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 3)

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

ARIB TR-B14
Version 6.7-E1

Vol. 3

DIGITAL TERRESTRIAL TELEVISION BROADCASTING

Specifications for Data Broadcasting Operations

(Part 2)

Contents

[Section 4] Operational provisions related to C-profile 1

1 Introduction 1

2 (Unused number) 2

3 Functions that basic receiver units that receive video, audio and multimedia data 3
3.1 Configuration of the receiver units 3
3.1.1 Hardware configuration 3
3.1.2 Receiver unit reference model 4
3.2 Presentation function 6
3.2.1 Presentation technique of receiver units 6
3.2.2 Pixel sizes and restrictions of each plane configuring screen 8
3.2.3 Visual mono-media coding 9
3.2.4 Audio playback functions 11
3.2.5 Fonts 11
3.3 Buttons (remote controller) 11
3.3.1 Keys to be used in C-profile reception 11
3.3.2 Key masks 14
3.4 Memory capability of receiver units 14
3.4.1 RAM 14
3.4.2 NVRAM 15
3.5 Communication function 15
3.6 Character input function 16
3.6.1 Function specification 16
3.6.2 User interface 16
3.6.3 Character types 16
3.6.4 Kana Kanji conversion function 16
3.7 TVlink 17
3.7.1 Introduction of TVlink 17
3.7.2 What is a TVlink? 17
3.7.3 Changed points from profile A bookmarks 18
3.7.4 NVRAM in TVlink 18
3.7.4.1 Number of TVlink areas 18
3.7.4.2 Sharing with other medias 18

3.7.5 Format and details of TVlink 19
3.7.5.1 Format of TVlink 19
3.7.5.2 Details of each parameters 19
3.7.6 Operation of TVlink types 21
3.7.7 Guidelines for the functions for each TVlink type and receiver unit 22
3.7.8 Writing of TVlinks 25
3.7.9 Guidelines of TVlink list functions 26
3.7.9.1 Start and end of the TVlink list by receiver's native application 26
3.7.9.2 Display of TVlink information 27
3.7.9.3 TVlink selection and transition to the destination link 27
3.7.9.4 Deletion of TVlink information 27
3.8 BML browser 28
3.8.1 Browser specific display 28
3.9 Browser for the C-profile contents on the internet 28

4 Operation of data transfer format 31
4.1 PSI/SI 31
4.1.1 Type of data broadcasting services 31
4.1.1.1 Channel service type operated by data broadcasting programs 31
4.1.2 Configuration of the data broadcasting service contents and operation of the components 31
4.1.2.1 Contents and local contents 31
4.1.2.2 Relationship between local contents and ES 32
4.1.2.3 Operation of component tags 32
4.1.2.4 Configuration of the ES transmitted by 1 service 32
4.1.2.5 Identification of entry components 32
4.1.2.6 Identification of event message components 33
4.1.2.7 Detailed specifications for section data transmission 33
4.1.2.8 Default maximum bit rate for data broadcasting programs 33
4.1.3 Operation of PMT 33
4.1.3.1 PMT operation of specific data broadcasting services 33
4.1.3.2 Operation of the Data Component Descriptor in the PMT 34
4.1.3.3 Target Resion Descriptor 35
4.1.4 Operation of the L-EIT Data Contents Descriptor 35
4.1.5 Related receiver operation 35
4.1.5.1 Receiver operation at the beginning of data broadcasting 35
4.1.5.2 Receiver behavior during update of the PMT 36

4.1.5.3 Specifications on the partial transport stream output-input 37
4.2 Data carousel transmission operation 37
4.2.1 Data carousel transmission operation 37
4.2.1.1 Introduction of data event and local contents 37
4.2.2 Operation of data events 38
4.2.2.1 Operation of data events 38
4.2.2.2 Initiation and termination of local contents 38
4.2.2.3 Receiver basic behavior during data broadcasting program presentation 38
4.2.2.4 Operation when DII cannot be obtained for long time 39
4.2.3 Operation of empty carousels 39
4.2.4 Operation of the DownloadInfoIndication (DII) message 40
4.2.5 Operation of DownloadDataBlock (DDB) 43
4.3 Operation of event messages 43
4.3.1 Purpose for operating event messages 43
4.3.2 Transmission of general event messages 44
4.3.2.1 Operation of general event message data_event_id, event_msg_group_id 44
4.3.2.2 Regarding component specifications of general event messages from the BML document 46
4.3.2.3 Event message processing in receivers 46
4.3.3 Operation of DSMCC_section() 47
4.3.4 Operation of the General Event Descriptor 48
4.4 Receiver performance 48
4.4.1 Filtering resources used during data broadcasting reception 48
4.4.2 Desirable operation of the receiver in relation to the acquisition of C-profile modules 49
4.4.3 Receiver operation concerning updating the version 49
4.5 Auxiliary information function using general event message (Optional) 49
4.5.1 Operation of event_msg_id and general event message transmission 49
4.5.2 Transmission of the specified character string in private_data_byte 50
4.5.3 Operation of content 52
4.5.4 Conditions for incorporation on the receiver 52
4.5.5 Conditions for presentation on the receiver 52
4.5.5.1 Presentation time 53
4.5.5.2 Presentation 53
4.5.5.3 Presentation update 54
4.5.5.4 Deletion timing 54
4.5.5.5 Character color, background color, character size and character type 54

4.5.5.6 Receiver operation in response to each specified character string 54

5 Operation of mono-media coding 58
5.1 Image coding 58
5.1.1 H.264|MPEG-4 AVC 58
5.1.1.1 Coding parameters 58
5.1.1.2 Picture format 58
5.1.1.3 Restrictions in the bit stream 58
5.1.1.4 Other restrictions 65
5.1.1.5 Identification of 16:9 pictures 65
5.1.1.6 Operation of PanScan in the picture display area 66
5.2 Still picture and bitmap figure coding 67
5.2.1 JPEG 67
5.2.1.1 Coding parameters 67
5.2.1.2 Other restrictions 67
5.2.1.3 Operation markers and marker segments 67
5.2.2 GIF 68
5.2.2.1 Usable blocks 68
5.2.2.2 Header 68
5.2.2.3 Trailer 68
5.2.2.4 Logical Screen Descriptor 68
5.2.2.5 Global Color Table 68
5.2.2.6 Image Descriptor 69
5.2.2.7 Local Color Table 69
5.2.2.8 Image Data 69
5.2.2.9 Graphic Control Extension 69
5.2.2.10 Comment Extension 70
5.2.2.11 animation GIF operation restrictions 70
5.3 Audio coding 71
5.3.1 MPEG-2 AAC 71
5.3.1.1 Coding parameters 71
5.3.1.2 Transmission of MPEG-2 AAC 71
5.3.1.3 Limitations of data carousel transmission 71
5.3.1.4 Data format of MPEG-2 AAC files 71
5.3.2 AIFF-C 72
5.3.3 Built-in sound 72
5.3.4 MIDI 72

5.3.5 Audiosynthesis of receiver units 72
5.4 Character codes 73
5.4.1 8-bit character codes for C-profile 73
5.4.2 Shift JIS 73

6 Operation of closed caption coding (option) 74
6.1 Definition and range of service 74
6.2 Organization and transmission operation 74
6.2.1 Restrictions on organization and transmission 74
6.2.2 PES transmission method used in closed caption 75
6.2.3 Operation of data groups 76
6.2.4 Operation of closed caption management data 77
6.2.5 Operation of closed caption text data 78
6.2.6 Operation of data units 78
6.2.7 Operation of PSI/SI 78
6.2.7.1 Operation of component tags 78
6.2.7.2 Operation of the PMT 78
6.2.7.3 Stream format identification 78
6.2.7.4 Descriptor operation 78
6.2.7.5 Data Component Descriptor 79
6.2.7.6 Target Region Descriptor 79
6.2.7.7 Data Contents Descriptor 79
6.3 Display format of closed caption 79
6.3.1 Display format 79
6.4 Characters used in closed caption 80
6.4.1 Character entity 80
6.4.2 Character fonts 81
6.4.3 Character size 81
6.4.4 Display partition 82
6.4.5 Non-spacing characters 82
6.5 Control code used in closed caption 83
6.5.1 Control code 83
6.5.1.1 Operation of color specifications 85
6.5.2 Operation of flashing 85
6.5.3 Extension control code 85
6.6 Operation of the DRCS 87
6.7 Operation of the initialization operation 87

6.7.1 Initialization by switching classes in the data group....88
6.7.2 Initialization by closed caption text....88
6.7.3 Initialization by main text data unit....88
6.7.4 Initialization by character control codes....88
6.7.5 Initialization by receiver operation....89
6.8 Mono-media used in closed caption....89
6.8.1 Operation of geometric....89
6.8.2 Operation of bitmap data....89
6.8.3 Operation of built-in sound....89
6.8.4 Additional sound....89
6.9 Ideal receiver unit operation....89
6.9.1 Initialization and termination of closed caption display....89

7 Operation of multimedia encoding....89
7.1 Introduction....89
7.2 Operation of NVRAM in Digital Terrestrial Television C-profile broadcasting....90
7.2.1 Writing frequency of the NVRAM area....91
7.2.2 Allocation of NVRAM....91
7.2.2.1 Allocation of the C-profile area for the affiliation....91
7.2.2.2 Allocation of individual operator areas in the C-profile area for the affiliation....92
7.2.3 Deletion function of the C-profile area for the affiliation....92
7.2.4 Identification of the C-profile area for the affiliation....93
7.3 Operation of keys from the Multimedia service....94
7.3.1 Values handled for used-key-list characteristics....94
7.3.2 Correspondence of keys, key codes and access keys....94
7.4 Operation of the BML version....95
7.5 Operation of character coding schemes....95
7.6 Operation range of media type and monomedia....95
7.7 Operation of the BML elements....95
7.7.1 Declaration of XML and DOCTYPE....95
7.7.2 Operation of the BML elements....95
7.7.3 Attributes....98
7.7.3.1 Restrictions on the order of elements in the head element....104
7.7.3.2 Operation of bml:beitem element....104
7.7.3.3 Expansion of object elements....104
7.7.4 Entity....104
7.7.5 Operation of the BML element extended module (event)....105

7.8 Operation of CSS 105
7.8.1 Element applied each properties 108
7.8.2 Selector operation 109
7.8.3 Media type 109
7.8.4 Box model 110
7.8.5 Visual formatting model 111
7.8.6 Other visual effects 112
7.8.7 Colors and backgrounds 112
7.8.8 Fonts 113
7.8.9 Text 114
7.8.10 Default style sheets 114
7.8.11 length specification 115
7.9 extended property specification 115
7.10 Operations for procedural descriptions 117
7.10.1 Operations DOM 117
7.10.1.1 DOM Core Fundamental interfaces 117
7.10.1.2 DOM HTML interfaces 119
7.10.1.3 BMLCSS2Properties interface 130
7.10.2 Operation area of built-in objects 132
7.10.3 Operation area of extension objects for broadcasting 135
7.10.4 Operation area of Navigator pseudo-objects 136
7.10.5 Operation area of browser pseudo-objects 136
7.10.5.1 Operational rule of each extended function 140
7.10.5.2 Operation guidelines of function that generate communication 141
7.10.6 Extended functions provided by digital terrestrial broadcasting (1) 142
7.10.7 Extended function provided by digital terrestrial broadcasting (2) 142
7.10.8 Execution operation of the extended function 159
7.11 Restrictions in BML document descriptions 159
7.11.1 Restrictions of the script element 159
7.11.2 Positioning and restrictions 160
7.11.3 Vertical writing 161
7.11.4 Restrictions of element positioning 161
7.11.5 Operation related to object elements 162
7.11.5.1 Operation related to the presentation operation of animation GIF 162
7.11.6 Operation related to nested elements 162
7.11.7 Operation related to the p element 163

7.12 Operational guidelines related to presentation 163
7.12.1 Operational guidelines of object presentation 163
7.12.2 Presentation plane 164
7.12.3 Operation of monomedia presentation 164
7.12.3.1 Positioning by object elements and img elements 164
7.12.4 Guidelineson clipping 165
7.12.5 Range of values where CSS2 properties is applicable 166
7.12.6 Provision of box models 166
7.12.6.1 Attribute operation related to stream presentation behaviour 167
7.12.6.2 Presentation behaviour of animation GIF and operation of the streamstatus attribute 168

8 Guidelines for browser's behaviors 169
8.1 Guidelines for presentation 169
8.1.1 Principle of presentation in C-profile receivers 169
8.1.1.1 Principle of simultaneous presentation of video, audio and multimedia data 169
8.1.2 Presentation of video and audio 169
8.1.2.1 Presentation and control of broadcasting video and broadcasting audio 169
8.1.2.2 Display position and display size of video 169
8.1.2.3 Optimizing video display size 169
8.1.2.4 Display of QVGA 16:9 video 170
8.1.2.5 Changing the video display size 170
8.1.3 Operation during the startup of the TV reception function and tuning 170
8.1.3.1 Principle of display status when the TV reception function startups and during tuning 170
8.1.3.2 Principle of BML browser key acquisition 170
8.1.4 Prohibition of mixed display and simultaneous display 171
8.1.4.1 Princilple of mixed display prohibition 171
8.1.4.2 Simultaneous display of broadcasting screens and screens other than broadcasting (browsers for the contents on the internet, applications, etc.) 171
8.1.4.3 Various operations during simultaneous display 172
8.1.4.4 Operations of broadcasting content during simultaneous display 172
8.1.5 Prohibition of simultaneous display while displaying a part of broadcasting content (video audio only, audio only or BML browsers) 172
8.1.5.1 Principle of prohibition of simultaneous display while displaying a part of the broadcasting contents 172
8.1.5.2 Exceptions to simultaneous display prohibition when presenting only a part of the broadcasting contents 172

8.1.6 Simultaneous display of messages presented by receiver systems 173
8.1.7 Specifications regarding communication contents presentation 173
8.1.7.1 Contents presented in BML browsers 173
8.1.7.2 Activation of communication content other than C-profile linked content 173
8.1.7.3 Specifications to display receiver specific browsers for the contents on the internet 174
8.1.7.4 Common use of BML browsers and Browser for the C-profile contents on the internets 174
8.1.7.5 Browser to present with TVlink 174
8.1.7.6 Display of videos acquired from communication (optional) 175
8.1.7.7 Permission during communication generation and personal information dispatch 176
8.1.8 Presentation of multimedia data 177
8.1.8.1 Multimedia data display area 177
8.1.8.2 Exceptions in multimedia data display 177
8.1.8.3 Optimizing presentation size 177
8.1.8.4 Changing the size of BML browser display area 178
8.1.8.5 Image size possible for display 178
8.1.8.6 Character fonts and possible number of characters for display 178
8.1.8.7 Position of CSS and contents 179
8.1.9 Guidelines for contents creation 179
8.1.9.1 Pixel size assumed in contents creation 179
8.1.9.2 Character size controls and layout to be assumed in contents 179
8.1.9.3 Size of images used 180
8.1.9.4 CSS 180
8.1.9.5 Acquiring and releasing “*”, “#” and number keys 180
8.1.9.6 Notes on communication charges generated and information transmission 180
8.1.9.7 Permission when acquiring position information 181
8.1.9.8 Permission when acquiring receiver-unique identifiers or televiewer-unique identifiers 181
8.1.10 Display position for caption 181
8.1.11 Guidelines for DOM operation 181
8.1.12 Guidelines for external characters operation 181
8.1.13 Guidelines for ECMA script implementation 181
8.1.14 Guidelines for operation of extended objects for broadcasting 182
8.1.14.1 Operational specifications for BinaryTable objects 182
8.1.15 Guidelines for browser pseudo-object operations 182
8.1.15.1 Operations of Ureg 182

8.1.15.2 Operations of Greg....182
8.1.15.3 Operation of EPG functions....183
8.1.15.4 Interaction channel communication – Operation of TCP/IP....183
8.1.15.5 Operation of the operational control function....183
8.1.15.6 Non-volatile memory functions and operation of the data carousel storage function....197
8.1.15.7 Operation of functions related to TVlinks....198
8.1.15.8 Operation of special function for terrestrials....198
8.1.15.9 Operation of print functions (optional)....201
8.1.16 Built-in objects....203
8.1.17 Other restrictions....204
8.2 Transmission, reference and name space of content....204
8.2.1 Scope mapping to the transmission system....204
8.2.2 Restrictions when a mono-media is referred across a different media....204
8.2.3 Operations of name space....204
8.2.3.1 Restriction for the number of resources which are managed by receiver....205
8.2.3.2 Name space regarding multiple ES module locks....205
8.2.3.3 Name space regarding multiple ES module version watch....205
8.2.3.4 Reception operation and contents guidelines in #fragment operations....205
8.2.4 Reference guidelines of contents transmitted by components different from the BML document being presented....206
8.3 Operation of C-profile communicating content....207
8.3.1 Guidelines regarding presentation for C-profile communicating contents....207
8.3.2 Guidelines for operations of external fonts in C-profile communicating contents....207
8.3.3 Operation of DOM in C-profile communicating contents....207
8.3.4 Operation of ECMAScript implementation script in C-profile communicating contents....207
8.3.5 External objects for broadcasting in C-profile communicating contents....207
8.3.6 Operation of browser pseudo-objects in C-profile communicating contents....207
8.3.6.1 “Ureg” function....208
8.3.6.2 EPG functions....208
8.3.6.3 Interaction channel function -TCP/IP....208
8.3.6.4 Operational control functions....211
8.3.6.5 Receiver sound control....214
8.3.6.6 Timer functions....215
8.3.6.7 Other functions....215
8.3.6.8 Data carousel storage functions (T.B.D.)....216
8.3.6.9 Digital terrestrial broadcasting specific functions....216

8.3.6.10 Non-volatile memory functions 218
8.3.7 Communication contents scope mapping 219
8.3.8 Guidelines for C-profile linked contents transmission 222
8.3.9 Resource references between C-profile linked contents and broadcasting contents 222
8.3.9.1 Resource references from broadcasting contents to C-profile linked contents 222
8.3.9.2 Reference from C-profile linked contents to broadcasting contents 222
8.3.10 Name space in C-profile linked contents 223
8.3.10.1 Restriction for URI 223
8.3.10.2 Operation of name spaces in servers 223
8.3.10.3 Operation of name spaces upon transition from broacasting contents to C-profile linked contents 223
8.3.10.4 Operation of name spaces upon transition from C-profile link content to broadcasting content 223
8.3.11 Guidelines for operations of C-profile linked contents 224
8.3.11.1 Note on operations of C-profile linked contents 224
8.3.11.2 Guidelines for description of C-profile linked contents 224
8.3.11.3 Guidelines for receiver unit status transition operations 226
8.3.11.4 Guidelines for failure behaviour of receiver units 229
8.3.12 Guidelines for connection and disconnection 229
8.3.12.1 Guidelines for receiver upon automatic connection 229
8.3.12.2 Guidelines for receiver upon disconnection 230
8.3.13 Operation of certificates 230
8.3.14 Operation of User-Agent for identification of BML browsers and a browser for the C-profile contents on internet 230
8.3.15 Operation of X_DPA_getComBrowserUA() 232
8.3.16 Content transition and browser startup 232

9 Functions with which C-profile basic receiver unit that only receive video and audio 236
9.1 Configuration of the receiver unit 236
9.1.1 Hardware configuration 236
9.2 Presentation functions 237
9.2.1 Video presentation function 237
9.2.2 Audio playback function 237
9.3 Guidelines for reception operation 238
9.4 Guidelines for presentation 238
9.4.1 Presentation of video / audio 238
9.4.1.1 Display position and display size of video 238

9.4.1.2 Optimizing video display size 238
9.4.1.3 Display of QVGA 16:9 video 239
9.4.2 Presentation upon TV reception function startup and tuning 239
9.4.3 Restriction of mixed display of non-broadcasting screen and broadcasting screen and simultaneous display 239
9.4.3.1 Principle of mixed display prohibition 239
9.4.3.2 Prohibition of simultaneous display on receiver units that only receive video and audio 239
9.4.3.3 Simultaneous display of messages displayed by receiver unit 240
9.4.3.4 Operation of broadcasting contents upon simulaneous display 240

Appendix 1 Regards for NVRAM access 241

Appendix 2 Module compression format 242

Appendix 3 DTD for operation scope checking for basic service 243

Appendix 4 Default style sheet 250

Appendix 5 Operation scope for browser pseudo-object in Browser for the C-profile contents on the internets 251

Appendix 6 Operation of the getBrowserStatus() argument 258

Appendix 7 Communication restrictions for carriers 259

Appendix 8 Arrangement of elementary stream and empty carousel operations in the PMT 260

Appendix 9 Version update of the BML version 262

Appendix 10 Guideline concerning the operation of personal information in NVRAM and communications in data broadcasting 264

Appendix 11 Guideline concerning the cache function and URI history 265

Appendix 12 Details of the communication function 267

Appendix 13 Governing organization 272

[Section 4] Operational provisions related to C-profile

1 Introduction

C-profile, transmitted by the partial reception layer is configured by the following data.

- Video data, audio data: Low-frame-rate and low-resolution picture (H.264, MPEG-4 AVC), MPEG-2 AAC
- Multimedia data: Data encoding complied with ARIB STD-B24, Vol. 2, Appendix 4
- Closed caption

Video, audio and data are integral as broadcasting content and, as a general rule, a C-profile receiver shall need to be able to present all of them. The functions that should be implemented on a C-profile receiver are described from Chapter 3 to Chapter 8. As an exception, the desired functions of a receiver presenting only video and audio are prescribed in Chapter 9 "Functions with which C-profile basic receiver unit that only receive video and audio".

An exception is when data broadcasting services cannot be completed for some reason, which the communication cannot be used even if a data browser is implemented. There, besides the functions that receiver units should have for reception playback including the multimedia data and the closed caption described in Chapter 3, are descriptions for the functions that the receiver units should have to only receive video and audio in Chapter 9.

