uri` of `startVOD( )`, see 4.2.1 in the IPTV Standard Specifications of IPTV Service Approach by Integration with Broadcasting. A document that is obtained via a communication (scheme: http or https) shall be specified to the URI of a BML document that is presented after completion of the VOD function specified with the query parameter of the argument of `metafile_uri`. A BML document obtained through broadcasting shall not be specified to this URI.

The format of the argument of option of `startVOD( )` is as provided in 7.2.5.1 (1) "Operation of `startVOD( )`". However, it should be noted that the operation after presenting VOD contents is a matter of product planning if `service_ref` for service that is different from the selected service is specified as `service_ref` of `array[0]`. Moreover, the function returns NaN if a `vod_service_type` that is not supported by the receiver is specified to `vod_service_type` of `array[1]`.

- Precautions while presenting VOD contents

The description in a BML content does not affect the presentation of VOD contents during the presentation of the VOD contents. For instance, a description that is included in a BML content for causing an event to occur while presenting VOD contents will be ignored, if any.

- Restrictions on BML documents before reestablishing the linked status

A receiver may stop obtaining a broadcasting stream while replaying VOD contents. It should be noted that the following restrictions provided in 7.2.5.6.1 apply to a BML document that is obtained and presented after completion of the VOD function until the linked status is reestablished.

- The operation when a document is moved is a matter of product planning.
- Only the extended functions for broadcasting shown in Table 7.3 can be used. The operation when another extended function for broadcasting is called is a matter of product planning.

The reestablishment of the linked status can be confirmed by checking that the return value is 1 when the argument of the extended function for broadcasting `getBrowserStatus( )` is called with "TerrP", "IRDState", and "Link" specified as `sProvider`, `statusname`, and `additionalinfo`, respectively. The process to confirm that the reestablishment of the linked status using a BML document can be executed by repeatedly calling the `getBrowserStatus( )` function above.

### **7.3 IPTV download/VOD (integrated service type 2) function for CDN scope services**

[T.B.D.]

## 8 AIT-controlled application

An AIT-controlled application is an application whose execution including activation and termination is controlled by application control information (AIT). Among AIT-controlled applications, this chapter describes operation related to broadcasting integrated HTML5 applications and broadcasting-related operation.

The implementation of broadcasting integrated HTML5 applications is optional. However, receivers supporting broadcasting integrated HTML5 applications shall implement the functions provided in this chapter in principle, excluding the items described as optional.

In order to realize broadcasting integrated HTML5 application service provision as soon as possible, an operation specification in which functions to realize the IPTVFJ STD-0010 IPTV Standard Integrated Broadcast-Broadband System Specification provided by the IPTV Forum are limited to some of the functions is defined as phase 0 operation specifications, and the operation specifications for all necessary functions are defined as phase 1 operation specifications. “<phase 1>” will be added to the description regarding phase 1 operation specifications. Phase 0 operation specifications are applicable to operation for acquiring an AIT via communication only. Phase 1 operation specifications are applicable to the operation of functions such as that to acquire an AIT and auto start application through broadcasting, which are the main functions provided in the IPTV Standard Integrated Broadcast-Broadband System Specification.

It is desirable that a receiver supports phase 1 operation specifications. However, in these guidelines, receivers that only support phase 0 operation specifications shall be allowed.

### 8.1 Receiver functions

#### 8.1.1 Receiver model

- See 6.2 "Receiver Model" in the IPTV Standard Integrated Broadcast-Broadband System Specification for the receiver models that support broadcasting integrated HTML5 applications.
- In addition, in displaying using receivers that support broadcasting integrated HTML5 applications, captions and superimpositions that are transmitted over broadcasting shall be displayed at the forefront of the screen without scaling regardless of the size of the broadcasted video displayed using a broadcasting integrated HTML5 application. However, cases in which captions and superimpositions cannot be displayed while displaying VOD contents are also allowed.

- It is desirable that the broadcasted video and audio are replayed without being interrupted in both cases in which the broadcasting integrated HTML5 application is displayed from a BML document and in which the BML document is displayed from the broadcasting integrated HTML5 application. However, cases in which video of reduced size is temporarily displayed on a full screen and temporarily interrupted when switching are allowed.
- Receivers supporting phase 0  
Receivers that support all items other than those with the indication of “<phase 1>” in the operation specifications described in this chapter
- Receivers supporting phase 1  
Receivers that support all operation specifications in this chapter including phase 1 operation specifications (excluding specifications with an indication of optional)

### 8.1.2 Communication function

Refer to the communication protocols described in 6.2.2 (3) "Communication protocol for Ethernet" in Vol. 6 of ARIB TR-B14.

### 8.1.3 Media type operation range <phase 1>

See Annex C Media Type of B-XML/BML Documents and Monomedia Data and Table 9-6 Media type (Content-Type) and Corresponding File Type and Format Type in 9.1.2.3 in Vol. 2 of ARIB STD-B24. In some cases, receivers that support broadcasting integrated HTML5 applications transmit the media-type files shown in the table below using a data carousel. These media-type files can be transmitted using a component that transmits data broadcasting. However, access from data broadcasting to these media-type files cannot be made. The operation when accessing these files shall be a matter of product planning.

Media name	Media type	File type	Format type
Section format AIT	application/X-arib-ait	0x05	0x4a1
ZIP	application/zip	0x05	0x060

## 8.2 Operation of PSI/SI for broadcasting integrated HTML5 applications

### 8.2.1 Operation of PMT

#### 8.2.1.1 Operation of activation priority information of the PMT

The priority for activating data broadcasting and the AIT-controlled application is indicated using start\_priority of additional\_arib\_bxml\_info() specified in the data component descriptor with a PMT component tag value of 0x40.

(1) Operation of receivers supporting phase 0 operation specifications

Any receivers that support phase 0 operation specifications can activate the data broadcasting browser regardless of the start\_priority value of additional\_arib\_bxml\_info() specified in the data component descriptor of the PMT. In this case, the value specified to start\_priority shall be correctly ignored.

(2) Operation of receivers supporting phase 1 operation specifications <phase 1>

- The start\_priority value is evaluated only when selecting a channel and when performing the operation as described in 8.5.2.1 "Operation when selecting a channel".
- If 1 is specified to start\_priority, the activation process for data broadcasting is prioritized even when an AIT is transmitted.
- If 0 is specified to start\_priority, the AIT-controlled application is activated based on autostart\_priority (activation priority order) determined with the activation priority information descriptor (autostart priority descriptor) for the AIT. The activation of data broadcasting is processed when an AIT cannot be obtained.

**8.2.1.2 Operation of the data component descriptor <phase 1>**

If an AIT is transmitted via a data carousel, transmission shall be performed using a component to which the data component identifier for data broadcasting is specified. An AIT shall not be transmitted using a dedicated data component via a data carousel. Therefore, the data component identifier for a dedicated data component that is used to transmit an AIT is not described in the data component descriptor for the PMT. In addition, ait\_identifier\_info() is not operated as described in 7.3 in Vol. 4 of ARIB STD-B24.

**8.2.1.3 Operation of the PMT when transmitting an AIT as a section**

This operation is not specified for phase 1 operation.

**8.3 Operation of application control information (AIT)**

**8.3.1 Description format of AIT**

There are the following two AIT description formats: section format and XML format. For details, see Chapters 5 and 6 in Vol. 4 "Application Control System" of ARIB STD-B24.

For major items to be described in an AIT, see 4.1 "Overview of Application Control Information" in Vol. 4 "Application Control System" of ARIB STD-B24.

**8.3.2 AIT transmission operation**

The AIT can be transmitted using the following methods.

- (1) Transmission via communication
- (2) Transmission using a data carousel <phase 1>

(3) Transmission as a section (not specified in phase 1 operation)

### 8.3.2.1 Operation of transmitting an XML-format AIT via communication

- The XML format is used when an AIT is transmitted via communication.
- The transmission of an XML-format AIT via communication shall conform to the guidelines for transmitting communication contents described in 5.14.8.
- For the operation of SSL/TLS, see 8.2 in Vol. 6 of ARIB TR-B14.

### 8.3.2.2 Operation of transmitting an AIT using a data carousel <phase 1>

- The section format is used when an AIT is transmitted via a data carousel.
- Regarding the AIT transmission using a data carousel, multiple AITs can be transmitted using each of the multiple components (in an arbitrary number).
- module\_id of each component that transmits AITs is fixed to 0x9FFF.
- A multi-part format shall be used for a module that transmits AITs.
- The AIT file extension shall be ".ait".
- The maximum capacity of a section-format AIT file shall be 4,096 bytes.
- The resource name of an AIT that is constantly monitored is fixed to 9FFF/default.ait in each component.
- A component that transmits an AIT that is constantly monitored can be specified using a binary table, /40/9FFF/component.btb. If /40/9FFF/component.btb does not exist, /40/9FFF/default.ait shall be the AIT to be constantly monitored.
- An arbitrary name except for default.ait can be used as the file name of an AIT that is not constantly monitored. Such an AIT that is not constantly monitored can be referenced only with replaceApplication().

Appendix 21 "Operation example of AIT module transmission among affiliate stations" provides an operation example of transmitting or replacing AITs among affiliate service providers based on the assumption that an AIT is transmitted using a data carousel.

#### 8.3.2.2.1 Method to distinguish a component that transmits a constantly monitored AIT when selecting a channel, etc.

A component that transmits a constantly monitored AIT is distinguished using the following methods. (See Figure 8-1.)

1. If a /40/9FFF module exists, it is determined that an AIT is transmitted using a data carousel.

If a /40/9FFF module does not exist, the data broadcasting start process provided in

2.1.10.2 is executed.

2. / If /40/9FFF/component.btb exists, it is determined that a constantly monitored AIT is transmitted using a component other than 0x40. The tag value of the component that transmits the AIT to be constantly monitored is obtained by checking component.btb. Then, 9FFF/default.ait that is transmitted using the component is regarded as the AIT to be constantly monitored.

If /40/9FFF/component.btb does not exist, /40/9FFF/default.ait is regarded as the AIT to be constantly monitored.

“structure” of component.btb is "0,U:1B", and one component tag value shall be stored.

If an error occurs in a step, the data broadcasting start process provided in 2.1.10.2 is executed.



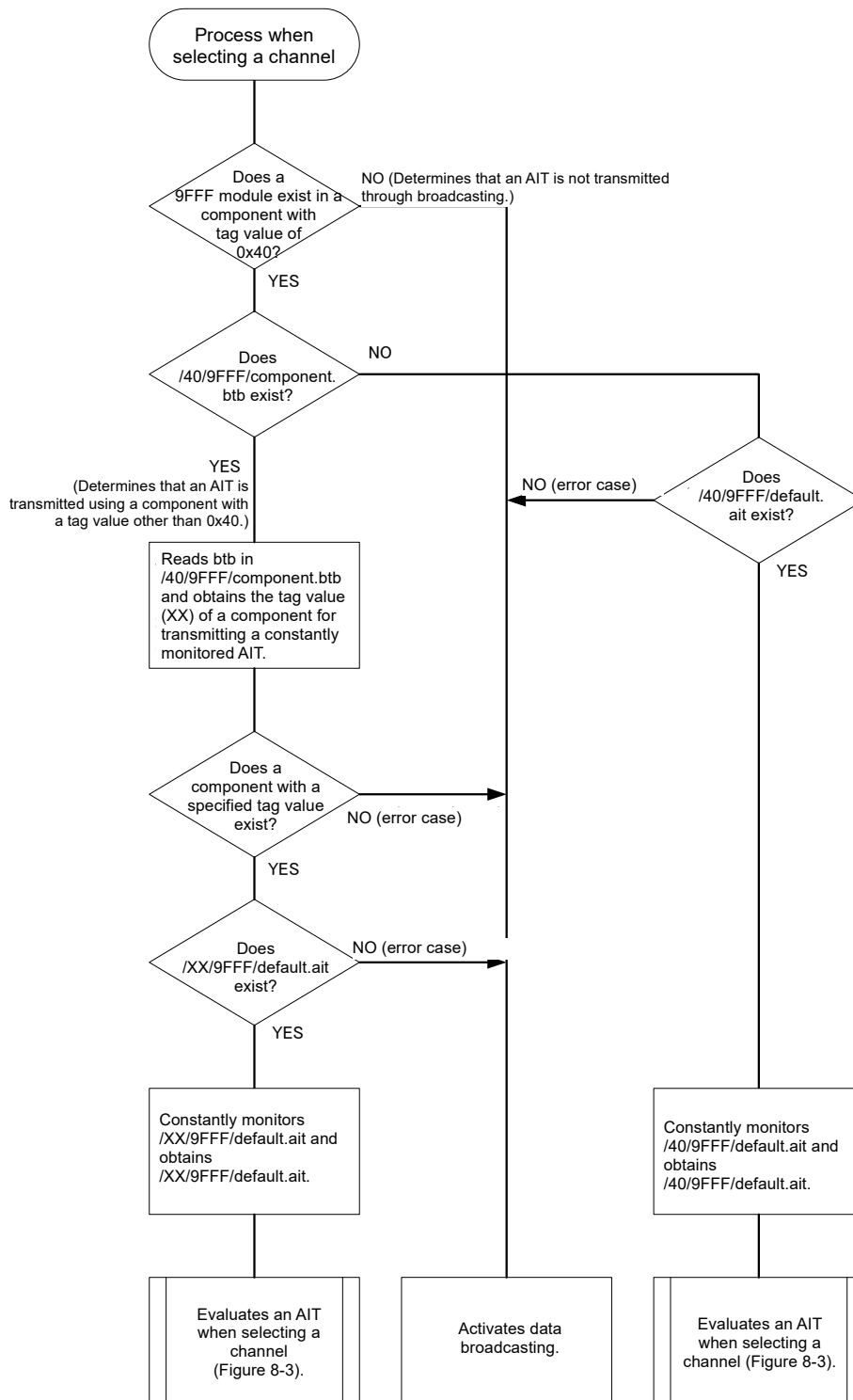


Figure 8-1 Process to obtain a constantly monitored AIT

### 8.3.2.3 Operation of transmitting an AIT using a section

This operation is not specified for phase 1 operation.

### 8.3.3 Operation of an AIT that is transmitted via communication

#### 8.3.3.1 Operation of an XML-format AIT that is transmitted via communication

Table 8-1 shows the structure of an XML-format AIT that is transmitted via communication.

Table 8-2 shows the definition and operation.

Table 8-1 Structure of an XML-format AIT transmitted via communication

Element and attribute names				Appearance frequency (B24)	Appearance frequency (operation)	Restriction on the number of characters and value
isdb:ServiceDiscovery				1	1	
isdb:ApplicationDiscovery				0..∞	1	
			@DomainName	1	1	Enter a character string of up to 255 characters.
			@Version	0..1	0	
isdb:ApplicationList				1..∞	1	
			isdb:Application	1..∞	1	
			isdb:applicationIdentifier	1	1	
			isdb:orgId	1	1	Enter a 16-bit length integer without a code as a numerical value.
			isdb:applId	1	1	Enter a 32-bit length integer without a code as a numerical value.
			isdb:applicationDescriptor	1	1	
			isdb:type	1	1	
			isdb:IsdbApp	1	1	Use the fixed value of "ISDB-HTML".
			isdb:controlCode	1	1	Use the fixed value of "AUTOSTART".
			isdb:visibility	0..1	0	
			isdb:serviceBound	0..1	0	
			isdb:priority	1	1	Enter an 8-bit length integer without a code as a character string without "0x" in hexadecimal notation (case insensitive). Use the fixed value of "FF".
			isdb:version	1	1	Enter an 8-bit length integer without a code as a character string without "0x" in hexadecimal notation (case insensitive).
			isdb:mhpVersion	0..1	1	
			mhp:profile	1	1	Enter a 16-bit length integer without a code as a character string without "0x" in hexadecimal notation (case insensitive).
			mhp:versionMajor	1	1	Enter an 8-bit length integer without a code as a character string without "0x" in hexadecimal notation (case insensitive).
			mhp:versionMinor	1	1	Enter an 8-bit length integer without a code as a character string without "0x" in hexadecimal notation (case insensitive).

Element and attribute names					Appearance frequency (B24)	Appearance frequency (operation)	Restriction on the number of characters and value
				mhp:versionMicro	1	1	Enter an 8-bit length integer without a code as a character string without "0x" in hexadecimal notation (case insensitive).
				isdb:icon	0..1	0	
				@filename	1	-	
				@size	0..1	-	
				@aspectRatio	0..1	-	
				isdb:storageCapabilities	0..1	0	
				isdb:applicationTransport	1..∞	1	
				@xsi:type	1	1	Use the fixed value of "isdb:HTTPTransportType".
				isdb:URLBase	1	1	Enter a character string of up to 255 characters. The character string must end with "/" (slash).
				isdb:URLExtension	0..∞	0	
				isdb:applicationLocation	1	1	Enter a character string of up to 255 characters.
				isdb:autostartPriorityDescriptor	0..1	0	
				@autostartPriority	1	-	
				isdb:cacheControlInfoDescriptor	0..1	0	
				@isdb:cachePriority	1	-	
				@isdb:applicationSize	1	-	
				@isdb:expireDate	1	-	
				@isdb:packageFlag	1	-	
				@isdb:applicationVersion	1	-	
				isdb:randomizeLatencyDescriptor	0..1	0	
				@isdb:rate	1	-	
				@isdb:range	1	-	
				@isdb:randomizationEndTime	0..1	-	
				isdb:applicationBoundaryAndPermissionDescriptor	0..1	1	
				isdb:boundaryAndPermission	1..∞	1..2	
				isdb:permissionBitmap	1..∞	1	Enter a 16-bit length integer without a code as a character string without "0x" in hexadecimal notation (case insensitive).
				isdb:managedURL	0..∞	0..10	Enter a character string of up to 255 characters.
				isdb:applicationExpirationDescriptor	0..1	0	
				isdb:boundaryAndPermissi	1..∞	-	

Element and attribute names				Appearance frequency (B24)	Appearance frequency (operation)	Restriction on the number of characters and value
			on			
			isdb:permissionBitmap	1..∞	-	
			isdb:managedURL	0..∞	-	
			isdb:ApplicationOnPlayback	0..∞	0	
			isdb:applicationIdentifier	1	-	Same structure as under element of isdb:Application
			isdb:applicationDescriptor	1	-	Same structure as under element of isdb:Application
			isdb:applicationTransport	1..∞	-	Same structure as under element of isdb:Application
			isdb:applicationLocation	1	-	Same structure as under element of isdb:Application
			isdb:autostartPriorityDescriptor	0..1	-	Same structure as under element of isdb:Application
			isdb:cacheControlInfoDescriptor	0..1	-	Same structure as under element of isdb:Application
			isdb:randomizeLatencyDescriptor	0..1	-	Same structure as under element of isdb:Application
			isdb:applicationBoundaryAndPermissionDescriptor	0..1	-	Same structure as under element of isdb:Application
			isdb:applicationExpirationDescriptor	0..1	-	Same structure as under element of isdb:Application
			isdb:ExternalApplicationControlDescriptor	0..1	0	
			isdb:externalApplication	0..∞	-	
			isdb:targetApplicationClass	1	-	
			isdb:targetApplicationIdentifier	1	-	
			@isdb:orgId	1	-	
			@isdb:applId	1	-	
			isdb:permissionBitmap	0..∞	-	
			isdb:overLayControl	1	-	
			@isdb:overLayAdmissionPolarity	1	-	
			isdb:overLayControlledArea	0..∞	-	
			isdb:overlayControlledAreaTag	1	-	
			isdb:horizontalPos	1	-	
			isdb:verticalPos	1	-	
			isdb:horizontalSize	1	-	
			isdb:verticalSize	1	-	
			isdb:blockedApplicationIdentifier	0..∞	-	
			@isdb:orgId	1	-	
			@isdb:applId	1	-	

Table 8-2 Definition and operation of the XML-format AIT transmitted via communication

Name	Definition and operation
isdb:ServiceDiscovery	
isdb:ApplicationDiscovery	
@DomainName	Operated as specified. Receivers shall not use this value.
@Version	Not used.
isdb:ApplicationList	
isdb:Application	Application information
isdb:applicationIdentifier	Application identification information
isdb:orgId	Organization identification: Indicates the organization that created the application. A worldwide unique number is assigned as identification. The governing organization as described in Appendix 20 provides the operation of the value.
isdb:appld	Application identification: Indicates a number to identify an application. A unique number is assigned to an organization identification.
isdb:applicationDescriptor	
isdb:type	
isdb:IsdbApp	Application format: Indicates the format of application that is controlled by an AIT. The fixed value of "ISDB-HTML" is used.
isdb:controlCode	Application control code: Indicates a control code to control the application statuses. The fixed value of "AUTOSTART" is used for transmission via communication.
isdb:visibility	Visibility: Indicates whether or not the application is visible to users/other applications while executing the application. Not used.
isdb:serviceBound	Service boundary flag: Indicates whether or not this application is available only for the current service. If this flag indicates 1, this application is associated with the current service alone and this application will be terminated when the current service is switched to another service. Not used.
isdb:priority	Application priority: Indicates relative priority among applications when multiple applications are operated. The fixed value of "FF" is used.
isdb:version	Value not specified. A receiver shall correctly ignore the value of this element.
isdb:mhpVersion	
mhp:profile	Application profile: Indicates the application profile of a receiver that can execute the application. The governing organization described in Appendix 20 specifies the operation of the profile value.
mhp:versionMajor	Major version: Indicates the major version of the profile above. The value is fixed to "1".
mhp:versionMinor	Minor version: Indicates the minor version of the profile above. The value is fixed to "1".
mhp:versionMicro	Micro version: Indicates the micro version of the profile above. The value is fixed to "1".
isdb:icon	Not used.

Name	Definition and operation
@filename	
@size	
@aspectRatio	
isdb:storageCapabilities	Not used.
isdb:applicationTransport	Describes protocol information to transmit an application.
@xsi:type	Protocol identification: Indicates a protocol to transmit an application. The fixed value of "isdb:HTTPTransportType" is used for transmission via communication.
isdb:URLBase	URL base: A character string written in the base part of a URL for obtaining an application.
isdb:URLExtension	URL extension: A character string written in the extension part of a URL for obtaining an application. Not used.
isdb:applicationLocation	Application URL: A character string indicating the URL of an entry point for a corresponding application. Describes a relative URL from URLBase of the isdb:applicationTransport.
isdb:autostartPriorityDescriptor	Not used.
@autostartPriority	Activation priority order: Indicates the priority order to activate an appropriate application in data broadcasting applications and all applications that are interlocked with the service currently being received.
isdb:cacheControlInfoDescriptor	Describes cache information when an application is expected to be used again. Not used.
@isdb:cachePriority	Cache priority: Indicates the priority order to retain application caches. The larger the value is, the higher the priority shall become.
@isdb:applicationSize	Application size: Indicates the entire application size in kbyte units. 0 is specified if the size is unknown.
@isdb:expireDate	Cache expiration date: Indicates the expiration date of application caches. The date is expressed as year/month/date using the lower 16 bits of MJD.
@isdb:packageFlag	Package flag: Indicates whether or not an application is packaged as one file. If the value is 1, it indicates that the application is packaged.
@isdb:applicationVersion	Application version: Indicates the version number of an application.
isdb:randomizeLatencyDescriptor	Describes probabilistically applied delay information to distribute server access loads. Not used.
@isdb:rate	Distribution number: Indicates the number of steps in the delay time until a control code, which is probabilistically set, is applied.
@isdb:range	Delay time range: Indicates the maximum delay time from the current time until a control code is applied. The time is specified in seconds.
@isdb:randomizationEndTime	Probabilistically applied end time: Indicates the time limit for executing a probabilistically applied delay process.
isdb:applicationBoundaryAndPermis	

Name	Definition and operation
isdb:boundaryAndPermission	
isdb:permissionBitmap	Permission bitmap: Indicates the accessibility to each broadcasting resource in bitmap data for each function. For the operation, see 8.3.3.4.
isdb:managedURL	Permission management area setting information: Indicates the character string of an Permission management area's URL. If the description is omitted, it indicates that a URL including an arbitrary location is specified.
isdb:applicationExpirationDescriptor	Not used
isdb:ApplicationOnPlayback	Not used
isdb:ExternalApplicationControlDescriptor	Not used

### 8.3.3.2 Character coding scheme for an AIT that is transmitted via communication

Basic Latin, which is a UCS (Unicode 2.0) subset, shall be used as the coded character set for an AIT that is transmitted via communication. UTF-8 with no BOM (byte order mark) shall be used as a character coding scheme.

### 8.3.3.3 Operation of XML declaration and name space

Table 8-3 shows the operation of the XML declaration and name space that needs to be described for an XML-format AIT.