2 (Unused number)

3 Functions that basic receiver units that receive video, audio and multimedia data

3.1 Configuration of the receiver units

C-profile basic receiver units that receive video, audio and multimedia data shall be compliant with ARIB STD-B24, Vol. 2, Appendix 4. The following are the specifications for each part of the process of C-profile basic receiver units as hardware configuration point of view, and the specifications for resources within the receiver units by the reference decoders.

It is a matter of product planning of each manufacturer not to implement specified functions, implement the same functions by other means or implement the functions which surpass specifications, and which are not restricted. When the issues occur because of the above, it is assumed broadcasters will not be able to handle the issues.

3.1.1 Hardware configuration

Refer to Figure 3-1 for the hardware configuration of C-profile basic receiver unit.

The digital broadcasting signal input into C-profile basic receiver units are transformed into transport streams by tuners and demodulators. The demodulated transport stream is, by the process of a transport stream decoder, demultiplexed into video, audio and other data and are then output as video stream by video decoding process, and audio stream by audio decoding process. By those processes, C-profile basic receiver unit playback video and audio in a stream format. The receiver transfers data within a data carousel in main memory or non-volatile memory and then process data with CPU. Also, not only the playback process of video and audio but also processes such as audio playback by presenting character and images are possible and, at the same time, by transferring data of main memory into the audio decoder. By using two-way communication lines better interactive performance is expected than analog TV viewing. From the respective viewpoint of the above mentioned hardware process performances, it is necessary to provide the specifications for the following functions.

(1) Receivable data by transport decoders
(2) Playback audio data of the stream system and stored audio data
(3) Playback video data of the stream system
(4) Presentation of video, still images, graphic images, characters, etc.
(5) Two-way communication function using communication lines
(6) Persistent storage data size
(7) Sufficient ROM size (for example, fonts, etc.)
(8) Memory size for decoding obtained data

(9) Guidelines regarding the handling the buttons and the remote controller

(1) TS decoder functions, (2) (3) (4) (7) for presentation functions, (5) for communication functions, (6) (7) (8) memory capacities, and (9) is for remote control functions are especially specified.

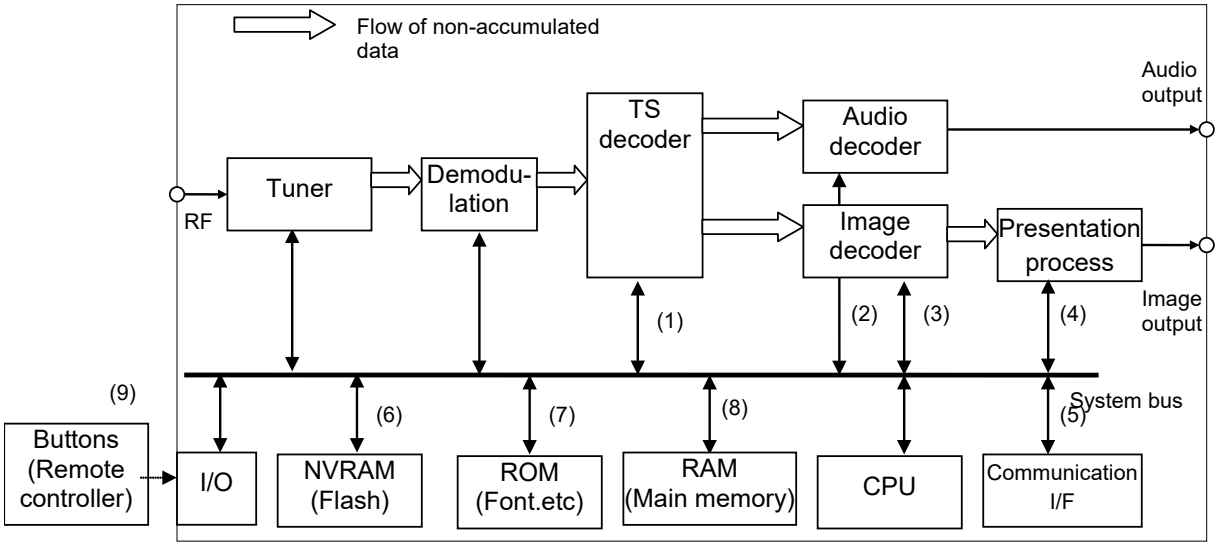


Figure 3-1 Hardware configuration of C-profile basic receiver units

3.1.2 Receiver unit reference model

The resources in the receiver units are classified according to the reference model of the receiver unit, and receivable limits are specified for data broadcasting in C-profile basic receiver units. Refer to Figure 3-2 for the reference model of the receiver unit .

The reference model is specified in compliance with the operation specifications based on the reference decoder models indicated on DAVIC 1.4 Part 9. If they are not specified in on this specifications, they shall be compliant with the definitions of DAVIC.

The received MPEG-2 TS, after filtering in each ES by the PID filter, is stored in the Bn, which is a main buffer, by the TBn, which is a transport buffer for the elementary stream of images and audio transmitted by PES. On the other hand, the elementary stream of multimedia contents transmitted by the data carousel is, after filtering by the PID filter, stored in the Bcontents by the TBn which has already been section filtered.

The received multimedia contents data will be executed by the viewer of the receiver unit. The multimedia engine loads data in the Bcontents following this command and executes multimedia contents using Bwork as execution memory. At this moment, mono-media contents transmitted by the data carousel are specified from the Bcontents to each decoder,

and mono-media contents transmitted by the stream are provided from the Bn to each decoder. The audio mono-media contents are presented by a presentation device of speakers after decoding. However, the visual mono-media contents is allocated in a display area after video, still images, and characters and images have been decoded respectively, and presented by the presentation devices. The presentation of closed caption specified in Chapter 6 is implementation dependent.

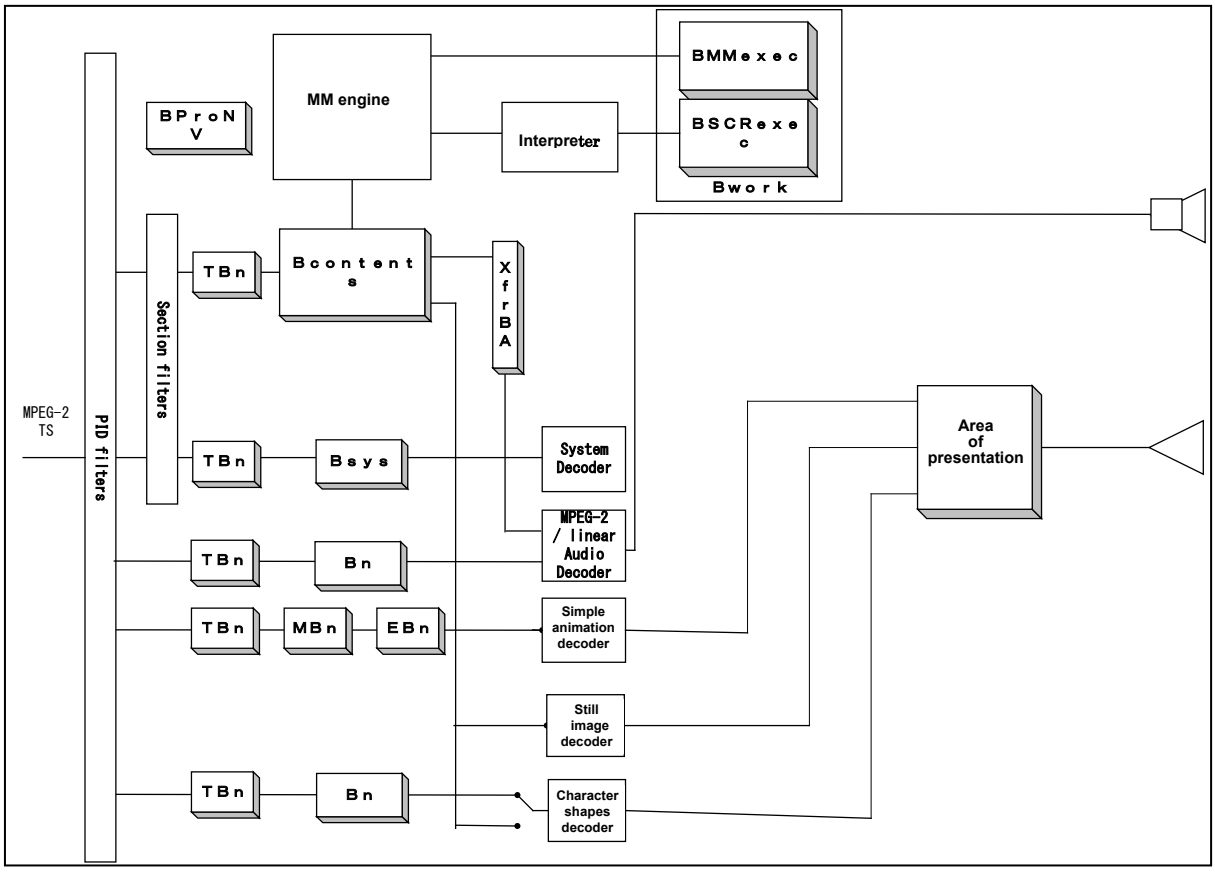


Figure 3-2 C-profile basic receiver unit reference model

TBn	Transport buffers TBn to deliver elementary stream n.
Bn	Main buffer of elementary stream n of decoders.
Bcontents	Buffer Bcontents to store multimedia contents data transmitted by a data carousel. When modules are transmitted as in a compressed format (allocates Compression Type Descriptor of DII), both data before and after compression (compression, decompression) are buffered.
XfrBA	Buffer used to transfer the file format of audio contents into the audio decoder.
BMMexec	Execution memory for multimedia code.
BSCRexec	Execution memory for scripts.
Bwork	Execution memory for multimedia contents in BMMexec and BSCRexec.
BproNV	Non-volatile memory to store information of each viewer and the information of each broadcaster.
Other buffers	Refer to ISO/IEC13818-1 and DAVIC 1.4 Specification Part 9 for details.

3.2 Presentation function

The presentation function of C-profile basic receiver units is modeled on both virtual planes which output the decoded results of each mono-media and a display buffer depend on the display device of the receiver units. The configuration of the display device is implementation dependent. Refer to the guidelines of ARIB STD-B24, Vol. 2, Appendix 4, section 5.1 for the presentation of video and the data in the display area, and Chapter 8 in this specifications.

3.2.1 Presentation technique of receiver units

The receiver unit should present video, closed caption and BML contents according following steps.

1. Decode video stream with a video decoder and output it to a video virtual plane (Y, Cb, Cr 4:2:0 format). The output result shall be (A).
2. Decode closed caption data with a closed caption decoder and output it to a closed caption virtual plane. This output result shall be (B).
3. Decode BML contents with BML browser and output it to a virtual plane for BML browser (width of 240 pixels, height of 480 pixels: each 8 bit format of RGB). This output result shall be (C).
4. Execute a scaling and color space conversion suitable for the display format of each receiver unit in each output from (A), (B) and (C) in order, then output them to the display buffer of the receiver unit. It is not necessary to overlap each virtual plane by alpha blending. Also, it is not mandate to display simultaneously both virtual planes for BML browser and closed caption virtual planes. This indicates that it is not mandate to execute parallel both BML browsers and closed caption decoders. Regarding BML

contents, it is permitted to display only a part of them. However, a function for browsing any part of the virtual plane by the scrolling representation by the receiver unit feature has to be offered.

Refer to Figure 3-3.

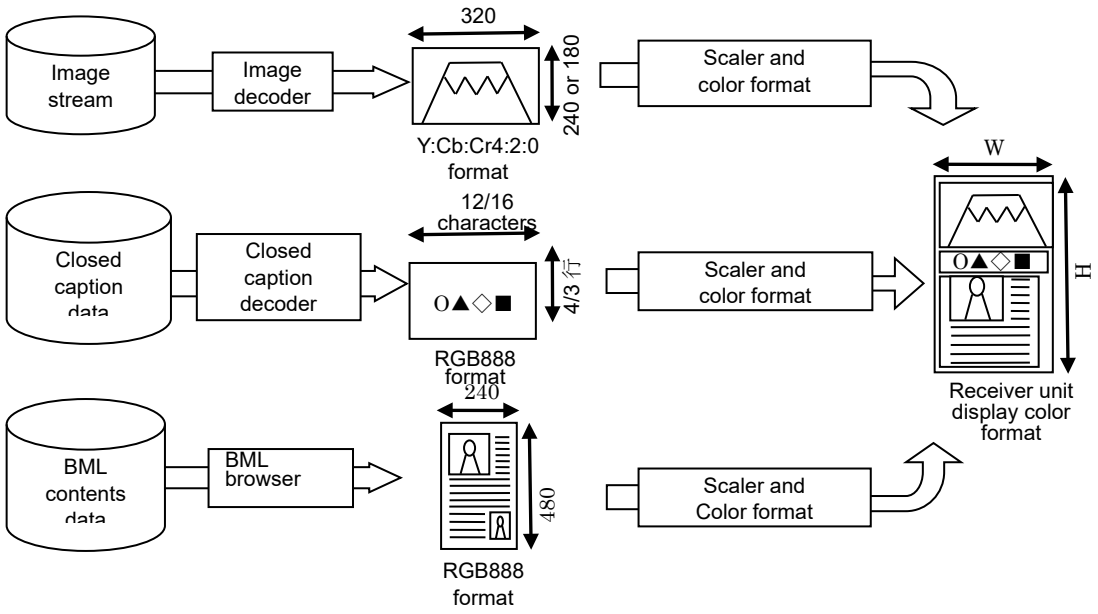


Figure 3-3 Presentation technique of receiver units

3.2.2 Pixel sizes and restrictions of each plane configuring screen

Table 3-1 indicates the specifications for the pixel size of each plane configuring display screen.

Table 3-1 Pixel size of screen planes

Item	Contents of specification	
Video virtual planes	Reso-lution	320 x 180 x 16, YCbCr (4:2:0), 16:9
		320 x 240 x 16, YCbCr (4:2:0), 4:3
BML browser virtual planes	Reso-lution	240 x 480 x 24, RGB each 8bit (*1)
Closed caption virtual planes	Reso-lution	Character writing direction normal size, more than 12 characters x 4 lines, or character writing direction normal size, more than 16 characters x 3 lines. 8 bits of each RGB (*2)

(*1) Regarding the color resolution of pixels; due to there may be receiver units that cannot be expressed by 8-bits resolution of each RGB ,it is recommended to consider using different distinctive colors for different color expression.

(*2) For details refer to section 6.3.

As for restriction matters on these planes, Table 3-2 indicates the specifications related to displayable mono-media codes, display position of mono-media codes, and size etc.

Table 3-2 Presentation of restricted matters in display planes

Item	Contents of specification	
Video virtual planes	Presentable mono-media codes	H.264|MPEG-4 AVC
	Presentation position	Implementation dependent. Not assigned by multimedia contents.
	Size	Same as the pixel configuration of virtual planes.
BML browser virtual planes	Presentable mono-media codes	JPEG, GIF, animation GIF
	Presentation position	Both x-y coordinate, from arbitrary pixels to arbitrary pixels
	Size	Both x-y coordinate, from arbitrary pixel count
	Overlapping	Restrictions on animation GIF and marquee (see section 7.11)
Closed caption virtual planes	Presentable mono-media codes	8-bit character codes for C-profile
	Presentation position	Not assigned by contents.
	Size	Medium size font (Implementation dependent)
Plane switching visual effect	Processing of the plane switching visual effect of presentation is implementarion dependent.	

3.2.3 Visual mono-media coding

The following are the specifications on the encoding method of visual mono-media desired to being presented in C-profile basic receiver unit. Refer to Chapter 5 for the operation details of each encoding method.

Table 3-3 indicates the specifications on the encoding method of visual mono-media desired to be presented in C-profile basic receiver unit.

Table 3-3 Visual mono-media desired to be presented

Encoding method		Contents of specicification	
Video coding	H.264| MPEG-4 AVC	Transmission method	Video PES; stream format identifier = 0x1B
		Video size	QVGA (320x240 (4:3) 320x180 (16:9)) (*1)
		Scaling	Implementation dependent. Not assigned by multimedia contents.

(*1) Order of numerical values is transversals and longitudinal

Table 3-4 indicates the specifications on mono-media encoding method desired to being presented in BML browsers in C-profile basic receiver units.

Table 3-4 Visual mono-media desired to be presented on BML browser

Encoding method		Contents of specification	
Still image coding	JPEG	Transmission method	Data carousel; stream format identifier = 0x0D
		Image size	Maximum 240x320 (*1)
		Scaling	Implementation dependent. Not assigned by multimedia contents.
Graphic coding	GIF	Transmission method	Data carousel; stream format identifier = 0x0D
		Image size	Maximum 240x320 (*1)
		Scaling	Implementation dependent. Not assigned by multimedia contents.
	animation GIF	Transmission method	Data carousel; stream format identifier = 0x0D
		Image size	Maximum 240x240 (*1) (*2)
		Scaling	Implementation dependent. Not assigned by multimedia contents.
Font coding	Shift JIS	Transmission method	Data carousel; stream format identifier = 0x0D
		Font	Desired to be selected among small, medium, and large.
		Others	For operation details refer to ARIB STD-B24, Vol. 2, Appendix 4.

(*1) Order of numerical values is horizontal and vertical

(*2) Maximum 240 for both x-y coordinates. Refer to section 5.2.2.11 for details.

Table 3-5 indicates the specifications on the encoding method of desired visual mono-media being presented by closed caption in C-profile basic receiver unit. Refer to Chapter 6 for the operation details of each encoding method. However, the adoption of closed caption is implementation dependent.

Table 3-5 Desired visual mono-media to be presented by closed caption

Encoding method		Contents of specification	
Character coding	8-bit character codes for C-profile	Transmission method	Independent PES; stream format identifier = 0x06
		Font	Medium (Implementation dependent)

3.2.4 Audio playback functions

The specifications on each encoding method for the desired audio mono-media being presented in C-profile basic receiver units in shown in Table 3.6. Refer to Chapter 5 for operation details of each encoding method.

Table 3-6 Desired audio mono-media to be presented

Encoding method	Contents of specification	
MPEG-2 AAC	Transmission method	Audio PES; stream format identifier = 0x0F Data carousel; stream format identifier = 0x0D
	Sampling frequency	24kHz, 48kHz
	Maximum file size for continuous playback	128KB
Built-in sound	The encoding method is implementation dependent.	

3.2.5 Fonts

Table 3-7 indicates the specifications of desired fonts to be implemented into receiver units.

Table 3-7 Fonts

Item	Contents of specification
Number of Font face	Implementarion dependent
Character type	Kanji (level 1, 2), Hirakana, Katakana, alphanumeric, symbol, etc. (*1)
Character size controls	Recommended to be selected among small, medium, and large. The number of dots is not specified.
Grayscale font	Implementation dependent (Not assigned by contents)

(*1) For details about on types, refer to ARIB STD-B24, Vol. 2, Appendix 4.

3.3 Buttons (remote controller)

3.3.1 Keys to be used in C-profile reception

Keys expected to be used in reception are classified into following 3 groups.

- Group 1: Key events handling by BML contents
- Group 2: Key events handling by BML browsers
  (Do not handle by BML contents)
- Group 3: To be used for receiver features.

Table 3-8 indicates the type of keys to configure each group, contents production, and the guidelines at the moment of designing receiver units. It is recommended that each key have its dedicated buttons however, the buttons can share similar functions. In this case, it is

assumed that the names of the following key types and names of buttons are different by implementation dependent, and that at least 2 different names, such as one name used in receiver units and another name used by data broadcasting, are used for a single button. Also, multiple buttons shall provide functions for the same key types. The implementation of TVlink button is optional.

The method for key input is assumed to be implemented on the hardware by using dedicated buttons, and on the software by using on a screen keyboard. It is implementation dependent which the receiver unit adopt method, however, this specifications are specified for an implementation method in view of simple user operation that are required in the contents. So it is mandatory implementation that the configuration of receiver units with suitable buttons on a remote controller.

On the other hand, it is permissible implementation on a software that the configuration of receiver units without a suitable buttons on a remote controller, but in this specifications, it is mandatory implementation that the configuraition of the receiver unit with suitable buttons on a remote contoroller, for example, displaying a list of keys on a popup screen which is selectable by a single user operation, should be implemented to allow of directly user operation as possible.

Table 3-8 keys for C-profile

Key types		Implementation		Guidelines
		Mandatory / Optional	Physical Button	
Group 1: Key events handling by BML contents				
	Selection	◎	◎	Such as “Selection” and “Execution”
	Return	◎	◎	Such as “Cancel” and “Return”. Cancel of the user operation. Back space of user input character. (Or batch delete process) (*) BML document can be used for Return purpose however, pay consideration to the existence of the return destination.
	*	◎	◎	Selection (execution) of the user operation
	#	◎	◎	Selection (execution) of the user operation
	0 to 9 (numerical keys)	◎	O	Numerical input. However, the contents are assumed that numerical keys are not allocated to receiver unit specification or user setting, and both options of “Execute by up and down focus movement” and “select directly by numerical keys” have to be necessary.
	TVlink	Δ	Δ	Writing of TVlink
Group 2: Key events bandling by BML browsers (Do not handle by BML contents)				
	↑, ↓(Up and down keys) Jog dial	◎	◎	Movement of scrolling and selection focus.
	←, → (Left and right keys)	Δ	Δ	Movement of focus.
Group 3: To be used for receiver features				
	TV key	◎	O	TV On-Off functions. Selection of active window.
	Volume UP/DOWN	◎	O	Volume control
	TVlink list	◎ (*1)	O (*1)	Call TVlink list function
	Channel UP/DOWN	O	O	Tuning
	EPG key	O	O	EPG On-Off function

◎: mandatory, O: desired, Δ: optional

(*1) These are basic functions in this specifications, however, receiver units not implemented with TVlinks are exceptions.

3.3.2 Key masks

Multimedia contents of C-profile comply with ARIB STD-B24, and can execute key masks. But keys (numerical keys, “*”, “#”) supposed to be used as one-touch tuning cannot be executed by key masks unless it is necessary. Also, depending on the receiver unit, ”*” and ”#” may be assigned to one-touch tuning numbered 10 or higher, so the assign and release of key masks on numerical keys, ”*” and ”#” are desired to be done simultaneously.