Table 8-3 Operation of XML declaration and name space

```
<?xml version="1.0" encoding="UTF-8"?>
<isdb:ServiceDiscovery
xmlns:isdb="urn:arib:isdb:2012"
xmlns:ipi="urn:dvb:metadata:iptv:sdns:2008-1"
xmlns:mhp="urn:dvb:mhp:2009"
xmlns:tva="urn:tva:metadata:2005"
xmlns:mpeg7="urn:tva:mpeg7:2005"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:arib:isdb:2012 isdb_xmlait.xsd">
```

### 8.3.3.4 Operation of permissionBitmap for an AIT that is transmitted via communication

A bitmap is composed for each function to indicate the availability of access to each broadcasting resource. The 3 highest bits are used to identify the bitmap. The 13 lowest bits are assigned as shown below. For details, see the IPTV Forum Hybridcast operational guideline.\*

\* Receivers that support phase 0 operation specifications interpret only bitmap0. If any value other than 000 is specified to the 3 highest bits, broadcasting resources cannot be referred.

●Bitmap0

bit15 to bit13 000

bit12 to bit0 All 1: All broadcasting resources can be referred.

All 0: Broadcasting resources cannot be referred.

\* If any value other than "all 1" or "all 0" is specified to bit12 to bit0, broadcasting resources cannot be referred.

●Bitmap1<phase 1>

bit15 to bit13 001

bit12 Broadcasting video and audio reference and change of reference destination.

1: Allows the reference; 0: Prohibits the reference

bit11 Placement of monomedia data on broadcasting video

1: Allows the placement; 0: Prohibits the placement\*

bit10 SI acquisition API 1: Allows the use; 0: Prohibits the use

bit9 NVRAM access API 1: Allows the use; 0: Prohibits the use

bit8 Boundary extension API 1: Allows the use; 0: Prohibits the use

bit7 Receiver ID acquisition API 1: Allows the use; 0: Prohibits the use

bit6 VOD-related API 1: Allows the use; 0: Prohibits the use

bit5 API for accessing other broadcasting resources and signals

1: Allows the use; 0: Prohibits the use

bit4 to bit0 Reserved

\* It is preferable that receivers that support phase 1 conform to the bit11 specification. However, placing monomedia data on the broadcasting video according to the specification of a document is allowed, regardless of a bit11 value.

**8.3.4 Operation of a section-format AIT that is transmitted using a data carousel <phase 1>**

The specifications provided in Chapter 5, Vol. 4 of ARIB STD-B24 apply to items that are not described here. An AIT that is transmitted in a section format via a data carousel shall follow the format described in 5.1, Vol. 4 of ARIB STD-B24 in accordance with the section format.

• Operation of version\_number

Updated when a change is made to the subtable contents. It is desirable that a value is incremented by 1 when changed. (However, receivers shall consider that the subtable contents have been updated each time the value is changed regardless of whether or not the value is incremented by 1.)



- Operation of `section_number` and `last_section_number`  
The values of `section_number` and `last_section_number` are fixed to 0.
- Operation of `application_type`  
The value of `application_type` is fixed to 0x0010 (fixed to broadcasting integrated HTML5 application).
- Operation of `common_descriptors_length`  
The loop of common descriptor is not used. Therefore the value of `common_descriptors_length` is fixed to 0.
- Operation of `application_control_code`  
The operation shall conform to the IPTV Forum operational specifications.  
As described in IPTV Forum's IPTV Standard Integrated Broadcast-Broadband System Specification (IPTVFJ STD-0010), only one application whose `application_control_code` is AUTOSTART exists in an AIT.
- Operation of the URL character string coding scheme  
Basic Latin, which is a UCS (Unicode 2.0) subset, shall be used as the coded character set.  
UTF-8 without BOM (byte order mark) shall be used as a character coding scheme.

#### **8.3.4.1 Operation of application identifier**

- Operation of `organization_id` and `application_id`  
The operation of `organization_id` and `application_id` shall conform to the IPTV Forum operational specifications.

#### **8.3.4.2 Operation of application descriptor**

- Operation of `application_profiles_length`  
Only one `application_profiles` shall be set in the `application_profiles` loop. Therefore, the value of `application_profiles_length` is fixed to 5.
- Operation of `application_profile`  
The operation of `application_profile` shall conform to the IPTV Forum operational specifications.
- Operation of `service_bound_flag`  
The value is fixed to 1. (If the current service is switched to another service, the application is terminated.)
- Operation of `visibility`  
The value is fixed to 11.
- Operation of `application_priority`  
The value of 0xFF indicates that priority is not specified.  
The value is fixed to 0xFF.

- Operation of transport\_protocol\_label

One transport\_protocol\_label is described for one application descriptor.

#### **8.3.4.3 Operation of transport protocol descriptor**

- The transport protocol descriptor is arranged only in the application information descriptor loop and one per application is arranged..

- Operation of protocol\_id

One of the following values is used.

0x0003: HTTP/HTTPS transport

0x0004: Data carousel transport

##### **8.3.4.3.1 Operation of the selector byte for data carousel transport**

- Operation of remote\_connection

The value is fixed to 0. (An AIT shall be transmitted in the same service.)

- Operation of component\_tag

This is operated as specified.

##### **8.3.4.3.2 Operation of the selector byte for http/https transport**

- Operation of URL\_base\_byte

This is operated as specified. The number of URL\_base\_bytes that can be arranged in the selector area is one.

This character string shall be end with "/"(slash).

- URL\_extension\_count

The value is fixed to 0. (URL extension is not used.)

#### **8.3.4.4 Operation of simple application location descriptor**

- Operation of initial\_path\_bytes

initial\_path\_bytes is a character string that indicates the entry point URL of an application. This indicates the location where the application specified by a transport protocol descriptor can be obtained using a relative path that shows the route to the location.

- Operation of initial\_path\_bytes when an application is transmitted using a data carousel

A receiver obtains (a specific resource of) a specific module from the application URL specified in the simple application location descriptor by identifying a component with component\_tag specified in the transmission protocol descriptor.

The application URL above shall be specified using a format of

<module\_id>/(<resource\_name>). (module\_id is described in hexadecimal format without "0x".)

Example:

Transmission protocol descriptor

component\_tag: 0x40

Simple application location descriptor

initial\_path\_bytes: 9FFE/index.html

In this case, the application can be obtained from a resource whose name is index.html. This resource is included in a module whose module\_id is 0x9FFE in the data carousel of a component that indicates component tag value of 0x40 that is used in the same service.

- Operation of initial\_path\_bytes when an application is transmitted over http/https  
The application URL is described with a character string of up to 255 characters.

#### 8.3.4.5 Operation of application boundary and permission descriptor

- One or more application boundary authority setting descriptors are arranged.
- One permission\_bitmap is arranged in one application boundary authority setting descriptor.
- Operation of permission\_bitmap\_count

The value is fixed to 1.

- Operation of permission\_bitmap

A bitmap is composed for each function to indicate the accessibility to each broadcasting resource. The 3 highest bits are used to identify the bitmap. Either 000 or 001 shall be specified to the 3 highest bits. If any value other than 000 or 001 is specified to the 3 highest bits, it is regarded that none of the resources can be referred.

The 13 lowest bits are assigned as follows. For details, see the IPTV Forum Hybridcast operational guideline.

- Bitmap0

bit15 to bit13 000

bit12 to bit0 All 1: All broadcasting resources can be referred.

All 0: Broadcasting resources cannot be referred.

\* If any value other than "all 1" or "all 0" is specified to bit12 to bit0, broadcasting resources cannot be referred.

- Bitmap1<phase 1>

bit15 to bit13 001

- bit12 Broadcasting video and audio reference and change of reference destination.  
1: Allows the reference; 0: Prohibits the reference
- bit11 Placement of monomedia data on broadcasting video  
1: Allows the placement; 0: Prohibits the placement\*
- bit10 SI acquisition API 1: Allows the use; 0: Prohibits the use
- bit9 NVRAM access API 1: Allows the use; 0: Prohibits the use
- bit8 Boundary extension API 1: Allows the use; 0: Prohibits the use
- bit7 Receiver ID acquisition API 1: Allows the use; 0: Prohibits the use
- bit6 VOD-related API 1: Allows the use; 0: Prohibits the use
- bit5 API for accessing other broadcasting resources and signals  
1: Allows the use; 0: Prohibits the use
- bit4 to bit0 Reserved

\* It is preferable that receivers that support phase 1 conform to the bit11 specification. However, placing monomedia data on the broadcasting video according to the specification of a document is allowed, regardless of a bit11 value.

- Operation of managed\_URL\_count  
Operated as specified.
- Operation of managed\_URL\_length  
Operated as specified.
- Operation of managed\_URL\_byte  
Operated as specified.

The maximum number of permission\_bitmaps that are defined in the application boundary authority setting descriptor and API (addPermissionManagedArea()) and retained in a receiver is 16. The maximum value of the total number of managed\_URLs that are associated with all permission\_bitmaps retained is 1,024.

The application boundary authority setting descriptor is arranged for an application that is transmitted via a data carousel. However, any URLs including those of documents to be transmitted to managed\_URL\_byte using data carousels are not described. See 8.4.4 for the authority of a document transmitted using a data carousel.

#### 8.3.4.6 Operation of autostart priority descriptor

Only one activation priority information descriptor can be inserted in the application information descriptor loop of the AIT whose application\_control\_code value is 0x01

(identification name: AUTOSTART). If this descriptor is not inserted, it is regarded that the priority of the information, including data broadcasting, is the lowest.

• Operation of autostart\_priority (activation priority order)

autostart\_priority indicates the priority order to activate an application among the data broadcasting in the service currently being received and all AIT-controlled applications. The smaller the value specified is, the higher the priority of activation shall become. A value from 1 to 254 is specified to autostart\_priority for each AIT-controlled application.

For data broadcasting, a value from 1 to 255 can be specified to determine the activation priority. For the data broadcasting activation priority, the smallest value shall be found excluding the values specified to all AIT-controlled applications.

#### **8.3.4.7 Operation of cache control info descriptor**

No cache information descriptor shall be used. An application that is transmitted through broadcasting shall perform cache operation in conformity with the specifications provided in 8.4.1.5. An application obtained via communication shall clear caches as described in Vol. 4 of ARIB STD-B24 when the application is closed.

#### **8.3.4.8 Operation of randomized latency descriptor**

The operation of probabilistically applied delay descriptors shall conform to the IPTV Forum Hybridcast operational guideline.

##### **8.3.4.8.1 Operation of probabilistically applied delay descriptors for an auto start application**

No probabilistically applied delay descriptor shall be inserted in the loop of an application that is transmitted using a data carousel.

#### **8.3.4.9 Operation of external application control descriptor**

External application control descriptor is not used.

#### **8.3.4.10 Operation of playback application descriptor**

Playback application descriptor is not used.

#### **8.3.4.11 Operation of simple playback application location descriptor**

Simple playback application location descriptor is not used.

#### **8.3.4.12 Operation of application expiration descriptor**

Application expiration descriptor is not used.

#### **8.3.4.13 Operation of the lengths of character strings for an AIT**

The lengths of character strings used for AITs are specified as follows.

Table 8-4 Operation of the lengths of character strings for section-format AITs

Section format AIT	Length of the character string
Transmission protocol descriptor URL_base_byte	A character string of up to 255 characters
Simple application location descriptor initial_path_bytes	A character string of up to 255 characters
Application boundary authority setting descriptor managed_URL_byte	A character string of up to 255 characters

It shall be noted that a section-format AIT may be subject to the restriction of a range that can be expressed by descriptor\_length.

### 8.3.5 AIT Monitoring

#### 8.3.5.1 Monitoring an AIT that is transmitted via communication

An XML-AIT that is transmitted via communication shall not be constantly monitored. Therefore, automatic update via receiver operation is not performed. The update will be performed when an XML-AIT is explicitly obtained using the BML or HTML application.

#### 8.3.5.2 Monitoring an AIT that is transmitted using a data carousel <phase 1>

If an AIT is transmitted using a data carousel and if the existence of a constantly monitored AIT is identified through the process performed when selecting a channel as described in 8.3.2.2.1, a receiver shall monitor the modules including the AIT (9FFF/default.ait) to be constantly monitored that is transmitted using a component specified when selecting the channel. When the version of the module has been upgraded, the receiver acquires a module again according to the process flow shown in Figure 8-2. When a constantly monitored AIT in the reacquired module has been updated, the AIT is evaluated again. (See Figure 8-3.) A constantly monitored AIT shall not be changed, if component.btb has been updated when the module is reacquired, except for cases where the process described in 8.5.2.1 "Operation when selecting a channel" is performed.

If the access right bitmap of an application in operation is changed in accordance with the update of a constantly monitored AIT, the application in operation is terminated at the time at which the application calls a prohibited API or at the time at which the application performs a prohibited operation, and the process described in 8.5.2.1 "Operation when selecting a channel" is then performed. However, it is allowed to have the change to the operation not reflected to the operation until page transition, if bit12 and bit11 of bitmap1 is changed.

See 8.5.2.3 for the operation of updating an AIT when the AIT is transmitted using a data carousel.

Update is confirmed in accordance with the following procedure for a component that is constantly monitored.

- 1) Detect the version upgrade of a data carousel or module that transmits a constantly monitored AIT.
- 2) Obtain the file of the AIT to be constantly monitored and compare version\_number with the detected version\_number.
- 3) If version\_number is different, it is decided that the AIT contents are updated, and the AIT is evaluated again using a reacquired AIT.

An AIT is not monitored when the application is activated with a URI specified using the API (replaceApplication()). If the application is activated without specifying a URI using the API (replaceApplication()), the process specified in 8.5.2.5 "Operation when an application is changed" is performed and the monitoring of the AIT is started again. Then, the specified application is activated.

AIT monitoring shall not be performed while data broadcasting is being displayed. This does not mean that an AIT cannot be acquired during data broadcasting.

Some receivers are not able to perform AIT monitoring while presenting VOD contents.

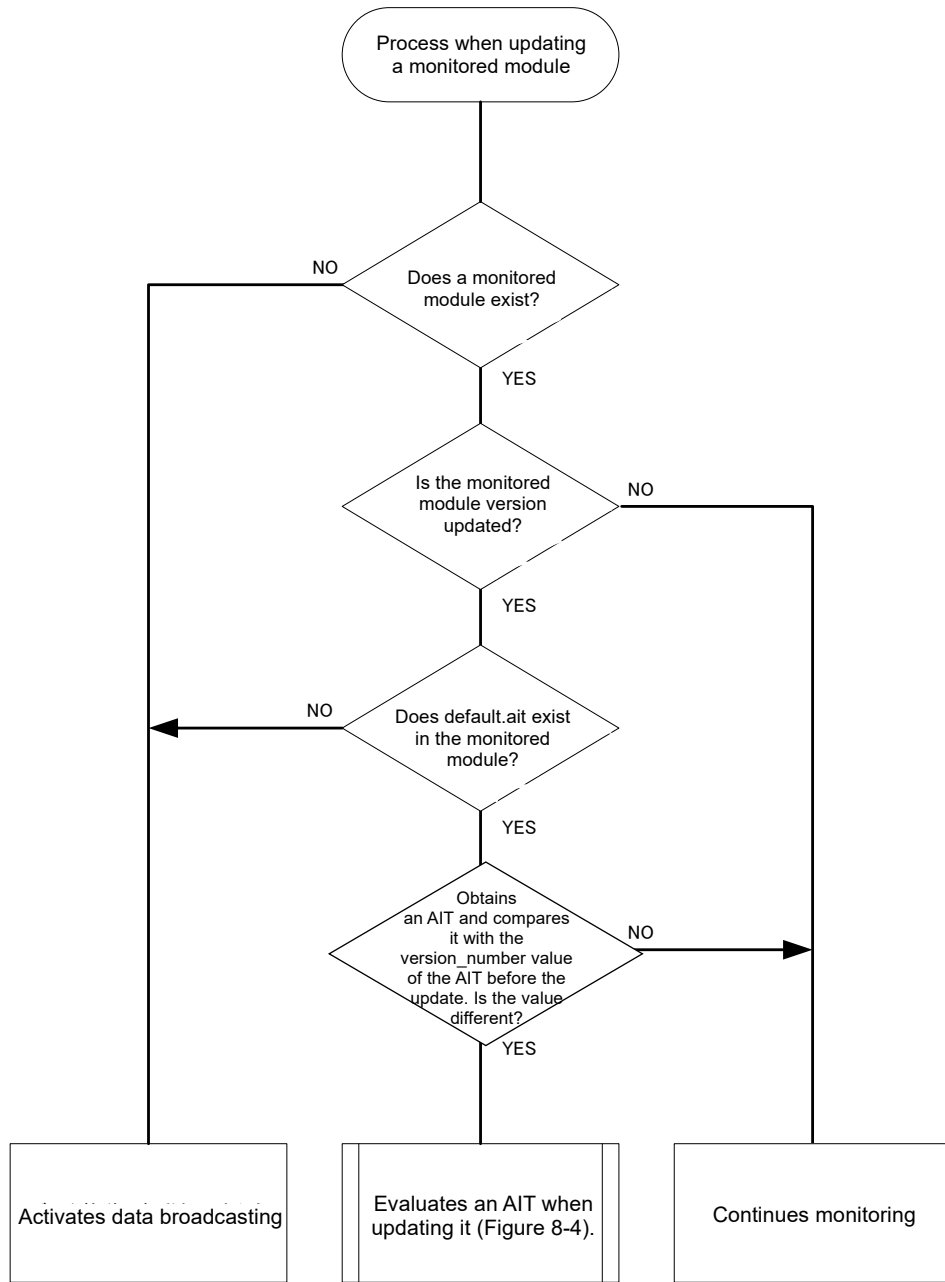


Figure 8-2 Process when a module to be monitored is updated



#### 8.3.5.2.1 Start and end monitoring of an AIT that is transmitted using a data carousel

The timing to start and end monitoring of an AIT that is transmitted using a data carousel is as follows. See 8.5.2 for the operation performed when selecting a channel after the monitoring of an AIT ends.

- Conditions for starting AIT monitoring
  - ✓ startAITControlledApp() is executed without specifying ait\_uri during data broadcasting.
  - ✓ start\_priority of a 0x40 data component in PMT is 0 when the process described in 8.5.2.1 "Operation when selecting a channel" is performed in the cases as provided in 8.5.2, such as at the time of selecting a channel, powering ON, and 0x40 data component emergence, and subsequently the AIT-controlled application that is specified based on the AIT is activated.
  - ✓ The specified AIT-controlled application is activated after executing replaceApplication without specifying URI for the application in operation when an AIT is not constantly monitored.
- Conditions for ending AIT monitoring
  - ✓ Data broadcasting is started.
  - ✓ Broadcasting service is switched.
  - ✓ The specified AIT-controlled application is activated after executing replaceApplication with the URI specified for an application in operation while an AIT is constantly monitored.

#### 8.3.5.3 Monitoring an AIT that is transmitted using a section

This is not specified in phase 1 operational specifications.

### 8.4 Operation for transmitting the broadcasting integrated HTML5 application using a data carousel <phase 1>

The broadcasting integrated HTML5 applications can be transmitted using data carousels. However, applications that are transmitted using data carousels shall be limited to auto start applications that are obtained based on the provisions described in 8.4.1 and applications included in modules that transmit the auto start applications. The operation of transmitting broadcasting integrated HTML5 applications that are not included in modules that transmit the auto start applications using data carousel is considered in the future.

#### **8.4.1 Operation for transmitting auto start applications**

An auto start application that is obtained based on the description in a monitoring target AIT when selecting a channel and automatically presented can be transmitted using a data carousel.

The location of a data carousel that is used to transmit the auto start application shall be specified in the transmission protocol descriptor and the simple application location descriptor for an AIT (limited to a constantly monitored AIT and to be transmitted through broadcasting).

See 8.3.4.3 and 8.3.4.4 for the operation of the transmission protocol descriptor and the simple application location descriptor.

For details of the transmission protocol descriptor, see 5.3.2 in Vol. 4 Application Control System of ARIB STD-B24, and for details of the simple application location descriptor, see 5.3.3 in Vol. 4 Application Control System of ARIB STD-B24.

An AIT version is not upgraded if there is no change in the description contents of an AIT when the auto start application is updated.

##### **8.4.1.1 Component to transmit the auto start application**

Any component within the range of a tag value for data broadcasting can be used to transmit the auto start application via a data carousel.

##### **8.4.1.2 Name of a module to transmit the auto start application**

The URL of an application that is specified in the simple application location descriptor needs to be described using the <module\_name>/<resource\_name> format. However, the URL shall be described as 9FFE /<resource\_name> by replacing module\_name with module\_id without 0x, because module\_name is not used in data broadcasting.

##### **8.4.1.3 Method to store resources in a module for the application**

A module transmits the auto start application as one zip file in the multi-part format by compressing all resources. The name of the zip file is autostart.zip.

A zip file shall only be compressed using the deflate format (compression method=8) without encryption.

Resources for data broadcasting cannot be contained in a module that transmits the application. The operation when the resources are contained in such a module shall be a matter of product planning.

##### **8.4.1.4 Capacity of autostart.zip**

The maximum capacity of an autostart.zip file is the maximum capacity (1 MB) that can be transmitted using one module. The maximum total file size before and after compressing an autostart.zip file is 5 MB. The maximum number of files included in an autostart.zip file is 1,024 files.

An autostart.zip file that is transmitted using a data carousel may be cached by a receiver during data broadcasting. Therefore, it is necessary for application creators to be aware that this may influence data broadcasting, such as delay in presenting data broadcasting if the capacity of an autostart.zip file is large.

#### **8.4.1.5 Auto start application caches**

If a document that is specified with `initial_path_bytes` of the simple application location descriptor for an application whose AUTOSTART is specified in a constantly monitored AIT is transmitted using a data carousel, a receiver shall constantly fix the module that transmits the auto start application to the inside of the memory while another broadcasting integrated HTML5 application is displayed via communication. The module is resident in the memory as long as the HTML browser is running until another channel service is selected.

If the ID of a module that transmits the auto start application is changed in connection with updating an AIT, the new module used after the change shall become resident in the memory.

If a module that is resident in the memory and that transmits the auto start application is updated, the new module shall be resident in the memory, regardless of the type of application currently being presented. The module used before the update shall be discarded after a new module is resident in the memory.

##### **8.4.1.5.1 Cache guidelines for application creators**

- Application creators shall keep in mind that if resources have been updated after reacquiring a module, the update is not automatically reflected to the broadcasting integrated HTML5 application that is being displayed.
- When a module that transmits the auto start application is updated, careful attention shall be given to prevent failure, taking into account the possibility that an application that is currently presented may be referring the resources included in the module.

#### **8.4.2 Operation of transmitting applications other than auto start applications**

An application that is not an auto start application can be transmitted using a module that contains an auto start application.

Such an application together with the auto start application is cached and can be referred through the broadcasting integrated HTML5 application.

The transmission of an application using a module that is not a module with an auto start application is not specified in the phase 1 specifications.

#### **8.4.3 Name space of an application that is transmitted using a data carousel**

The name space used when the broadcasting integrated HTML5 application that is transmitted via communication and broadcasting refers an application and its resources that are

transmitted over broadcasting and then transition is described below.

Individual resources that are transmitted in a manner zipped into an autostart.zip file are considered as being unzipped to the <component\_tag>/<moduleName> folder on the local http server that is using TCP port number 40080 in the receiver. Individual resources are uniquely identified by the name shown below.

Please note that resources that are transmitted as an autostart.zip file shall not be accessed from any browser other than browsers that conform to these operation specifications (excluding accesses that are made as part of operation implemented in a receiver).

`http://localhost:40080/<component_tag>/<moduleName>/<resourceName>`

Here, <component\_tag> is the name of a component that transmits the autostart.zip file, and <moduleName> is the name of a module that transmits the autostart.zip file.

The <resourceName> shall conform to the character string of the file name in the local file header of the zip file.

The alphabetic characters used for <resourceName> are not case sensitive.

The characters that can be used for <resourceName> of a resource included in an autostart.zip are as follows. Unlike <resourceName> of a resource that is stored in a module using an entity format, "/" shall not be used as echar.

```
resourceName = startChar *echar
echar        = startChar | "-" | "." | "!" | "~" | "" |
              "(" | ")" | ";" | "@" | "=" | "+" | "$" | ","
startChar    = lowalpha | upalpha | digit | "_"
lowalpha     = "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" |
              "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" |
              "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z"
upalpha      = "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" |
              "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" |
              "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z"
digit        = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
```

If resources in `http://localhost:40080/<component_tag>/<moduleName>/<resourceName>` are referred, the receiver shall perform reception processing considering that these resources are included in an autostart.zip file in the

`arib-dc://<original_network_id>.<transport_stream_id>.<service_id>/<component_tag>/`

<moduleName>/autostart.zip that is transmitted through broadcasting.

<original\_network\_id>, <transport\_stream\_id>, and <service\_id> are original\_network\_id, transport\_stream\_id, and service\_id of the service currently being used.