Also, in C-profile basic receiver units, it is assumed that main functions are not in C-profile basic receiver units but in mobile phones. In this case, in order to provide functions to devices in adaptive ways, it is assumed that the keys being acquired are not by multimedia contents.

3.4 Memory capability of receiver units

In C-profile basic receiver unit refer to section 3.1.2 for the specifications of each memory on receiver unit.

3.4.1 RAM

In the reference model, receiver units implement various memorys. As the use of RAM memory is expected, Bcontents and XfrBA memories are specified. See Table 3-9 for each RAM size. For details refer to section 3.1.2.

Table 3-9 RAM

Item	Contents of specification
Bcontents	512KB
XfrBA	128KB

For the buffer size of closed caption, refer to Chapter 6

3.4.2 NVRAM

BproNV, the mainly non-volatile memory implemented in receiver units of data broadcasting, as indicated in the reference model, stores information on each receiver unit user and specific information of each broadcaster. The area type and capacity usable as BproNV is indicated in Table 3-10. The area of BproNV, apart from the area of mandatory implementation, is capable of the implementation of the dynamically allocation areas where data is stored.

Table 3-10 Type and capacity of BproNV

Type	Capacity
C-profile memory area for the affiliation	288KB (12 affiliations x 24KB)
TVlink area	Over 12.8KB (More than 50 cases)

For details of BproNV, refer to section 7.2. Regarding about the area shown in Table 3-10, for security protection of receiver units, it should be configured in a way that the viewer cannot read or write related provisions, function of receiver units not specified in this specifications, and function of devices connected to receiver units. In order to satisfy the above constraints, a corresponding area type should be allocated within the memory device inside the receiver unit like NVRAM. For details refer to section 3.2.1.

3.5 Communication function

Assumed communication function by two-way communication on data broadcasting services of C-profile is shown in Table 3-11.

Since the communication functions related to mobile phone are dependent on the telecommunication line (i.e. telecommunications carrier), it shall be necessary to confirm them individually with the telecommunications carrier.

Table 3-11 Communication functions

Item	Contents of specification
Non secure communication	Use of communication functions nominating the namespace starting by “http://”. Communication with an http server on the Internet is required.
Secure communication	Use of communication functions nominating the namespace starting by “https://”. Communication with an https server on the Internet is required.

- The receiver shall implement HTTP/1.1 as defined in RFC2616.
- The receiver shall implement "Keep-Alive". Implementation of other headers shall be implementation dependent.

The receiver shall implement the TLS1.0 and SSL3.0 or TLS1.2 protocols for encrypted communication. See Vol. 6 for the implementation of TLS and SSL on receivers.

- Cache-Control and pragma:no-cache are expected to be instructed explicitly due to program configuration. Therefore, if Cache-Control or pragma:no-cache has been specified, it shall be desirable that the receiver interprets it and operates accordingly. Refer to Appendix 11 "1. Cacheing by the receiver function" for the details of the cache function.

Refer to Appendix 12 for the details of the communication function.

3.6 Character input function

The character input function is specified as a receiver's native application to assist character input to BML contents by user operation. The application supports character input into the input element and the textarea element of the BML contents.

The basic functions to be implemented for the application are mentioned below.

3.6.1 Function specification

The function specified by the application basically implementation dependent. However, the specifications are for character types that can be assigned in BML contents.

3.6.2 User interface

The character input application is permitted to receive exclusive key input and does not on the value of the used-key-list characteristic specified by the BML contents currently being presented.

3.6.3 Character types

Characters that can be entered into the input element and the textarea element refer to ARIB STD-B24, Vol. 1, Part 2, 7.3. However, the Kanji set allocated in Row 90 to Row 94 is excluded. For details refer to section 7.9 in this specification.

3.6.4 Kana Kanji conversion function

The implementation of the function and the function specifications are implementation dependent.

3.7 TVlink

3.7.1 Introduction of TVlink

In C-profile, the load on the functions and implementation required in mobile reception is considered, and an cross-media “bookmark” provided in ARIB STD-B24 and adopted by profile A is not used but TVlink is specified.

3.7.2 What is a TVlink?

This is a function to write the link information to a communication sites where information related to the programs stored in BML contents previously and is operated by the viewer pressing buttons.

The flow of the TVlink service is as follows.

1. There is a communication site where information related to the contents of broadcasting is offered, and when the broadcaster provides this service, the broadcasting contents present a TVlink icon on BML browser following the timing specified previously.
2. The viewer permits the writing of the TVlink by pressing a button when they are interested in the program contents (When the TVlink icon that is stored previously is displayed on the screen). The contents to write the TVlink are wrote to the URI and NVRAM of the communication site that provide the related information.
3. After that the viewer specifies the desired TVlink from the TVlink list screen offered as a receiver features, and browses the communication contents of the site with related information.

The contents for writing the TVlink is assumed in broadcasting as a part of the BML contents associated with A/V program and CM contents.

The destination link to be written as a TVlink is assumed to be the external HTTP server.

3.7.3 Changed points from profile A bookmarks

The changed points from profile A bookmarks of the TVlink are mainly the following three points. See Table 3-12.

(1) "TVlink list" by receiver features transits to the destination link when the viewer selects the desired TVlinks from the displayed list and the list of the TVlink. The TVlink list cannot be displayed by the BML contents, or read the TVlink. Therefore, the implementation of a TVlink list by a receiver features is mandatory.

(2) Transition to the destination link when the viewers select the desired TVlink from the TVlink list, including transition to the type of TVlink that presents communication contents depends on the link conditions. (In profile A, when presenting the communication contents as linked status, implementation is done by a combination of the receiver features and BML contents. In C-profile, it is done only by receiver features.)

(3) Arranging the write items and the extended zone is omitted.

Table 3-12 Comparison between TVlinks and bookmarks of profile A

	Write	Read	List display / transition to destination link	Extension area
"TVlink"	BML contents	Receiver unit function	Receiver unit function	None
"bookmark" of profile A	BML contents / (receiver unit function)	Receiver unit function / BML contents	Receiver unit function (remarks) / BML contents	Exist

Remarks: When the bookmark is bmType=009, it is a combination between tuning by receiver features and transition by BML contents

3.7.4 NVRAM in TVlink

3.7.4.1 Number of TVlink areas

The number of TVlinks is implementation dependent, but writing more than 50 is recommended.

3.7.4.2 Sharing with other medias

The NVRAM area of TVlinks may be shared with bookmark writing service of other media if there are compatible with the specifications of TVlink type, parameter type, data length, and recordable/non-recordable. However, sharing is not possible with bookmark areas of profile A receiver units etc.

3.7.5 Format and details of TVlink

3.7.5.1 Format of TVlink

The parameters used by TVlink and the length of its data, necessity in writing to NVRAM, necessity in displaying the TVlink list function of receiver units are indecated in Table 3-13. The maximum required number of bytes for the NVRAM area necessary for 1 TVlink is assumed to be 256. However, the write format is arbitrary, and the length of the table data (bytes) is a reference because it is not read from BML contents.

Table 3-13 Data configuration of TVlinks

Parameter	Length (bytes)	CproBMtype (*1)					Remarks
		0	1	2	3	4	
title	Max. 40	O/O	O/O	O/O	O/O	O/O	
dstURI	Max. 60	Δ/Δ	O/Δ	O/Δ	O/Δ	O/Δ	(*2)
outline	Max. 130	O/O	O/O	◇/◇	◇/◇	◇/◇	List display not required
CproBMtype	1	O/Δ	O/Δ	O/Δ	O/Δ	O/Δ	
expire	5(*3)	◇/◇	◇/◇	◇/◇	◇/◇	◇/◇	YYYYMMDDHHmm format
network_id	2	Δ/X	O/X	Δ/X	Δ/X	Δ/X	
service_id	2						
affiliation_id	6(*4)						
reserved	10						

(Write/Display) O: Mandatory, ◇: Desired, Δ: implementation dependent, X: Prohibited

(*1) Necessity of support for each CproBMtype is implementation dependent.

(*2) Character used in URI specifications in ARIB STD-B24, Vol. 2, section 9.2.

(*3) Assumed to be written as a numerical value.

(*4) The affiliation_id described in the BIT is 1 byte each and maximum 6.

3.7.5.2 Details of each parameters

- Details of titles (title)
  - The character string displayed in the title column of the TVlink list is written.
  - The title is a maximum of 40 bytes. (The length code is not included). The title when exceeding 40 bytes, the receiver unit behavior is implementation dependent. However, it is recommended that the receiver unit writes 40 bytes, and does not write 41 bytes or more. Moreover, if the 40th byte is the first byte of two-byte character, it is recommended not to write the first byte.
  - It is necessary to write this information into NVRAM.

- Details of URI of destination links (dstURI)
  - The URI of communication contents of destination links is written.
  - The URI of the destination link is maximum of 60 bytes (The length code is not included). The URI of the destination link when exceeding 60 bytes, the receiver unit does not write them, and returns NaN as the return value.
  - The null character string is specified for memo information (CproBMtype=0).
  - It is necessary to write the information into NVRAM.

- Details of TVlink outline (outline)
  - The outline of communication contents of destination links is written.
  - If no TVlink outline is given, a null character string is specified.
  - The TVlink outline is to be a maximum of 130 bytes (The length code is not included). The TVlink outline when exceeding 130 bytes, the receiver unit behavior is implementation dependent. However, it is recommended that the receiver unit write 130 bytes, and does not write 131 bytes or more. Moreover, if the 130th byte is the first byte of two-byte character, it is recommended not to write the first byte.
  - It is necessary to write the TVlink outline into NVRAM, TVlink type in memo information (CproBMtype=0), and the TVlink of C-profile linked contents (CproBMtype=1). Other types are optional. However, it is recomemened to write the TVlink outline for the improvement of the user operations.

- Details of TVlink type (CproBMtype)
  - The TVlink type indicates the TVlink of the destination link contents. The receiver unit can select a browser that can display the communication contents specified by the URI destination link using this information.
  - The response of the TVlink type is shown in Table 3-14.
  - TVlink types are specified by numerical values from 0 to 255.
  - It is necessary to write this information into NVRAM. Display is optional.

- Details of expiration dates (expire)
  - The expiration date shall be written as a final date of the TVlink.
  - The expiration date is always specified by year, month, day and time.
  - The argument of the write function (X_DPA_writeCproBM()) can be omitted. When omitted, the TVlink is always available.
  - Writing this information into NVRAM is optional.

3.7.6 Operation of TVlink types

Table 3-14 indicates the TVlink types.

Table 3-14 TVlink types

	TVlink type
0	Memo information (no destination link)
1	C-profile linked contents (BML contents)
2	C-profile unlinked contents (BML contents)
3	HTML contents
4	Communication contents assumed to use specific networks
5 to 255	Reserved

The operation of TVlink types is as follows.

- Operation of memo information (CproBMtype: 0)
  - This type is used to write text information related to the broadcast program in the memo.
  - URI destination link is not specified in this type. A null character string is specified for the argument of the write function (X_DPA_writeCproBM()). When characters other than null characters are specified, it fails in writing.
  - The writing and reading of the TVlink of memo information (CproBMtype=0) are necessary.

- Operation of C-profile linked contents (CproBMtype: 1)
  - This type is assigned to write link information to C-profile linked contents (BML contents) assumed to be presented by BML browsers.
  - The writing and reading of the TVlinks of C-profile linked contents (CproBMtype=1) are necessary.

- Operation of C-profile unlinked contents (CproBMtype:2)
  - This type is assigned to write link information to C-profile unlinked contents (BML contents) assuming the presentation by the browser for the C-profile contents on the internet.
  - The writing and reading of the TVlinks of C-profile unlinked contents (CproBMtype=2) are optional. It is recommended to be able to write the TVlinks of this type in receiver units which the browser for the C-profile contents on the internet.

- Operation of HTML contents (CproBMtype:3)

- This type is, for example, used to assign link information to HTML contents that are assumed to be presented in HTML browser implemented in PDA's etc.
- The writing and reading of the TVlinks of the HTML contents (CproBMtype=3) are optional. It is recommended to be able to write the TVlinks of the types in receiver units which the browser that can display HTML.

- Operation of contents on proprietary network (CproBMtype: 4)
  - This type is assigned to write link information of communication contents that are assumed to be presented in HTML browsers using specific networks implemented in mobile phones, etc.
  - The writing and reading of the TVlinks of contents on proprietary network (CproBMtype=4) are optional. It is recommended to be able to write the TVlinks of this type in the receiver units which carrier's proprietary browser assumed to use a specific network with receiver units such as mobile phones, etc. On the other hand, it is recommended not to write the TVlinks of this type in receiver units without carrier's proprietary browser assumed to use a specific network.

3.7.7 Guidelines for the functions for each TVlink type and receiver unit

The following indicates the read operation of each TVlink type of the TVlink list function implemented as receiver features and the behavior of the TVlink write function (X_DPA_writeCproBM()) of each TVlink types.

- In case of memo information (CproBMtype=0)

  (Function behavior when writing)
  - When this type is assigned to the argument, it writes without conditions.
  - Writing fails when a null character string is assigned to the argument (URI of the destination link).
  - When a TVlink area of writing NVRAM cannot be secured, -1 is returned as a return value.

  (Receiver unit behavior when calling)
  - When this type is selected from the TVlink list by the user operation, only the TVlink outline (outline) is displayed, and the transition is not executed.

- In case of C-profile linked contents (CproBMtype=1)

  (Function behavior when writing)

- In the case where this type is assigned to the argument, when the TVlink is written the receiver unit writes the network_id, service_id and affiliation_id of that broadcaster with individual TVlinks.
- When the TVlink area of the NVRAM that writes cannot be allocated, -1 is returned as a return value.

(Receiver unit behavior when calling)

- When this type is selected from the TVlink list by the user operation, some of following behaviors are executed.

(Behavior 1)

- Tuning is by the writed network_id and service_id. The display of the Video display area, when the broadcaster is outside the reception area or when tuning cannot be executed, is implement dependent. The presentation of the communication contents is executed even if tuning fails.
- The receiver unit obtain the contents specified for URI via communication line, and presented in BML browser as linked status. However, after tuning, and until the communication contents are presented, the broadcasting contents are not presented in BML browser. The behavior of not being able to transit by URI is implementation dependent.
- The access control of NVRAM is used by the wrote network_id and affiliation_id as the TVlink information regardless of the success or failure of tuning.
- The time handled by the Date object of ECMAScript, when tuning fails is recommended to be the TOT obtained by the receiver unit at last or the time revised by other means, or an error may be returned.

(Behavior 2)

- The receiver unit is not executed always reception or tuning and the contents specified in the URI via communication lines are obtained and presented in BML browser in a linked status.
- The access control of NVRAM is used by the wrote network_id and affiliation_id.
- In this case, it is recommended that BML browser display in full screen.
- The time handled by the Date object of ECMAScript is recommended to be a time revised by the TOT obtained by the receiver unit at last or the time revised by other means, or an error may be returned.

- In case of C-profile unlinked contents (CproBMtype=2)

(Function behavior when writing)

- When the receiver unit does not implement the browser for the C-profile contents on the internet, in principle it does not write and -2 is returned as a return value. However, when the presence of the implemented browser for the C-profile contents on the internet cannot be confirmed from BML browser, it does not follow this process.
- When the TVlink area of writing NVRAM cannot be allocated, -1 is returned as a return value.

(Receiver unit behavior when calling)

- When this type is selected from the TVlink list by the user operation, the browser for the C-profile contents on the internet starts and the C-profile unlinked contents are displayed. The receiver units behavior that cannot start the browser for the C-profile contents on the internet is implementation dependent.

- In case of HTML contents (CproBMtype=3)

(Function behavior when writing)

- The receiver units, in principle, do not write when they do not have browsers that can display HTML, and return -2 as a return value. However, when the presence of an implemented HTML browser cannot be confirmed from BML browser, it does not follow this process.
- When the TVlink area of writing NVRAM cannot be allocated, -1 is returned as a return value.

(Receiver unit behavior when calling)

- When this type is selected by the user operation from the TVlink list, the HTML browser starts, and the HTML contents are displayed. The receiver units behavior that cannot start browsers which can display HTML is implementation dependent.

- In case of specific network communication contents (CproBMtype=4)

(Function behavior when writing)

- The receiver units, which do not implement the browser that can display contents for a specific network like mobile phones, etc., in principle, do not write and return -2 as a return value. However, when the presence of the implemented browser cannot be confirmed from BML browser, it does not follow the process.
- When the TVlink area of writing NVRAM cannot be allocated, -1 is returned as a return value.

(Receiver behavior when calling)

- When this type is selected by the user operation from the TVlink list, a browser that can display contents for a specified network starts, and the communication contents are

displayed. The receiver units behavior that cannot start the broadcaster is implementation dependent.

3.7.8 Writing of TVlinks

- The writing into NVRAM of TVlinks is done by the broadcasting contents and C-profile linked contents.
- The TVlink is written without fail based on the permission of the viewer. The contents should not be written automatically without the permission of the viewer.
- When the TVlink is wrote specifying CproBMtype for the option, it is recommended to display the TVlink once the presence of the correspondence of such receiver unit concerned for CproBMtype is examined by contents.
- From the viewpoint of the protection of personal information, the implementation of functions that write or update by receiver features for the TVlink of C-profile linked contents (CproBMtype=1) is not recommended.
- It is recommended not to implement the writing or the update function of TVlinks of other types by receiver features.
- Below, the contents that write the TVlink indicate the receiver behavior when the write function (X_DPA_writeCproBM()) is called.
  - The receiver unit judges whether the receiver unit can display the communication contents according to the TVlink type specified by the argument, and when it is not possible to display, it is recommended not to write it. In this case, the return value of the function returns -2. Moreover, it is recommended to display the reason why the viewer cannot write.
  - When there is no TVlink area available, the following wrote the TVlink can be deleted before writing new TVlinks. In this case, present the title of the TVlink to be deleted to the viewers and obtain user permission.

    (1) The ones that have expired.

    (2) The oldest TVlink.

    1 is returned as the return value of the function when successfully writing the TVlink (after obtaining permission from the viewers). When user permission is not obtained, a failure (NaN) is returned.

    Moreover, in receiver units that do not execute such deletion processes, when no TVlink area available, -1 is returned as a return value.

3.7.9 Guidelines of TVlink list functions

A list of TVlinks is displayed, and it is necessary that the "TVlink list" have a function that can transit the communication contents desired by the viewer for implementation as a receiver features. The function is indicated as follows. Issues not mentioned are implementation dependent.

3.7.9.1 Start and end of the TVlink list by receiver's native application

- Initiation of the TVlink list by the receiver's native application is done by using the buttons on the remote control unit of the receiver unit and the function (X_DPA_startResidentApp()) of the multimedia contents. This function is described later.
- When the TVlink list starts by the receiver's native application during the presentation of multimedia contents, the display of the multimedia contents may be interrupted. However, it is recommended that the presentation of the multimedia contents continues.
- The TVlink list screen display method, for example, is repeatedly displayed in the data area. At this time, it is recommended that the display of video continues.
- Note that key input may not be obtained by BML browser while the TVlink list screen is displayed when authoring contents.
- When the display of multimedia contents is interrupted and the TVlink list is displayed, and a TVlink is selected by the viewers, and the TVlink list ends without being transited, it is recommended that the interrupted multimedia contents be displayed again, and the presentation continues. The key input also returns to the multimedia contents.
- When a TVlink list starts during the display of multimedia contents, BML browser continues the asynchronized operation with the TVlink list while presenting the TVlink list, and the screen presented when the multimedia contents is displayed again after the TVlink list ends, and it depends on BML browser behavior while presenting the TVlink list. BML browser continue asynchronous operations with the TVlink list during the presentation of multimedia contents. Therefore, when the TVlink list starts during the presentation of multimedia contents and the TVlink list finishes to redisplay multimedia contents, the screen displayed depends on BML browser behavior executed during the presention of the TVlink list. For example, when the data event changes while displaying the TVlink list, the display contents of the data broadcasting at the moment that the TVlink list starts, and the display contents when the TVlink list ends are different. Even if the data screen is displayed, when the TVlink list display ends, the data broadcasting might not be displayed.

3.7.9.2 Display of TVlink information

- The TVlink list has the title list display function of the TVlink.
- The display of "URI destination link" is optional.
- "TVlink explanation" cannot necessarily be displayed in the list, and it can only be displayed by the user operating the buttons, etc.
- The display of the "Expiration date" is optional, however, display is recommended.
- The information indicated by "TVlink type" is optional.
- It is recommended that the "Bookmark" wrote by the communication browser other than BML browser should be distinguished clearly in view of the viewer. When the bookmark wrote by the TVlink and a communication browser is displayed by one application, special consideration is necessary. (Especially, TVlinks of C-profile linked contents (CproBMtype=1)).
- In the list display, it is recommended to display, for the understanding of the viewr, that the TVlink type is a memo information (CproBMtype=0) item without a destination link.
- Specification for other displays is implementation dependent.

3.7.9.3 TVlink selection and transition to the destination link

- The viewer can select the TVlinks from the TVlink list and display the communication contents shown in the URI link destination included in the selected TVlink.
- At this time, when necessary, by referring to the TVlink types listed in the TVlink, one can start a browser that can present the communication contents of the specified type.
- When an expired TVlink is selected by the viewer, it has to specify to the viewer if the expiration date has passed, and not to transit to specified communication contents. At this time, it is recommended to delete the TVlink with permission of the viewer.

3.7.9.4 Deletion of TVlink information

- TVlinks are deleted by a TVlink list function (receiver features). As it cannot be deleted from contents, implementation of the deletion function is necessary.
- Deletion of the TVlink is based on the permission of the viewer.
- It is recommended in adequate timing, that expired TVlinks be automatically deleted with permission of the viewer. (*)The timing of automatic deletion is implementation dependent.
- Additionally, the specification of the deletion function in the user interface, etc., is implementation delendent.

* As the writing of expiration dates is optional, it is necessary to note that TVlinks that have expired may not be deleted implementation dependent even if the expiration date is specified at the time of writing. In this case, a TVlink once wrote remains in the receiver unit until the viewer deletes it, so special consideration needs to be taken by

broadcasting operators in order not to fail the communication contents at the destination link.