Directory structure which appears when autostart.zip is decompressed refers to IPTV Forum Operational Guideline,

#### **8.4.4 Operation of the application boundary and access right for an application transmitted using a data carousel**

Receivers shall process documents that are transmitted using a data carousel considering that they are always included in the application boundary and that the receivers have the access rights for all of them.

### **8.5 Activation and termination of AIT-controlled applications**

#### **8.5.1 Activation and termination of AIT-controlled applications using an AIT that is not constantly monitored**

Method B defined in 7.5.1.2 in the IPTV Standard Integrated Broadcast-Broadband System Specification shall be used. An AIT that is transmitted through broadcasting is not constantly monitored while displaying an application that is activated using an AIT that is not constantly monitored. The method for activating an AIT-controlled application first after selecting a channel for service is activation from a BML document and external activation from AIT-controlled application (optional). The process is performed without being influenced by the activation priority information. When an AIT-controlled application that has been activated using this method is terminated, the termination method described in 7.5.3 (1) "Termination at the instruction of application control information" in the IPTV Standard Integrated Broadcast-Broadband System Specification is not applied. The subsequent sections will explain the activation and termination processes for receivers for each case in which an application activation or termination may occur.

##### **8.5.1.1 Operation when activating using a BML document**

The AIT-controlled application is activated using a BML document after obtaining an AIT from the URI specified using arguments based on the specified functions and then analyzing the AIT. The operation corresponds to a case of (b) "With AIT URI specification" provided in 7.5.2.3. "Fetch/launch from data broadcasting content" in the IPTV Standard Integrated Broadcast-Broadband System Specification. For details of the operation of a function used to activate the application, see 8.9 "Broadcasting integrated HTML5 application linkage function

used on a BML document".

In phase 1 operation, an AIT transmitted via communication and an AIT transmitted using a data carousel can be specified in the AIT URI.

#### **8.5.1.2 Operation when an application is changed**

If another application is activated using the AIT-controlled application, the AIT-controlled application is activated after obtaining an AIT from the URI specified using arguments based on the function defined in the HTML document of the AIT-controlled application and then analyzing the AIT. The operation corresponds to a case of (b) "With AIT URI specification" provided in 7.5.2.2. "Fetch/launch from another application" in the IPTV Standard Integrated Broadcast-Broadband System Specification.

#### **8.5.1.3 Operation when the AIT-controlled application terminates by itself**

The operation when an AIT-controlled application in operation terminates itself based on its own description is as follows.

##### 1) Phase 0 operation specifications

If data broadcasting component exists, data broadcasting shall be activated based on auto\_start\_flag.

##### 2) Phase 1 operation specifications <phase 1>

The receiver shall reacquire the PMT, and at the same time reevaluate the activation priority including data broadcasting.

#### **8.5.1.4 Operation when activating a BML document**

If a function is used to activate a BML document on the HTML document of an AIT-controlled application that is being operated, the application is terminated first, and a BML document specified is acquired after receiving a data carousel on the broadcasting stream, and the BML document is then activated.

#### **8.5.1.5 Operation when receiving a return flag**

##### 1) Phase 0 operation specifications

When a return flag is received, the flag shall be correctly ignored.

##### 2) Phase 1 operation specifications <phase 1>

If a return flag that is in a predetermined location of a data carousel included in the service being received is received while an AIT-controlled application is being operated, the process to activate data broadcasting described in 2.1.10.2 shall be executed after the application is immediately terminated. In addition, evaluation for return flag shall not be performed, when AIT-controlled application is activated. Attention shall be paid to the fact

that some receivers cannot monitor a return flag while VOD contents are being viewed and cannot return.

#### **8.5.1.6 Operation when switching a broadcasting service**

If a broadcasting service is switched, in principle, the AIT-controlled application in operation shall be terminated and broadcasting service shall then be selected. See 8.5.2.9 for the operation when switching a service using the AIT-controlled application for phase 1 operation.

#### **8.5.1.7 Operation when broadcasting is suspended**

If broadcasting is suspended, the AIT-controlled application shall execute termination operation. However, it is not necessary to terminate the application while presenting VOD contents. The AIT-controlled application shall perform termination operation if broadcasting has been stopped when the display of the VOD contents ends.

The same operation shall be performed if broadcasting signals cannot be received for any reason.

### **8.5.2 Activation/termination of an AIT-controlled application based on an AIT that is constantly monitored <Phase 1>**

Method A, which is defined in 7.5.1.1 in the IPTV Standard Integrated Broadcast-Broadband System Specification, shall be used. In this case, an AIT is transmitted through broadcasting and a receiver constantly monitors it. Control for activation/termination of the AIT-controlled application is performed in accordance with AIT status changes based on the monitoring results. It is allowed to perform the AIT operation defined in 8.5.1 while transmitting an AIT using broadcasting signals based on this method. In this case, the receiver shall execute termination process based on the method used for activation.

Specifically, a termination method defined in 8.5.2 is used for the application activated using method A, and a termination method defined in 8.5.1 is used for the application activated using method B.

See 8.3.5.2 for how to specify and obtain a monitoring target AIT. The subsequent sections will specify the activation and termination processes of receivers for each case in which an application may be activated/terminated.

#### **8.5.2.1 Operation when selecting a channel**

Whether data broadcasting or the AIT-controlled application should be activated is determined based on 7.3.2. "Start priority control operations" in the IPTV Standard Integrated Broadcast-Broadband System Specification in the process performed when selecting a channel for the target service. If it is determined that priority should be given to the AIT-controlled application, then an application for which AUTOSTART is specified in the AIT shall be activate.

(See Figure 8-3 "Evaluation of an AIT when selecting a channel".) A method to obtain a constantly monitored AIT shall conform to the specifications provided in 8.3.2.2.1. If the AIT to be constantly monitored cannot be received, or if an application for which AUTOSTART is not specified in the obtained AIT, then it shall be determined that the priority is given to data broadcasting and data broadcasting shall be activated. It is desirable to activate data broadcasting if it is determined that a receiver has evident difficulty in obtaining the auto start application for any reason.

If the auto start application that is specified in an AIT is out of the application boundary, data broadcasting shall be activated as described in 2.1.10.2. The same operation shall be performed if a location redirected from the URL of the auto start application that is specified in an AIT is out of the application boundary.



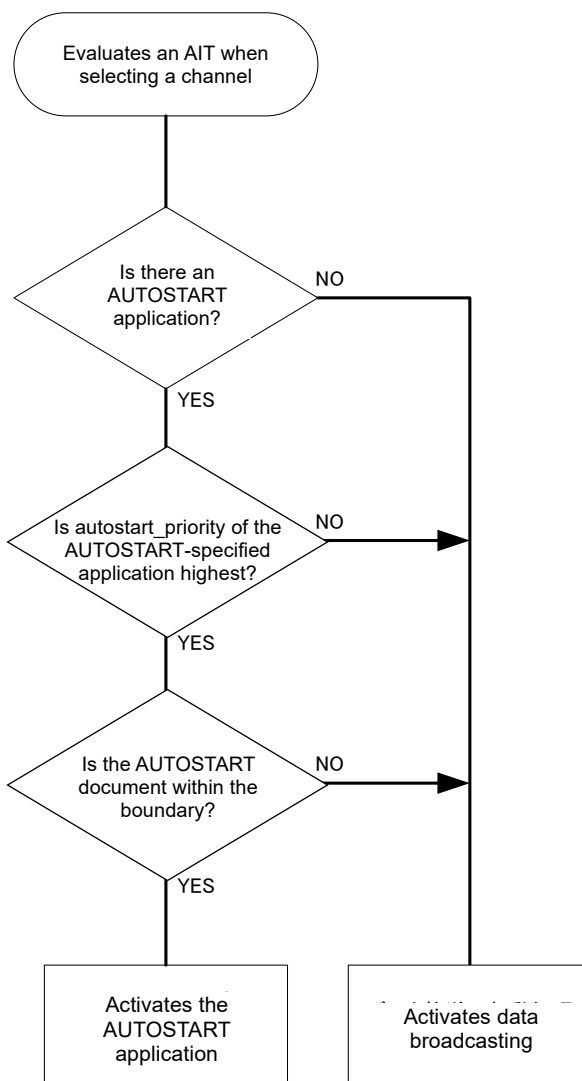


Figure 8-3 Evaluation of an AIT when selecting a channel

### 8.5.2.2 Operation when updating a PMT

One of the following processes shall be performed based on the status when updating a PMT.

- If a component that transmits an AIT with a tag value other than 0x40 is removed
  - If a component that transmits an AIT under monitoring with a tag value other than 0x40 is removed, and if the AIT-controlled application that operates based on the AIT is being operated, the application shall be terminated and the process specified in 8.5.2.1 "Operation when selecting a channel" shall be performed.
- If the default data broadcasting component (0x40) disappears
  - If data broadcasting is being operated, the display of the data broadcasting contents shall be terminated. The AIT-controlled application shall not be activated as the monitoring target is unknown.
    - The application shall be terminated if the AIT-controlled application is being operated and if a monitoring target AIT exists in a component with a tag value of 0x40. If a monitoring target AIT exists in a component whose tag value is other than 0x40, the application shall be continuously operated.
- Operation when a 0x40 component appears
  - If an application is not activated, and if a component with a tag value of 0x40 appears when such component whose tag value is other than 0x40 does not exist, start\_priority for a PMT shall be evaluated. If the evaluation result value is 0, the process specified in 8.5.2.1 "Operation when selecting a channel" shall be performed.

### 8.5.2.3 Operation when updating a constantly monitored AIT

A receiver shall detect the update of an AIT that is constantly monitored. The receiver shall acquire and retain the updated AIT to evaluate the AIT as follows. (Figure 8-4 shows the process flow when updating an AIT.)

- If AUTOSTART is specified to another application after cancelling AUTOSTART or PRESENT that is specified to the application in operation
  - After terminating the application in operation, the autostart\_priority value that is specified in the activation priority information descriptor for an application to which AUTOSTART is newly specified shall be evaluated. The application to which new AUTOSTART is specified shall be activated if the evaluation result of the application is determined to have the highest priority.
- If AUTOSTART is not specified to another application after cancelling AUTOSTART or PRESENT that is specified to the application in operation
  - The application in operation shall be terminated and the process specified in 8.5.2.1 "Operation when selecting a channel" shall be performed.

- If the application in operation falls out of the application boundary range
  - If the application in operation falls out of the application boundary range, the application shall be terminated and the process specified in 8.5.2.1 "Operation when selecting a channel" shall be performed.

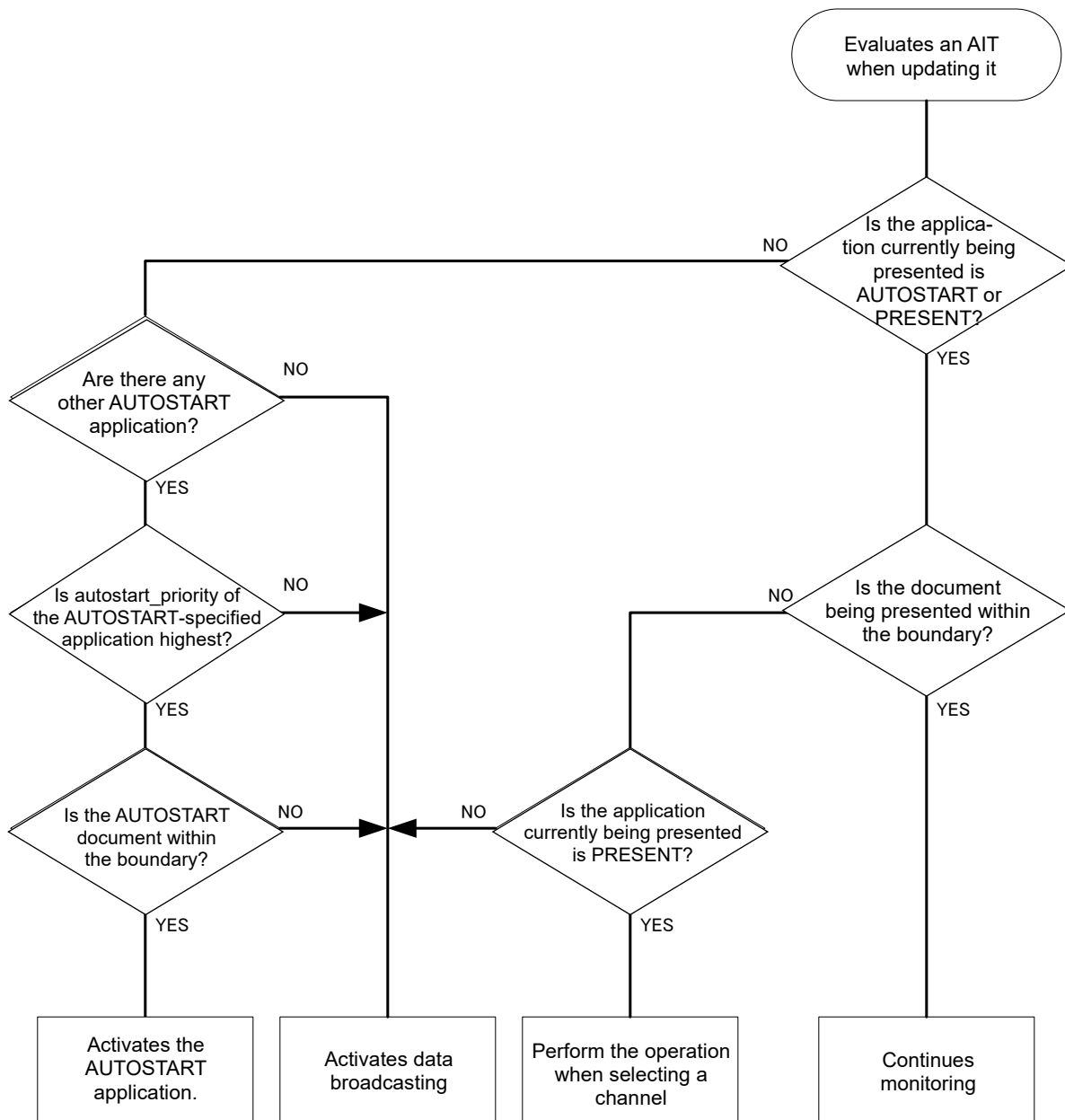


Figure 8-4 Evaluation of an AIT when updating an AIT

#### 8.5.2.4 Operation when activating using a BML document

If startAITControlledApp() is executed without specifying the argument of ait\_uri when using a BML document, an AIT that is constantly monitored shall be obtained based on the provisions in 8.3.2.2.1 and the AIT-controlled application shall be activated. In this case, the process is performed without being influenced by the activation priority information.

#### 8.5.2.5 Operation when using another application

One of the following two types of processes is applied based on whether or not the argument of uri is specified, if a transition is made from an AIT-controlled application to another application, which means that the replaceApplication function to activate another application is executed using an HTML document for the AIT-controlled application. (For details, see the IPTV Standard HTML5 Browser Specification.)

➤ When uri is not specified

If an AIT is constantly monitored, the application currently being operated shall be terminated based on the latest AIT information held and a specified application shall be activated.

If the application is being operated when an AIT is not constantly monitored, the application currently being operated shall be terminated and a specified application shall be activated after obtaining a constantly monitored AIT based on the specifications in 8.3.2.2.1. At the same time as activating the specified application, constant monitoring of the AIT shall be also started.

➤ When uri is specified

The operation specified in 8.5.1.2 shall be performed. If an AIT is constantly monitored, the constant monitoring of the AIT is terminated after transition to another application is executed with this function. The activation time can be shortened when an application activation is instructed by specifying an AIT for constant monitoring using API and so on, if obtaining the constantly monitored AIT is continued after the monitoring ends.

See 8.5.2.12 for the operation when an application that falls out of the application boundary is detected.

#### 8.5.2.6 Operation when an AIT-controlled application is terminated by itself

When an AIT-controlled application in operation terminates itself based on its own description, the process specified in 8.5.2.1 "Operation when selecting a channel" shall be performed after the application is terminated.

#### **8.5.2.7 Operation when activating a BML document**

If a function to activate a BML document is executed for an AIT-controlled application in operation, the process of activating data broadcasting that is specified in 2.1.10.2 shall be performed after the application is terminated. In this case, the process is performed without being influenced by the activation priority information.

#### **8.5.2.8 Operation when a return flag is received**

If a return flag that is in a predetermined location of a data carousel included in the service being received is received while an AIT-controlled application is being operated, the process to activate data broadcasting described in 2.1.10.2 shall be executed after the application is immediately terminated. In this case, the process is performed without being influenced by the activation priority information. In addition, evaluation for return flag shall not be performed, when AIT-controlled application is activated. Attention shall be paid to the fact that some receivers cannot monitor a return flag while VOD contents are being viewed and cannot return.

#### **8.5.2.9 Operation when switching broadcasting service**

If a broadcasting service is switched, in principle, the AIT-controlled application in operation shall be terminated and then broadcasting service shall be selected. In accordance with the provision in 8.5.2.1, application activation shall be performed as necessary under the condition in which the broadcasting service has been switched. However, the application shall be continuously operated if the broadcasting service is switched with unbound specified to call the tuneTo function, and also if a constantly monitored AIT is acquired at the selected channel, and if AUTOSTART or PRESENT is specified in the AIT as specified for the application presented before selecting the channel. In this case, the application shall be transmitted via communication both before and after selecting the channel.

#### **8.5.2.10 Operation when a monitored AIT is removed**

If an AIT that is being monitored through the DII of a data carousel is removed for a reason described below while operating an AIT-controlled application, the application shall be terminated and the process specified in 8.5.2.1 "Operation when selecting a channel" shall be performed. However, it is not required if such an operation cannot be performed while presenting VOD contents. In this case, the application shall be terminated if a monitoring target AIT is removed when the display of VOD contents ends.

- 1) A component that transmits an AIT that is being monitored uses an empty carousel.
- 2) A module that transmits an AIT that is being monitored is not transmitted.
- 3) The file of an AIT that is being monitored disappears from the resource list of a module that transmits AIT files.

#### **8.5.2.11 Operation when broadcasting is suspended**

The application shall be terminated if a broadcasting service is suspended while operating an AIT-controlled application. However, it is not required to perform such an operation while presenting VOD contents. In this case, the application shall be terminated if broadcasting is in a suspended state when the display of VOD contents ends.

The same operation shall be performed if broadcasting signals cannot be received for any reason.

#### **8.5.2.12 Operation when the application deviates from the application boundary range**

If the application deviates from the application boundary range, the process specified in 8.5.2.1 "Operation when selecting a channel" shall be performed immediately. The same process shall be performed when a deviation from the application boundary range occurs when redirected from the URI.

#### **8.5.2.13 Operation when a URI that is out of the application boundary is specified to the argument of replaceApplication**

If a URI that is out of the application boundary range is specified to the argument of `replaceApplication()`, the process specified in 8.5.2.1 "Operation when selecting a channel" shall be performed. The same process shall be performed when the destination redirected from the URI is out of the application boundary range.

### **8.6 Referring broadcasting resources using a broadcasting integrated HTML5 application**

Access from a broadcasting integrated HTML5 application, which is a broadcasting managed application, to broadcasting resources is described below. The access method shall conform to the IPTV Forum operational specifications. For details of access control, see 7.6. "Application boundary and broadcast resource access control" in the IPTV Standard Integrated Broadcast-Broadband System Specification.

It should be noted that while presenting VOD contents, in some cases, an error occurs when referring broadcasting resources or the reference of broadcasting resources may be disabled.

#### **8.6.1 Operation of a Ureg value using a broadcasting integrated HTML5 application**

A Ureg value cannot be accessed from a broadcasting integrated HTML5 application. A Ureg value is not guaranteed after activating a broadcasting integrated HTML5 application.

### **8.6.2 Operation of Greg using a broadcasting integrated HTML5 application**

A value can be written to/read from Greg using a broadcasting integrated HTML5 application. A Greg value is retained after activating a broadcasting integrated HTML5 application. A Greg value is also retained when data broadcasting contents are activated using an API from the broadcasting integrated HTML5 application. The coding scheme for the character-string data written from data broadcasting is EUC-JP. Therefore, receivers shall convert a value into a character code specified in the operation of broadcasting integrated HTML5 applications, and shall return it as a return value. On the other hand, when character-string data is written using the broadcasting integrated HTML5 application, receivers shall convert the data into EUC-JP and shall write it in Greg.

### **8.6.3 Referring video and audio components using a broadcasting integrated HTML5 application**

In addition to TV video and audio, video and audio that can be referred only from data broadcasting can be referred using a broadcasting integrated HTML5 application based on the specifications in 2.1.2.8.

### **8.6.4 Acquisition of information related to EIT [p/f]**

Information on the program that is currently being received as well as on the next program can be obtained.

### **8.6.5 Operation of NVRAM**

In order to share the NVRAM area for data broadcasting using a broadcasting integrated HTML5 application, data can be written to/read from the "A-profile memory area for the affiliation", "A-profile memory area for the specified broadcaster", and "A-profile memory area for communication purposes for the specified broadcaster" in NVRAM. Broadcasters shall appropriately manage contents to be written in these areas, taking into account that they can be accessed from AIT-controlled applications. In addition, viewers' resident area information can be read.

The coding scheme for character-string data written from data broadcasting is EUC-JP. Therefore, receivers shall convert a value into a character code specified in the operation of broadcasting integrated HTML5 applications, and return it as a return value. On the other hand, when character-string data is written using a broadcasting integrated HTML5 application, receivers shall convert the data into EUC-JP and shall write it in NVRAM.

### **8.6.6 Referring the general event descriptor**

The general event descriptor can be referred using the broadcasting integrated HTML5 application.

However, event messages that can be referred shall be limited to event messages whose event\_msg\_group\_id is 0x001. Event messages whose event\_msg\_group\_id is 0x001 can be also used for data broadcasting.

Event messages in three arbitral components can be referred simultaneously using a broadcasting integrated HTML5 application.

### **8.6.7 Referring the NPT reference descriptor <phase 1>**

The NPT reference descriptor that is transmitted using a component for data broadcasting is referred as the NPT, and is also used for broadcasting integrated HTML5 applications. For the operation related to the NPT, see 2.3.4.4, 2.3.4.5, and 2.3.4.8.

Operation to stop NPT stepping is not performed. in Phase 1. In other words, scaleNumerator/scaleDenominator = 0/1 is not operated in the NPT reference message.

When a receiver receives the NPT reference message of scaleNumerator/scaleDenominator = 0/1, the receiver shall either normally ignore the message, or operate as specified in the message.

STCmax reference message is not transmitted. The receiver shall handle STC properly even when STC has the zero circulating.

### **8.6.8 Referring the built-in sound of a receiver**

The built-in sound of a receiver can be referred using a broadcasting integrated HTML5 application.

### **8.6.9 Operation of return flag <phase 1>**

A return flag shall be monitored even when a broadcasting integrated HTML5 application is displayed. For the receiver operation, see 8.5.1.5 and 8.5.2.8.

## **8.7 Displaying broadcasting video and audio and displaying video and audio via communication**

### **8.7.1 Using VOD contents**

See Chapter 9 in the IPTV Standard Integrated Broadcast-Broadband System Specification. Broadcasting video and audio can be suspended while presenting VOD contents.

Application creators shall note that audio may be mixed in receivers that can present broadcasting video and VOD contents at the same time. In such a receiver, it is necessary to



appropriately control the audio using an application.

## **8.7.2 Using audio via communication**

### **8.7.2.1 Mixing broadcasting audio and audio via communication**

Whether or not to mix broadcasting audio and audio via communication when providing audio via communication shall be determined in accordance with the IPTV Forum operational specifications.

## 8.8 Operation of browser pseudo-objects

### 8.8.1 Operation of operation control function

(1) Operation of getBrowserSupport( )

Table 8-5 Character strings that can be specified to the getBrowserSupport( ) argument and the definitions

functionname	additionalinfo	Operation of getBrowserSupport( )
AITControlledAppEngineFunction	"IPTV-F","HTML5_ph0"	1 is returned if the broadcasting integrated HTML5 application can be executed and all phase 0 operational specifications are supported.
	"IPTV-F","HTML5_ph1"	1 is returned if the broadcasting integrated HTML5 application can be executed and all phase 1 operational specifications are supported.
AITTransportMethod	"XML","HTTP"	1 is returned if an XML-format AIT can be transmitted via HTTP as an AIT transmission method.
	"Section","DataCarousel" <Phase 1>	1 is returned if a section-format AIT can be transmitted using a data carousel as an AIT transmission method.

(2) Operation of getResidentAppVersion( )

When AITControlledApp is specified to the argument of appName, a character string\* that indicates the execution environmental profile of the broadcasting integrated HTML5 application that is supported by the receiver as the broadcasting integrated HTML5 application linkage function is returned as the return value of Array[4].

\* The governing organization shown in Appendix 20 shall define a character-string value that is returned as a return value.

### 8.8.2 Operation of AIT-controlled application linkage functions

For the operation of AIT-controlled application linkage functions to obtain an AIT via communication, see 8.9.1.1 "Operation of the AIT-controlled application linkage functions when obtaining an AIT via communication".

For the operation of AIT-controlled application linkage functions to obtain an AIT through broadcasting, see 8.9.1.2 "Operation of the AIT-controlled application linkage functions when obtaining an AIT through broadcasting".

## 8.9 Broadcasting the integrated HTML5 application linkage function used on a BML document

This section describes the specifications for realizing the broadcasting integrated HTML5 application linkage function using a BML document.

Receivers that support phase 0 operation specifications shall obtain AITs described in XML format stored on a communication server.

### 8.9.1 Operation of the BML browser used in receivers that support the broadcasting integrated HTML5 application linkage function

#### 8.9.1.1 Operation of functions used for the AIT-controlled application linkage function when obtaining an AIT via communication

(1) startReceivingAIT()

- Communication via an IP connection needs to be available to start obtaining an XML-format AIT that is stored on a communication server using this function. Therefore, it is desirable that the availability of communication via an IP connection is confirmed before executing this function. If this function is executed when communication via an IP connection is disabled, a return value indicating an error will be returned.
- The format of ait\_uri that is specified with arguments of this function is as follows.  
SCHEME “://” HOST [ “:” PORT ] “/” PATH [ “?” QUERY ]  
SCHEME = “http” | “https”  
HOST Conforms to RFC3986.  
PORT Conforms to RFC3986.  
PATH Conforms to RFC3986.  
QUERY Conforms to RFC3986.  
\* The PORT or QUERY that is enclosed with [ ] in the format above can be omitted.
- The maximum length of a URI specified by ait\_uri shall be 255 bytes.
- The minimum number of AITs that can be obtained at the same time as executing this function is 10. The maximum number shall be a matter of product planning.
- This function may return an error if an AIT acquisition process is executed with a number of AITs that exceeds the above-mentioned maximum number specified.
- The minimum total number of AITs and error histories that can be held at the same time using this function is 10. The maximum number shall be a matter of product planning.
- When an AIT is obtained exceeding the maximum number of AITs that can be held, one of the AITs can be removed from the retained AITs. Which AIT is removed from the retention target and the timing shall be a matter of product planning. In order to ensure the

acquisition of AIT information using this function, application creators need to call `getReceivedAIT()` before calling `startReceivingAIT()` with another AIT specified after calling `startReceivingAIT()` for a specific AIT.

- An AIT is retained in a BML document scope.
- The acquisition operation that is started with this function is only effective in the BML document used for calling this function. If another BML document is used, the acquisition operation needs to be executed again for the same URI using this function.
- It is not required to execute this function to activate an AIT-controlled application.

(2) `getReceivedAIT ()`

- In order to obtain an XML-format AIT that is stored on a communication server using this function, a process to start obtaining the AIT using `startReceivingAIT()` needs to be executed in advance.
- The `ait_uri` argument shall be specified to obtain an XML-format AIT that is stored on a communication server.
- The format of `ait_uri` that is specified to an argument shall conform to the operation of `startReceivingAIT()`.
- It is not required to execute this function for activating an AIT-controlled application.

(3) `startAITControlledApp()`

- The `ait_uri` argument shall be specified to obtain an XML-format AIT that is stored on a communication server.
- The format of `ait_uri` that is specified to an argument shall conform to the operation of `startReceivingAIT()`.
- If `app_id` specified with the argument and the application ID described in the obtained AIT are different, the AIT-controlled application is not activated and an error is returned.
- If this function is executed with 0 specified (a character string "0") to the arguments of both `organization_id` and `application_id`, a process is performed deeming that `organization_id` and `application_id` of the application are specified, if only one application whose `application_control_code` value is 0x01 (identification name: AUTOSTART) exists among the applications described in the AIT specified with the argument of `ait_uri`. If no applicable application exists, an error is returned and execution of the script is continued. The broadcasting integrated HTML5 application will not be activated. The operation when 0 is specified to one of the arguments of the above two shall be a matter of product planning.
- If the values specified to the arguments of `organization_id` and `application_id` are not 0, and if an application with such value combination is not described in the AIT, this function returns an error and execution of the script is continued. The broadcasting integrated HTML5 application will not be activated.

### 8.9.1.2 Operation of functions used for the AIT-controlled application linkage function when obtaining an AIT through broadcasting <phase 1>

#### (1) startReceivingAIT()

- Acquisition of an AIT that is transmitted through broadcasting can be started with this function.
- If the acquisition of an AIT that is transmitted through broadcasting can not be started with any reason, it returns failure as a return value.
- See 8.3.2.2 for the format when specifying an AIT that is transmitted through broadcasting to the argument of ait\_uri of this function.
- If the ait\_uri argument is omitted, it shall be regarded that a constantly monitored AIT is specified. However, in this case, this does not mean that constant monitoring of the AIT starts.
- The minimum number of AITs that can be obtained at the same time as executing this function is 10. The maximum number shall be a matter of product planning.
- The function may return an error if an acquisition process for an AIT is executed exceeding the above-mentioned maximum number.
- This function may return an error if an AIT acquisition process is executed with a number of AITs that exceeds the above-mentioned maximum number specified.
- The minimum total number of AITs and error histories that can be held at the same time using this function is 10. The maximum number shall be a matter of product planning.
- When an AIT is obtained exceeding the maximum number of AITs that can be held, one of the AITs can be removed from the retained AITs. Which AIT is removed from the retention target and the timing shall be a matter of product planning. In order to ensure the acquisition of AIT information using this function, it is necessary for application creators to call getReceivedAIT() before calling startReceivingAIT() with another AIT specified after calling startReceivingAIT() for a specific AIT.
- An AIT obtained through broadcasting is retained in a BML document scope.
- The acquisition operation that is started with this function is only effective in the BML document used for calling this function. If another BML document is used, the acquisition operation needs to be executed again for the same URI using this function.
- It is not required to execute this function to activate an AIT-controlled application.

(2) getReceivedAIT ()

- AIT information that is obtained and retained with startReceivingAIT() and transmitted through broadcasting can be obtained using this function.
- In order to obtain AIT information (excluding information of a constantly monitored AIT) that is transmitted through broadcasting using this function, a process to start obtaining the information with startReceivingAIT() needs to be executed in advance.
- See 8.3.2.2 for the format when explicitly specifying an AIT that is transmitted through broadcasting without omitting the ait\_uri argument of this function. If the argument is omitted, it is deemed that a constantly monitored AIT is specified in conformity with the specification.
- It is not required to execute this function for activating an AIT-controlled application.
- The return values of this function are defined as follows. The definition of Array[n][3] is newly added.

Array[n]: Control information regarding the n<sup>th</sup> application

Array[n][0]: Organization identifier (String type)

Array[n][1]: Application identifier (String type)

Array[n][2]: Application control code (Number type)

Array[n][3]: Autostart priority of the activation priority information descriptor (Number type)

Array[n][3] is assumed to be used by broadcasters to determine the priority of the application based on data broadcasting contents. For instance, Array[n][3] may be used to set a specific value that represents the priority of applications to be activated soon after a program starts, such as a data broadcast application linked to a program.

(3) startAITControlledApp()

- A broadcasting integrated HTML5 application can be activated specifying an AIT that is transmitted through broadcasting using this function.
- See 8.3.2.2 for the format when explicitly specifying an AIT that is transmitted through broadcasting, without omitting the argument of ait\_uri for this function. If the ait\_uri argument is omitted, it is deemed that a constantly monitored AIT is specified in conformity with the specification. After activating the application, monitoring of the AIT update is started. An application is activated regardless of the priority order specified in the AIT. If an argument is explicitly specified, the AIT is not constantly monitored.
- If this function is executed with 0 specified (a character string "0") to the arguments of both organization\_id and application\_id, a process is performed deeming that organization\_id and application\_id of the application are specified, if only one application whose

application\_control\_code value is 0x01 (identification name: AUTOSTART) exists among the applications described in the AIT specified with the argument of ait\_uri (including constantly monitored AITs that are specified by omitting the argument of ait\_uri). If no applicable application exists, an error is returned and execution of the script is continued. The broadcasting integrated HTML5 application will not be activated. The operation when 0 is specified to one of the arguments of the above two shall be a matter of product planning.

- If the values specified to the arguments of organization\_id and application\_id are not 0, and if an application with such value combination is not described in the AIT, this function returns an error and execution of the script is continued. The broadcasting integrated HTML5 application will not be activated.

#### **8.9.1.3 Activation of a broadcasting integrated HTML5 application from data broadcasting**

To activate a broadcasting integrated HTML5 application from BML ccontents, startAITControlledApp() shall be used. Whether or not to terminate the BML browser in accordance with the activation of a broadcasting integrated HTML5 application shall be a matter of product planning. If the BML browser is not terminated, attention shall be paid so that the contents are not displayed in a mixed manner. When displaying a broadcasting integrated HTML5 application, the provisions in 9.3 "Guarantee of Uniqueness of Broadcasting Programs and Contents" in Vol. 2 of these guidelines shall be taken into account.

In this case, the broadcasting integrated HTML5 application is not displayed on the "Internet browser screen" as described in 9.3 "Guarantee of Uniqueness of Broadcasting Programs and Contents" in Vol.2. Phase 0 operation specifications are defined based on the premise that the BML browser is terminated in accordance with the activation of a broadcasting integrated HTML5 application.

#### **8.9.1.4 Returning to the BML browser from a broadcasting integrated HTML5 application**

When BML contents that are transmitted via a data carousel over API using a broadcasting managed application are activated, the BML browser shall be displayed in the broadcasting state.

#### **8.9.1.5 Confirmation of the broadcasting integrated HTML5 application linkage function from a BML document**

Whether or not a BML document can use the broadcasting integrated HTML5 application linkage function can be confirmed using the following procedure.

- 1) Confirms the implementation of the functions used for the AIT-controlled application linkage function with getBrowserSupport ("ARIB", "APIGroup", "AITControlledApp.Start").

- 2) Confirms whether phase 0 and phase 1 functions are supported with `getBrowserSupport` ("ARIB", "AITControlledAppEngineFunction", *additionalinfo1*, *additionalinfo2*). A character string provided in Table 8-5 is specified to *additionalinfo1* and *additionalinfo2*.
- 3) Confirms a supported AIT transmission method with `getBrowserSupport` ("ARIB", "AITTransportMethod", *additionalinfo1*, *additionalinfo2*). The character string provided in Table 8-5 is specified for *additionalinfo1* and *additionalinfo2*.
- 4) Obtains the execution environment profile information of the broadcasting integrated HTML5 application with `getResidentAppVersion` ("AITControlledApp") and selects an appropriate broadcasting integrated HTML5 application, as necessary.

#### **8.9.1.6 Operation of a BML document using the broadcasting integrated HTML5 application linkage function**

- 1) Process to obtain the information of an AIT that is transmitted via communication

A process using the `startReceivingAIT()` function shall be performed in advance in order to obtain AIT information from a BML document. After the execution of the `startReceivingAIT()` function has succeeded, AIT information shall be obtained with the `getReceivedAIT()` function. However, the timing for having an AIT loaded to a receiver is uncertain due to the asynchronous operation of the `startReceivingAIT()` function as well as the loads on a communication server and depending on the communication path status. Therefore, the `getReceivedAIT()` function needs to be called repeatedly using the `setInterval()` function, etc., in order to obtain AIT information if the return value of the `getReceivedAIT()` function is 0.

It is optional to obtain AIT information using the `getReceivedAIT()` function when activating a broadcasting integrated HTML5 application.

- 2) Activation of a broadcasting integrated HTML5 application

A broadcasting integrated HTML5 application shall be activated with the `startAITControlledApp()` function via a BML document. A broadcasting integrated HTML5 application can be activated without performing the obtaining operation of AIT information using the `getReceivedAIT()` function. In this case, it is necessary to be aware that if acquisition of AIT information fails when activating a broadcasting integrated HTML5 application, it may take a long time to identify the failure. It is desirable to confirm whether or not an AIT can be obtained using the `startReceivingAIT()` and `getReceivedAIT()` functions in advance, because the `startAITControlledApp()` function is a synchronous function and the BML browser's operation is blocked until the determination of failure is returned.



## 9 AIT operation for external application control

External application is the application whose activation and termination are controlled by means such as application authentication performed outside broadcasting signals and AIT transmitted by means other than broadcasting signals. Available application is managed mainly by the receiver function through communications. The application begins to activate mainly by user's operations. On the other hand, AIT transmitted by broadcasting can restrict the execution of external application for coexistence between external application and AIT-controlled application controlled by a broadcast program or a broadcast station (e.g. a broadcasting integrated HTML5 application, etc.).

However, since operation of external application is not supposed, no particular standard is provided for AIT operation to control the application execution.

## Appendix-1 Receiver unit common fixed colors

Receiver unit common fixed colors are indicated in Table:Appendix-1. There are 64 basic colors including a half transparent color and a transparent color.

This was created based on the following policies.

- 1) First 16 colors are in accordance with the palette colors of 8-bit character codes. 1 color is transparent.
- 2) The remaining will be allocated equally in the color space.
- 3) The Alpha values are allocated equally as well.
- 4) According to the above policy, there will be 129 colors, so R,G,B,Alpha=255,255,170,128 is deleted.
- 5) Gamma correction is assumed.
  - Allocation level of RGB  
64 colors with 4 values of RGB= 0,85,170,255
  - Allocation level of Alpha  
3 values of Alpha=0,128,255

Values of RGB need to be converted to Y, Cb, Cr at the end, however, they are kept as RGB here for ease of understanding.

**Table:Appendix-1 Receiver unit common fixed colors**

**(R,G,B = 0, 85, 170, 255 Alpha= 0, 128, 255)**

Index value	R	G	B	Alpha	Name/Comments
0	0	0	0	255	Black
1	255	0	0	255	Red
2	0	255	0	255	Green
3	255	255	0	255	Yellow
4	0	0	255	255	Blue
5	255	0	255	255	Magenta
6	0	255	255	255	Cyan
7	255	255	255	255	White
8	0	0	0	0	Transparent
9	170	0	0	255	Half brightness Red
10	0	170	0	255	Half brightness Green
11	170	170	0	255	Half brightness Yellow
12	0	0	170	255	Half brightness Blue
13	170	0	170	255	Half brightness magenta
14	0	170	170	255	Half brightness Cyan
15	170	170	170	255	Half brightness White(Gray)
16	0	0	85	255	
17	0	85	0	255	
18	0	85	85	255	

Index value	R	G	B	Alpha	Name/Comments
19	0	85	170	255	
20	0	85	255	255	
21	0	170	85	255	
22	0	170	255	255	
23	0	255	85	255	
24	0	255	170	255	
25	85	0	0	255	
26	85	0	85	255	
27	85	0	170	255	
28	85	0	255	255	
29	85	85	0	255	
30	85	85	85	255	
31	85	85	170	255	
32	85	85	255	255	
33	85	170	0	255	
34	85	170	85	255	
35	85	170	170	255	
36	85	170	255	255	
37	85	255	0	255	
38	85	255	85	255	
39	85	255	170	255	
40	85	255	255	255	
41	170	0	85	255	
42	170	0	255	255	
43	170	85	0	255	
44	170	85	85	255	
45	170	85	170	255	
46	170	85	255	255	
47	170	170	85	255	
48	170	170	255	255	
49	170	255	0	255	
50	170	255	85	255	
51	170	255	170	255	
52	170	255	255	255	
53	255	0	85	255	
54	255	0	170	255	
55	255	85	0	255	
56	255	85	85	255	
57	255	85	170	255	
58	255	85	255	255	
59	255	170	0	255	
60	255	170	85	255	
61	255	170	170	255	
62	255	170	255	255	
63	255	255	85	255	
64	255	255	170	255	
65	0	0	0	128	Black
66	255	0	0	128	Red
67	0	255	0	128	Green
68	255	255	0	128	Yellow
69	0	0	255	128	Blue
70	255	0	255	128	magenta

Index value	R	G	B	Alpha	Name/Comments
71	0	255	255	128	Cyan
72	255	255	255	128	White
73	170	0	0	128	Half brightness Red
74	0	170	0	128	Half brightness Green
75	170	170	0	128	Half brightness Yellow
76	0	0	170	128	Half brightness Blue
77	170	0	170	128	Half brightness magenta
78	0	170	170	128	Half brightness Cyan
79	170	170	170	128	Half brightness White(Gray)
80	0	0	85	128	
81	0	85	0	128	
82	0	85	85	128	
83	0	85	170	128	
84	0	85	255	128	
85	0	170	85	128	
86	0	170	255	128	
87	0	255	85	128	
88	0	255	170	128	
89	85	0	0	128	
90	85	0	85	128	
91	85	0	170	128	
92	85	0	255	128	
93	85	85	0	128	
94	85	85	85	128	
95	85	85	170	128	
96	85	85	255	128	
97	85	170	0	128	
98	85	170	85	128	
99	85	170	170	128	
100	85	170	255	128	
101	85	255	0	128	
102	85	255	85	128	
103	85	255	170	128	
104	85	255	255	128	
105	170	0	85	128	
106	170	0	255	128	
107	170	85	0	128	
108	170	85	85	128	
109	170	85	170	128	
110	170	85	255	128	
111	170	170	85	128	
112	170	170	255	128	
113	170	255	0	128	
114	170	255	85	128	
115	170	255	170	128	
116	170	255	255	128	
117	255	0	85	128	
118	255	0	170	128	
119	255	85	0	128	
120	255	85	85	128	
121	255	85	170	128	
122	255	85	255	128	

Index value	R	G	B	Alpha	Name/Comments
123	255	170	0	128	
124	255	170	85	128	
125	255	170	170	128	
126	255	170	255	128	
127	255	255	85	128	

## Appendix-2 Module compression format

In case of module compression, the same compression format as PNG is used based on RFC-1950(ZLIB Compressed Data Format Specification version 3.3). Refer to Table:Appendix-2.

**Table:Appendix-2 Detailed operation of the zlib compression format**

Field	Operation
Compression Method(4bit)	8 ("deflate") only
Compression Info(4bit)	Less than or equal to 7 (less than or equal to window 32KB)
Flags	
FCHECK(5bit)	(Values specified in RFC-1950)
Preset Dictionary(1bit)	0 (no preset dictionary) only
Compression Level(2bit)	(As desired. It is ignored at the time of decoding)

### Appendix-3 Clip function in video plane

Of all the side panels attached at the time of sending video materials in a 4:3 aspect ratio in format of a 16:9 aspect ratio, the main objective of this function is for side panels in either the left side or right side to be in non-display mode (video is shifted to the left side or right side.) Therefore, the only video materials that this function can be applied to are the ones with “O” for the relationship of the following two in the following Table.

- “display\_horizontal\_size” of “sequence\_display\_extension”
- “horizontal\_size\_value” of “sequence\_header” and “aspect\_ratio\_information”

The same function for the y-coordinate direction is not operated.

		display_horizontal_size		
		No specification of value <sup>(*1)</sup>	The same value as horizontal_size_value <sup>(*2)</sup>	Different value as horizontal_size_value <sup>(*2)</sup>
aspect_ratio _information	2(4:3)	x	X	O
	3(16:9)	O	O	x

(\*1) In case sequence\_display\_extension is not used.

(\*2) In case the respective values are 540 and 544, then they are regarded as equal.

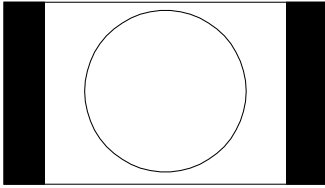
Also, continuous location changes of video object put a load on the display function of the receivers, therefore, it is not preferred.

This function is applicable only if the scaling ratio is 128/128 and regarding HD video of HD data pixel size or SD video of SD data pixel size and the width, height attribute of object elements referring to TV video, which is the clip target, will match the size of data broadcasting pixel size respectively, in other words, if the data broadcasting pixel size is 960x540, then the corresponding attributes are 960px,540px respectively, or if data broadcasting pixel size is 720x480, then the corresponding attributes are 720px,480px respectively. This function is realized in models where the “div” element is clipped by the video plane frame. Therefore, the functions are specified by the left attribute of ancestor “div” elements of corresponding video objects. Operational restrictions on other clips are in accordance with ARIB STD B-24 Appendix 2 5.1.4. For example, in case the side panel on the left side of the screen is switched to a non-display mode, “-” value is specified in the left attribute of the “div” element. BML descriptions and an example of display are indicated in the following. In case of using this function in contents, a separate display object should be located in the equivalent area on character shapes or the still image plane for the area where the video object is not located on the video plane.

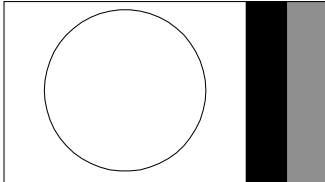
BML description example

Object allocation  
on video plane

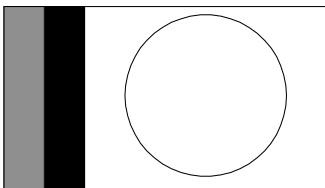
BML記述例



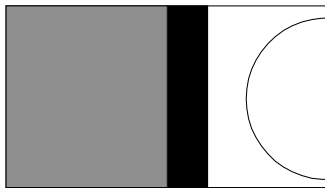
```
(*)In case it is not clipped.
<body>
<div style="left:0; top:0; width:960px; height:540px;">
<object id="HD_video"
  style="left:0; top:0; width:960px; height:540px;"
  type="video/X-arib-mpeg2" data="/-1" remain="remain"/>
</div>
</body>
```



```
<body>
<div style="left:-120px; top:0; width:960px; height:540px;">
<object id="HD_video"
  style="left:0; top:0; width:960px; height:540px;"
  type="video/X-arib-mpeg2" data="/-1" remain="remain"/>
</div>
</body>
```



```
<body>
<div style="left:120px; top:0; width:960px; height:540px;">
<object id="HD_video"
  style="left:0; top:0; width:960px; height:540px;"
  type="video/X-arib-mpeg2" data="/-1" remain="remain"/>
</div>
</body>
```



```
<body>
<div style="left:480px; top:0; width:960px; height:540px;">
<object id="HD_video"
  style="left:0; top:0; width:960px; height:540px;"
  type="video/X-arib-mpeg2" data="/-1" remain="remain"/>
</div>
</body>
```



Side panel part (video)



Background-color of body elements

Figure Appendix-3 BML description and display example



#### Appendix-4 Regarding the scaling of videos

The scaling ratio is mapped to the relationship between the size of object referring to the video in the BML document and the MPEG pixel size of the video referred to by the object. The following indicates the height, width attributes that are specifiable in object elements and a display example of videos that refer to video in BML documents.

The “display\_vertical\_size” and “display\_horizontal\_size” are specified by “sequence\_display\_extension”, and if those values are different from the “vertical\_size\_value” specified by the “sequence\_header” or values specified by “horizontal\_size\_value”, then which one the scaling will be based on depends on implementation. Therefore, in object elements that refer to corresponding video, it is assumed that the size of the area where the video is actually displayed may be different. In consideration for this case, produce contents with care to other elements covering up the object element referring to the video.

#### Appendix 4-1. “Height”, “width” attributes that are specifiable in object elements

(1) In case data broadcasting pixel size is 960x540(16:9).

Number of lines (Vertical size)	Horizontal size		Operation for each signal type				Explanation
	16:9 signal	4:3 signal Excluding HD	HD 1080 lines	SD 480 lines	SIF 240lines	QSIF 120lines	
1080	1920	1440	O	- note)	-	-	HD standard
960	1706	1280	-	O	-	-	Maximum SD
944	1680	1260	O	O	-	-	
810	1440	1080	O	O	-	-	
674	1200	900	O	O	-	-	
540	960	720	O	O	-	-	
480	852	640	-	O	O	-	SD standard Maximum SIF
404	720	540	O	O	O	-	
270	480	360	O	O	O	-	
240	426	320	-	O	O	O	SIF standard Maximum QSIF
120	212	160	-	O	O	O	QSIF standard

O · · Operated - · · Not operated

In case a 16:9 size is specified at the time of a 4:3 signal, then the receiver unit will automatically execute operations such as the insertion of side panels, etc. and the receiver unit needs to display the video part while maintaining the aspect ratio.