3.8 BML browser

Regarding the functions that BML browser should have, the hardware resources and the software resources are specified in these sections. In the following sections, the specification of transmission, mono-media and multimedia encoding is mentioned. This section, only provides specifications that are not included in the following sections.

3.8.1 Browser specific display

The browser should not display a browser specific display because these contents may hide the intended presentation content. This is a restriction to avoid display loss of the contents by the allocation of the browser's behavior button for the URI display frame done in general and by a browser of the computer and the browser's behavior. However, this restriction doesn't intend to get rid of channel banners and permission etc, when executing the browser. Note that the regulations of mixed display apply to the display of the browser for the C-profile contents on the internet in operation, as well as to the browser for the C-profile contents on the internet. (Refer to section 8.1.4.)

For BML browser, individual display outside the display frame is not assumed. However, this does not restrict implementing a function by implementation dependent to assist the function required by the browser specified in this specifications or for the intention of assisting it. For example, the display of the scroll bar related to the scrolling representation etc., is assumed for this. On the other hand, the functions not provided in this volume, like of the display of URI contents, should not be displayed.

3.9 Browser for the C-profile contents on the internet

This chapter, in receiver units that receive video, audio and multimedia data specified by this chapter, specifies options that may be implemented in the browser for the C-profile contents on the internet as well as BML browser specified in this section etc.The use of each browser and each relationship is specified follows.

1) BML browser

BML browser is work based on C-profile specifications in this volume. The browser is executed in the data broadcasting reception status and linked status, and are used for presenting the provided contents when broadcasters services by broadcasting business, via broadcasting signals and communication lines.

2) Communication browser

The browsers for the contents on the internet are classified into the following the browser for the C-profile contents on the internet and the carrier's proprietary browser. In this volume, they are positioned as one of the receiver's native applications from the viewpoint of BML browser. Therefore, the start by BML browser as the specifications in Chapter 8, uses the same method (start by X_DPA_startResidentApp()) for starting applications. The browsers are used to present contents located on servers that supply services other than broadcasting. Mixing display with the broadcasting contents is limited.

a) Browser for the C-profile contents on the internet

The browser is the specifications (element, attribute, CSS, DOM, and ECMAScript, etc.) related to the BML of C-profile specicications in this volume. However, it is necessary to reatrict some of the extended functions. Moreover, it is prohibited to refer to and transit the broadcasting contents. (Refer to Appendix 5.)

b) Carrier's proprietary browser

It is assumed that browser is implemented based on the specifications defined by the carriers. In this volume, the specifications only for the area used as receiver's native applications in C-profile receiver unit are specified, and the specifitions for the internal part are not specified.

c) HTML browser

It is assumed that browsers are implemented for viewing HTML contents. In this volume, the specifications only for the area used as receiver's native applications in C-profile basic receiver unit are specified, and the specifications for the internal part are not specified.

The following figure indicates the relationship of each browser.
HTML contents can be presented by HTML browser implemented in the receiver unit besides the browser for the C-profile contents on the internet and the carrier's proprietary browser shown in the figure.

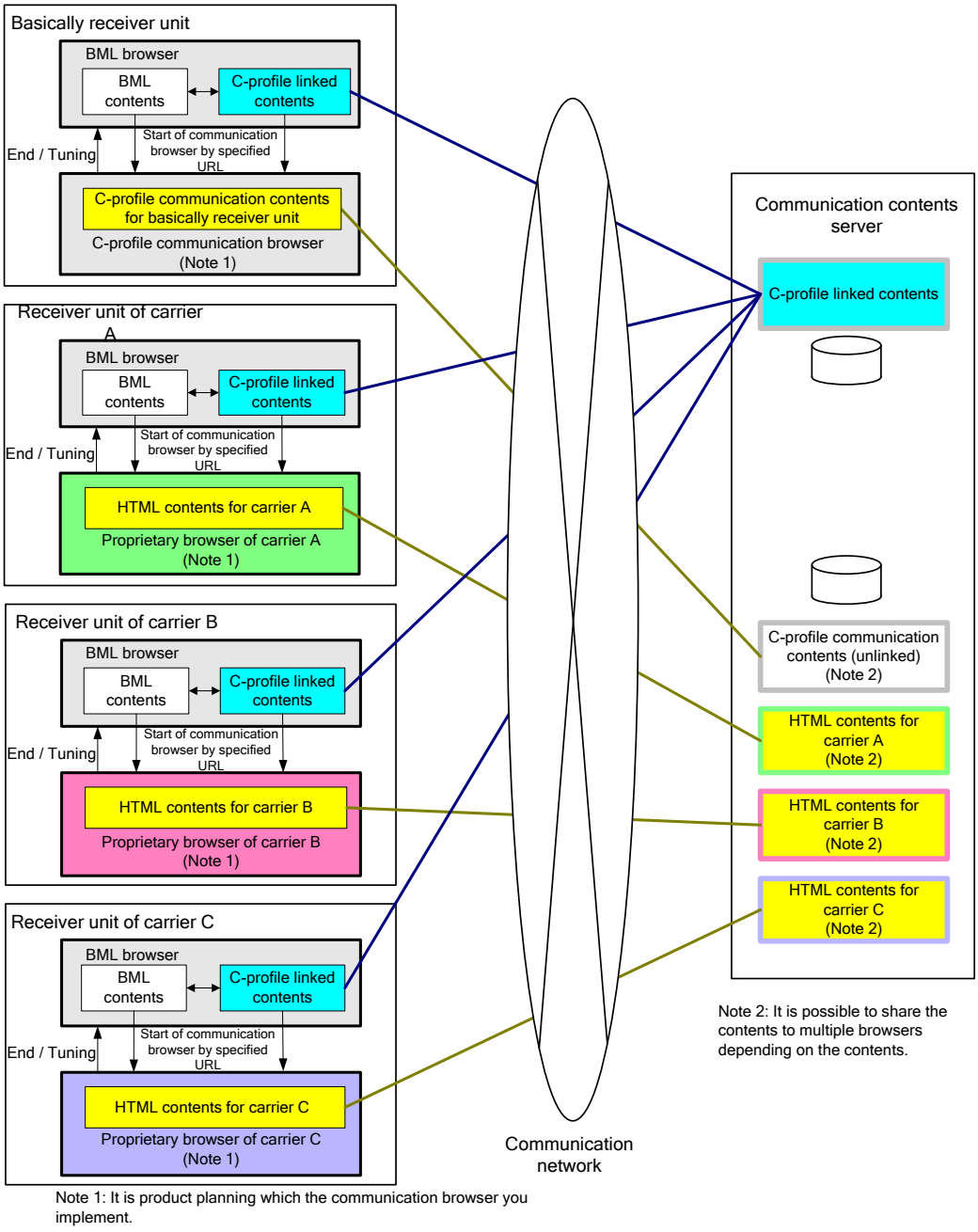


Figure 3-4　Relationship between BML browser and the browser for the contents on the internet

4 Operation of data transfer format

In this chapter, newly defined items, differences or operational limits related to the operation of the transmission method of the multimedia encoding method mainly as an XML base are described based on the provision contents with the described applicable document, as follows.

In this chapter, the data broadcasting service indicates the multimedia data broadcasting service by the multimedia encoding method of the XML base, unless otherwise mentioned.

Items without the following special description are applied as is for the provision contents within the applicable document.

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

4.1 PSI/SI

4.1.1 Type of data broadcasting services

4.1.1.1 Channel service type operated by data broadcasting programs

The value of the service_type is 0xC0.

4.1.2 Configuration of the data broadcasting service contents and operation of the components

4.1.2.1 Contents and local contents

The definition of contents and local contents and their relationship to events are shown in Table 4-1 and Figure 4-1　Local contents and events.

Table 4-1　Definition of contents and local contents

	Definition	Operation
Contents	Shows the sets of local contents transmitted in the event period of a certain component. It is identified by the Data Contents Descriptor.	The operation of the contents cannot be specified.
Local contents	Sets of BML documents transmitted in data events with a certain component.	Multiple local contents might be broadcast, alternatively one by one, in process of time, within one component.

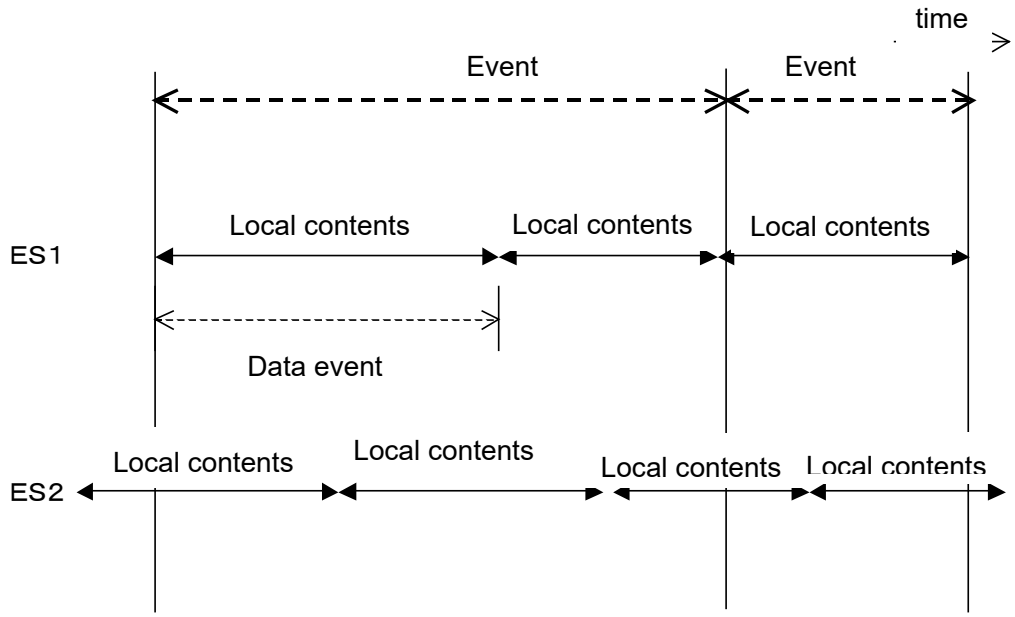


Figure 4-1 Local contents and events

4.1.2.2 Relationship between local contents and ES

A local content is transmitted by one data carousel (ES). The stream identification of the component that transmits the data carousel and the event message is always 0x0D.

4.1.2.3 Operation of component tags

The operation of the component tag value used by data broadcasting services including closed caption follows Vol. 7 section 5.1.1(2).

4.1.2.4 Configuration of the ES transmitted by 1 service

- Basic moving picture (H.264|MPEG-4 AVC): maximum 1ES
- Audio (MPEG-2 AAC): maximum 2ES
- Data carousel for C-profile: maximum 2ES
- Event massage for C-profile: maximum 2ES (do not operate NPT)
- Closed caption: maximum 1ES
- PCR: maximum 1ES

4.1.2.5 Identification of entry components

Default components of data-broadcasting programs are always transmitted by the tag value component of 0x80 in C-profile. Identification of the entry component that transmits a startup document when a channel is selected, is not the entry_point_flag of the Data Component Descriptor and is this tag value. The entry component transmits the entry module (moduleId=0x0000), and the entry module includes one startup document.

4.1.2.6 Identification of event message components

When general event messages are transmitted by the component that transmits the data carousel, they are transmitted by components with a component tag value of 0x80 and 0x8B. When a general event message is transmitted with an ES other than the component transmitting the data carousel, it is always transmitted by components with a tag value 0x89 and 0x8A. 0x89 and 0x8A are the ES for exclusive use general event messages.

4.1.2.7 Detailed specifications for section data transmission

- During the data carousel and the event message transition, multi section transmission shall not be conducted (transmitting two or more sections by 1 TS packet).
- The following standards are established in the transmission bit rate when each section data of the data carousel and the event message is transmitted.
  - Six TS packets or more are not transmitted continuously with the same PID.
  - The total bit rate of the component (4PID or less) received at the same time when one content is received is to be 650kbit/s or less. These include DII, DDB, and event messages.
  - The maximum bit rate of each 1 subtable is also 650kbit/s (2kB±100% for 32 milliseconds) or less.
  - It should be transmitted without exceeding the above-mentioned standards because when transmission exceeds these standards it is assumed that the reception efficiency of the section decreases depending on the receiver and the time required for the acquisition of necessary sections increases.

4.1.2.8 Default maximum bit rate for data broadcasting programs

Follows Vol. 7.

4.1.3 Operation of PMT

4.1.3.1 PMT operation of specific data broadcasting services

- Refer to section 4.1.5.2 for the receiver operation during the update of the PMT.
- The components that allow the ES not to exist although it is registered in the PMT are the following three.
  - Component transmitting data carousels
  - Component transmitting closed caption
  - Component transmitting only event messages
- However, the display indicating the data under reception might keep appearing for sometime due to the fact that receivers try to acquire data when the ES of the data carousel doesn't exist even though it is registered in the PMT. Empty carousels should be

kept sending into components of at least 0x80 even if there is no sending data on the transmission side in order to avoid this status. Refer to Appendix 8.

4.1.3.2 Operation of the Data Component Descriptor in the PMT

The Data Component Descriptor is allocated in the following components.

- Component transmitting closed caption
- Refer to Chapter 6 for detailed operation of the Data Component Descriptor allocated in the closed caption component.
- Component transmitting data carousels
- The Data Component Descriptor is not allocated in other components.
- The operation of the Data Component Descriptor is shown in Table 4-2.

Table 4-2 Operation of Data Component Descriptor

Flag	Operation
data_component_id	0x000D
Contents of additional_data_component_info(additional_arib_bxml_info())	
transmission_format	Shall be 00 (Data carousel transmission method and event message transmission method).
entry_point_flag	• Component component_tag=0x80 shall be 1. (When the data broadcasting program is selected, the component of component_tag=0x80 transmits the module including the document initially starts.) The rest of the components shall be 0. • When the receiver selects stations, acquires and presents a startup document of the data carousel transmitted by the component of component_tag=0x80.
auto_start_flag	Always auto_start_flag=1.
document_pixel size	Always 1111.
use_xml	0 (XML that uses the application dependence tag is not transmitted.).
default_version_flag	Always 0.
independent_flag	Always 1.
style_for_tv_flag	No operation. Value is fixed in 0.
bml_major_version, bml_minor_version	Operated as specified. If the bml_major_version is 12 and when these fields are allocated, the mobile basic receiver judges reception as possible. bml_minor_version always specifies 0. For receiver behavior concerning the version, refer to Appendix 9.
ondemand_retrieval_flag	Always 1.
file_storable_flag	Always 0.
Operation of additional_arib_carousel_info()	
data_event_id	Not operating in PMT. Value is fixed at 0xF(1111).
event_section_flag	Always 1.

4.1.3.3 Target Resion Descriptor

Not operated.

4.1.4 Operation of the L-EIT Data Contents Descriptor

Not operated.

4.1.5 Related receiver operation

Prior condition of data transmission operation specifications

- The data_event_id allocated in the Data Component Descriptor of the PMT data is assumed as fixed (0xF), and is not used.
- The component of the component_tag=0x80 is an entry component, and the data carousel transmitted by the entry component is called an entry carousel.
- Entry component (0x80) of C-profile may not exist. However, the PID value doesn't change while it exists. The receiver behavior when PID value changes is specified in section 4.1.5.2.

4.1.5.1 Receiver operation at the beginning of data broadcasting

1. Playback is executed a stream of basic moving picture (H.264|MPEG-4 AVC) if the component of the component_tag=0x81 is included in the PMT 2nd loop. Moreover, Playback is executed as an audio stream of MPEG-2 AAC (sampling frequency = 24KHz) if the component of the component_tag=0x83 or 0x84 is included. Similarly, Playback is executed out as an audio stream of MPEG-2 AAC (sampling frequency = 48KHz) if the component of the component_tag=0x85 or 0x86 is included.
2. Identifies the entry component (component of the component_tag=0x80) among the components placed in the PMT 2nd loop.
3. If the data_componet_id of the Data Component Descriptor is 0x000D, and the entry component (0x80) exists in the PMT 2nd loop, it is considered as a data broadcasting program by the multimedia encoding method (C profile) of XML base, then proceed to following process. The data broadcasting service doesn't begin if the data encoding method doesn't correspond to the receiver.
4. Based on the version number of the BML specified by the Data Component Descriptor of the entry component, the presentation of data broadcasting is judged. The data broadcasting service is not presented when reception is judged as impossible.
5. The BML-engine begins, and acquisition and presentation of the startup document of the entry component starts after initialization of the BContents and Ureg.
6. However, when the entry component is an empty carousel after the BML-engine starts, the switching of data event of the entry component is monitored, and when a startup

module appears due to data event switching, then the acquisition and presentation of the startup document is done. (Refer to section 4.2.3 for empty carousels.)

4.1.5.2 Receiver behavior during update of the PMT

Receiver behavior (for the receiver that receives C-profile) when the PMT is updated during data broadcasting program reception.

- When the component of C-profile disappears during viewing

  - Initiates the presentation of a startup document of the C-profile entry carousel by destroying the document under presentation.

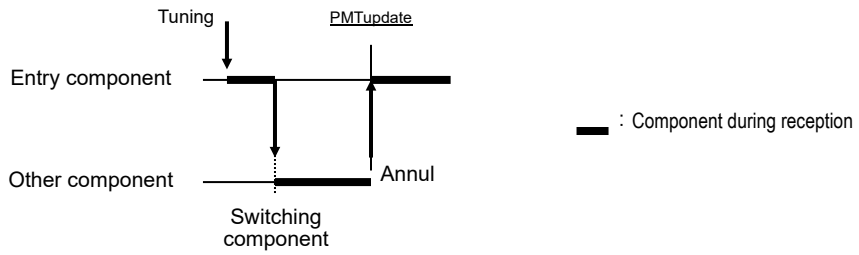


Figure 4-2 Operation when components disappear when the PMT is updated during reception

- When the entry component of C-profile (tag value = 0x80) disappears

  Since the component in the data broadcasting program disappears, the BML-engine ends. However, appendix 8 makes some recommendations on operation for when the link contents are presented.

- When the entry component of C-profile (tag value = 0x80) appears

  When the BML-engine has not started, the BML-engine starts, and behavior are done at the same time as tuning.

- When the PID of the component transmitting data carousel (including empty carousels) is changed during reception

  The receiver judge if the contents of the data program are changed and a process the re-tuning corresponding to the service displayed below is carried out.

  - The document being presented is destroyed.
  - All acquired resources are deleted from the BContents memory.
  - Re-acquisition and presentation of the startup document of the entry component.

4.1.5.3 Specifications on the partial transport stream output-input

When the receiver outputs the data component (component tag value 0x80 to 0x8F) to the partial transport stream, the output follows the descriptors shown in Table 4-3. (Follows the specifications Vol. 2)

Table 4-3 Descriptor output in the partial TS

Descriptor name	Description table	Allocation standard
Data Component Descriptor	PMT 2nd loop	Required
Extended Broadcaster Descriptor	SIT 2nd loop	Required
Broadcast ID Descriptor	SIT 2nd loop	Required

- When the partial transport stream is input and multimedia contents are playbacked, operations that are used when the function required for SI information not included in the partial transport stream and the function to tune implementation dependant. For example, the following broadcasting enhancing function is given as an example.
  - Tuning (X_DPA_tuneWithRF) and reservation (epgReserve)

4.2 Data carousel transmission operation

4.2.1 Data carousel transmission operation

- The maximum number of modules transmitted with one data carousel is 64 for C-profile.
- The module configuration of the data carousel may change according to the event under process and the time (the module increases and decreases). In that case, the DII version is updated.
- The transmission frequency will vary depending on the module that configures the data carousel.

4.2.1.1 Introduction of data event and local contents

- The concept of a data event that is not time related directly to the event is introduced to enable the switching of contents items at anytime, regardless of in a program or between programs, and switches the contents presentation by each data event unit. The contents transmitted in one data event are called local contents.
- The data event is identified with the data_event_id of DII.

4.2.2 Operation of data events

4.2.2.1 Operation of data events

- The data_event_id is updated when local contents change. That means the data_event_id is different before and after local contents switching. (Does not necessarily increase by 1.) (Figure 4-3)
- Data_event_id is not necessarily updated before and after the stop when the ES stops (the component description disappears from the PMT). Start of the ES implies the beginning new local contents. It is not necessary to memorize the data_event_id before the ES stops. (Figure 4-3)
- The data_event_id is managed and updated individually in each component. The data_event_id is operated by values other than 15(0xF).

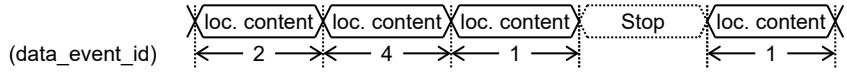


Figure 4-3 Local contents and data_event_id

4.2.2.2 Initiation and termination of local contents

- When the data_event_id of DII is updated, it is considered as a switching of local contents, and the document under presentation is destroyed and a startup document of new local contents is acquired and presented. (Refer to section 4.2.2.3.) The initialization and termination of local contents are ocurred, as a rule, synchronized with the update of the data_event_id of DII in the carousel during reception.

4.2.2.3 Receiver basic behavior during data broadcasting program presentation

- The receiver receiving C-profile will observe constantly DII of all the data components transmitted.
- Receiver processing when the data event of components during reception switches
  The BML-engine sends a "DataEventChanged" message to the document being presented. Afterwards, the document currently being presented is destroyed and the startup document included in the moduleID=0000 module of the component being presented is acquired and presented. When it is an empty carousel after switching the data event, the DII update of the relevant component is monitored after the document being presented isdestroyed, and a startup document is acquired and presented at the moment when it is no longer an empty carousel by switching data events. However, the synchronous/asynchronous event ocurred after "DataEventChanged" is annulled excluding the unload event. Moreover, when the launchDocument()/reloadActiveDocument()/

X_DPA_tuneWithRF()/quitDocument() function is used in the "DataEventChanged" event handler, a function is assigned without the acquisition of a startup document.

4.2.2.4 Operation when DII cannot be obtained for long time

Receivers do not process time-outs for contents presented by PMT and DII reception, etc., when the reception conditions are not stable for certain moments, even in the condition of broadcasting reception or linked condition. However, this is not to eliminate the warning display function, in order for users to choose whether or not to maintain broadcasting reception or tuning. Here, when the contents presentation ends, it is necessary to end all contents that originate in broadcasting. For example, data broadcasting only ends but video and audio broadcasting continues.

The operation of contents might be destroyed because the receiver cannot acquire the event message sent during the period when the stream cannot be acquired. The contents configuration shall be considered, in order not to destroy operation of contents by temporarily breaking the stream during the transmission period of event messages.

4.2.3 Operation of empty carousels

- Data carousels that do not include a DDB are called empty carousels when the numberOfModules field is only DII of 0.
- An empty carousel is used to switch data transmission and termination of the component without changing the description of the ES in the PMT when certain components are necessary at specific periods of an event. Refer to Appendix 8.
- When switching a data carousel that is not empty and an empty carousel, the data_event_id is switched.
- When an empty carousel is transmitted, the minimum transmission interval specification of DII (section 4.2.4) is applied.
- A newer version of DII may be ocurred while transmitting an empty carousel. Moreover, an update of the data event might be ocurred, too.
- When an empty carousel is detected by tuning and the switching of data events, it is not considered an error and the update of DII is monitored, and when a startup document appears the presentation of a startup document starts.
- When the carousel containing the document on reception switches to an empty carousel during multimedia contents presentation, an event (Identify by a "DataEventChanged" event and status=1) that shows "Change into an empty carousel" to the BML document occurs. This operation is the same whether this is an entry component or not.

- When an empty carousel is transmitted by the component that transmits the assigned BML document during the document change specifying the BML document of the transition destination, like launchDocument(), it is processed as an error equal to when the reference module is not included in the usual carousel.

4.2.4 Operation of the DownloadInfoIndication (DII) message

- When presenting multimedia contents, the reception of DII of all carousels is necessary.
- Refer to 4.2.1.1 for the relationship of DII and local contents/data event.
- For restrictions on the performance of the receiver, the minimum transmission interval of DII of each component (excluded the moment of the switching if carousel) is to be 300 milliseconds.
- DSMCC_section transmitting DII is operated as the standard.
- Module information stored in the DII message should store the moduleId's in ascending order (however, moduleId's are not necessarily continuous).
- The operation of userNetworkMessage() is shown in Table 4-4.

Table 4-4 Operation of DII:userNetworkMessage()

Field	Operation	Remarks
dsmccMessageHeader()		
protocolDiscriminator	Operated as specified.	0x11
dsmccType	Operated as specified.	0x03
messageId	Operated as specified.	0x1002
transaction_id	Operated as specified. Incremented by one on Transaction Number (lower 30bits of transaction_id) as a rule for the following cases. When the value is changed, then the receiver unit should judge that the contents of the DII have been changed, this is not limited to when it is incremented by 1 • When the data event switches • When at least 1 carousel configuration module is updated • When the number of module configuring carousels changes (include before and after numberOfModules=0.)	
dsmccAdaptionHeader()	Not operated.	
downloadId	Operated as specified. It is updated in the switch timing of the data event. bit31-28 data_event_id bit27-0 all 1 Operate for identification and switching of the local contents, and response to event message and local contents.	In order to avoid the mis-reception of the event message on local contents adjoined timewise or to switch the data event, the data_event_id is operated.
blockSize	Operated by a fixed value.	4066
windowSize	Operated as specified.	0
ackPeriod	Operated as specified.	0
tcDownloadWindow	Operated as specified.	0
tcDownloadScenario	Operated. The cycle of a longest module of the transmission cycle, among carousel configuration modules, is described.	The time-out time setting based on this value implementation dependantl.
compatibilityDescriptor()	Follows the standard of when operation is done without contents.	compatibilityDescriptorLength=2 descriptorCount=0
numberOfModules	Maximum number of modules transmitted with one data carousel is 64. Moreover, to indicate that it is an empty carousel, numberOfModules=0 is used. Refer to 4.2.3 for empty carousels.	

Field	Operation	Remarks
moduleId, moduleVersion	Operated as specified.. However, it is not guaranteed that +1 moduleVersion increases continuously.	
moduleSize	The maximum value of the module size is 256KB. Refer to 4.2.5 for details.	
Module information area	The descriptor described later is allocated.	
Private data area	Not operated.	
Descriptor stored in the module information area		
Type Descriptor	When mapping 1 resource to 1 module directly, allocation is necessary. It is not necessary for modules to store resources by entity format.	
Name Descriptor	Not operated.	
Info Descriptor	Not operated.	
Module_Link Descriptor	Not operated.	
CRC Descriptor	Not operated.	
Estimated Download Time Descriptor	It can be operated. The maximum transmission cycle of the corresponding module is specified when operating.	Response is optional.
Expire Descriptor	Not operated.	
ActivationTime Descriptor	Not operated.	
CompressionType Descriptor	Not operated.	
Control Descriptor	Not operated.	
RootCertificate Descriptor	Not operated.	

4.2.5 Operation of DownloadDataBlock (DDB)

- DSMCC_section transmitting the DDB message is operated according to standards.
- The maximum size of the module transmitted by the DDB message is 256KB. In detail, the number of DSMCC_section transmitting DDB's is to be 64 as a maximum. (maximum module size is 4066 x 64 = 260224 bytes)
- The module may be compressed by zlib format and transmitted. In this case, the CompressionType Descriptor is allocated in the module information area of DII corresponding to the pertinent module, and the compression type is to be 0. For details of the compression format refer to Appendix 2.
- The sum total module size before and after compression should not exceed the above-mentioned maximum module size when the module is compressed and sent.
- The operation of DDB (downloadDataMessage()) is shown in Table 4-5.

Table 4-5 Operation of DDB: downloadDataMessage()

Field	Operation	Remarks
dsmccDownloadDataHeader()		
protocolDiscriminator	Operated as specified.	0x11
dsmccType	Operated as specified.	0x03
messageId	Operated as specified.	0x1003
downloadId	Operated as specified.	The same value as the downloadId of DII is stored.
adaptationLength	dsmccAdaptationHeader() is not operated in case of multimedia.	0
moduleId	No special specifications for the moduleId value.	
moduleVersion	Operated.	It is not guaranteed that +1 value is given when updated.
blockNumber	Operated as specified..	Value obtained from moduleSize/blockSize.

4.3 Operation of event messages

Only general event messages are operated. Refer to section 4.5 for the optional "auxiliary information function using general event message".

4.3.1 Purpose for operating event messages

- Event messages (hereinafter general event message), including the General Event Descriptor, are operated to transmit data that accompanies events at the same time events are asynchronously caused in multimedia contents.

4.3.2 Transmission of general event messages

- The general event message is transmitted by components with component tag values 0x80, 0x89, 0x8A, and 0x8B. It transmits by components with a component tag value = 0x89 and 0x8A when general event messages only are transmitted. The event_section_flag allocated in the Data Component Descriptor of the PMT is not operated. It is always one.
- The last_section_number of the DSMCC_section transmitting the event message is always 0. That is, the sub-table that transmits one event message is always transmitted by one section.
- The private_data_byte is operated.
- The maximum number of General Event Descriptors that can be allocated in 1 DSMCC_section is 8.
- When setting to acquire the arbitrary message_id by specifyingthe message_id=255 or omitting message_id, it is not set at the same time as the (0~254) assignment that specifies message_id. Moreover, it is assumed that the message version is omitted without fail in this case (Or, 255 is specified).
- When the message_id is specified, the maximum number of general event messages that can subscribe at the same time is 8.
- Multiple sub-tables with the same contents may be transmitted in order to prevent the receiver from failing to receive the general event message. Neither the transmission interval nor the transmission frequency are specified in this case.
- It should be considered that the receiver can fail to receive the general event message, and to secure 200ms or more at update intervals of the DSMCC_section that transmits the general event message in the same ES. This indicates the recommended value at the intervals from sending the first DSMCC_section of an arbitrary version to sending DSMCC_section of the following updated version, and is not the one that provides the sending interval of the DSMCC_section for a different adjacent version during update.
- Only 0x00 (immediate firing) is operated as a time mode to specify the firing time.
- It is assumed that there are cases when the receiver cannot acquire the event message, due to the temporary deterioration of the receiving status. It is necessary to consider that the receiver may not be able to acquire the event message when authoring contents.

4.3.2.1 Operation of general event message data_event_id, event_msg_group_id

The value of the event_msg_group_id of the general event message uses 0 or 1. The receiver behaves as follows.

- Only when the value of the data_event_id of the event message is the same as the data_event_id of local contents during presentation ,the general event message that is the

event_msg_group_id=0 processed as an effective event message. When the value of the data_event_id is different, the event message is ignored.

- General event messages with an event_msg_group_id=1 are processed as effective event messages regardless of the value of the data_event_id.

The above specification applies to both when the general event message is transmitted by the same component as local contents that use the general event message of among four components shown in section 4.3.2, or the general event message is transmitted by other components.

Event_msg_group_id=0 should be specified usually. This is because, for example, there is a possibility of being mis-received, when event_msg_group_id=1 is specified by adjacent local contents and when the event message is transmitted close to same time as the switching of the local contents as in Figure 4-4.

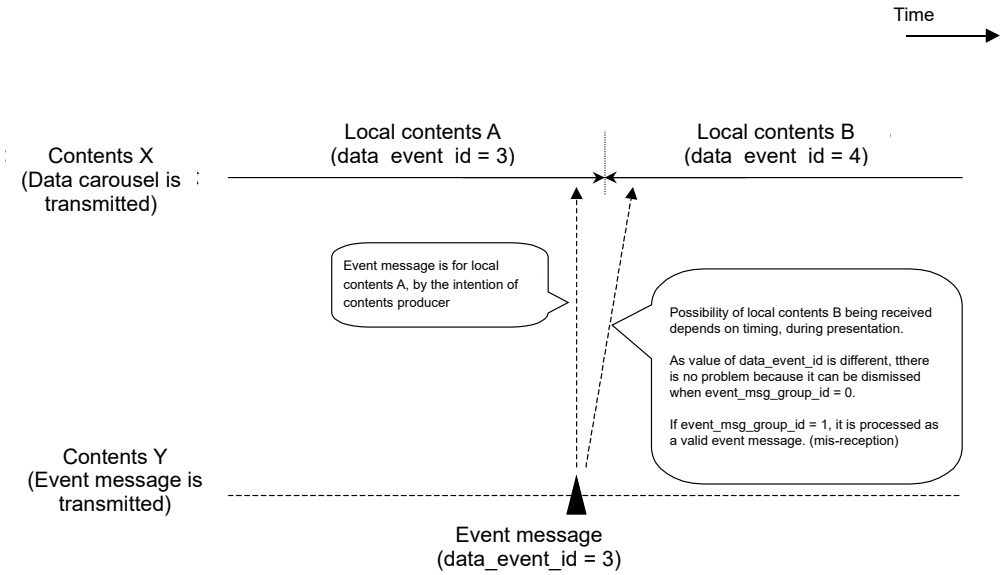


Figure 4-4 Mis-reception of general event messages

For example,The data carousel uses the one sent from the transmission station in network programs etc. and the general event message uses it as if it is transmitted by a reception station, the specification of event_msg_group_id=1 is used when it is difficult to match the value of data_event_id. (Figure 4-5).

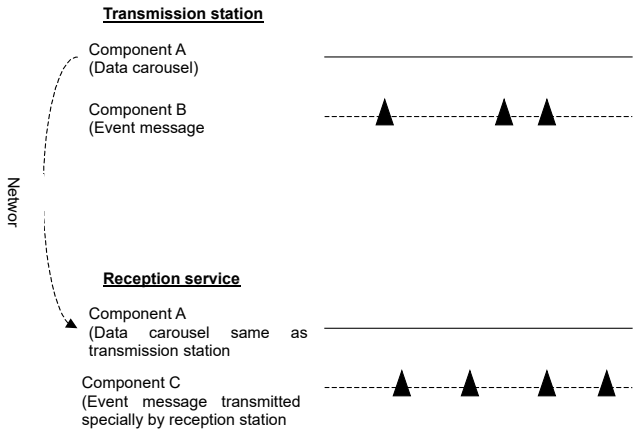


Figure 4-5 When the data_event_id values are not unified

4.3.2.2 Regarding component specifications of general event messages from the BML document

Among the four components shown in section 4.3.2, the component_tag of the general event message that can be observed by 1 BML document, at the same time, is a maximum of 2 types.

4.3.2.3 Event message processing in receivers

(1) General event messages

- When events specified "EventMessageFired" as the type attribute in the BML document, receiver filter the DSMCC_section that transmits the general event message is carried out to comply with either of the following conditions.
  (Condition 1)
  -- component_tag is specified by the es_ref attribute
  -- event_msg_group_id=0x000
  -- data_event_id of local contents during presentation
  (Condition 2)
  -- component_tag is specified by the es_ref attribute
  -- event_msg_group_id=0x001
- For the DSMCC_section to transmit event messages, the components that can filter at the same time is a maximum of two.
- When es_ref is omitted, it is interpreted as the component of local contents currently being presented.

- When the first acquisition of the DSMCC_section or a version-up of the DSMCC_section is detected, an event is carried out according to regulations in the multimedia encoding method based on the message_id/message_version specified by BML document and the the event_msg_id of the General Event Descriptor included in the relevant DSMCC_section
- About the operated time mode
  Only 0x00 (immediate firing) is used as the time mode to specify the firing time.
- The receiver carries out the event in the multimedia contents as quickly as possible after receiving the event message. The target time is 100 milliseconds.
- When multiple general event messages are received, they are fired in the order of reception. The timing of processing is not specified.
- When general event messages that specified immediate multiple firing specification is transmitted by the same DSMCC_section is acquired, the firing is in the order the descriptors located in the DSMCC_section.
- The valid period of the subscribe for the general event message is the period in which the document that specifies subscribe is presented.

4.3.3 Operation of DSMCC_section()

Table 4-6 Operation of DSMCC_section() transmitting event messages

Field	Operation	Remarks
table_id	Operated as specified..	0x3D
section_syntax_indicator	Operated as specified..	1
private_indicator	Operated as specified..	0
dsmcc_section_length	Operated as specified..	
data_event_id	General event message: When event_msg_group_id=0x000, the value is the same as the data_event_id of responding local contents. When event_msg_group_id=0x001, the value is 0x0.	
event_msg_group_id	General event message: 0x000 or 0x001.	
version_number	As a rule, it is incremented by +1 at same subtable identified by the combination of table_id,data_event_id and event_msg_group_id, but continuity is not guaranteed.	
current_next_indicator	Always 1.	

4.3.4 Operation of the General Event Descriptor

Table 4-7 Operation of the General Event Descriptor

Field	Operation	Remarks
event_msg_group_id	Same value as event_msg_group_id of DSMCC_section. (0x000 or 0x001)	
time_mode	Always 0x00 (firing event as soon as reception occurs).	
event_msg_type	Always 1.	
event_msg_id (message_id, message_version)	Operated as specified..	
private_data_byte	Operated as specified..	Max. 244 bites.

4.4 Receiver performance

4.4.1 Filtering resources used during data broadcasting reception

Filtering resources necessary for obtaining a carousel

Required transmission operations

- DII of all carousels is monitored.
- The lock by the lockModuleOnMemoryEx() is enabled for the module of the component that transmits all carousels. (Refer to section 8.1.15.5).
- It is enabled the monitoring of the ModuleUpdated for the module of the component that transmits all carousels.

Table 4-8 Filtering resources required for acquiring the data carousel

Object ES	PID filter number	Section filter number	
		DII	DDB
ES of tag value =0x80	1	1	1
ES of tag value =0x8B	1	1	1

Filtering resources necessary for general event messages

Required transmission operations

The general event message in C-profile is transmitted by components with component tag values of 0x80, 0x89, 0x8A, and 0x8B. When only the general event message is transmitted, transmit them by components with component tag values of 0x89 and 0x8A.

Moreover, two section filters are necessary, respectively, because the value of the event_msg_group_id can be sent by 0 or by 1.

Table 4-9 Filtering resources required for acquiring the general event message

Object ES	PID filter number	Section filter number
ES of general event message tag value =0x89	1	2
ES of general event message tag value=0x8A	1	2

4.4.2 Desirable operation of the receiver in relation to the acquisition of C-profile modules

Improve presentation response by pre cache, obtaining unconditionally modules transmitted in C-profile receivers. In C-profile receivers., the presentation response can be improved by unconditional acquisition, and the pre cache of the module transmitted on a maximum extent. Moreover, the response of the presentation can be improved by acquisition of the multiple ES modules at one time, and the pre cache even if multiple ES transmits the contents.

4.4.3 Receiver operation concerning updating the version

Refer to Appendix 9 for the receiver behavior receiving the major_version excluding 12.

4.5 Auxiliary information function using general event message (Optional)

The following shows the auxiliary information function using general event message. This function handles the information in private_data_byte as "auxiliary information using general event message", using a specific character string in event_msg_id and private_data_byte (hereafter referred to as "specified character string"). Implementation of this function shall be optional.

The image of the operation using the relevant auxiliary information shall be used for purposes such as encouraging the viewer to view data broadcasting when it is not being displayed and providing auxiliary information to draw viewer's attention to a certain program.

4.5.1 Operation of event_msg_id and general event message transmission

event_msg_id (message_id and message_ version) shown below shall be used exclusively for the "auxiliary information function using general event message".

Table 4-10 Specification of ID, etc. to be transmitted

Target ES	message_id	message_version	event_msg_group_id
0x80,0x8B,0x89,0x8A	200	0～255	1

- In the case when a general event message in which event_msg_id (message_id and message_version) shown in Table 4-10 has been specified has been transmitted by multiple ES, only the latest auxiliary information shall be presented (Refer to section 4.5.5 for the operation of the receiver.).
- General event messages shall be transmitted using a component with component tag values of 0x80, 0x89, 0x8A, and 0x8B. When transmitting a general event message only, it shall be transmitted using a component with component tag values of 0x89 and 0x8A. event_section_flag, which will be allocated to the Data Component Descriptor of PMT, shall not be operated. It shall always be set to 1.
- 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.
- “private_data_byte” is operated.
- The maximum number of General Event Descriptors that can be located in 1 DSMCC_section is 8.
- 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 200ms. 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.
- Only 0x00 (immediate firing) shall be operated as the time mode for specifying the time of firing.
- It is conceivable that the receiver may fail to acquire the event message due to temporary disruption of the reception status. During content development, it shall be necessary to sufficiently consider the possibility of the receiver failing to acquire the event message.
- 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.
- Refer to section 4.3.3 for the operation of DSMCC_section. However, event_msg_group_id shall only be operated with 1
- Refer to section 4.3.4 for the operation of the General Event Descriptor.
- event_msg_id in Table 4-10 shall not be used as a normal general event message.

4.5.2 Transmission of the specified character string in private_data_byte

Table 4-11 below shows the specified formats for private_data_byte.

Table 4-11 Specified formats

Byte position	No. of bytes	Field name	Specified character string	Other
0 -9	10	HEADER	”DPA-EMSUBI”: Indicates that it is an "auxiliary information function using event message".	
11-15	5	SC1 (specified code 1)	”DMARK”: Displays a mark.	Duplicate specification of the same specified character string shall be prohibited. However, this excludes "NONSC".
			”STEXT” : Displays the character string of the STEXT item.	
			”FUNC1” : (Reserved for future expansion)	
			”FUNC2” : (Reserved for future expansion)	
			”FUNC3” : (Reserved for future expansion)	
			”NONSC” : Indicates "No specification".	
			”CLEAR” : Clears the display.(Only SC1 can be specified)	
17-18	2	SC1T	Describes the length of time (period) in second, for which SC1 will be presented. It shall be described in single-byte alphanumeric characters. The range of values that can be specified shall be "00" and "05" - "99". Refer to Table 4-13 for the operation of the receiver.	"00" shall only be specified when "NONSC" and "CLEAR" are specified in SC1-SC3.
20-24	5	SC2	Same as for SC1.	Same as for SC1.
26-27	2	SC2T	Describes the length of time (period) in second, for which SC2 will be presented.Others shall be the same as for SC1T.	Same as for SC1T.
29-33	5	SC3	Same as for SC1.	Same as for SC1.
35-36	2	SC3T	Describes the length of time (period) in second, for which SC3 will be presented.Others shall be the same as for SC1T.	Same as for SC1T.
38	1	LOCATION1	Display position: Indicates top or bottom ”T” : Align to top of the screen ”B” : Align to bottom of the screen	It shall always be specified.
40	1	LOCATION2	Display position: Indicates left or right ”L” : Align to left of the screen ”R” : Align to right of the screen	It shall always be specified.
42-	Max. 60	STEXT	A character string of up to 60 bytes (30 double-width characters) shall be specified. If "STEXT" has been specified in SC1-SC3, the character string in this field must always be specified. The character code shall follow the Shift-JIS standard defined in 5.4.2 in [Section 4] of this volume. However, the use of a delimiter ("|" (0x7C)) shall not be permitted.	Empty string shall be specified if there is no specification of "STEXT" in SC1-SC3.
	3	TRAILER	"END" must always be specified at the end.	

*1 "|" (0x7C) shall be used for delimiting fields.

Example: DPA-EMSUBI|DMARK|05|STEXT|15|NONSC|00|T|R|In data broadcasting, ...|END
Example: DPA-EMSUBI|DMARK|10|NONSC|00|NONSC|00|T|R||END (Mark only)
Example: DPA-EMSUBI|CLEAR|00|NONSC|00|NONSC|00|T|R||END (Clear)

*2 Combinations of SC1-SC3 that will be operable for the time being are as follows.

Table 4-12 Operable combinations

	SC1	SC2	SC3
Only DMARK is specified	DMARK	NONSC	NONSC
DMARK and STEXT are specified	DMARK	STEXT	NONSC
Clearing of information is specified	CLEAR	NONSC	NONSC

*3 "NONSC" shall always be specified if there is no specification in SC1-SC3.
If "CLEAR" has been specified in SC1, "NONSC" shall always be specified in SC2 and SC3.