**ARIB TR-B14**  
**Version 6.7-E1**

note 1) The following values are specified for the width attribute and height attribute of object elements in BML documents.

height: 1/2 of the number of lines(vertical size)

width: 1/2 of the horizontal direction

(2) In case data broadcasting pixel size is 720x480(16:9) or 720x480(4:3)

Number of lines (Vertical size)	Horizontal size	Operation by signal types				Explanation
		HD 1080 lines	SD 480 lines	SIF 240 lines	QSIF 120 lines	
480	720	-	O	O	-	SD standard Maximum SIF
420	630	-	O	-	-	
360	540	-	O	O	-	
300	450	-	O	O	-	
240	360	-	O	O	O	SIF standard Maximum QSIF
180	270	-	O	O	O	
150	224	-	-	O	O	
120	180	-	O	O	O	QSIF standard
90	134	-	-	O	O	
74	112	-	-	-	O	
60	90	-	-	O	O	
44	66	-	-	-	O	
30	44	-	-	-	O	

O · · Operated - · · Not operated

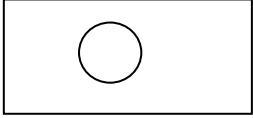
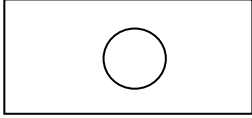
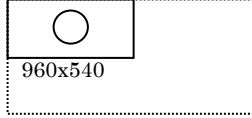
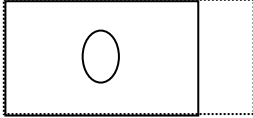
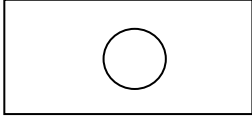
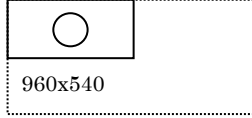
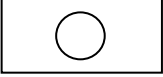
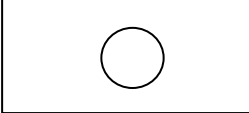

note 1) The following values are specified for the width attribute and height attribute of object elements in BML

height: number of lines (vertical size)

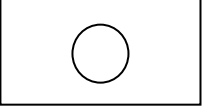
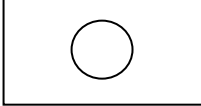


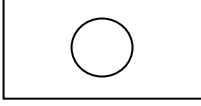
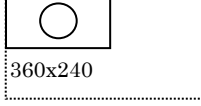
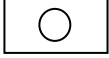
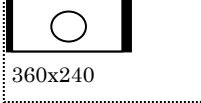
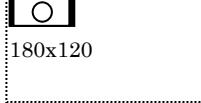

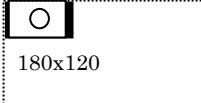
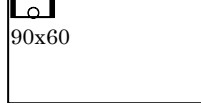
width: horizontal size

**Appendix 4-2. Display example of videos**

(Example 1) Presentation example by scaling of HD video data

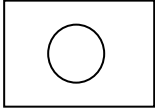
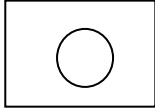
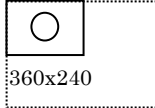
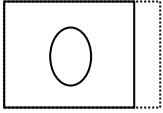
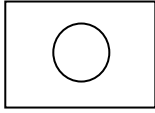
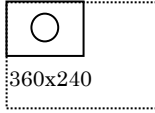

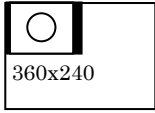
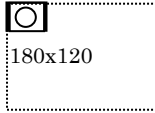

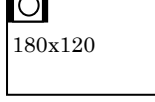
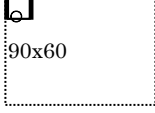
Video data and MPEG pixel size	Presentation on the video plane in case of data broadcasting pixel size 960x540 (16:9) (Scaling ratio =128/128)	Presentation on the video plane in case of data broadcasting pixel size 960x540 (16:9) (Scaling ratio =64/128)
<p>16:9</p>  <p>1920x1080</p>	 <p>1920x1080</p>	 <p>960x540</p> <p>1920x1080</p>
 <p>1440x1080</p>	 <p>1920x1080</p> <p>Vertical direction is kept as 1080, and horizontal direction is mapped to 1440-&gt;1920 to make an aspect ratio of 16:9 and is presented.</p>	 <p>960x540</p> <p>1920x1080</p>
 <p>1280x720</p>	 <p>1920x1080</p> <p>Mapped on 1920x1080(16:9) plane.</p>	 <p>960x540</p> <p>1920x1080</p>

(Example 2) Presentation example by scaling of SD video data (16:9)

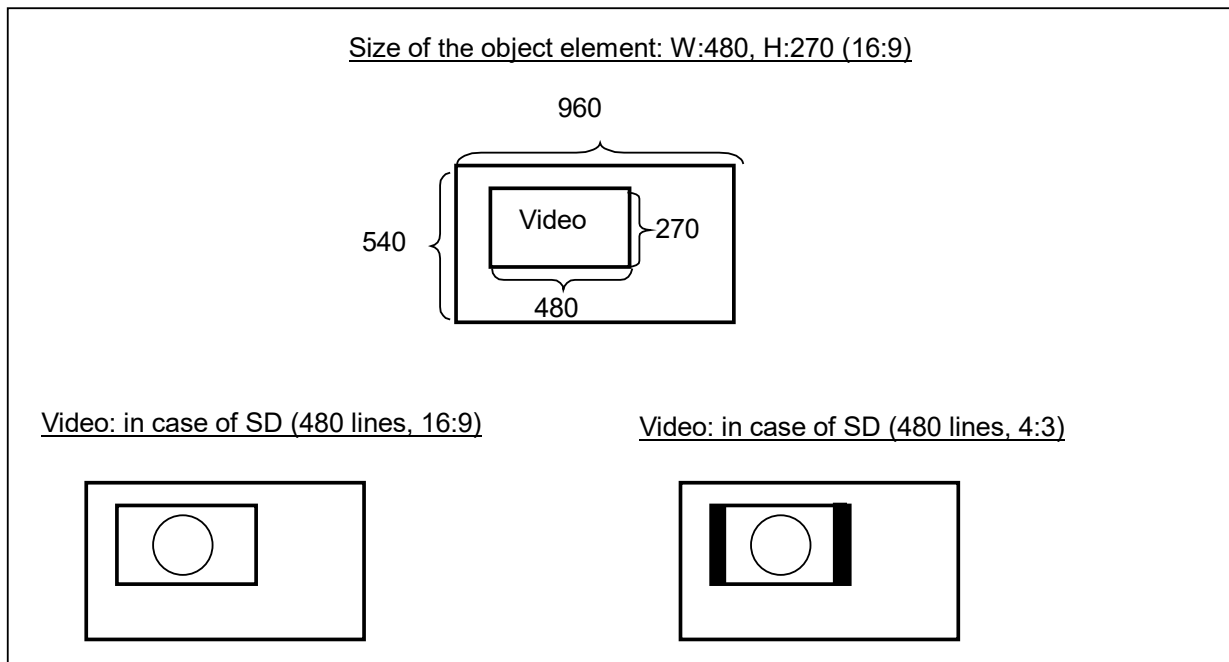
16:9	Video data and MPEG pixel size	Presentation on the video plane in case of data broadcasting pixel size 720x480 (16:9) (Scaling ratio =128/128)	Presentation on the video plane in case of data broadcasting pixel size 720x480 (16:9) (Scaling ratio =64/128)
	 <p>720x480</p>	 <p>720x480</p>	 <p>360x240 720x480</p>
	 <p>544x480 480x480</p>	 <p>720x480</p> <p>Vertical direction is kept as 480 and the horizontal direction is mapped to {544,480,352}-&gt;720 and is presented.</p>	 <p>360x240 720x480</p>
	 <p>352x240</p>	 <p>360x240 720x480</p> <p>Side panels of 4 picture elements in the left and right respectively are attached and presented in a presentation area of 360x240.</p>	 <p>180x120 720x480</p>
	 <p>176x120</p>	 <p>180x120 720x480</p> <p>Side panels of 2 picture elements in the left and right respectively are attached and presented in a presentation area of 180x120.</p>	 <p>90x60 720x480</p>

(\*) In data broadcasting, the base point of the video display image is the upper left corner.

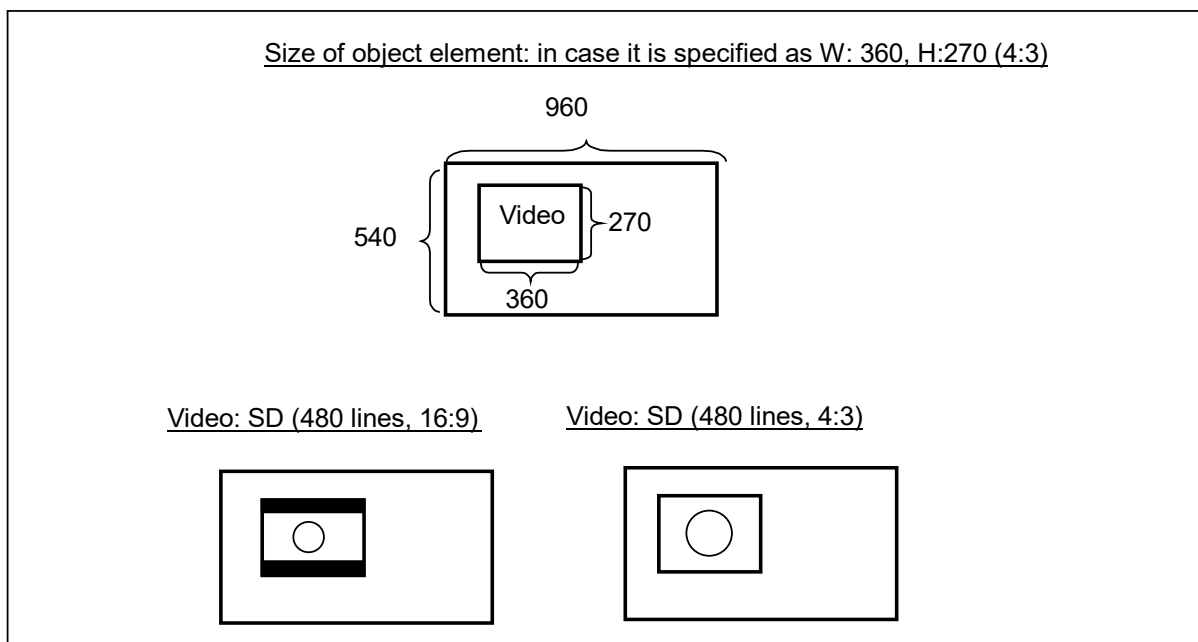
(Example 3) Presentation example by scaling of SD video data(4:3)

	Video data and MPEG pixel size	Presentation on the video plane in case of data broadcasting pixel size 720x480 (4:3) (Scaling ratio = 128/128)	Presentation on the video plane in case of data broadcasting pixel size 720x480 (4:3) (Scaling ratio = 64/128)
4:3	 <p>720x480</p>	 <p>720x480</p>	 <p>360x240 720x480</p>
	 <p>544x480 480x480</p>	 <p>720x480</p> <p>Vertical direction is kept as 480 and the horizontal direction is mapped to {544,480,352}-&gt;720 and is presented.</p>	 <p>360x240 720x480</p>
	 <p>352x240</p>	 <p>360x240 720x480</p> <p>Side panels of 4 picture elements in the left and right respectively are attached and presented in a presentation area of 360x240.</p>	 <p>180x120 720x480</p>
	 <p>176x120</p>	 <p>180x120 720x480</p> <p>Side panels of 2 picture elements in the left and right respectively are attached and presented in a presentation area of 180x120.</p>	 <p>90x60 720x480</p>

(Example 4) In case of data broadcasting pixel size 960x540(16:9), when presenting combinations of SD video



(Example 5) In case of data broadcasting pixel size 960x540(16:9), when presenting combinations of SD video (In case of 4:3 size specification)



## Appendix-5 DTD for checking the operation area for basic services

### Appendix 5-1 DTD

The objective of the following DTD is to check whether or not the BML document is in compliance with the operation area specified in this provision.

```
<!-- ===== Broadcast Markup Language (BML) 3.0 DTD [OPERATABLE] ===== -->
<!ENTITY %      ContentType "CDATA">
<!ENTITY %      Charset "CDATA">
<!ENTITY %      Character "CDATA">
<!ENTITY %      LanguageCode "NMTOKEN">
<!ENTITY %      Number "CDATA">
<!ENTITY %      URI "CDATA">
<!ENTITY %      Script "CDATA">
<!ENTITY %      StyleSheet "CDATA">
<!ENTITY %      Text "CDATA">
<!ENTITY % Events.attrib
"onclick %Script;      #IMPLIED
onkeydown %Script;    #IMPLIED
onkeyup %Script;      #IMPLIED">

<!ATTLIST a
      onfocus %Script; #IMPLIED
      onblur %Script; #IMPLIED
>
<!ATTLIST input
      onfocus %Script; #IMPLIED
      onblur %Script; #IMPLIED
      onchange %Script; #IMPLIED
>
<!ATTLIST body
      onload %Script; #IMPLIED
      onunload %Script; #IMPLIED
>
<!ATTLIST div
      onfocus %Script; #IMPLIED
      onblur %Script; #IMPLIED
```

```
    accesskey %Character; #IMPLIED
>
<!ATTLIST p
    onfocus %Script; #IMPLIED
    onblur %Script; #IMPLIED
    accesskey %Character; #IMPLIED
>
<!ATTLIST object
    onfocus %Script; #IMPLIED
    onblur %Script; #IMPLIED
    accesskey %Character; #IMPLIED
>
<!ATTLIST span
    onfocus %Script; #IMPLIED
    onblur %Script; #IMPLIED
    accesskey %Character; #IMPLIED
>

<!ENTITY % Core.attrib
" id ID #IMPLIED
class CDATA #IMPLIED
style %StyleSheet; #IMPLIED"
>
<!ENTITY % Common.attrib
"%Core.attrib;
%Events.attrib;"
>
<!ENTITY % Inlstruct.class "br | span">
<!ENTITY % Inline.class "%Inlstruct.class;
| a">
<!ENTITY % Inline-noa.class "%Inlstruct.class;">
<!ENTITY % Blkstruct.class "p | div">
<!ENTITY % Block.class "%Blkstruct.class;">
<!ENTITY % Boxed.mix "%Block.class;
| object
| input">
<!ENTITY % Br.content "EMPTY">
```



```
<!ELEMENT br %Br.content;>
<!ATTLIST br
    %Core.attrib;
    %Style.attrib;
>
<!ENTITY % Span.content "( #PCDATA | %Inline.class; )*">
<!ELEMENT span %Span.content;>
<!ATTLIST span
    %Common.attrib;
>
<!ENTITY % Div.content "( %Boxed.mix; )*">
<!ELEMENT div %Div.content;>
<!ATTLIST div
    %Common.attrib;
>
<!ENTITY % P.content "( #PCDATA | %Inline.class; )*">
<!ELEMENT p %P.content;>
<!ATTLIST p
    %Common.attrib;
>
<!ENTITY % Script.content "( #PCDATA )">
<!ELEMENT script %Script.content;>
<!ATTLIST script
    src %URI; #IMPLIED
>
<!ENTITY % Style.content "( #PCDATA )">
<!ELEMENT style %Style.content;>
<!ENTITY % A.content "( #PCDATA | %Inline-noa.class; )*">
<!ELEMENT a %A.content;>
<!ATTLIST a
    %Common.attrib;
    href %URI; #IMPLIED
    accesskey %Character; #IMPLIED
>
<!ENTITY % Object.content "EMPTY">
<!ELEMENT object %Object.content;>
```

```
<!ATTLIST object
  %Common.attrib;
  data %URI; #IMPLIED
  type %ContentType; #IMPLIED
  remain (remain) #IMPLIED
  streamposition %Number; "0"
  streamstatus (stop | play | pause) #IMPLIED
>
<!ENTITY % InputType.class "( text | password )">
<!ENTITY % Input.content "EMPTY">
<!ELEMENT input %Input.content;>
<!ATTLIST input
  %Common.attrib;
  type %InputType.class; "text"
  value CDATA #IMPLIED
  disabled (disabled) #IMPLIED
  readonly (readonly) #IMPLIED
  maxlength %Number; "40"
  accesskey %Character; #IMPLIED
  inputmode (direct | indirect | none) "none"
  charactertype (all|number|alphabet|hankaku|zenkaku|katakana|hiragana) "all"
>
<!ENTITY % Title.content "( #PCDATA )">
<!ELEMENT title %Title.content;>

<!ENTITY % Meta.content "EMPTY">
<!ELEMENT meta %Meta.content;>
<!ATTLIST meta
  name NMTOKEN #IMPLIED
  content CDATA #REQUIRED
>
<!ENTITY % Head.content "( title, meta?, style?, link?, script*, bevent? )">
<!ELEMENT head %Head.content;>
<!ENTITY % Body.content "( div | p )+ ">
<!ELEMENT body %Body.content;>
<!ATTLIST BODY
  %Core.attrib;
  %Style.attrib;
  invisible (invisible) #IMPLIED
```

```
>
<!ENTITY % Bml.content "( head, body )">
<!ELEMENT bml %Bml.content;>
<!ENTITY % bevent.content "( beitem )+ ">
<!ELEMENT bevent %bevent.content;>
<!ATTLIST bevent
    id ID #IMPLIED
>
<!ENTITY % BMLEventType "(EventMessageFired|ModuleUpdated|ModuleLocked|TimerFired
|DataEventChanged|CCStatusChanged|MainAudioStreamChanged|NPTReferred
|MediaStopped|DataButtonPressed|IPConnectionTerminated)">
<!ENTITY % BMLTimeMode "(absolute|origAbsolute|NPT)">
<!ENTITY % beitem.content "EMPTY">
<!ELEMENT beitem %beitem.content;>
<!ATTLIST beitem
    id ID #REQUIRED
    type %BMLEventType; #REQUIRED
    onoccur %Script; #REQUIRED
    es_ref %URI; #IMPLIED
    message_id %Number; #IMPLIED
    message_version %Number; #IMPLIED
    message_group_id %Number; "0"
    module_ref %URI; #IMPLIED
    language_tag %Number; #IMPLIED
    time_mode %BMLTimeMode; #IMPLIED
    time_value CDATA #IMPLIED
    object_id CDATA #IMPLIED
    subscribe (subscribe) #IMPLIED
>
<!ENTITY % link.content "EMPTY">
<!ELEMENT link %link.content;>
<!ATTLIST link
    href %URI; #IMPLIED
>
<! End of BML DTD -->
```

## Appendix 5-2 Description of the DTD declaration section

Description of the DTD declaration section is in accordance with the following.

```
<?xml version="1.0" encoding="EUC-JP" ?>  
<!DOCTYPE bml PUBLIC  
    "+//ARIB STD-B24:1999//DTD BML Document//JA"  
    "http://www.arib.or.jp/B24/DTD/bml_1_1.dtd">  
<?bml bml-version="3.0" ?>
```

## Appendix-6 Precautions for Non-volatile memory access

- Regarding the lifespan of Non-volatile memory

Non-volatile memory is assumed to be implemented using the semiconductor memory device called “Flash memory” in general. This device has a limit on how many times it can be written and that is the lifespan of this device. The maximum limit number of times of writing as of year 2000 is about 100,000 times.

If information is to be accumulated along with time of contents presentation, then global variables and Ureg should be used. Regarding temporary data memory for the passing of data between multiple services, if the Greg memory is supported, then the Greg memory should be used. Using Non-volatile memory for temporary memory of data is not recommended.

## Appendix-7 Root Certificate Descriptor

Format of Root Certificate Descriptor located in the module information area of DII is as follows.

Root Certificate Descriptor

Data structure	Bit number	Bit string display
<pre> root_certificate_descriptor(){   descriptor_tag   descriptor_length   root_certificate_type   reserved   if ( root_certificate_type == 0 ){     for ( i=0; i&lt;8; i++){       root_certificate_id       root_certificate_version     }   } else {     for ( i=0; i&lt;8; i++){       reserved     }   } } </pre>	<p>8</p> <p>8</p> <p>1</p> <p>7</p> <p>32</p> <p>32</p> <p>64</p>	<p>uimsbf</p> <p>uimsbf</p> <p>bslbf</p> <p>bslbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>bslbf</p>

root\_certificate\_type:

Root certificate type. If this field is 1, then it means the corresponding root certificate is a broadcaster exclusive certificate. If it is 0, then it means the corresponding root certificate is an general-purpose certificate. (The meanings of broadcaster exclusive certificate and general-purpose certificate are explained later.)

The following fields are located 8 times repeatedly and they correspond to the 8 memory areas of a root certificate.

root\_certificate\_id:

If the value of this field is "0xFFFFFFFF", then it means that the root certificate stored in the corresponding root certificate memory area is not included in this module.

If the value of this field is other than "0xFFFFFFFF", then it means that the root certificate stored in the corresponding root certificate memory area is included in this module. In such case, the value in this field means the ID to identify the root certificate. A unique value is allocated to root certificates operated in domestic digital broadcasting.

root\_certificate\_version:

If the value of the corresponding "root\_certificate\_id" is "0xFFFFFFFF", then value of this field is "0xFFFFFFFF" at all times. In other cases, this field indicates the version number of the root certificate. (It is not a version number of the certificate format)

## Appendix-8 getBookmarkInfo2 function

- getBookmarkInfo2(): Information of bookmarked area specified by filename is obtained.

Syntax:

```
Array getBookmarkInfo2 (  
    input String region_name  
)
```

Argument:

region\_name : URI that is indicating the entire bookmark area

Return value:

Array to store bookmark information: successful

Array[0]: Number, number of all bookmark areas that are implemented

Array[1]: Number, number of remaining bookmarks that can be registered.

Array[2]: String, URI indicating the bookmark areas that can be registered as new.

null: failure

Explanation:

Information relating to the entire bookmark area specified by “region\_name” is returned as the return value of array data. The specification method of “region\_name” is a URI indicating the entire Non-volatile memory area that is secured as the bookmark area such as “nvram://bookmark(bookmark area)” or “nvram://denbun(registration transmission area)”.

### Appendix-9 Operation of argument of “getBrowserStatus()”

The functions that can be inspected with the combinations of “statusname” and “additionalinfo” in case sProvider is "TerrP" are shown in the table below.

Character strings that can be specified as arguments of “getBrowserStatus()”

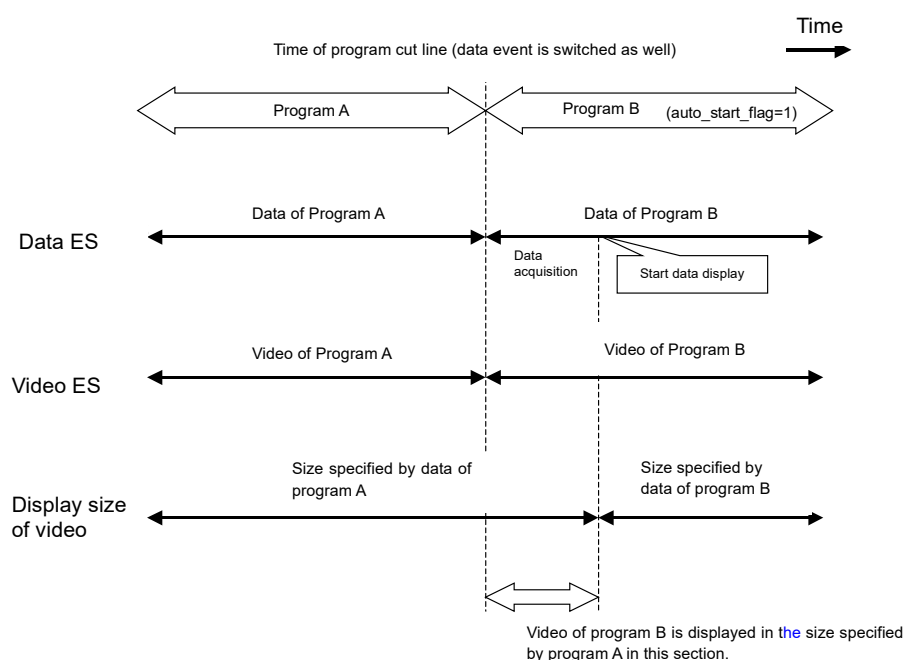
statusname	additionalinfo	Performance of “getBrowserStatus()”
IRDState	One of the following: “Broadcast” ”Link” ”UnLink”	If the browser is in a status specified by “additionalinfo”, then return 1. “Broadcast”:Reception condition of data broadcasting “Link”: Linked status “UnLink”: Unlinked status



## Appendix-10 Precautions at the time of switching video pixel size

Precautions at the time of switching video pixel size are explained.

The following figure represents the transmission and presentation of data/video at the cut line of data broadcasting programs. In this figure, a case where data broadcasting program A is finished and the data event is switched and data broadcasting program B (auto\_start\_flag=1) is started is explained as an example. In order to simplify the explanation, only 1 data ES is used for both program A and B.

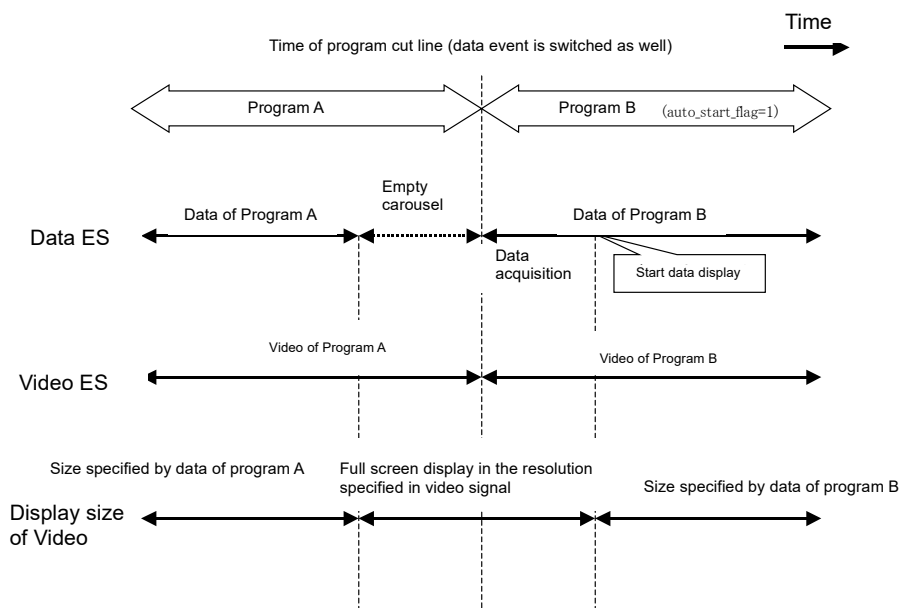


At the time of the program cut line, the display of video of program B is started, however, the display of data of program B is started with a short delay. (This is because acquisition of the startup document takes a certain length of time) During document acquisition, video continues to be displayed in the size specified before acquisition. (Refer to section 5.12.1) Therefore, video of program B is displayed in the size specified by the data broadcasting of program A during the period from the time of program cut line to the start of data display of program B.

In the above mentioned status, some attention may be required not to make scaling of the video incorrect. (Scaling not permitted in Appendix-4. For example, to display SD video of 480 lines in 1920x1080 pixel size, etc.) An example of a sending operation to avoid incorrect scaling is indicated in the following.

(Operation example 1)

It takes a certain amount of time before the video pixel size is switched, data ES is switched to empty carousel. (Figure below) At the reception time of an empty carousel, video is displayed in full screen in the original display pixel size specified to the video signal, therefore, incorrect scaling does not occur.



In case multiple data ES's are included in program A, all data ES's may be switched to empty carousel, or keep return flag of entry ES as 1 and only entry ES may be switched to an empty carousel. The same effect can be obtained by transparent contents (refer to section 1.2.2.) as well instead of empty carousels.

(Operation example 2)

Data broadcasting immediately before the switching of video pixel size, it should be combinable with all types of operated video signals. For example, if there are 2 types of video signals to be operated, which are HD1080 lines and SD480 lines, then

- Pixel size of data broadcasting itself is 960x540
- Video size specified by data broadcasting is 1440x810. (Specification of the object element in the BML document is width=720,height=405)

If data broadcasting such as the above is transmitted, then no matter which one of the 2 types of video signals is combined, incorrect scaling does not occur. (Refer to the table in Appendix-4 "(1) In case data broadcasting pixel size is 960x540(16:9)").

## Appendix-11 Operation guidelines of information on data broadcasting non-volatile memory

### 1. Basic concepts relating to the handling of data broadcasting non-volatile memory

The following concepts are the basics of non-volatile memory used in broadcasting.

- Physical functions and performances are the target of manufacturer warranties.
- Personal information that are written belong to the users.
- Management of information and licensing with users are done by broadcasters.

### 2. Definition of personal information on non-volatile memory

Personal information on non-volatile memory handled in this guideline is in accordance with the definition of JIS Q 15001 "Requirement of compliance program relating to the protection of personal information".

Personal information is information relating to individuals; information that corresponds to individuals can be identified by names, dates of birth, other descriptions, numbers allocated for each individual, symbols and other types of codes, images and audio included in corresponding information etc. (Includes corresponding information that this information alone cannot be used for identification, but can be easily referred to in combination with other information to identify the corresponding individual.)

Also, data such as "points" used for prizes/games/ premium exchanges for free gifts needs to be carefully handled from the perspective of protecting viewers, and it is regarded as personal information handled in this guideline.

### 3. Specification method of viewers

Viewers should be specified, or consent from viewers is assumed for the execution of presentation, data deletion, restoration, etc. of personal information by data broadcasting contents. As a system to identify viewers, the following are conceivable.

- Specification of individuals by passwords
- Specification of receivers by CAS card ID
- Specification of households by caller ID.

### 4. Presentation functions guidelines of specifications, etc. for handling personal information by data broadcasting contents

- The contents of the following should be described in “membership specifications”, “FAQ” or “help”, etc.
  - (1) Registered information is stored in receivers.
  - (2) Personal information stored in receivers is managed by service users (members) themselves, and in case of transferring or discarding the receiver unit, it needs to be deleted by service users (members) themselves.
  - (3) Broadcasters of corresponding services manage the registered information and maintain confidentiality and their usages should be clearly explained.
  - (4) Broadcasters who use authorization by CAS card ID should clarify the procedure of non-volatile memory data handover (restoration) at the time of exchanging CAS cards, etc.
  - (5) At the time of member registration, a disclaimer regarding the handling of points, etc. in case of data deletion should be presented to users and should be agreed to by the users.
  
- Contents should have a function to present the contents of non-volatile memory data (personal information, etc.) stored in receivers.
  - (1) Section for “confirmation of memory contents” should be created on the menu screen for member registration, etc.
  - (2) For presentation of non-volatile memory data, identification by password, for example, should be executed.

All contents of non-volatile memory data (personal information, etc.) should be presented.

However, for highly confidential information such as passwords and credit card numbers, the actual contents (numbers) will not be presented, and “\*\*\*\*” or a “stored” message should be displayed.

#### 5. Guidelines for the non-volatile memory data deletion function by data broadcasting contents

In case of operating services to store personal information in receivers in data broadcasting, each broadcaster (source of program or group) should have the following deletion functions.

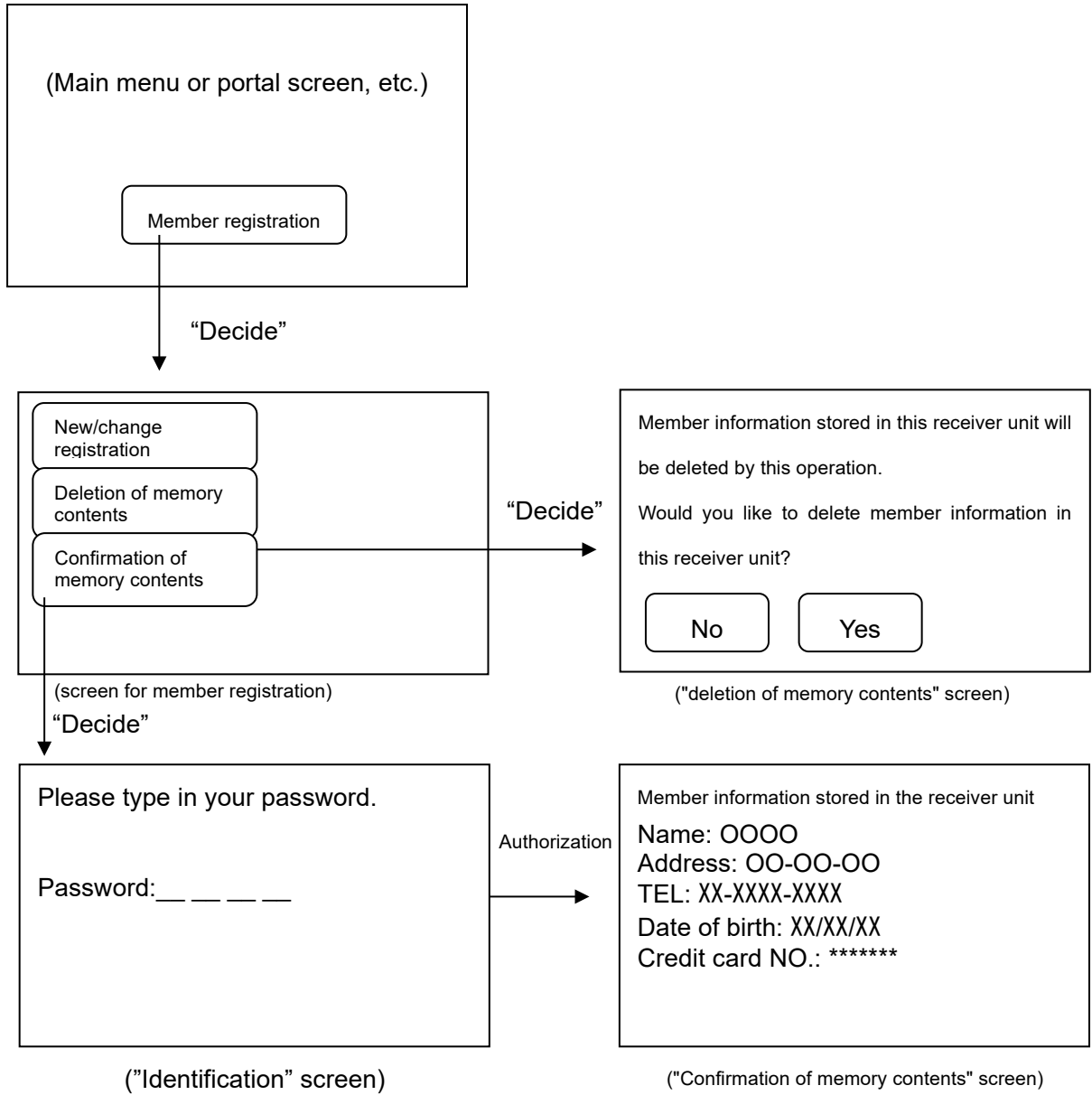
- Deletion target: Parts that the corresponding broadcaster will use exclusively among the following, area of non-volatile memory exclusively for Digital Terrestrial Television Broadcasters, Digital Terrestrial Television Broadcaster exclusive broadcasting communication common area and area exclusively for Digital Terrestrial Television Broadcaster groups.
- User interface: The following points should be implemented to improve operability for viewers and to prevent wrong operation. (Refer to the operation screen example.)

- “Memory contents deletion” section should be created in the menu screen such as “member registration”.
- For deletion of actual area of non-volatile memory, final confirmation of the viewer’s intention should be executed.
- Considering the possibilities that non-volatile memory is accessed by procedures other than the official procedures of data broadcasting and receivers, 0x00 or 0x20 should be written in all deletion target areas.

Operation to execute only if there is an instruction for withrenderal from viewers without presenting the deletion function contents at all times is possible.

- Display example of specifications for handling personal information
  - Our company will have the responsibility to manage personal information such as membership numbers, addresses, names and credit card numbers.
  - Information of members will not be disclosed to any third parties without consent from our members.
  - Our company will secure the privacy protection of all members.
  - (In case of sending data in an uplink) Among the information entered just now and information that is already registered, information of \*\*\*, \*\*\* (Example: name, address, TEL, credit card number) will be sent. The use of this information by our company is limited to this service (Example: TV shopping) only.

- Non-volatile memory data deletion operation screen example



**Appendix-12 Interaction channel communication based on connection status - Guidelines of return values for TCP/IP related functions**

	Setup for each type of priority usage line	Connection status			
		(1) Disconnected status	(4) Automatic connection / Constant connection	(2) Explicit connection	(3) Abort
Return value of "isIPConnected()"	Dial-up	0	1	2	0 *1
	Constant connection	0	1	2	0 *1
Return value of "connectPPP()"	Dial-up	NaN, 1 to -8	-100	-100	-100 *3
	Constant connection	NaN, 1 to-8 or -200 *5	NaN, 1~-8 or -200 *5	-100	-100 *3
Return value of "connectPPPWithISPPParams()"	Dial-up	NaN, 1~-8	-100	-100	-100 *3
	Constant connection	-200	-200	-100 or -200 *2	-100 or -200 *2
Return value of "disconnectPPP()"	Dial-up	-1	Return 1 and disconnect the automatic connection.	Return 1 and disconnect the explicit connection and connection parameter is deleted.	Return 1 and connection parameter is deleted.
	Constant connection	-200 or -1 *4	-200 or -1 *4	Return 1 and disconnect "connectPPP" and return to constant connection setup.	Return 1 and delete connection parameter of "connectPPP" and return to constant connection setup.

\*1 It is judged by whether the IP connection is established or not at the time of the function query.

\*2 It can be interpreted as connected by "connectPPP" even though it is a constant connection receiver or it can also be interpreted as calling "connectPPPWithISPPParams" is incorrect because of the constant connection receiver therefore, return values of -100 and -200 are permitted. Which one is returned depends on implementation.

\*3 Connection parameters cannot be overwritten in aborted status. PPP connection setup is regarded as already established and returns -100.

\*4 An interpretation to return -200 for reason that the connection setup is not dial-up, or another interpretation to return -1 because the call is not established to begin with, therefore, return values of -1 and -200 are permitted. Which one is returned depends on implementation

\*5 For receivers that cannot perform dial-up setup, then a return value of -200 is permitted.

**Appendix-13 Timing to initialize closed caption**

	At the time of updating closed caption management	Closed caption management text Format selection (SWF)	At the reception time closed caption text	At the reception time Closed caption text Data unit	Closed caption text, text screen deletion (CS)	Closed caption text, text , format selection (SWF)
Format (SWF)	Format field of management data	Value specified by corresponding SWF	Not initialized	Value specified by management data texts <sup>(*1)</sup>	Value specified by management data texts <sup>(*1)</sup>	Value specified by corresponding SWF
Display location (SDF,SDP)	Table 4-13	Value specified by management data texts <sup>(*2)</sup>	Not initialized	Value specified by management data texts <sup>(*2)</sup>	Value specified by management data texts <sup>(*2)</sup>	Value specified by management data texts <sup>(*2)</sup>
Character size controls (SSM)	Table 4-15	Value specified by management data texts <sup>(*3)</sup>	Not initialized	Value specified by management data texts <sup>(*3)</sup>	Value specified by management data texts <sup>(*3)</sup>	Value specified by management data texts <sup>(*3)</sup>
interval between characters/lines (SHS,SVS)	Table 4-15	Value specified by management data texts <sup>(*3)</sup>	Not initialized	Value specified by management data texts <sup>(*3)</sup>	Value specified by management data texts <sup>(*3)</sup>	Value specified by management data texts <sup>(*3)</sup>
Other status instruction	STD-B24	STD-B24	STD-B24	STD-B24	STD-B24	STD-B24

(\*1) Value of format field of management data if not specified by management data text

(\*2) Value of Table 4-13 if not specified by management data text

(\*3) Value of Table 4-15 if not specified by management data text.



**Appendix-14 Arguments and return values for confirming whether the receiver implements the IPTV browser and its functions**

(1) Character strings that can be specified as argument "browserName" to startExtraBrowser( )

browserName	Remarks
netTVBrowser	Browser standards represented by the browserName string are defined by the IPTV Forum (*).
IPTVBMLBrowser	

(2) Character strings that can be specified as argument to getBrowserSupport( )

functionname	additionalinfo	Remarks
ResidentApp	netTVBrowser	Browser standards represented by the additionalinfo string are defined by the IPTV Forum (*).
	IPTVBMLBrowser	
ExtraBrowserFunction	netTVVOD	Browser functions represented by the additionalinfo string are defined by the IPTV Forum (*).
	netTVDownload	
	IPTVBMLVOD	
	IPTVBMLDownload	

(3) Character strings that can be specified as argument "appName" to getResidentAppVersion( )

appName	Remarks
netTVBrowser	Browser standards represented by the appName string are defined by the IPTV Forum (*).
IPTVBMLBrowser	

(4) Character strings to be used as item names in the return value Array[4] of getResidentAppVersion( )

appName	Character strings to be used as item names
netTVBrowser	Defined by the IPTV Forum (*).
IPTVBMLBrowser	

(\*) Refer to the IPTV Forum specifications (URL <http://www.iptvforum.jp/standard/>).

## **Appendix 15 Operation when a required function is changed to an optional function due to revision**

Implementation of a function that is required in ARIB STD-B24 cannot be confirmed using a method to confirm the implementation of an extended function for broadcasting specifying "APIGroup" to functionname of getBrowserSupport() based on the specifications defined in 5.6.7 in Appendix 2, Vol. 2 of ARIB STD-B24. If a required function is changed to an optional function due to revision of the specifications, it is necessary to confirm that the function is not implemented specifying "Unsupported" to functionname and specifying a value shown in the table in 5.12.6.9 (14) "Operation of getBrowserSupport()" to additionalinfo.

The return values of getBrowserSupport() when "Unsupported" is specified to functionname are described below.

Return value:

- 1: The specified function is not implemented.
- 0: The specified function is implemented or the specified additionalinfo is not defined.

A required function can be changed to an optional function by newly defining an additionalinfo value to be used. A receiver returns 0 before newly defining the additionalinfo value, as each character string specified with the argument is unknown. Therefore, returning 0 means that the specified function is implemented.

However, it is necessary to be aware that whether the implementation of a function is required or optional may be different between digital terrestrial television broadcasting and BS digital satellite broadcasting.

## Appendix 16 Method to confirm the implementation of the BASIC procedure

The table below shows the return values of `getBrowserSupport()` to determine whether a communication function based on the BASIC procedure is implemented by taking a case where the BASIC procedure is optional as an example.

Implementation of the BASIC procedure and the return values of `getBrowserSupport()`

Receiver type	BS digital satellite broadcasting		Digital terrestrial television broadcasting	
	Before the addition of "Unsupported"	After the addition of "Unsupported"	Before the addition of "Unsupported"	After the addition of "Unsupported"
Implementation of the BASIC procedure	Required	Optional	Required for receivers with a modem	Optional
<code>getBrowserSupport()</code> return values when specifying "ARIB" to <code>sProvider</code> , "APIGroup" to <code>functionname</code> , and "Com.BASIC.Basic" to <code>additionalinfo</code>	Not yet specified	Not yet specified	Not yet specified	Not yet specified
<code>getBrowserSupport()</code> return values when specifying "ARIB" to <code>sProvider</code> , "Unsupported" to <code>functionname</code> , and "Com.BASIC.Basic" to <code>additionalinfo</code>	0 (Not yet specified) (Note 1)	BASIC procedure implemented: 0 BASIC procedure not implemented: 1	0 (Not yet specified) (Note 2)	BASIC procedure implemented: 0 BASIC procedure not implemented: 1
<code>getBrowserSupport()</code> return values when specifying "ARIB" to <code>sProvider</code> , "TransmissionProtocol" to <code>functionname</code> , and "datalink" and "BASIC" to <code>additionalinfo</code>	BASIC procedure implemented: 1 BASIC procedure not implemented: 0	BASIC procedure implemented: 1 BASIC procedure not implemented: 0	BASIC procedure implemented: 1 BASIC procedure not implemented: 0	BASIC procedure implemented: 1 BASIC procedure not implemented: 0

(Note 1) It can be determined that the BASIC procedure is implemented because the implementation of a modem is required.

(Note 2) Whereas there may be receivers with a modem that does not support the BASIC procedure, whether or not the BASIC procedure is implemented cannot be determined since 0 will be returned in both cases as the return value of `getBrowserSupport()`.

Based on the table above, the implementation of the BASIC procedure can be determined for BS digital satellite broadcasting and digital terrestrial television broadcasting, specifying the character strings as shown in the table below to the argument of `functionname`.

Argument values and return values for each medium

	Arguments of the getBrowserSupport()		Return value	
	functionname	additionalinfo	With BASIC procedure	Without BASIC procedure
BS digital satellite broadcasting (note)	TransmissionProtocol	“datalink” and “BASIC”	1	0
	Unsupported	“Com.BASIC.Basic”	0	1
Digital terrestrial television broadcasting	TransmissionProtocol	“datalink” and “BASIC”	1	0

(Note) It is desirable to specify "Unsupported" to functionname.

## Appendix 17 How to realize the VOD (integrated service type 2) function

- Overview of the VOD (integrated service type 2) function and implementation by receivers

The VOD (integrated service type 2) function allows for the use of the VOD function as part of the data broadcasting service. In this specification, a method to directly activate the VOD function from data broadcasting in the linked status, not through the IPTV browser, is provided.

A reference model for receivers that this specification assumes to be one that can realize the above-mentioned method is a "two-resource model" that has special resources for the VOD function as provided in 7.2.2.4.1 "Reference model for receivers" and that can replay VOD contents while retaining the service provided before activating the VOD function. However, actual receiver implementation is not limited to this. For instance, the above-mentioned method can be realized by a receiver (hereinafter referred to as resource model 1) that apparently behaves like the two-resource model above, terminating a service being provided in connection with the activation of the VOD function and reestablishing the linked status for data broadcasting after the VOD function is terminated. Precautions for resource model 1 receiver implementation are described below.

- Precautions when starting the VOD function

The VOD function shall be activated based on 7.2.5.2 "Execution sequence to activate the VOD function". Resource model 1 receivers shall perform the following processes when activating the VOD function.

- Termination of a service being displayed

A service that has been being provided can be terminated when activating the VOD function. The BML browser can also be stopped when terminating a service being provided.

- Storing the base URI directory

This specification considers that as VOD contents are part of the data broadcasting service, linked status continues after VOD contents are replayed. Therefore, the URI of a BML document that is included in the first argument of the extended function of startVOD() for broadcasting as a query parameter and is displayed after the VOD function ends, needs to be located under the base URI directory of the communication contents that are used to activate the VOD function. In addition, the URI will remain in the linked status after the display of the VOD contents is terminated. Therefore, the base URI directory shall be retained even when the BML browser is stopped while replaying the VOD contents.

- Precautions when replaying VOD contents

It is expected that the time taken to redisplay BML documents after the termination of the VOD function can be reduced by avoiding stopping the BML browser while VOD contents are being replayed. However, it is necessary to perform the following processes in order to avoid stopping the BML browser while replaying VOD contents.

- Disables the BML browser while replaying the VOD contents in order to prevent the BML browser's operation from influencing on the VOD contents display, such as video and audio.
- Avoids performing the timeout process for the display in connection with the PMT or DII reception while VOD contents are being replayed.

- Precautions when VOD function ends

It is desirable to redisplay a BML document and then reestablish the linked status as soon as possible after the VOD function ends in order to maintain the continuity of contents and to reduce discomfort to users. The VOD function shall end in conformity with the sequence described in 7.2.5.5 "Sequence to terminate the VOD function". However, it is necessary for resource model 1 receivers to perform the following process when the VOD function ends.

- Reestablishment of a link

After the VOD function ends, the BML browser is activated if it is stopped. Then, a BML document specified in the URI that is included in the first argument of the extended function for broadcasting of startVOD() as a query parameter and that is displayed after the VOD function ends shall be obtained and displayed. At the same time, service that is specified with a service identifier is set to Array[0] that is the second argument of startVOD() and a link is reestablished as described below.

If a BML document to be displayed after the VOD function ends cannot be obtained, service shall be reselected. The receiver operation when a link cannot be reestablished shall be a matter of product planning.

Basic operation of a receiver when reestablishing a link (guidelines)

- 1) If the TV video and audio (TV video: component\_tag = 0x00 to 0x0F; TV audio: component\_tag = 0x10 to 0x2F) are included in the PMT second loop, they are replayed.

- 2) An entry component (component with a component\_tag of 0x40) is identified among the components inserted in the PMT second loop.
- 3) If data\_component\_id of the data component descriptor for the entry component is 0x000C, it is determined that a data broadcasting program that uses the XML-based multimedia coding scheme is received. If a data coding scheme that is not supported by the receiver is used, the BML browser is terminated.
- 4) A content group identifier (service\_id expressed in a character string in hexadecimal in "0XXXXX" format) is written to Ureg[0].
- 5) The monitoring of DII that is transmitted using a component specified in 2.3.1.7 "Basic behavior of the receiver during data broadcasting program presentation" is started.
- 6) The restrictions as described in "Restrictions on BML documents before reestablishing the linked status" in 7.2.5.7 "Operation of BML contents for VOD (integrated service type 2) services" are cancelled.
- 7) After steps 1 to 6 above, the receiver changes the return values of getBrowserStatus(), which is an extended function for broadcasting. Specifically, the return values when the function is called specifying "TerrP", "IRDState", and "Link" as sProvider, statusname, and additionalinfo, respectively, for the getBrowserStatus( ) arguments are changed to 1.

After VOD contents are terminated, NaN shall be returned as a return value when getBrowserStatus( ) is called by specifying the combination of the arguments above until the return values are changed in this step.

## Appendix 18 Operation of a CorrelatingID

A "CorrelatingID", which is an ID of a receiver that is used as hardware and which is used for the receiver if the receiver supports the IPTV download/VOD (integrated service type 2) function, shall be operated as follows.