* 4 If "NONSC" and "CLEAR" have been specified in SC1-SC3, "00" shall be specified in SC1T-SC3T.

Refer to section 4.5.5 for the operation of the receiver.

4.5.3 Operation of content

event_msg_id of the target ES as shown in Table 4-10 shall be used exclusively for the "auxiliary information function using general event message". The content, time and position of the presentation shall be specified to private_data_byte according to the format specified in Table 4-11.

Moreover, the relevant event_msg_id will not be used as a normal general event message.

4.5.4 Conditions for incorporation on the receiver

A receiver incorporating this function shall observe the following incorporation conditions.

- The receiver must incorporate a data broadcasting browser. This function shall not be implemented if the receiver does not incorporate a data broadcasting browser.
- If PID filtering resources for general event messages will be implemented for this function, the number of PID filtering resources shall be 4 excluding the 2 resources shown in Table 4-9.

4.5.5 Conditions for presentation on the receiver

A receiver incorporating this function shall observe the following presentation conditions.

O In case of data broadcasting not being presented

- A receiver not presenting data broadcasting shall present information content using this function. It shall be presented in the video area.

- Refer to section 4.5.5.2 for the presentation relationship with subtitles and system messages of the receiver.

○ In case of data broadcasting being presented

- If the receiver presents data broadcasting, the presentation of information content using this function shall be a matter of product planning. If information content will be presented using this function, the above conditions for when the receiver is not presenting data broadcasting shall be observed.

4.5.5.1 Presentation time

The presentation time shall be specified using SC1T-SC3T. The presentation shall be terminated when the specified amount of time has elapsed. If the receiver does not support the SC1T-SC3T specifications ("05" - "99" seconds), it must terminate the presentation in less than 15 seconds.

4.5.5.2 Presentation

- The position of the presentation will be specified by the specified character string.
  Refer to Figure 4-6 in section 4.5.5 for the presentation positions of the mark and the character string.
  If the receiver does not support specification of the presentation positions of the mark and the character string, the information shall always be presented at the "top right" of the screen.
- The method of presentation shall be a matter of product planning. While it is recommended that the entire character string be presented by scrolling, it shall also be permissible to split the character string into multiple parts and present them separately.
- All character strings specified in "STEXT" shall be presented. No character string shall be displayed if it is not possible.
- The priority order of the presentation by this function shall be the lowest compared to the presentation of subtitles, etc. (including system messages of the receiver).

  For example, the presentation by this function shall not be performed if viewability will be lowered by the overlapping of the subtitles, etc. with the presentation by this function. (It shall also be possible to adopt a means for improving the visibility of the subtitle display, etc., such as showing a frame around the subtitle.)
- A function for turning this function on and off shall not be incorporated.

4.5.5.3 Presentation update

The presentation shall be updated if message_version has been changed. The update timing shall be dependent on the receiver.

In the event when the same message_version is specified in the same ES after displaying the "auxiliary information using general event message", the information shall not be displayed again.

In the case when a general event message in which event_msg_id shown in Table 4-10 has been specified has been transmitted by multiple ES, only the latest auxiliary information shall be presented regardless of which ES the message was transmitted by (i.e. the latest write has priority). Auxiliary information specified by 2 or more ES must not be displayed simultaneously.

4.5.5.4 Deletion timing

If "CLEAR" has been specified in SC1, all presentations of "auxiliary information using general event message" shall be deleted.

The presentation shall be removed when the time period specified in specified character strings SC1T-SC3T have elapsed. If the receiver does not support the SC1T-SC3T specifications ("05" - "99" seconds), it must terminate the presentation in less than 15 seconds.

4.5.5.5 Character color, background color, character size and character type

Character color, background color, character size and character type shall be a matter of product planning. If the number of characters exceeds the limit specified in Table 4-13, the receiver shall ignore the entire general event message as invalid format.

4.5.5.6 Receiver operation in response to each specified character string

The following shows the receiver operation in response to each specified character string.

Table 4-13 Receiver operation in response to each specified character string

Field	Specified character string	Receiver operation	Support by the receiver incorporating this function
HEADER	”DPA-EMSUBI”	The receiver shall operate as "auxiliary information function using general event message".	Mandatory
SC1～SC3	”DMARK”	The receiver shall display a mark representing data broadcasting. Refer to *6 for the position of the presentation. Contact the governing organization described in Appendix 13 regarding the recommended mark.	Mandatory
	”STEXT”	The receiver shall display the character string.	Optional
	”FUNC1”～”FUNC3”	These shall be reserved for future expansion. The receiver shall ignore the specified strings and not generate any errors.	Optional
	”NONSC”	The receiver shall not display any character string.	Mandatory
	”CLEAR”	The receiver shall clear the information that was specified in SC1-SC3 and displayed on the receiver.	Mandatory
SC1T～SC3T	”05”～”99”	The receiver shall terminate the presentation when the specified amount of time has elapsed. If the receiver does not support the SC1T-SC3T specifications ("05" - "99" seconds), it must terminate the presentation in less than 15 seconds.	Mandatory
	”00”	If "00" is specified in SC1T-SC3T, "NONSC" and "CLEAR" will be specified in SC1-SC3.	Mandatory
LOCATION1 LOCATION2	”T”、”B”、”L”、”R”	The receiver shall present the information at the top left, top right, bottom left or bottom right of the screen as specified. If the receiver does not support specification of the presentation position, the information shall always be presented at the "top right" of the screen.	Mandatory
TRAILER	”END”	The "END" character string. The receiver shall interpret this as the end of the instruction.	Mandatory

*1 Fields shall be delimited using "|" (0x7C).

Example: DPA-EMSUBI|DMARK|05|STEXT|15|NONSC|00|T|R|In data broadcasting, ...|END

Example: DPA-EMSUBI|DMARK|10|NONSC|00|NONSC|00|T|R||END (Mark only)

Example: DPA-EMSUBI|CLEAR|00|NONSC|00|NONSC|00|T|R||END (Clear)

*2 The receiver shall operate as follows when an unsupported or undefined character string has been specified in the SC1-SC3 fields.

- The receiver shall ignore the unsupported/undefined SC1-SC3 fields and their corresponding SC1T-SC3T (and STEXT) fields, without generating any error.

  Example : DPA-EMSUBI|DMARK|05|STEXT|15|NONSC|00|T|R|In data broadcasting, ...|END

  - If the relevant receiver does not support STEXT specified in SC2, it shall ignore SC2, SC2T and STEXT fields and, as a result, only present DMARK that has been specified in SC1.

  Example: DPA-EMSUBI|DMARK|15|SOUND|10|NONSC|00|T|R||END

  - Since SOUND is not defined, the receiver shall ignore SC2 and SC2T fields and, as a result, only present DMARK that has been specified in SC1.

*3 Combinations of SC1-SC3 that will be operated for the time being are as shown in Table 4-12.

*4 As a rule, the receiver shall operate as described in *2 if any combination other than those shown in Table 4-12 has been specified. However, the receiver may also be implemented such that it will ignore the relevant general event message.

*5 If multiple specifications have been made using SC1-SC3 and the receiver supports multiple specifications, they shall be displayed by the receiver simultaneously. If simultaneous display is not possible, the receiver shall display one of the specifications based on the following order of priority. The order of presentation priority shall be as follows (from left to right): SC1 > SC2 > SC3.

*6 The presentation shall be positioned according to LOCATION1 and LOCATION2. For example, the presentation will be aligned to top and left edges of the screen if "T" and "L" are specified. The following shows the presentation image examples based on the specification of marks and character strings.

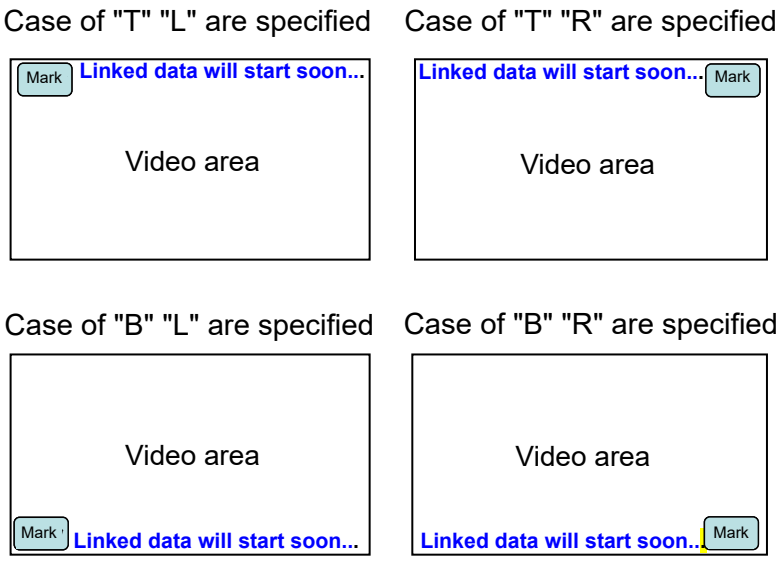


Figure 4-6 Presentation image based on the specification of presentation positions

*7 The receiver shall ignore a general event message if it does not conform to the specified format, such as when (1) the position and/or character code of the delimiter does not conform to the guidelines, (2) a character string other than the specified character strings has been specified in each field (excluding SC1-SC3 fields), or, (3) the size of each field differs from the prescribed number of bytes.

5 Operation of mono-media coding

5.1 Image coding

5.1.1 H.264|MPEG-4 AVC

- Follows the specifications of ITU-T Rec. H.264/ISO/IEC 14496-10:2003.

However, for MPEG2 Systems, please refer to ITU-T Rec. H.222.0|ISO/IEC 13818-1:2000/AMD3(FDAM3). Moreover, the AVC Video Descriptor and AVC timing and HRD Descriptor are not operated.

5.1.1.1 Coding parameters

- For coding parameter restrictions, follow ARIB STD B24 Vol. 1, Part 2, 4.4
- Profile-level is Baseline-level 1.2.

5.1.1.2 Picture format

- Picture format is QVGA (4:3) (screen size : 320x240), QVGA (16:9) (screen size : 320x180).

However, restrictions on PES packets are as below.

- PTS_DTS_flags of the PES packet header is always ‘10’.
- IDR AU is always the first AU of the PES packet.
- PES packet is configured by n pieces of AU (n is an integer 1 or larger).
- The PTS difference of two consecutive PES packets is within 0.7 seconds.

5.1.1.3 Restrictions in the bit stream

(1) Transmission cycle of the IDR access unit

- IDR access units (abbreviated below as AU) are inserted into the bit stream at intervals of every two seconds, normally, to shorten the required time of reproduction. However, they should be inserted in intervals not exceeding five seconds even when extending the IDR-AU interval in order to maintain image quality.
- Each IDR-AU is an elementary stream access point described in ISO/IEC 13818-1:2000 FDAM3.

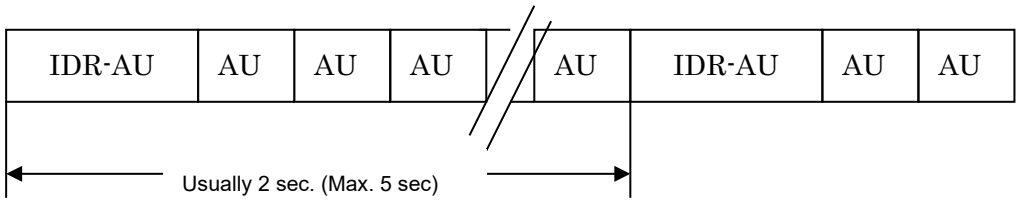


Figure 5-5-1 IDR-AU intervals

(2) AU configuration

- The number and order of NAL units to configure the IDR AU and non-IDR AU are as follows. However, NAL units other than the following are not operated.

Table 5-1 AU configuration

Type and order of NAL Unit	Quantity	
	IDR AU	non-IDR AU
access unit delimiter	1	1
sequence parameter set (SPS)	1	0
picture parameter set (PPS)	1	0 or 1
supplemental enhancement information (SEI)	0 or 1 (Note 1)	0 or 1(Note 1)
coded slice of an IDR picture	1 and over	0
coded slice of a non-IDR picture	0	1 and over
filler data	0 or 1	0 or 1
end of sequence	0 or 1	0 or 1

(Note1)Insertion condition of SEI NAL unit is Table 5-10.

(3) Operation restrictions on syntax

Restrictions of the flag in the syntax are shown in the tables below. However, the ID of the SPS (Sequence Parameter Set) and PPS (Picture Parameter Set) can be operated by a fixed value regardless of changes in the contents described in the parameters.

Table 5-2 Access unit delimiter restrictions

Flag denomination	Operation	Remarks
primary_pic_type	0 or 1	IDR picture is 0, non IDR picture is 1

Table 5-3 SPS (Sequence Parameter Set) restrictions

Flag denomination	Operation	Remarks
profile_idc	66	Baseline profile
constraint_set0_flag	1	FMO, ASO, RS are not used.
constraint_set1_flag	1	
constraint_set2_flag	1	
level_idc	12	Level 1.2
seq_parameter_set_id	0 - 31	Configure during operation
log2_max_frame_num_minus4	0 - 12	Value not circulating among IDR is configured during operation
pic_order_cnt_type	2	POC type is 2 only

Flag denomination	Operation	Remarks
num_ref_frames	1 or 2 or 3	Reference frame number is a maximum of 3
gaps_in_frame_num_value_allowed_flag	0	Missing frames compensation process is not carried out
pic_width_in_mbs_minus1	19	320 / 16 – 1 =19
pic_height_in_map_units_minus1	14 or 11	When 320x180, it is 11 When 320x240, it is 14
frame_mbs_only_flag	1	Only frame MB
direct_8x8_inference_flag	1	No meaning when Baseline
frame_cropping_flag	0 or 1	When 320 x 180, it is 1 When 320 x 240, it is 0
frame_crop_left_offset	0	When it is 320 x 180
frame_crop_right_offset	0	When it is 320 x 180
frame_crop_top_offset	0	When it is 320 x 180
frame_crop_bottom_offset	6	When 320 x 180, it is a 12 line portion crop
vui_parameters_present_flag	1	VUI is mandatory

Table 5-4 VUI parameters restrictions

Flag denomination	Operation	Remarks
aspect_ratio_info_present_flag	0	Aspect ratio 1:1 is specified on operation (default value is unspecified)
overscan_info_present_flag	0	No operation
video_signal_type_present_flag	0	Under specifications of STD-B24, value of color_primaries, transfer_characteristics, matrix_coefficients are interpreted as 1 (Rec.ITU-R BT 709) at the decoder side
chroma_loc_info_present_flag	0	Uses default values
timing_info_present_flag	1	
num_units_in_tick	1001 * N	N≥1, Max. 15fps (Note 1)
time_scale	24000 or 30000	Max. 15fps (Note 1)
fixed_frame_rate_flag	0 or 1	
nal_hrd_parameters_present_flag	0 or 1	
vcl_hrd_parameters_present_flag	0 or 1	
low_delay_hrd_flag	0	Low delay mode is prohibited
pic_struct_present_flag	0	
bitstream_restriction_flag	0 or 1	(Note 2)
motion_vectors_over_pic_boundaries_flag	0 or 1	
max_bytes_per_pic_denom	0 - 16	
max_bits_per_mb_denom	0 - 16	
log2_max_mv_length_horizontal	0 - 9	Horizontal direction ±128
log2_max_mv_length_vertical	0 - 9	
num_reorder_frames	0	Prohibition of reordering
max_dec_frame_buffering	1 - 3	Reference frame number is a maximum of 3

(Note 1) Specifications on the picture display interval are mentioned below

- When the fixed_frame_rate_flag is 0, the interval of decoding and displaying the neighbouring picture shoud be any value more than 1001/15000, and the multiples of num_units_in_tick / time_scale for num_units_in_tick, time_scale specified by VUI.

  (Operation example 1)

  When time_scale=30000, num_units_in_tick=1001, num_units_in_tick/time_scale=1001/30000=1/29.97, in other words, the unit value of cpb_removal_delay becomes 1/29.97 seconds. Moreover, since there are restrictions on 1001/15000 or more, the difference of the cpb_removal_delay (interval of adjacent pictures) is two or more.

  (Operation example 2)

  When time_scale=24000,num_units_in_tick=1001, the unit of cpb_removal_delay becomes 1 / 23.9 (sec.). Also, since there are restrictions on 1001/15000 or more, the difference of cpb_removal_delay becomes 2 or more.

- When the fixed_frame_rate_flag is 1, the decoding and display interval of the adjacent picture is 1001/15000 or more and 2 * num_units_in_tick / time_scale for num_units_in_tick, time_scale is specified by VUI.

  (Operation example 1)

  When time_scale=30000,num_units_in_tick=1001, 2 * num_units_in_tick / time_scale = 2 * 1001 / 30000 = 2 / 29.97, in other words the decoding and display interval of the adjacent picture is 2/29.97 (sec.).

  (Operation example 2)

  When time_scale=24000,num_units_in_tick=1001, the interval of the adjacent picture is 2/23.9 (sec.).

(Note 2)The search area of the movement vector is limited in the horizontal direction to ±128. Moreover, num_reorder_frames=0 is to prohibit reordering.

Table 5-5 HRD parameters restrictions

Flag denomination	Operation	Remarks
cpb_cnt_minus1	0	CPB type is 1
bit_rate_scale	0 - 15	Configure during operation
cpb_size_scale	0 - 15	Configure during operation
bit_rate_value_minus1	(bit_rate_value_minus1 + 1) * $2^{(6 + bit\_rate\_scale)}$ <= 384000 or 460800	Configure based on the bit rate used for real operation
cpb_size_value_minus1	(cpb_size_value_minus1 + 1) * $2^{(4 + cpb\_size\_scale)}$ <= 1000000 or 1200000	Configure based on the CPB size used for real operation
cbr_flag	0 or 1	
initial_cpb_removal_delay_length_minus1	0 - 31	Configure during operation
cpb_removal_delay_length_minus1	0 - 31	Configure during operation
dpb_output_delay_length_minus1	0 - 31	Configure during operation
time_offset_length	0	

Table 5-6 PPS (Picture Parameter Set) restrictions

Flag denomination	Operation	Remarks
pic_parameter_set_id	0 - 255	Configure during operation
seq_parameter_set_id	0 - 31	Assign reference id of SPS
entropy_coding_mode_flag	0	Only CAVLC
pic_order_present_flag	0	Only Type 2
num_slice_groups_minus1	0	FMO prohibited
num_ref_idx_l0_active_minus1	0 or 1 or 2	Reference frame is 1 to 3 immediately before.
num_ref_idx_l1_active_minus1	0	No B frame
weighted_pred_flag	0	WP prohibited
weighted_bipred_idc	0	No B frame
pic_init_qp_minus26 /* relative to 26 */	-26 - 25	Configure during operation
pic_init_qs_minus26 /* relative to 26 */	0	Not used
chroma_qp_index_offset	-12 - 12	Configure during operation
deblocking_filter_control_present_flag	0 or 1	Configure during operation, no limitations
constrained_intra_pred_flag	0	Do not restrict on intra forecast
redundant_pic_cnt_present_flag	0	RS prohibited

- Restrictions in the SEI (Supplemental Enhancement Information) are shown in the table below. Buffering period, Picture timing, Pan-scan, and Filler payload SEI message can only be inserted in the SEI. (Note 1)

Table 5-7 Buffering period SEI message (Note 2)

Flag denomination	Operation	Remarks
seq_parameter_set_id	0 - 31	Assign reference id of SPS
initial_cpb_removal_delay	initial_cpb_removal_delay + initial_cpb_removal_delay_offset ≤ 135000(recommendation value)	Configure during operation when NalHrdBpPresentFlag and VclHrdBpPresentFlag are ”1” each
initial_cpb_removal_delay_offset		Configure during operation when NalHrdBpPresentFlag and VclHrdBpPresentFlag are "1" each

Table 5-8 Picture timing SEI message (Note 3)

Flag denomination	Operation	Remarks
cpb_removal_delay	0 - 150	Configure during operation when CpbDpbDelaysPresentFlag is "1"
dpb_output_delay	0	Configure during operation when CpbDpbDelaysPresentFlag is "1"

Table 5-9 Pan-scan rectangle SEI message (Note 4)

Flag denomination	Operation	Remarks
pan_scan_rect_id	0	
pan_scan_rect_cancel_flag	0 or 1	
pan_scan_cnt_minus1	0	
pan_scan_rect_left_offset	0 or 640	Refer to section 5.1.1.6
pan_scan_rect_right_offset	0 or -640	Refer to section 5.1.1.6
pan_scan_rect_top_offset	480 or 0	Refer to section 5.1.1.6
pan_scan_rect_bottom_offset	-480 or 0	Refer to section 5.1.1.6
pan_scan_rect_repetition_period	1	

(Note 1)The existence and the order of each AU of SEI message are as follows.

- IDR-AU

  Buffering period SEI message, Picture timing SEI message, Pan-scan rectangle SEI message, and Filler payload SEI message can be inserted in the relevant AU, and inserted if necessary. The insertion order is as follows.

  1 Buffering period SEI message

  2 Picture timing SEI message

3 Pan-scan rectangle SEI message

4 Filler payload SEI message

- non IDR-AU

  Only the Picture timing SEI message, Filler payload SEI message can be inserted in the relevant AU, and inserted if necessary. The insertion order is as follows.

  1 Picture timing SEI message

  2 Filler payload SEI message

In addition, in either of IDR-AU and non IDR-AU, existence or not of the Filler payload SEI is not specified.

(Note 2) (Note 3) Restrictions regarding the insertion of the Buffering period SEI message and Picture timing SEI message are as below.