- A CorrelatingID is an identifier (ID) to identify a terminal that is shared between the IPTV service and broadcasting service.
- Broadcasters can associate a receiver with information that is related to a terminal managed by an IPTV service provider by obtaining a CorrelatingID using a BML document and sending it to the IPTV service provider's server.\*
- A CorrelatingID value is assigned to each IPTV service supported by a receiver. If the receiver supports multiple IPTV services, multiple CorrelatingID values corresponding to individual services are used. In this case, multiple IPTV services may have the same CorrelatingID value.\*
- A CorrelatingID of each IPTV service is obtained by specifying a value corresponding to each IPTV service to the argument of type for getIRDID( ).
  - For the value of the argument of type and return value format to obtain a CorrelatingID of specific broadcasting integrated IPTV service, see the IPTV Standard Specifications of IPTV Service Approach by Integration with Broadcasting.
  - The value range of the argument of type that is used to obtain CorrelatingID shall be 0xE0001 to 0xE0FFF.

\* The "IPTV service" here refers to the service that is associated to each character string in the return value of Array[4] of getResidentAppVersion( ) in a one-to-one manner.

When broadcasting integrated IPTV service is provided in collaboration with an IPTV service provider and a broadcaster, it is necessary for both parties to specify the methods for operating CorrelatingID beforehand.

A CorrelatingID is used to associate the information registered for IPTV service with each receiver that support the IPTV download/VOD (integrated service type 2) function. IPTV service providers and broadcasters shall not use Correlating ID for other purposes. If a CorrelatingID is sent from a receiver to an external server or another location over the Internet, the IPTV service provider/broadcaster who provides the IPTV download/VOD (integrated service type 2) services shall take responsibility for the transmission and management of the CorrelatingID.



## Appendix 19 Operation of a TVsetID

A "TVsetID", which is an ID of a receiver as hardware and which can be obtained by a receiver that supports the method as provided in Part 2, Vol. 5 of these guidelines as a method used for protected free broadcasting programs, shall be operated as follows. The operation of a TVsetID in receivers that do not support the method of Part 2, Vol. 5 of these guidelines shall be a matter of product planning.

- A TVsetID shall be 128 bits. The structure is as follows.

ID to identify a manufacturer: 16 bits

ID to identify a receiver: 80 bits

CRC: 32 bits

- An ID (16 bits) that identifies a manufacturer is placed as the highest 16 bits of a TVsetID in order to avoid duplication between receiver manufacturers. The ID is operated as follows.

0x0000 to 0x00FF A value (1 byte; assigned by ARIB) that is assigned to maker\_id used for table\_id\_ext in the software download trigger table defined in ARIB STD-B21 12.2.1.1 is placed at the lowest 8 bits of the ID that identifies a manufacturer.

0x0100 to 0xFFFF (Reserved)

- An ID (80 bits) that identifies a receiver needs to be assigned such that duplication in all receivers that use a TVsetID for each receiver manufacturer is avoided. Whereas the implementation method is not specified, if a random number is used to create an ID, it is necessary to ensure sufficient accuracy to practically leave the possibility of duplication out of consideration, following the guidelines described below.
- CRC shall be placed at the lowest 32 bits of a TVsetID. The CRC is appended to 96 bits consisting of an ID for identifying a manufacturer and an ID for identifying a receiver. As the generator polynomial, CRC-32 shall be used.
- The implementation of a function to initialize and change a TVsetID value when a receiver is transferred or discarded shall be a matter of product planning. However, if a receiver implements an initialization function, it is necessary that an ID value that is different from that before initialization is assigned, and also that an ID value that will not be duplicated in all receivers is assigned after initialization. If a random number is used to create an ID when initializing, it is

necessary to ensure sufficient accuracy to practically leave the possibility of duplication out of consideration, following the guidelines described below.

- It is necessary to be aware that the possibility of a TVsetID being duplicated is not zero when a TVsetID is used for services such as those that involve personal information for which the duplication of an ID is completely unacceptable.

Guidelines when a random number is used:

If a random number is used to create an ID, a random number longer than 64 bits needs to be generated in order to reduce the possibility of creating a duplicated to the extent possible. (If IDs with 64-bit length each are created and are randomly assigned to about 4.3 billion [ $2$  raised to the 32nd power] receivers, the same ID for one or more receivers will be created with a probability of 50%.)

Methods for generating random numbers are as follows: a physical random number that is generated using a physical means (noise generated using an electric circuit, etc.); an arithmetic random number generated using arithmetic expressions (program, etc.); or a combination of these. In either case, a method that can generate a random number with greater randomness needs to be employed.

## Appendix 20 Governing organization

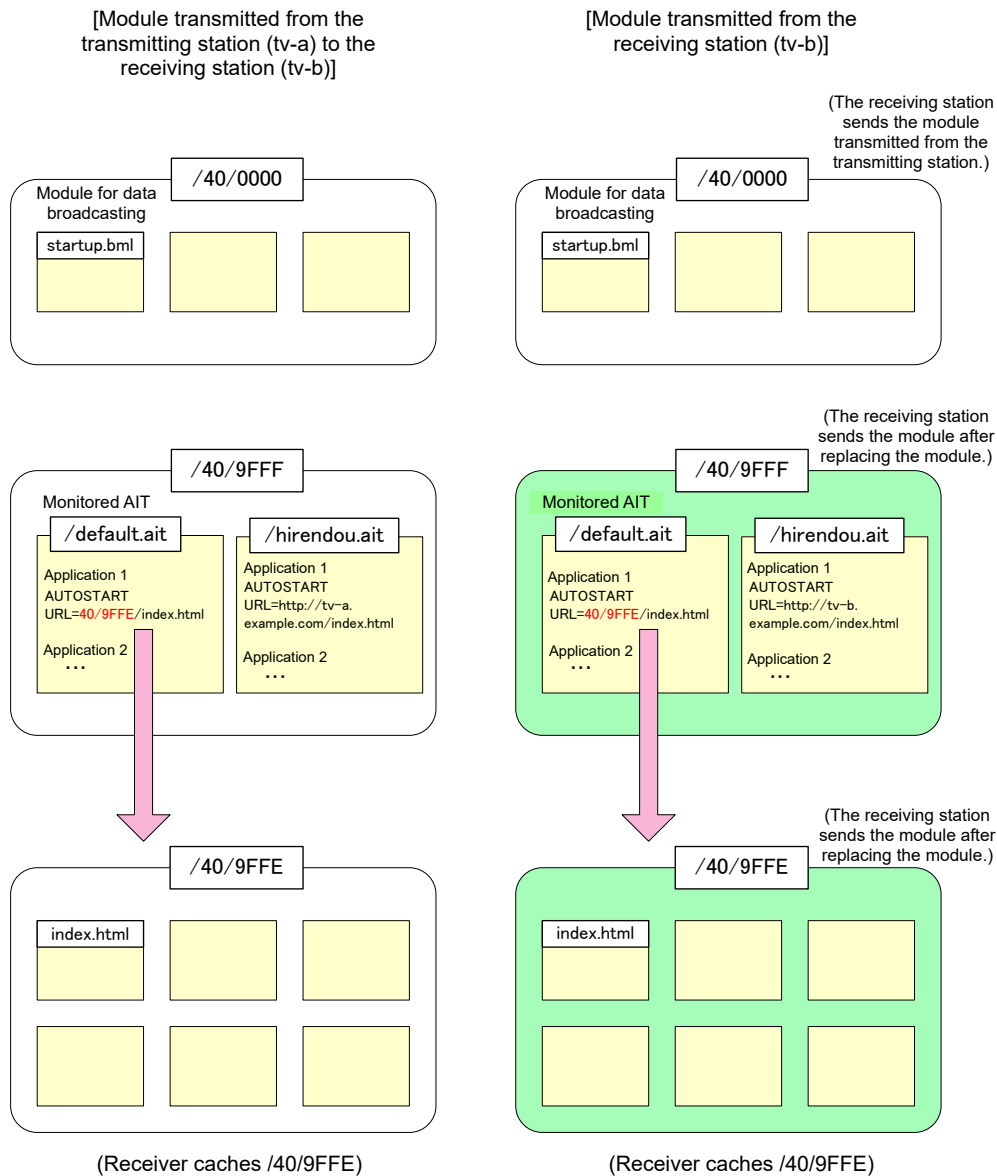
IPTV Forum Japan (URL: <http://www.iptvforum.jp/>)

## **Appendix 21 Operation example of AIT module transmission among affiliate stations**

The figures below shows an outline of operation regarding AIT transmission among service provider stations in individual affiliations. Each figure shows operation examples of modules that are transmitted from a transmitting station to a receiving station and modules that are transmitted from the receiving station. These figures show only examples of conceivable operation and they do not represent actual operations.

### Operation example 1:

This shows an example of operation in which the receiving station substitutes a module to transmit the auto start application with a module to transmit an AIT and thereby activates a unique application.



Operation example 2:

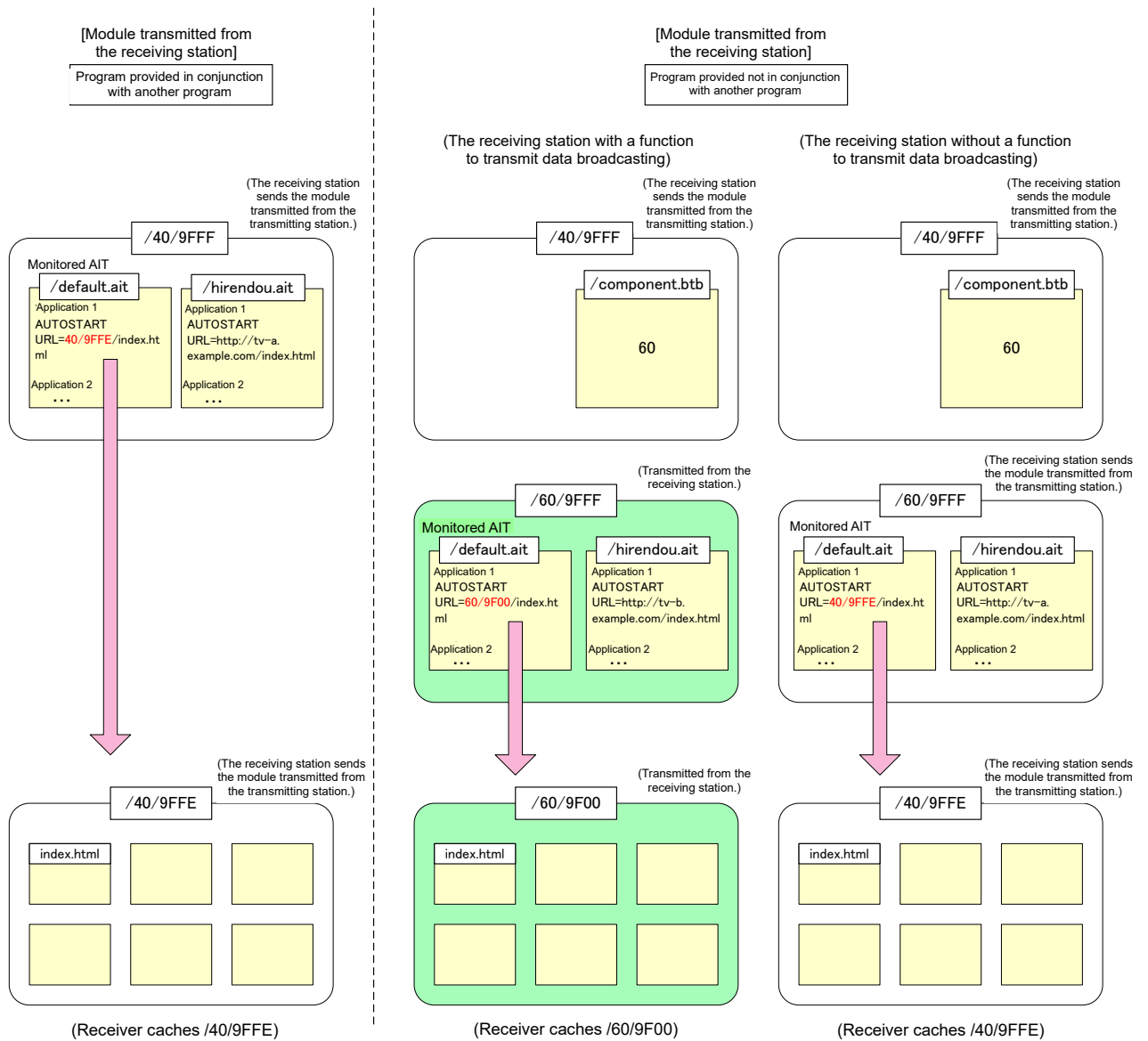
This shows an operation example (1) in which the ES to transmit an AIT is changed based on whether or not a program is provided in conjunction with another program.

If a program is provided in conjunction with another program, an AIT is transmitted using a component with a component tag of 0x40 without using component.btb for a /40/9FFF module. If a program is provided not in conjunction with another program, the component with a component tag of 0x60 is specified to a component that transmits an AIT, using component.btb.

If a program is provided not in conjunction with another program, a receiving station with a data transmission device displays its own contents that are not in conjunction with other contents after replacing the entire ES (0x60) that transmits an AIT for a non-conjunctive

application. A receiving station that does not have a transmission device sends the ES as-is to a transmitting station.

In this example, a component that has a component tag of 0x40 transmitted from the transmitting station will always be transmitted as-is by the receiving station.



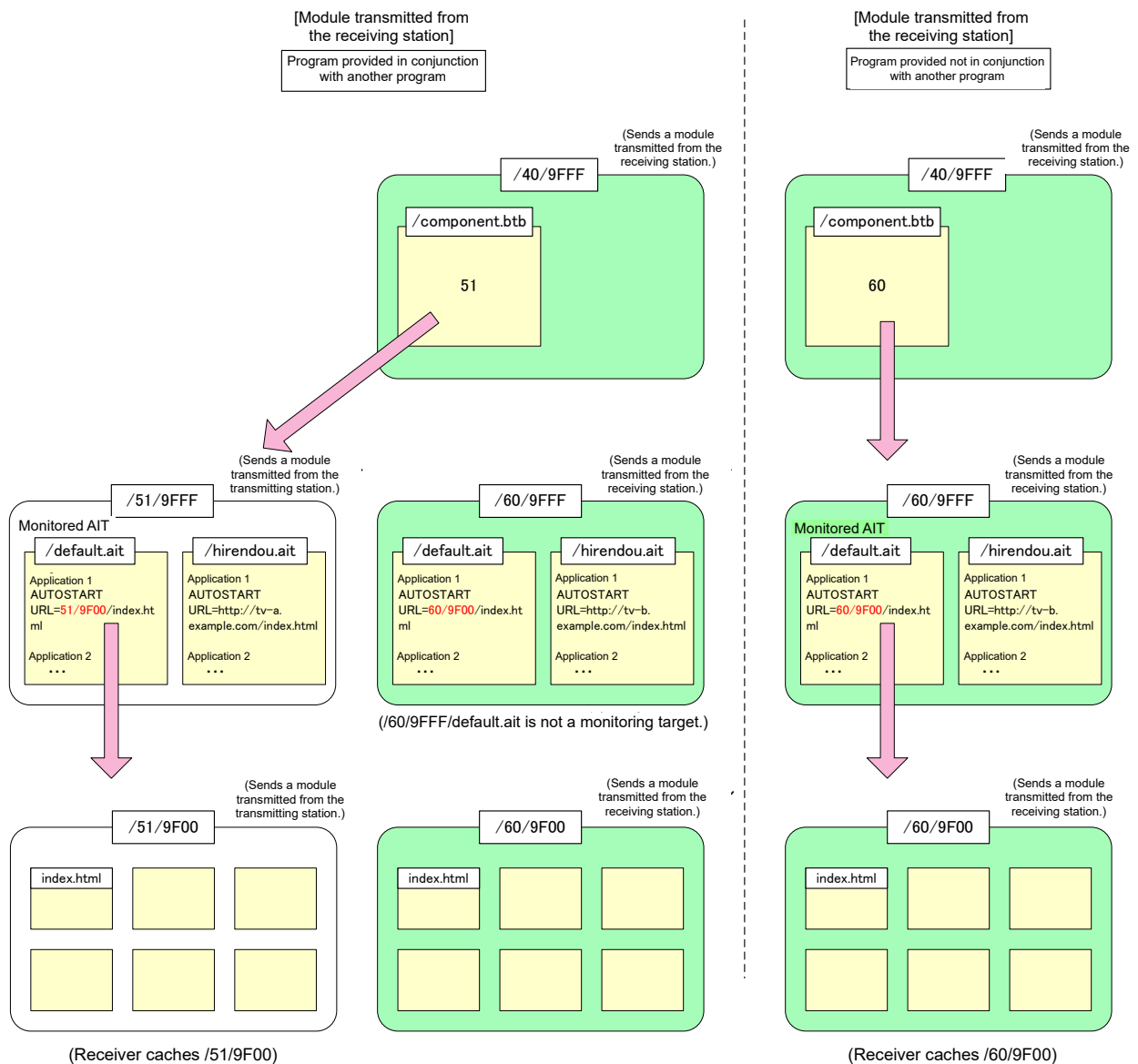
Operation example 3:

This shows an operation example (2) in which an ES to transmit an AIT is changed based on whether or not a program is provided in conjunction with another program.

In component.btb, a component with a component tag of 0x51 is specified during the provision of a program provided in conjunction with another program, and a component with a

component tag of 0x60 is specified during the provision of a program not in conjunction with another program.

An ES (0x51), including an AIT for a program conjunction application created by a transmitting station, is also transmitted from a receiving station for a program that is provided in conjunction with another program. As for a program that is provided not in conjunction with another program, an ES (0x60), including an AIT that is uniquely created by a receiving station, is transmitted to display the receiving station's own non-conjunctive contents.



<Intentionally blank>



**[Section 3] Operational specifications related to profile B  
(T.B.D.)**

<Intentionally blank>

## **[Section 5] Operational provisions of common digital receiver units for terrestrial/BS/wide bandwidth CS**

### **1 Introduction**

[Section 5] provides functions required from digital terrestrial data broadcasting reception of common digital receiver units (common receiver unit hereafter) for terrestrial/BS/wide bandwidth CS.

### **2 Functions required in common receiver units**

Common receiver units should at least be equipped with functions described in TR-B14 and TR-B15 Part 2 regarding the media they are capable of receiving (terrestrial/CS/BS).

#### **2.1 RAM**

Greg should be shared in all receivable media. Greg values should be kept even while presenting communication contents that do not support Greg. It is recommended to keep Greg values even while presenting other media that do not support Greg.

If Greg values cannot be kept for some reason during presentation of other media, etc., the receiver unit should initialize a Greg value upon first presenting media that support Greg afterwards.

#### **2.2 NVRAM**

Bookmark areas, registration calling areas and general-purpose root certificate areas should be shared with each media.

Access availability from other media in NVRAM areas for each media is as provided in Table 2-1.

**Table 2-1 NVRAM's inter-media correlation in common receiver units**

Media	Area name	Access from BS	Access from CS	Access from terrestrial
BS	Broadcaster common area	RW	×	R
	Broadcaster specific area	RW	×	×
	Broadcaster specific, broadcasting and communication common area (2KB per broadcaster)	RW (Only BML3.0)	×	×
CS	Broadcaster common area	×	RW	×
	Broadcaster specific area	×	RW	×
	Broadcaster specific, broadcasting and communication common area	×	RW (Only BML3.0)	×
Terrestrial	Broadcaster common area	R (Only BML3.0))	×	RW
	Broadcaster specific area	×	×	RW
	Affiliate-specific area	RW (Only BML3.0) (Only BS broadcasting companies defined as same affiliate by the terrestrial Extended Broadcaster Descriptor)	×	RW
	Broadcaster specific, broadcasting and communication common area	×	×	RW
Common	General-purpose root certificate area	RW	RW	RW
	Registration calling area	RW (Only level 3)	RW (Only level 3)	RW
	Bookmark area	RW (Only level 2 and 3)	RW	RW

RW : Reading and writing both available  
R : Only reading available  
× : Reading and writing both unavailable

### **3 Contents operation guidelines assuming common receiver units**

This chapter provides guidelines for contents descriptions assuming common receiver units.

#### **3.1 Judgment of common receiver units**

Contents can confirm whether a receiver unit corresponds with media other than terrestrial digital media (i.e. is a common receiver unit) or not by utilizing `isSupportedMedia()`. To utilize functions unique to common receiver units such as NVRAM reference in other media, channel selection in other media, etc., it should always confirm that the receiver unit corresponds with the object media by utilizing `isSupportedMedia()` beforehand.

#### **3.2 NVRAM access for other media**

When receiver units correspond with media other than terrestrial digital media, that media can access the NVRAM area within the range described in 2.2. However, pre-confirmation by `isSupportedMedia()` is always necessary. Refer to TR-B15 Part 2 Vol. 3 “8.2 Operation of NVRAM for common use in MM service in BS” for operation of name spaces upon accessing the NVRAM area of BS digital.

#### **3.3 Channel selection of other media services**

If the receiver unit corresponds with media other than terrestrial digital media, channel selection can be performed for such media services by `epgTune()` and `epgTuneToComponent()`. Refer to ARIB STD-B24 Vol. 2 Appendix 1 “8.5.1 Operation of operational control functions” for details.

#### **3.4 Viewing reservation and programmed recording of other media services**

If the receiver unit corresponds with media other than terrestrial digital media, viewing reservation and programmed recording functions (`epgIsReserved()`, `epgReserve()`, `epgCancelReservation()`, `epgReclsReserved()`, `epgRecReserve()`, `epgRecCancelReservation()`) for such media services can be implemented. However, there is no guarantee that the receiver unit will acquire the SI information of the object media, and reservation will fail without acquisition.

#### **3.5 Information sharing between media**

Since the `Greg[]` value is shared with all receivable media forms, it can be utilized for information delivery between contents broadcasted in different media forms. It can also be utilized for information delivery between communication contents and broadcasting contents

<Intentionally blank>

## **[Section 6] Operational provisions of common digital receiver units for terrestrial/BS/wideband CS/Advanced BS/Advanced wideband CS**

### **1 Introduction**

Section 6 provides functions required for data broadcasting of common digital receiver units (hereinafter referred to as “common receiver units”) for terrestrial/BS/wideband CS digital broadcasting (hereinafter referred to as “digital broadcasting”) and advanced BS/wideband CS digital broadcasting (hereinafter referred to as “advanced wideband satellite digital broadcasting”).

## **2 Functions required for common receiver units**

### **2.1 Functions installed in common receiver units**

Common receiver units should be at least equipped with functions described in TR-B14, TR-B15 and TR-B39 regarding the media (for terrestrial/BS/CS/Wideband CS/Advanced BS/Advanced wideband CS) that can be received by receiver units.

### **2.2 RAM**

Greg should be shared in all media that can be received by receiver units. It is recommended to keep Greg values even while presenting other media that do not support Greg.

If Greg values cannot be kept for some reason during presentation of other media, etc., the receiver units should initialize a Greg value upon the first presenting media that support Greg afterwards.

### **2.3 NVRAM for digital broadcasting**

Bookmark areas, registration calling areas and general-purpose root certificate areas should not be shared areas for advanced wideband digital satellite broadcasting. Table 2-1 shows access availability from data contents of advanced wideband digital satellite broadcasting in NVRAM areas for each media of digital broadcasting.



Table 2-1: Accessible NVRAM Areas from Advanced Wideband Satellite Digital Broadcasting

Media	Area name	Access from advanced BS digital broadcasting	Access from advanced wideband CS digital broadcasting	Operation of related broadcaster description (*)
BS	Broadcaster common area	X	X	
	Broadcaster specific area	RW	RW	- network_id of BS digital broadcasting - broadcaster_id required for specifying access areas
	Broadcaster specific, broadcasting and communication common area	RW	RW	
Wideband CS	Broadcaster common area	X	X	
	Broadcaster specific area	RW	RW	- network_id of wideband CS digital broadcasting - broadcaster_id required for specifying access areas
	Broadcaster specific, broadcasting and communication common area	RW	RW	
Terrestrial	Broadcaster common area	X	X	
	Broadcaster specific area	RW	RW	- original_network_id required for specifying access areas
	Broadcaster specific, broadcasting and communication common area	RW	RW	
	Affiliate-specific area	RW	RW	- affiliation_id required for specifying access areas

RW: Both reading and writing available

X: Both reading and writing available unavailable

(\*): See TR-B39 for operations of related broadcaster description

## 2.4 Nonvolatile storage area for advanced wideband satellite digital broadcasting

Nonvolatile storage area for advanced wideband satellite digital broadcasting shall not be the shared area, because data broadcasting contents do not handle written data.

## 2.5 Character code conversion

### 2.5.1 Conversion from EUC-JP to UCS

Conversion from characters encoded by EUC-JP defined in Section 4.1.1, Part 2, STD-B24 to UCS shall conform to the following specification.

- JIS X0201 YEN SIGN (Code value 5/12) is converted to UCS REVERSE SOLIDUS (Code value U+5C).
- JIS X0201 OVERLINE (Code value 7/14) is converted to UCS TILDE (Code value U+7E).
- Each character in a graphic character set for Katakana of JIS X0201 is converted to the referenced UCS character by an alternative name shown in JIS X0213 Annex 5 Table 1.
- The character\* listed in JIS X0213 Annex 5 Table 2, which is a character in a coded Kanji character set of JIS X0208, is converted to the referenced UCS character by an alternative name specified in the Table.