Table 5-10 Restrictions on the Buffering period SEI message and Picture timing SEI message insertion

AU configuration in 1PES	fixed_frame_ rate_flag	Type of AU	Buffering period SEI message	Picture timing SEI message
1PES=1AU fixed operation	0	IDR	Δ	Δ
		Non IDR	×	Δ
	1	IDR	×	×
		Non IDR	×	×
1PES=1AU other than fixed operation	0	IDR	O	O
		Non IDR	×	O
	1	IDR	×	×
		Non IDR	×	×

O: Input is mandatory Δ: Input is optional ×: Input prohibited

- When the Buffering period SEI message and Picture timing SEI message are inserted, at least nal_hrd_parameters_present_flag or vcl_parameters_present_flag is "1".
- When neither the Buffering period SEI message nor the Picture timing SEI message are inserted, both the nal_hrd_parameters_present_flag and vcl_parameters_present_flag are "0".
- When fixed_frame_rate_flag = 1, the framerate can be obtained in time_scale / num_units_in_tick / 2. (Example: when time_scale = 30000, num_units_in_tick = 1001, the framerate is 29.97 / 2 (frames/sec)

(Note 4) Restriction on inserting the Pan-scan rectangle SEI message

- When operating PanScan, insert it in IDR-AU, without fail.

Table 5-11 Slice header restrictions

Flag denomination	Operation	Remarks
first_mb_in_slice	0 - 299 or 0 - 239	Configure during operation
slice_type	0, 5 or 2, 7	7 in IDR slice, and 0, 2, 5, 7 in non IDR slice
pic_parameter_set_id	0 - 255	Assign reference id of PPS
frame_num	0 ~ MaxFrameNum	Do not circulate between IDR
idr_pic_id	0 - 65535	Configure during operation
num_ref_idx_active_override_flag	0 or 1	Configure during operation
num_ref_idx_l0_active_minus1	0 or 1 or 2	Reference frame is 1 to 3 immediately before
slice_qp_delta	-51 - 51	Configure during operation
disable_deblocking_filter_idc	0 or 1	Modes that do not filter on the slice boundary are prohibited
slice_alpha_c0_offset_div2	-6 - 6	Configure during operation
slice_beta_offset_div2	-6 - 6	Configure during operation

Table 5-12 Reference picture list reordering restrictions

Flag denomination	Operation	Remarks
ref_pic_list_reordering_flag_l0	0	Reordering of reference frame is prohibited

Table 5-13 Decoded reference picture marking restrictions

Flag denomination	Operation	Remarks
no_output_of_prior_pics_flag	0	Do not clear DPB when IDR is input (Due to PTS=DTS, no meaning)
long_term_reference_flag	0	Do not use longtime memory
adaptive_ref_pic_marking_mode_flag	0	

5.1.1.4 Other restrictions

- The stream input in the CPB is recommended to be set to be decoded within 1.5 seconds.

5.1.1.5 Identification of 16:9 pictures

- Broadcasters may transmit images of 16:9 though the image format is only QVGA. They are 16:9 pictures when pic_height_in_map_units_minus1 of the H.264 stream SPS is 11.
  - As a rule, the pic_height_in_map_units_minus1 is not modified for a program, they should be operated at semi fixed status by broadcasters, however, during the period of simultaneous broadcasting with analog spectrum, the aspect program of 4:3

remains, therefore there is a possibility that it can be changed for a program. It can be changed in each CM unit.

5.1.1.6 Operation of PanScan in the picture display area

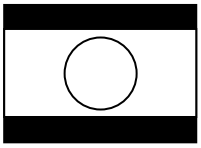
It is possible to not display black borders (picture frame) depending on the picture angle of the receiver unit by configuring the following PanScan parameters when delivering by a different aspect ratio for actual picture source, like side panels and letterboxes.

(1) When displaying a part of an original picture source (320×180) in a 16:9 full screen picture display area, and when the delivery picture format is QVGA 4:3(320×240) and the original image source is 16:9 (letterbox).

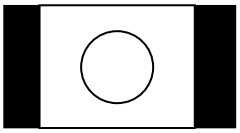
(2) When displaying a part of original picture source (240×180) in 4:3 full screen picture display area, and when the delivery picture format is QVGA 16:9 (320×180) and the original picture source is 4:3 (side panel).

It is necessary to send the center of the delivery image format and the center of the image source corresponding in both horizontal and vertical directions.

The value of each parameter is indicated as follows, when the above-mentioned operation is carried out.



Reference figure (1)



Reference figure (2)

Table 5-14 Restrictions of each parameter during PanScan operation

Definition part	Syntax	Reference figure (1)	Reference figure (2)
Sequence Parameter Set	pic_width_in_mbs_minus1	19	19
	pic_height_in_map_units_minus1	14	11
Pan-scan rectangle SEI message	pan_scan_rect_left_offset	0	640
	pan_scan_rect_right_offset	0	-640
	pan_scan_rect_top_offset	480	0
	pan_scan_rect_bottom_offset	-480	0

PanScan can be turned on-off by the unit of coded video sequence, and when the above PanScan operation is not carried out, the encoding of the Pan-scan rectangle SEI message shall not be done. (When a PanScan operation is done, the Pan-scan rectangle SEI is always included in IDR-AU.)

(*) The receiver unit response to the PanScan is optional.

5.2 Still picture and bitmap figure coding

5.2.1 JPEG

- In conformity with the BaseLine method of ISO/IEC 10918-1(ITU-T T.81).
- For the colorimetryof JPEG, ARIB STD B24 Vol. 1, Part 1, 7.2 "colorimetry" is applied.

5.2.1.1 Coding parameters

- Order of coding: interleaved system
- Loss permitted base line system
- Sampling factor: It is YCBCR = 4:2:0. However, it shall not fail when receiving the 4:2:2 formats. JPEG of the 4:2:0 format indicates the value of SOF0 marker information (H1,V1), (H2,V2), (H3,V3) is each (2,2), (1,1), (1,1), and at the 4:2:2 format indicates, similarly, the value of (H1,V1), (H2,V2), (H3,V3) corresponding to (2,1), (1,1), (1,1) respectively.

5.2.1.2 Other restrictions

- Do not operate progressive mode.

5.2.1.3 Operation markers and marker segments

Markers and marker segments operated by JPEG are shown in Table 5-15.

Table 5-15 Markers and marker segments operated by JPEG

Marker	Observation	Receiver process
SOI	Initiation of image	General process
DQT	Definition of quantization table	General process
DRI	Definition of re-start interval	General process(Note 1)
SOFn	Initiation of frame Only SOF0 (FFC0) is the subject of encoding.	General process
DHT	Definition of Huffman	General process
SOS	Initiation of scan	General process
RSTm	Termination of re-start interval	General process(Note 1)
EOI	Termination of image	General process
COM	Comment	General process
APPn	Used for application	Ignore
DNL	Assignation of special size	Ignore

(Note 1) Response to the error of DRI and RSTm is implementation dependent.

In order for the above SOI – DNL only to appear in the base line system, markers other than the above are treated as errors.

5.2.2 GIF

The graphics file format in GIF uses specifications based on “GRAPHICS INTERCHANGE FORMAT Version 89a" specified by Compuserve Inc. America.

5.2.2.1 Usable blocks

In this chapter, the blocks that can be used are specified, and when the value of each field is limited, the limitations are shown together.

"O" in the following table shows operations as specified.

5.2.2.2 Header

1 will always exist at the start.

Field	Operation	Remarks
Signature	O	Character string and fixed value of “GIF”
Version	89a	File format is Version89a

5.2.2.3 Trailer

1 will always exist at the end.

Field	Operation	Remarks
GIF Trailer	O	Fixed value 0x3B

5.2.2.4 Logical Screen Descriptor

1 will always exist after the Header block.

Field	Operation	Remarks
Logical Screen Width	O	Width of logical screen
Logical Screen Height	O	Height of logical screen
Global Color Table Flag	O	When this bit stands, the Global Color Table block exists right after this block.
Color Pixel size	O	
Sort Flag	O	
Size of Global Color Table	O	Global Color Table size
Background Color Index	O	
Pixel Aspect Ratio	O	

5.2.2.5 Global Color Table

This block is operated according to standards, and the existence is controlled by the Global Color Table Flag of the Logical Screen Descriptor. When it exists, only one exists after the block of Logical Screen Descriptor. Moreover, the Size of Global Color Table of the same block specifies the size.

5.2.2.6 Image Descriptor

Multiple descriptors are possible. Image data follows after this block.

Field	Operation	Remarks
Image Separator	O	Fixed value 0x2C
Image Left Position	O	
Image Top Position	O	
Image Width	0-240	Width of image
Image Height	0-320	Height of image
Local Color Table Flag	O	When this bit stands, the Global Color Table block exists right after this block.
Interlace Flag	O	1 when display interlace, and 0 if not
Sort Flag	O	
Size of Local Color Table	O	Local Color Table size

5.2.2.7 Local Color Table

This block is operated according to standards, and the existence is controlled by the Local Color Table Flag of the Image Descriptor. When it exists, only one exists after the Image Descriptor block. Moreover, the Size of Local Color Table of the same block specifies the size.

5.2.2.8 Image Data

Multiple existences are possible. If there is Image Descriptors or Local Color Tables, this block always exists after it.

Syntax is operated according to standards, Multiple sub-blocks to store the coloration scheme image data compressed in LZW format exist after the 1byte LZW Minimum Code Size field which indicate the size of LZW minimum code used for recovering data.

5.2.2.9 Graphic Control Extension

The operation of this block is arbitrary, but it shall be operated whenever a transparent color and delay time are specified. In that case, it exists only one in front of the Image Descriptor.

Field	Operation	Remarks
Extension Introducer	O	Fixed value 0x21
Graphic Control Extension Label	O	Fixed value 0xF9
Block Size	O	Fixed value 0x04
Disposal Method	O	
User Input Flag	0	0: Do not accept user input at the moment of data display
Transparent Color Flag	O	1 for transparent processing, and 0 when this process is not done. (animation GIF is fixed at 0)
Delay Time	20-500	200 - 5000ms, specified in100ms units
Transparent Color Index	O	Color of transparent color is Global Color Table, or assign by Index from the Local Color Table
Block Terminator	O	Fixed value 0

5.2.2.10 Comment Extension

Operation is arbitrarily and can be multiple. The syntax is operated according to standards.

5.2.2.11 animation GIF operation restrictions

- Total data size

  File size is not specified. However, the total size (regardless of display or not)of each 1 document after decoding shall not surpass 76800 for the total of width*height*frame number for all objects.

- Display size

  Size of 1 object is vertical <=240, horizontal <=240

- Number of stored images

  Maximum of 16

- Update cycle specified values

  Minimum: 200 (millisecond)

  Maximum: 5000 (millisecond)

  Specified unit: 100 (millisecond)

- Repetition

  Infinite. But, the actual repetition number is implementation dependent.

- Number of objects for each screen

  Maximum 4. But, each object shall not overlap.

- Rewritable display size for each unit time

  Shall not exceed 38.4KB in 1 second.

- Others
  - Do not change the stored GIF object size and position. In other words it consists of the following.

    LogicalScreenWidth = ImageWidth = width characteristic of object element.

    LogicalScreenHeight = ImageHeight = height characteristic of object element.

    ImageLeftPosition = 0

    ImageTopPosition = 0
  - Even if delays occur in the update of animation GIF that should be generated at the same time as other rendering execution etc., by C-profile basic receivers, do not cut out the GIF images and display the GIF image in order.
  - Transparent processing is not used in animation GIF.

5.3 Audio coding

5.3.1 MPEG-2 AAC

Follows Vol. 7, section 4.2.

5.3.1.1 Coding parameters

Table 5-16 Coding parameters of MPEG-2 AAC

Sampling frequency	Bit length
24kHz (half-rate) 48kHz	16 bit 16 bit

- Quality indicator (quality_indicator) during half-rate sampling is operated at mode 3.
- Multichannel stereo is not operated.

5.3.1.2 Transmission of MPEG-2 AAC

- Audio encoded with MPEG-2 AAC is transmitted by audio PES (stream format identification 0x0F) and the data carousel (stream format identification 0x0D).

5.3.1.3 Limitations of data carousel transmission

- File size is 128KB or less.
- When start orstop controls are received, playing the audio shall start or stop.
- The operation of SBR by data carousel transmission shall be an optional, and the encoding sampling rate is half-rate (24kHz) during SBR operation.

5.3.1.4 Data format of MPEG-2 AAC files

- MPEG-2 AAC Elementary Stream format

- The audio frame composed of audio data that corresponds to the ADTS header as shown in Figure 5-2 is assumed to be one unit, and the format is composed of a single unitor multiple units units. (Because one audio frame becomes a unit of 1024 samples in PCM, it is about 21.3 milliseconds in 48kHz sampling.)

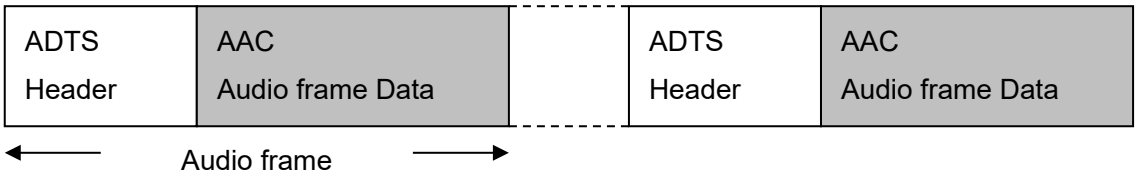


Figure 5-2 Data format of MPEG-2 AAC files

5.3.2 AIFF-C

AIFF-C is not operated.

5.3.3 Built-in sound

The encoding method for the built-in sound of receivers are implementation dependent. The allocation of a built-in sound sources is shown in Table 5-17.

Table 5-17 Allocation of receiver unit built-in sound

0: button operation sound A	1: button operation sound B	2: button operation sound C	3: button operation sound D
4: button operation sound E	5: button operation sound F	6: button operation sound G	7: button operation sound H
8: button operation sound I	9: button operation sound J	10: button operation sound K	11: button operation sound L
12: Alert sound A	13: Alert sound B	14: Alert sound C	15: Alert sound D

The numbers in the table show the sound_id when they are specified from multimedia codes.

5.3.4 MIDI

MIDI is not operated.

5.3.5 Audiosynthesis of receiver units

Simultaneous playback of multiple sounds is optional, but when simultaneous playback is not possible, follow Table 5-18. However, MPEG-2 AAC stream and built-in sound should be output in combination.

Table 5-18 Priority order during multiple audio playback presentation

	MPEG-2 AAC File (storage)	Built-in sound (4)
MPEG-2 AAC Stream (main-line)	Storage priority destination (2)(3)	Built-in sound priority destination (1)
MPEG-2 AAC File (storage)	Afterward priority destination	Built-in sound priority destination
Built-in sound (4)	Built-in sound priority destination	Afterward priority destination

(1) When the MPEG-2 AAC stream is interrupted by built-in sound playback, the interruption time should be minimized in designing the receiver.When mixing both sounds, a sound volume should be 1:1.

(2) When the MPEG-2 AAC file is instructed to playback during MPEG-2 AAC stream playback, the MPEG-2 AAC stream playback stops and priority is given to playback of the MPEG-2 AAC file. However, in this case, the playback of the main line image stream may be stopped.

(3) MPEG-2 AAC stream and main line image stream playback is done after the MPEG-2 AAC file playback terminates however, it is necessary to create contents by noting there is a possibility switching may take some time.

(4) Operations for the timing of audio (for example, button sounds such as mobile phones) and built-in sound overlaps, is implementation dependent. Moreover, the tone quality of a built-in sound is not especially specified. Therefore, the output audio should correspond to the receiver performance.

5.4 Character codes

5.4.1 8-bit character codes for C-profile

Refer to Chapter 6 of this vol..

5.4.2 Shift JIS

Refer to Chapter 3 of this vol..

6 Operation of closed caption coding (option)

6.1 Definition and range of service

In C-profile, the following closed caption services are provided .

Closed caption :Closed caption service (for example, translated character closed caption, etc.) synchronized with main video, audio and data.

This volume complies with [Section 2] "Operational provisions related to profile A", Chapter 4 "Operation of caption and superimpose encoding".

6.2 Organization and transmission operation

6.2.1 Restrictions on organization and transmission

(1) Transmission method

It transmits by the independent PES transmission method (stream format identification 0x06).

(2) Organization

Transmitted by independent ES. Moreover, it is transmitted by the PMT which is the same as the main service, at the same time, and the delivery of closed caption data within the same program or before the program starts is not done.

(3) The Number of ES

Closed caption that can be transmitted simultaneously is 1 ES.

(4) Transmission of multiple languages

The number of languages that are transmitted simultaneously is maximum of two or less per 1ES, and the language identification is done by the closed caption management data and the data group identification of the data group in the ES. However, the display of closed caption for the second language is not necessary in the receiver.

(5) Bitmap data

Bitmap data is not operated.

(6) Display modes that can be used

This is a fixed operation only for the "the choice of caption and displayduring reception, and the choice of caption and displayduring the playback of recording". It is a same when multiple languages are transmitted.

(7) Operation of built-in sound and additional sound

Neither the sound with built-in the receiver nor additional sound is operated.

(8) Existence confirmation of closed caption

The closed caption management data and the closed caption data show that the closed caption exists in the stream. Display of closed caption text is carried out after the closed caption management data or closed caption data is received but because the operation of the data group is fixed (Refer to 6.2.3), the display of the closed caption text after the

closed caption data is received is possible even if the closed caption management data is not received.

Standard delivery frequency 1 time/10 sec. (permissible deviation of ±5 sec.)
However, the delivery of the following closed caption management data can be omitted when the closed caption text data is delivered, within the above-mentioned minimum delivery frequency (5 sec.), however, in this case, both the closed caption text data and closed caption management data should meet requirements of the above-mentioned standard delivery frequency. Moreover, the delivery of the closed caption management data may be interrupted by CM's etc.

6.2.2 PES transmission method used in closed caption

The synchronous type PES transmission method is applied and the timing synchronization is maintained by the PTS. Table 6-1 shows the parameters set in the PES packet.

- Configuration parameters : Refer to Table 6-1
- Maximum number of ES's transmitted to the same layer simultaneously : 1ES
- Maximum number of languages per ES : 2 languages
- PES configuration unit : 1 data group
- PES maximum size : 640bytes (However, 3 TS packets is the upper limit)
- PES packet minimum delivery interval : 1000 milliseconds (However, CS: for PES only for the screen erasure does not follow this.)
- Maximum ES rate : 3Kbit/s
- Reception buffer : 1280bytes or more (for both 1language and 2 languages)

Apart from this, 256bytes is needed for the DRCS. Refer to 6.6.

Table 6-1 Configuration parameters of PES packets in closed caption

Field	Operation
Stream_id	0xBD(private_stream_1)
PES_packet_length	Number of bytes in a PES packet follow later. *1
data_identifier	0x80
private_stream_id	0xFF
PES_data_packet_header_length	Shows the length of PES_data_private_data_byte. Normally 0x00 is input. *2
PES_data_private_data_byte	This field can be skipped. *2
Synchronized_PES_data_byte	Stores data of the closed caption data group.

*1 Input 0 into this value, and operations not provided in the PES packet length are prohibited.

*2 When operating PES_data_private_data_byte, specify the correct length of PES_data_private_data_byte in the PES_data_packet_header_length, without fail.

For PES packet delivery, the following restrictions are established.

- Delivery order of PES packets and the time order of PTS should not be interchanged.

- In the ‘n’th order of the PES packet at PTS time, the total information volume of PES packets starting delivering in ‘n’th order should not exceed the reception buffer capacity (1280bytes).
- Complete sending PES packets before Td, and from the PTS time. Here, Td is the time from reception completion to presentation completion, almost 0.5 seconds as a rough estimate.
- Regarding the interval of PES packet delivery of the closed caption text data group, the interval of the PTS time of delivery order ‘n’th PES packet and PTS time of ‘n-1’th PES packet should be bigger than the Td of nth data.

  Transmission of synchronized with Video are possible on the transmission side when the image of the total delay T meets the following requirements.

  T > LX 8 / R + Td

  Here, L is the maximum PES packet length and R is the ES bit-rate.
- The movement of receiver exceeding the reception buffer is implementation dependant. The movement of receiver exceeding the reception buffer for the DRCS is implementation dependant, however, it is recommended to have movements described in 6.6.

6.2.3 Operation of data groups

Data group transmission shall be operated as shown in Table 6-2. The “data_group_version” will not be operated.

When class A is received, the receiver shall judge the number of languages to be 1 and process only the 0x01 closed caption text of class A (body text and DRCS). When class B is received, the receiver shall judge the number of languages to be 2 and, if the user selects the closed caption of the first language, process only the 0x21 closed caption text of class B (body text and DRCS). If the user selects the closed caption of the second language, the receiver shall process only the 0x22 closed caption text of class B (body text and DRCS).

Table 6-2 Data group parameters

Field	Operation
data_group_id	• If the number of closed caption languages is 1, the closed caption management data and the closed caption text shall be operated with 0x00 and 0x01 of class A, respectively. 0x20, 0x21 and 0x22 of class B will not be operated. • If the number of languages is 2, the closed caption management data shall be operated with 0x20 of class B and the first and second closed captions shall be operated with 0x21 and 0x22 of class B, respectively. 0x00 and 0x01 of class A will not be operated.
data_group_version	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 640bytes.
data_group_data_byte	Data group data (closed caption management data, closed caption text data) is stored.
CRC_16	Operate error checking by CRC16. On detecting errors, receiver delete aforesaid data group.

6.2.4 Operation of closed caption management data

Data units are not allocated in closed caption management data. The movement of receiver that do not receive closed caption management data is implementation dependant. Table 6-3 shows the parameters that can be specified for the closed caption management data used in closed caption.

Table 6-3 Closed caption management data parameters

Field	Operation
TMD	‘00’ (free)
num_languages	1 - 2
language_tag	0 - 1
DMF	‘1010’ (the choice of caption and display on reception and the choice of caption and display on recording playback)
ISO_639_language_code	Used language code (“jpn” fixed)
Format	Not operated ‘1111’
TCS	‘00’ (8-bit character codes)
rollup_mode	‘00’ (non-rollup)
data_unit_loop_length	‘0’
data_unit	Do not allocate data units.

6.2.5 Operation of closed caption text data

In the same closed caption text data, multiple data units of the same or different data unit parameters can be allocated. When multiple data units exist in the same closed caption text data, they are processed in order of appearance of the data units.