\*Considered as JIS X0208 Ku-Ten disregarding the plane of both Ku-Tens in the Table

- Each character in an additional symbol set is converted to the UCS character shown in Tables 5-2 and 2-2 in STD-B62 Volume 1, Part2: However,
  - Each character in the range of Row 92, Cells 26 to 31 and Cells 56 to 85 is converted to UCS GETA MARK (U+3013).
- Characters which are not applicable to the preceding paragraphs are converted to the referenced UCS characters according to character names.

Table 2-2 Correspondence of Characters and Symbols defined in STD-B24 to UCS Code

STD-B24 Ku Ten (Row-Cell)	UCS Code value	UCS Character name	Example Glyph
91-8	U+3012	POSTAL MARK	〒
91-17	U+2668	HOT SPRINGS	♨
91-43	U+260E	BLACK TELEPHONE	☎
92-7	U+5E74	CJK UNIFIED IDEOGRAPH-5E74	年
92-8	U+6708	CJK UNIFIED IDEOGRAPH-6708	月
92-9	U+65E5	CJK UNIFIED IDEOGRAPH-65E5	日
92-10	U+5186	CJK UNIFIED IDEOGRAPH-5186	円
92-11	U+33A1	SQUARE M SQUARED	m <sup>2</sup>

STD-B24 Ku Ten (Row-Cell)	UCS Code value	UCS Character name	Example Glyph
92-13	U+339D	SQUARE CM	cm
92-44	U+3232	PARENTHESES IDEOGRAPH HAVE	(有)
92-45	U+3231	PARENTHESES IDEOGRAPH STOCK	(株)
92-46	U+3239	PARENTHESES IDEOGRAPH REPRESENT	(代)
92-48	U+25B6	BLACK RIGHT-PORTING TRIANGLE	▶
92-49	U+25C0	BLACK LEFT-PORTING TRIANGLE	◀
92-50	U+3016	LEFT WHITE LENTICULAR BRACKET	[
92-51	U+3017	RIGHT WHITE LENTICULAR BRACKET	]
92-53	U+00B2	SUPERSCRIP TWO	²
92-54	U+00B3	SUPERSCRIP THREE	³
93-9	U+337E	SQUARE ERA NAME MEIZI	明治
93-10	U+337D	SQUARE ERA NAME TAISYOU	大正
93-11	U+337C	SQUARE ERA NAME SYOUWA	昭和
93-12	U+337B	SQUARE ERA NAME HEISEI	平成
93-13	U+2116	NUMERO SIGN	№
93-14	U+2121	TELEPHONE SIGN	☎
93-39	U+2113	SCRIPT SMALL L	ℓ
93-40	U+338F	SQUARE KG	kg
93-43	U+339E	SQUARE KM	km
93-48	U+00BD	VULGAR FRACTION ONE HALF	½
93-50	U+2153	VULGAR FRACTION ONE THIRD	⅓
93-51	U+2154	VULGAR FRACTION TWO THIRD	⅔
93-52	U+00BC	VULGAR FRACTION ONE QUARTER	¼
93-53	U+00BE	VULGAR FRACTION THREE QUARTERS	¾
93-54	U+2155	VULGAR FRACTION ONE FIFTH	⅕
93-64	U+2600	BLACK SUN WITH RAYS	☀
93-65	U+2601	CLOUD	☁
93-66	U+2602	UMBRELLA	☂
93-68	U+2616	WHITE SHOGI PIECE	♟
93-69	U+2617	BLACK SHOGI PIECE	♞
93-75	U+2660	BLACK SPADE SUIT	♠
93-72	U+2666	BLACK DIAMOND SUIT	♣
93-73	U+2665	BLACK HEART SUIT	♥
93-74	U+2663	BLACK CLUB SUIT	♣

STD-B24 Ku Ten (Row-Cell)	UCS Code value	UCS Character name	Example Glyph
93-78	U+203C	DOUBLE EXCLAMATION MARK	!!
93-79	U+2049	EXCLAMATION QUESTION MARK	?!
93-83	U+2603	SNOWMAN	☃
93-90	U+266C	BEAMED SIXTEENTH NOTES	♪
93-91	U+260E	BLACK TELEPHONE	☎
94-1	U+2160	ROMAN NUMERAL ONE	I
94-2	U+2161	ROMAN NUMERAL TWO	II
94-3	U+2162	ROMAN NUMERAL THREE	III
94-4	U+2163	ROMAN NUMERAL FOUR	IV
94-5	U+2164	ROMAN NUMERAL FIVE	V
94-6	U+2165	ROMAN NUMERAL SIX	VI
94-7	U+2166	ROMAN NUMERAL SEVEN	VII
94-8	U+2167	ROMAN NUMERAL EIGHT	VIII
94-9	U+2168	ROMAN NUMERAL NINE	IX
94-10	U+2169	ROMAN NUMERAL TEN	X
94-11	U+216A	ROMAN NUMERAL ELEVEN	XI
94-12	U+216B	ROMAN NUMERAL TWELVE	XII
94-13	U+2470	CIRCLED NUMBER SEVENTEEN	⑰
94-14	U+2471	CIRCLED NUMBER EIGHTEEN	⑱
94-15	U+2472	CIRCLED NUMBER NINETEEN	⑲
94-16	U+2473	CIRCLED NUMBER TWENTY	⑳
94-29	U+3251	CIRCLED NUMBER TWENTY ONE	㉁
94-30	U+3252	CIRCLED NUMBER TWENTY TWO	㉂
94-31	U+3253	CIRCLED NUMBER TWENTY THREE	㉃
94-32	U+3254	CIRCLED NUMBER TWENTY FOUR	㉄
94-59	U+3255	CIRCLED NUMBER TWENTY FIVE	㉅
94-60	U+3256	CIRCLED NUMBER TWENTY SIX	㉆
94-61	U+3257	CIRCLED NUMBER TWENTY SEVEN	㉇
94-62	U+3258	CIRCLED NUMBER TWENTY EIGHT	㉈
94-63	U+3259	CIRCLED NUMBER TWENTY NINE	㉉
94-64	U+325A	CIRCLED NUMBER THIRTY	㊀
94-65	U+2460	CIRCLED DIGIT ONE	①
94-66	U+2461	CIRCLED DIGIT TWO	②
94-67	U+2462	CIRCLED DIGIT THREE	③

STD-B24 Ku Ten (Row-Cell)	UCS Code value	UCS Character name	Example Glyph
94-68	U+2463	CIRCLED DIGIT FOUR	④
94-69	U+2464	CIRCLED DIGIT FIVE	⑤
94-70	U+2465	CIRCLED DIGIT SIX	⑥
94-71	U+2466	CIRCLED DIGIT SEVEN	⑦
94-72	U+2467	CIRCLED DIGIT EIGHT	⑧
94-73	U+2468	CIRCLED DIGIT NINE	⑨
94-74	U+2469	CIRCLED NUMBER TEN	⑩
94-75	U+246A	CIRCLED NUMBER ELEVEN	⑪
94-76	U+246B	CIRCLED NUMBER TWELVE	⑫
94-77	U+246C	CIRCLED NUMBER THIRTEEN	⑬
94-78	U+246D	CIRCLED NUMBER FOURTEEN	⑭
94-79	U+246E	CIRCLED NUMBER FIFTEEN	⑮
94-80	U+246F	CIRCLED NUMBER SIXTEEN	⑯
94-81	U+2776	DINGBAT NEGATIVE CIRCLED DIGIT ONE	①
94-82	U+2777	DINGBAT NEGATIVE CIRCLED DIGIT TWO	②
94-83	U+2778	DINGBAT NEGATIVE CIRCLED DIGIT THREE	③
94-84	U+2779	DINGBAT NEGATIVE CIRCLED DIGIT FOUR	④
94-85	U+277A	DINGBAT NEGATIVE CIRCLED DIGIT FIVE	⑤
94-86	U+277B	DINGBAT NEGATIVE CIRCLED DIGIT SIX	⑥
94-87	U+277C	DINGBAT NEGATIVE CIRCLED DIGIT SEVEN	⑦
94-88	U+277D	DINGBAT NEGATIVE CIRCLED DIGIT EIGHT	⑧
94-89	U+277E	DINGBAT NEGATIVE CIRCLED DIGIT NINE	⑨
94-90	U+277F	DINGBAT NEGATIVE CIRCLED NUMBER TEN	⑩
94-91	U+24EB	NEGATIVE CIRCLED NUMBER ELEVEN	⑪
94-92	U+24EC	NEGATIVE CIRCLED NUMBER TWELVE	⑫
94-93	U+325B	CIRCLED NUMBER THIRTY ONE	⑳

STD-B24 Ku Ten (Row-Cell)	UCS Code value	UCS Character name	Example Glyph
91-8	U+3012	POSTAL MARK	〒
91-17	U+2668	HOT SPRINGS	♨
91-43	U+260E	BLACK TELEPHONE	☎
92-7	U+5E74	CJK UNIFIED IDEOGRAPH-5E74	年

STD-B24 Ku Ten (Row-Cell)	UCS Code value	UCS Character name	Example Glyph
92-8	U+6708	CJK UNIFIED IDEOGRAPH-6708	月
92-9	U+65E5	CJK UNIFIED IDEOGRAPH-65E5	日
92-10	U+5186	CJK UNIFIED IDEOGRAPH-5186	円
92-11	U+33A1	SQUARE M SQUARED	m <sup>2</sup>
92-13	U+339D	SQUARE CM	cm
92-44	U+3232	PARENTHESIZED IDEOGRAPH HAVE	(有)
92-45	U+3231	PARENTHESIZED IDEOGRAPH STOCK	(株)
92-46	U+3239	PARENTHESIZED IDEOGRAPH REPRESENT	(代)
92-48	U+25B6	BLACK RIGHT-PORTING TRIANGLE	▶
92-49	U+25C0	BLACK LEFT-PORTING TRIANGLE	◀
92-50	U+3016	LEFT WHITE LENTICULAR BRACKET	[
92-51	U+3017	RIGHT WHITE LENTICULAR BRACKET	]
92-53	U+00B2	SUPERSCRIPIT TWO	²
92-54	U+00B3	SUPERSCRIPIT THREE	³
93-9	U+337E	SQUARE ERA NAME MEIZI	明治
93-10	U+337D	SQUARE ERA NAME TAISYOU	大正
93-11	U+337C	SQUARE ERA NAME SYOUWA	昭和
93-12	U+337B	SQUARE ERA NAME HEISEI	平成
93-13	U+2116	NUMERO SIGN	№
93-14	U+2121	TELEPHONE SIGN	☎
93-39	U+2113	SCRIPT SMALL L	ℓ
93-40	U+338F	SQUARE KG	kg
93-43	U+339E	SQUARE KM	km
93-48	U+00BD	VULGAR FRACTION ONE HALF	½
93-50	U+2153	VULGAR FRACTION ONE THIRD	⅓
93-51	U+2154	VULGAR FRACTION TWO THIRD	⅔
93-52	U+00BC	VULGAR FRACTION ONE QUARTER	¼
93-53	U+00BE	VULGAR FRACTION THREE QUARTERS	¾
93-54	U+2155	VULGAR FRACTION ONE FIFTH	⅕
93-64	U+2600	BLACK SUN WITH RAYS	☀
93-65	U+2601	CLOUD	☁
93-66	U+2602	UMBRELLA	☂
93-68	U+2616	WHITE SHOGI PIECE	♞
93-69	U+2617	BLACK SHOGI PIECE	♟

STD-B24 Ku Ten (Row-Cell)	UCS Code value	UCS Character name	Example Glyph
93-75	U+2660	BLACK SPADE SUIT	♠
93-72	U+2666	BLACK DIAMOND SUIT	♠
93-73	U+2665	BLACK HEART SUIT	♥
93-74	U+2663	BLACK CLUB SUIT	♣
93-78	U+203C	DOUBLE EXCLAMATION MARK	!!
93-79	U+2049	EXCLAMATION QUESTION MARK	?!
93-83	U+2603	SNOWMAN	☎
93-90	U+266C	BEAMED SIXTEENTH NOTES	♪
93-91	U+260E	BLACK TELEPHONE	☎
94-1	U+2160	ROMAN NUMERAL ONE	I
94-2	U+2161	ROMAN NUMERAL TWO	II
94-3	U+2162	ROMAN NUMERAL THREE	III
94-4	U+2163	ROMAN NUMERAL FOUR	IV
94-5	U+2164	ROMAN NUMERAL FIVE	V
94-6	U+2165	ROMAN NUMERAL SIX	VI
94-7	U+2166	ROMAN NUMERAL SEVEN	VII
94-8	U+2167	ROMAN NUMERAL EIGHT	VIII
94-9	U+2168	ROMAN NUMERAL NINE	IX
94-10	U+2169	ROMAN NUMERAL TEN	X
94-11	U+216A	ROMAN NUMERAL ELEVEN	XI
94-12	U+216B	ROMAN NUMERAL TWELVE	XII
94-13	U+2470	CIRCLED NUMBER SEVENTEEN	⑰
94-14	U+2471	CIRCLED NUMBER EIGHTEEN	⑱
94-15	U+2472	CIRCLED NUMBER NINETEEN	⑲
94-16	U+2473	CIRCLED NUMBER TWENTY	⑳
94-29	U+3251	CIRCLED NUMBER TWENTY ONE	㉑
94-30	U+3252	CIRCLED NUMBER TWENTY TWO	㉒
94-31	U+3253	CIRCLED NUMBER TWENTY THREE	㉓
94-32	U+3254	CIRCLED NUMBER TWENTY FOUR	㉔
94-59	U+3255	CIRCLED NUMBER TWENTY FIVE	㉕
94-60	U+3256	CIRCLED NUMBER TWENTY SIX	㉖
94-61	U+3257	CIRCLED NUMBER TWENTY SEVEN	㉗
94-62	U+3258	CIRCLED NUMBER TWENTY EIGHT	㉘
94-63	U+3259	CIRCLED NUMBER TWENTY NINE	㉙

STD-B24 Ku Ten (Row-Cell)	UCS Code value	UCS Character name	Example Glyph
94-64	U+325A	CIRCLED NUMBER THIRTY	⑳
94-65	U+2460	CIRCLED DIGIT ONE	①
94-66	U+2461	CIRCLED DIGIT TWO	②
94-67	U+2462	CIRCLED DIGIT THREE	③
94-68	U+2463	CIRCLED DIGIT FOUR	④
94-69	U+2464	CIRCLED DIGIT FIVE	⑤
94-70	U+2465	CIRCLED DIGIT SIX	⑥
94-71	U+2466	CIRCLED DIGIT SEVEN	⑦
94-72	U+2467	CIRCLED DIGIT EIGHT	⑧
94-73	U+2468	CIRCLED DIGIT NINE	⑨
94-74	U+2469	CIRCLED NUMBER TEN	⑩
94-75	U+246A	CIRCLED NUMBER ELEVEN	⑪
94-76	U+246B	CIRCLED NUMBER TWELVE	⑫
94-77	U+246C	CIRCLED NUMBER THIRTEEN	⑬
94-78	U+246D	CIRCLED NUMBER FOURTEEN	⑭
94-79	U+246E	CIRCLED NUMBER FIFTEEN	⑮
94-80	U+246F	CIRCLED NUMBER SIXTEEN	⑯
94-81	U+2776	DINGBAT NEGATIVE CIRCLED DIGIT ONE	⓪
94-82	U+2777	DINGBAT NEGATIVE CIRCLED DIGIT TWO	⓫
94-83	U+2778	DINGBAT NEGATIVE CIRCLED DIGIT THREE	⓬
94-84	U+2779	DINGBAT NEGATIVE CIRCLED DIGIT FOUR	⓭
94-85	U+277A	DINGBAT NEGATIVE CIRCLED DIGIT FIVE	⓮
94-86	U+277B	DINGBAT NEGATIVE CIRCLED DIGIT SIX	⓯
94-87	U+277C	DINGBAT NEGATIVE CIRCLED DIGIT SEVEN	⓰
94-88	U+277D	DINGBAT NEGATIVE CIRCLED DIGIT EIGHT	⓱
94-89	U+277E	DINGBAT NEGATIVE CIRCLED DIGIT NINE	⓲
94-90	U+277F	DINGBAT NEGATIVE CIRCLED NUMBER TEN	⓳
94-91	U+24EB	NEGATIVE CIRCLED NUMBER ELEVEN	⓴
94-92	U+24EC	NEGATIVE CIRCLED NUMBER TWELVE	⓵
94-93	U+325B	CIRCLED NUMBER THIRTY ONE	㉁



## 2.5.2 Conversion from UCS to EUC-JP

Conversion to encode repertoire characters defined in Sections 5.2 and 5.3, STD-B62 Volume 1, Part 2 according to EUC-JP defined in Section 4.1.1, Part 2, STD-B24 Volume 2 shall conform to the following specification.

- UCS REVERSE SOLIDUS (Code value U+5C) is converted to JIS X0201 YEN SIGN (Code value 5/12).
- UCS TILDE (Code value U+7E) is converted to JIS X0201 OVERLINE (Code value 7/14).
- Each character for UCS code value U+FF61 ~ U+FF9F is converted to the JIS X0201 character corresponding to the character name by JIS X0213 Annex 5, Table 1.
- Each character for UCS code value U+FF01, U+FF03 ~ U+FF06, U+FF08 ~ U+FF0C, U+FF0E ~ U+FF5D, U+FFE3 and U+FFE5 is converted to the JIS X0208 character\* corresponding to the character name by JIS X0213 Annex 5, Table 2.

\*Considered as JIS X0208 Ku-Ten disregarding the plane of both Ku-Tens in the Table

- The UCS character described in Tables 5-2 and 2-2 in STD-B62 Volume 1, Part2 is converted to the character in an additional symbol set shown in the Tables. However,
  - POSTAL MARK (Code value U+3012) shall be converted to Row 2, Cell 9 and shall not be converted Row 91, Cell 8.
  - CJK UNIFIED IDEOGRAPH-5E74 shall be converted to Row 39, Cell 15 and shall not be converted Row 92, Cell 7.
  - CJK UNIFIED IDEOGRAPH-6708 shall be converted to Row 23, Cell 78 and shall not be converted Row 92, Cell 8.
  - CJK UNIFIED IDEOGRAPH-65E5 shall be converted to Row 38, Cell 92 and shall not be converted Row92, Cell 9.
  - CJK UNIFIED IDEOGRAPH-5186 shall be converted to Row 17, Cell 63 and shall not be converted Row 92, Cell 10.
  - SQUARED CJK UNIFIED IDEOGRAPH-4E8C (Code value U+1F214) shall be converted to Row 90, Cell 58 and shall not be converted Row 93, Cell 30.
  - BLACK TELEPHONE (Code value U+260E) shall be converted to Row 91, Cell 43 and shall not be converted Row 93, Cell 91.
- USC characters which are not applicable to the preceding paragraphs are converted to JIS X0201 or JIS X0208 characters if referenced characters according to character names are the elements in the coded character set defined in X0201 or JIS X0208. Otherwise, they are converted to Row 2, Cell 14 in JIS X0208 (geta mark).

### 3 Contents operation guidelines assuming common receiver units

This chapter provides guidelines for contents descriptions assuming common receiver units.

#### 3.1 Judgment of common receiver units

Contents can confirm whether a receiver unit corresponds to media other than digital broadcasting (i.e. a common receiver unit) or not by utilizing `isSupportedMedia()`. In order to use functions unique to common receiver units such as channel selection in other media, it should be always confirmed that the receiver unit corresponds to the target media by using `isSupportedMedia()` in advance.

In addition, common receiver units judged by this function are required to support all of the operations unique to the receivers as shown below.

#### 3.2 Greg access

Data contents in advanced wideband digital satellite broadcasting can read data written by data broadcasting contents in digital broadcasting, and data broadcasting contents in digital broadcasting can read data written by data contents in advanced wideband digital satellite broadcasting.

It is noted the function shown in section 2.5 converts character codes, however, there is a case where written data are not read out as they are because of the difference between character repertoires corresponding to digital broadcasting and those corresponding to advanced wideband digital satellite digital broadcasting.

There are the following four kinds of cases where character codes cannot be converted, or all characters cannot be read out in Greg.

- 1) Characters applied in digital broadcasting only
  - Row 92, Cells 26 ~ 31
  - Row 92, Cells 56 ~ 85
- 2) Characters applied in advanced wideband satellite digital broadcasting only
  - These correspond (Corresponding) to characters in JIS X0213:2004 implementation level 3 (excluding characters operated/applied as additional symbols in digital broadcasting) and those in implementation level 4.
- 3) Characters assigned/allocated with more than one character codes
  - Characters shown in Tables 3-1 and 3-2

Table 3-1 Characters which define individual additional symbols set on characters defined in JIS X0208-1990

Character	(A)JIS X0208-1990 Ku Ten code	(B) Additional symbol set Ku Ten code
〒	2-9	91-8
年	39-15	92-7
月	23-78	92-8
日	38-92	92-9
円	17-63	92-10

Table 3-2 Characters with multiple Ku Ten codes assigned as additional symbols

Character	((A)JIS X0208-1990 Ku Ten code	(B) Additional symbol set Ku Ten code
☐	90-58	93-30
☎	91-43	93-91

4) Character string for written characters exceeding 256 bytes

- Characters after 256 bytes when they exceed 256 bytes after the conversion to EUC
- However, character after 255 bytes if the 256th byte is the first byte of two-byte character (code)

Greg operation for common receivers is shown as follows /below, when characters cannot be converted, or readout of all characters is unable in this manner.

- Data broadcasting contents in digital broadcasting writes a character string including characters in case 1) and data contents in advanced wideband satellite digital broadcasting read it. However, if the string is not written, there is the implemented receiver in which the corresponding character is converted to geta mark and read out.
- If data broadcasting contents in digital broadcasting write a character string including characters in case 1), and the read string by data contents in advanced wideband satellite digital broadcasting is written as it is, the corresponding character is converted to geta mark and read out.
- If data contents in advanced wideband satellite digital broadcasting write a character string including characters in case 2), the corresponding character is converted to geta mark and read out.
- Data broadcasting contents in digital broadcasting writes a character string including

characters in case 3) and data contents in advanced wideband satellite digital broadcasting read it. However, if the string is not written, there is the implemented receiver in which the corresponding character read out by a different character code\*1.

- If data broadcasting contents in digital broadcasting write a character string including characters in case 3), and the read string by data contents in advanced wideband satellite digital broadcasting is written as it is, the corresponding character read out by a different character code\*1.
- For the case 4), it is not possible to read all of character strings written by data contents in advanced wideband satellite digital broadcasting.

\*1: Converted to the character code shown in Table 3-1 (A) and Table 3-2 (A)

### 3.3 NVRAM access

When the receiver unit corresponds to advanced wideband satellite digital broadcasting, data contents of the satellite digital broadcasting can access the NVRAM area of each media within the range described in Table 2-1. However, it is necessary to transmit the related broadcaster descriptor shown in Table 2-1 containing the identification value required to access the descriptor to allow access from data contents of advanced wideband satellite digital broadcasting.

### 3.4 Channel selection of advanced wideband satellite digital broadcasting services

When the receiver unit corresponds to advanced wideband satellite digital broadcasting, channel selection can be performed for the satellite digital broadcasting services by `epgTune()`. Namespace specifying a service operates that defined in ARIB STD-B62 Part 2 Section 4.2.

Note that channel selection by `epgTuneToComponent()` is not operated.

### 3.5 Viewing reservation and programmed recording of advanced wideband satellite digital broadcasting services

When the receiver unit corresponds to advanced wideband satellite digital broadcasting, viewing reservation and programmed recording function (`epgIsReserved()`, `epgReserve()`, `epgCancelReservation()`, `epgReclsReserved()`, `epgRecReserve()` and `epgRecCancelReservation()`) for the satellite digital broadcasting services can be executed. Namespace specifying events operates that defined in ARIB STD-B62 Part 2 Section 4.2.

Note that there is no guarantee that the receiver unit will acquire SI information of the target media, and reservation will fail if the information cannot be acquired.

---

OPERATIONAL GUIDELINES FOR  
DIGITAL TERRESTRIAL TELEVISION BROADCASTING

ARIB TECHNICAL REPORT

ARIB TR-B14 VERSION 6.7-E1  
(Fascicle 2)  
(September 28th, 2020)

---

This Document is based on the ARIB technical report of  
“Operational Guidelines for Digital Terrestrial Television  
Broadcasting” in Japanese edition and translated into English in  
March, 2021.

---

Published by

Association of Radio Industries and Businesses

Nittochi Bldg. 11F  
1-4-1 Kasumigaseki, Chiyoda-ku, Tokyo 100-0013, Japan

TEL 81-3-5510-8590  
FAX 81-3-3592-1103

Printed in Japan  
All rights reserved

---