Configuration parameters for closed caption text data is shown in Table 6-4.

Table 6-4 Parameters of closed caption text data

Field	Operation
TMD	'00' (free)
STM	Not operated.
data_unit_loop_length	Operated as defined. However, 1 PES packet should not exceed 640bytes.
data_unit	Data unit (text, 1 byte DRCS) is stored.

6.2.6 Operation of data units

Parameters that can be configured in data units are shown in Table 6-5.

Table 6-5 Data unit parameters

Field	Operation
unit_separator	0x1F as defined.
data_unit_parameter	0x20 (text) 0x30 (1 byte DRCS)
data_unit_size	Operates as defined. However, 1 PES packet should not exceed 640bytes.
data_unit_data_byte	Stores data unit data.

6.2.7 Operation of PSI/SI

6.2.7.1 Operation of component tags

Component tag value of the closed caption ES is 0x87.

6.2.7.2 Operation of the PMT

Updating the PMT is recommended to add and delete ES information at the moment of initiation and termination of closed caption. However, an operation to enable the description of ES information at all times is needed.

6.2.7.3 Stream format identification

The stream_type of the closed caption ES is 0x06 (independent PES_packet).

6.2.7.4 Descriptor operation

Descriptor operation of the PMT and EIT for closed caption is shown in Table 6-6.

Table 6-6 Descriptor operation of the PMT and EIT

Descriptor	PMT	EIT
Stream Identifier Descriptor	Mandatory	-
Data Component Descriptor	Mandatory	-
Data Contents Descriptor	-	Not operated

In the Short Event Descriptor of EIT, it is recommended to descript character information that can inform users that this program comes with a closed caption function.

6.2.7.5 Data Component Descriptor

The data_component_id of the Data Component Descriptor is 0x0012. And Table 6-7 shows the parameters to configure the additional information identification.

Table 6-7 Configuration parameters of additional information identification for the Data Component Descriptor

Field	Operation
DMF	’1010’
Timing	Closed caption: ’01’ (Program synchronization)

6.2.7.6 Target Region Descriptor

The Target Region Descriptor is not operated.

6.2.7.7 Data Contents Descriptor

The Data Contents Descriptor is not operated.

6.3 Display format of closed caption

6.3.1 Display format

The broadcasting station considers two kinds of rectangular areas, a lateral direction of a normal size of 12 characters or more with 4 lines (in case a portrait display device is used on a vertical screen) and a lateral direction of a normal size of 16 characters or more with three lines (in case of a portrait display device is placed horizontally and used on a horizontal screen), as display formats to present closed caption in the receiver , and to produce the closed caption text in consideration of displaying all characters of the closed caption text data, even if the receiver provides a display area with the above-mentioned features.

The receiver should have one of above-mentioned display areas, and displays with a linefeed in the place of the control code of the active position line feed of the closed caption text, and displays with line feed for the character writing direction edge in the display area of the receiver . Only 1 linefeed, not 2, is processed when the line feed of the operation position linefeed control code and the display area character writing direction edge overlaps.
The display area size of the receiver depends on the implementation dependant of the receiver , but when receiving the closed caption text produced according to the above-mentioned assumption, all characters should be able to be displayed.

Overlapping of the closed caption text and the image is implementation dependant. Moreover, it is recommended, in implementation dependant, to display the closed caption text for receiver that do not meet the display format of the above-mentioned assumption appropriately.
The image chart in Figure 6-1 shows the conversion and production of closed caption from profile A to C-profile.

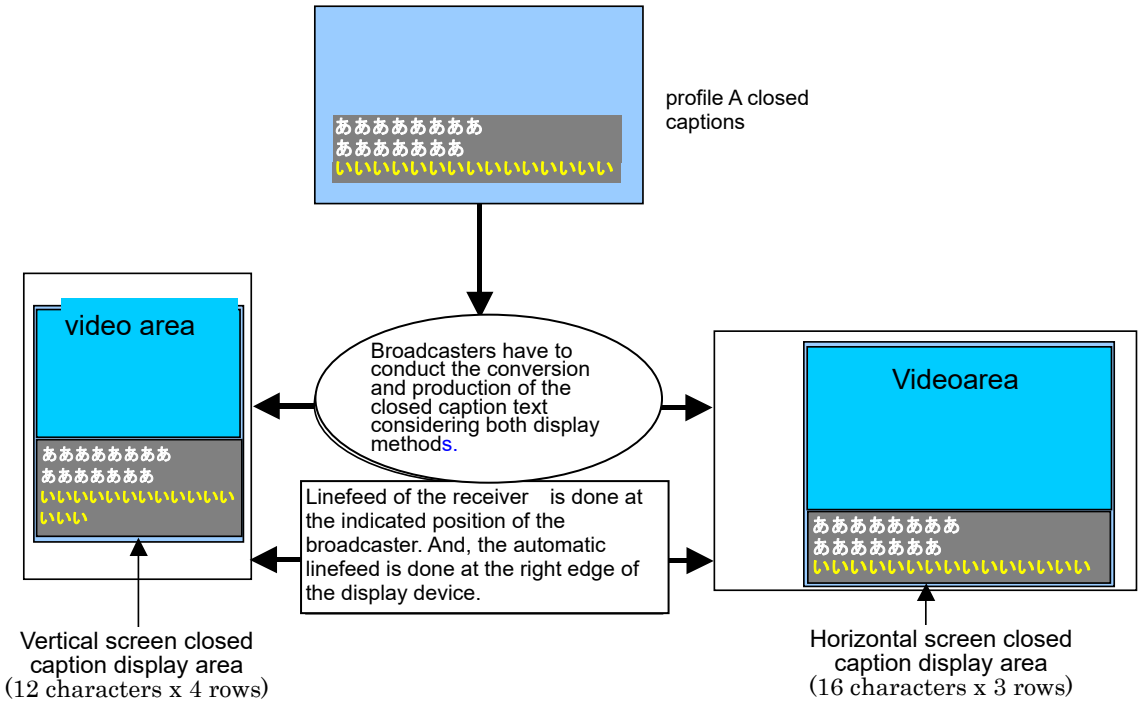


Figure 6-1 Conversion and production image of C-profile closed caption

6.4 Characters used in closed caption

6.4.1 Character entity

The character encoding method used in closed caption uses 8-bit character codes for C-profile with a subset of 8-bit character codes for C-profile. In 8-bit character code for C-profile, the GL code area is fixed to the DRCS character set (one byte code) provisioned by ARIB STD-B24, the GR code area is fixed to the Kanji character set (two byte code, Row 1 to Row 94) provisioned by ARIB STD-B24, and the character set table is not changed. Moreover, (3) below not used as a character code is used as a control code.

(1) Kanji character set range

First byte A1 - FEh

Second byte A1 - FEh

(2) DRCS set range

21 - 7Eh

(3) Control code range

00 - 1Fh, 20h, 7Fh, 80 - 9Fh

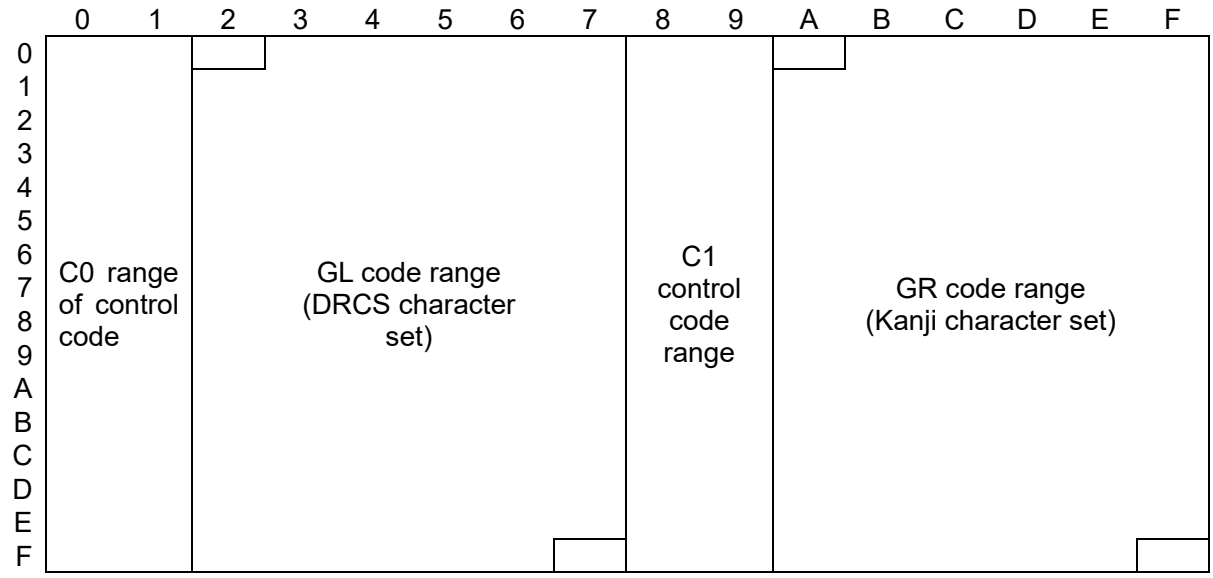


Figure 6-2 Table of 8-bit character codes range for C-profile

6.4.2 Character fonts

This is implementation dependant of the receiver .

6.4.3 Character size

Character size displayed in closed caption is assumed as medium font. Only standard and medium character size are operated during transmission.

Normal size: medium font equivalent

Medium size: Only the size of the horizontal direction is half the size of characters from the standard.

Medium size is not operated except for characters specified by operation of medium size characters shown in Table 6-8. Moreover, the voiced sound mark specified as medium may be converted into medium size double characters in the receiver .

Table 6-8 shows the restrictions of character display.

Table 6-8 Range of code set used to specify the display size format and characters

Character	2 byte code Row (Cell)	Standard	Medium
Symbol	1, 2	O	O*1 *2
Alphanumeric	3	O	O
Hirakana	4	O	O*2
Katakana	5	O	O*2
Greek alphabet	6	O	
Russian character	7	O	
Ruled line	8	O	
Kanji	16 - 84	O	
Additional symbols	90 (1 - 6) (8 - 11) (16 - 17) (20 - 40) (64 - 65)		
	91 (1 - 49)		
	92 (1 - 4) (5 - 12) (13 - 15) (16 - 25) (26 - 31) (32 - 41) (42 - 47) (48 - 52) (53 - 54) (55 - 91)	O O O O O O O O O	
	93 (1 - 45) (48 - 91)	O O	
	94 (1 - 93)	O	
DRCS	Normal size pixel configuration Medium size pixel configuration	O	
Special code (space, delete)		O	O

*1 Exclude Row 2, Cell 94.

*2 This may be displayed by the normal size in the receiver .

6.4.4 Display partition

As this is a letter space and line space without a specified operation, the definition of display partition is not done.

6.4.5 Non-spacing characters

Non-spacing characters are not used.

6.5 Control code used in closed caption

6.5.1 Control code

The control code used in the closed caption is compliant with ARIB STD-B24, Vol. 1, Part 2, 7.1.2. However, the operation limitations in Table 6-9, Table 6-10, Table 6-11 are established.

Table 6-9 C0 control range

C0 Control code	Control function	Yes or no, of use	Control item and supplement
NUL	Space	O	
BEL	BEL	X	
APB	Active position backward	X	
APF	Active position forward	X	
APD	Operation line advance	X	
APU	Operation line retreat	X	
APR	Active position return	O	
PAPF	Specified operation position advance	X	
APS	Active position specification	X	
CS	Clear screen	O	
CAN	Cancel	X	
ESC	Escape	X	
LS1	Locking-shift 1	X	
LS0	Locking-shift 0	X	
SS2	Single-shift 2	X	
SS3	Single-shift 3	X	
RS	Data header identification code	X	
US	Data unit identification code	O	Used for identification of data units, but cannot be used at 8-bit character codes for C-profile.

O: possible use Δ: possible use with limitations X: cannot be used

Table 6-10 C1 control range

C1 Control code	Control function	Yes or no, of use	Control item and supplement
BKF (CFLA0)	Foreground color Black	O	Specify Black to foreground (index value =0)
RDF (CFLA1)	Foreground color Red	O	Specify Red to foreground (index value =1)
GRF (CFLA2)	Foreground color Green	O	Specify Green to foreground (index value =2)
YLF (CFLA3)	Foreground color Yellow	O	Specify Yellow to foreground (index value =3)
BLF (CFLA4)	Foreground color Blue	O	Specify Blue to foreground (index value =4)
MGF (CFLA5)	Foreground color Magenta	O	Specify Magenta to foreground (index value =5)
CNF (CFLA6)	Foreground color Cyan	O	Specify Cyan to foreground (index value =6)
WHF (CFLA7)	Foreground color White	O	Specify White to foreground (index value =7)
COL	Color specification	X	
POL	Pattern polarity	X	
SSZ	Small size	X	
MSZ	Medium size	Δ	Operation complies with Table6-8.
NSZ	Normal size	Δ	Operation complies with Table6-8.
SZX	Specification size	X	
FLC	Flashing control	Δ	Flashing speed is so that users can recognize the flashing of characters. Operation of flashing, refer to 6.5.2.
CDC	Conceal control	X	
WMM	Writing mode modification	X	
TIME	Time control	Δ	Only usable for process waiting.
MACRO	Macro specification	X	
RPC	Character repeat	X	
STL	Start underline and mosaic separation	X	
SPL	Stop underline and mosaic separation	X	
HLC	Enclose control	X	
CSI	Control sequence introducer	X	

O: possible use Δ: possible use with limitation X: cannot be used

6.5.1.1 Operation of color specifications

The color specification is only operated for the foreground color, and specified from 8 colors of index0 to index7 of the CLUT common fixed colors allocated in CFLA0 to CFLA7 of C1 control code in Table 6-10. The background color, the half foreground colors and backgrounds neutral color are not specified.

It is recommended that the broadcaster operate the color specification in the same line within 3 times.

Refer to Appendix 1 of [Section 2] related to profile A for the CLUT common fixed colors.

6.5.2 Operation of flashing

Flashing of 8-bit character codes character string for C-profile should flash at a speed which the user can recognize the character flashing. Timing in which the flashing is started is the time when the character of flashing specification is shown. Moreover, flashing is assumed to be only a positive-phase-sequence. When the closed caption initialization operation defined in 6.7 is performed and the deletion of the display screen is indicated by the CS control code, the flashing of an aforesaid character is terminated.

6.5.3 Extension control code

Operation of the extension control code is shown in Table 6-11.

Table 6-11 Extension control code (CSI)

Character	Control function	Yes or no, of use	Control item and supplement
SWF	Format selection	X	
CCC	Synthesis control	X	
RCS	Luster color control	X	
ACPS	Operation position coordinates specification	X	
SDF	Display configuration dot specification	X	
SDP	Display position specification	X	
SSM	Character configuration dot specification	X	
PLD	Partial line down	X	
PLU	Partial line up	X	
SHS	Character interval specification	X	
SVS	Line interval specification	X	
GSM	Character transformation	X	
GAA	Coloring division	X	
SRC	Luster specification	X	
TCC	Switch control	X	
CFS	Character font setting	X	
ORN	Character decorations specification	X	
MDF	Character format at specification	X	
PRA	Built-in sound playback	X	
XCS	External character set alternative code string definition	X	
SCR	Scrolling specification	X	

O: possible use Δ: possible use with limitation X: cannot be used

6.6 Operation of the DRCS

Calling of the DRCS uses only one arbitrary set from DRCS-1 to DRCS-15 in the one byte DRCS sets in ARIB STD-B24, Vol. 1, Part 2, 7.1.1.5, and the two byte DRCS set DRCS-0 is not used. Moreover, the encoding of the DRCS pattern operates only the pattern transmission, and operation by geometric is not done. Moreover, the fontId (font identification) should specify 0. When specified numbers except 0, the receiver considers it as 0. Buffers ensured by the receiver for DRCS is 256 bytes in DRCS for closed caption. The maximum number of DRCS used, at the same time, in closed caption should be 5. When 6 or more are delivered, it is mandatory that the first 5 be kept in the buffer of the receiver , and reading continues repeatedly. The handling of the 6th depends on the receiver . The patternData should send two tone data of the design frame based on the dot configuration (horizontal and vertical of 16×18) in ARIB STD-B37 "Structure and operation of closed caption data conveyed by ancillary data packets", Supplement "Operational guidelines for closed captions".

Table 6-12 Active parameters of DRCS figure coding

Field	Operation
NumberOfCode	Operated as defined in ARIB STD-B37
CharacterCode	Operated as defined in ARIB STD-B37
NumberOfFont	Operated as defined in ARIB STD-B37
FontId	Specification of only 0 is possible. (0, even if specified other than 0)
Mode	Specification of only 0000 is possible.
Depth	Specification of only 0 is possible.
Width	Operated as defined in ARIB STD-B37
Height	Operated as defined in ARIB STD-B37
patternData	Operated as defined in ARIB STD-B37

6.7 Operation of the initialization operation

The receiver shall perform the closed caption initialization operation shown in 6.7.1 to 6.7.5.

Table 6-13 Initialization status

Item	Target	Initial status
Display screen	Display image	Character delete status
Definition data	DRCS	Data clear
Operation instruction	Character coding	Time control operation End status Operation position Left top corner of display area
Status instruction	Character coding	Character size controls 1X1 (standard) Foreground color Maximum white brightness (CMLA7) Background color Transparent (CMLA8) Flashing control Flashing end (FLC04/15)

6.7.1 Initialization by switching classes in the data group

The receiver shall initialize all items prescribed in Table 6-13 when it receives closed caption text data or closed caption management data of a different data group class than those currently being processed for presentation.

6.7.2 Initialization by closed caption text

The receiver shall initialize all items prescribed in Table 6-13 when it receives closed caption text data of the same data group class and language as those currently being processed for presentation.

6.7.3 Initialization by main text data unit

The receiver shall initialize the items in "Status instruction" in Table 6-13 when the received closed caption text data of the same data group class and language as those currently being processed for presentation which contains the body text data unit, immediately before processing the body text data unit for presentation on the receiver.

6.7.4 Initialization by character control codes

The receiver shall initialize the items under "Display screen", "Operation instruction" and "Status instruction" in Table 6-13 immediately before executing the clear screen (CS) process.

6.7.5 Initialization by receiver operation

The receiver shall initialize all items prescribed in Table 6-13 when a station selection operation or switching of the closed caption language has been performed.

6.8 Mono-media used in closed caption

6.8.1 Operation of geometric

Geometric is not operated.

6.8.2 Operation of bitmap data

Bitmap data is not used.

6.8.3 Operation of built-in sound

Built-in sound is not operated.

6.8.4 Additional sound

Additional sound is not used.

6.9 Ideal receiver unit operation

It is preferable that the receiver unit judges the presence of closed caption data reception by the presence of the closed caption management data. The presence of the closed caption management data or PMT data is set as standard, when displaying symbols that inform viewers of closed caption reception, and it is based on the presence of the title management data or the PMT data.

6.9.1 Initialization and termination of closed caption display

The receiver unit controls initialization and termination of the closed caption function by operation of the user. The control of initialization and termination of closed caption function tied with other data like the title management data or PMT data, etc. is a matter of product planning.

7 Operation of multimedia encoding

7.1 Introduction

The operation of the multimedia encoding follows,

- "Appendix 1, Operational Guidelines" and
- "Appendix 4, Operational Guidelines for Implementing Services for Portable Receiving System",
  in ARIB STD-B24, Vol. 2 " XML-based Multimedia Coding Scheme ".

However if there is contradiction, the specification in this volume have a priority to above normative references. The normative reference that is not operated in the specification in this volume is not applicable.

When the data broadcasting reception function is implemented in receivers that receive C-profile data broadcasting contents, it is assumed that the communication function is implemented at the same time. Because, in portable reception, the bandwidth that can be used by data broadcasting is narrow, it is assumed that contents are produced on the assumption of a tie-up with communication. Therefore, when such contents are presented in receivers that do not have the communication function, only incomplete functions and information may be provided to the users.

However, if receiver unit designers are aware of the above assumption, they may design receivers without the communication function.

7.2 Operation of NVRAM in Digital Terrestrial Television C-profile broadcasting

Digital Terrestrial Television C-profile area for the affiliation (hereafter referred to as C-profile area for the affiliation) and TVlink area shown in Table 7-1 shall be set up for allocation as NVRAM to be used for storing permanent information in the C-profile digital terrestrial television broadcasting.

Table 7-1 NVRAM used by Digital Terrestrial Television C-profile broadcasting

Class	Purpose	Capacity of NVRAM
Digital Terrestrial Television C-profile area for the affiliation	Area used commonly by operators belonging to the same affiliation area	• 24KB per affiliation (64 byte fixed blockx 384) • 8KB out of24KB is for inner affiliation common area, and the remaining 16KB is divided by 8 and used as the individual operator area. • Affiliation number: 12 affiliations
TVlink area	Area used for TVlink service	• Maximum of 256 bytes variable block • Writable block number :50 or more

C-profile area for the affiliation and TVlink area shall ideally be allocated statically on the receiver; however, implementation in which they will not be allocated statically shall also be acceptable. In other words, the non-volatile physical memory area can be shared with other applications on the receiver.

In the case of a receiver on which the above areas have not been allocated statically, the writing operation to the area(s) from a BML document may fail if the relevant memory area is in use by another application. On such a receiver, relevant area is assumed to be allocated at the time of the first writing operation to each affiliation area. And, the unit (capacity) for the C-profile area for the affiliation shall ideally be allocated in the unit of 1 affiliation (24KB).

7.2.1 Writing frequency of the NVRAM area

NVRAM implemented in the receiver uses a device that has an upper limitation in the writing frequency. These devices may breakdown, and results in shortening the life of the receiver when the writing frequency is exceeded the limit. Therefore, it is recommended to consider not generating excessive writing frequency to NVRAM. This is explained in appendix 1.

7.2.2 Allocation of NVRAM

7.2.2.1 Allocation of the C-profile area for the affiliation

- The maximum number of affiliations to which a C-profile area for the affiliation can be allocated shall be 12. In addition to the allocation to affiliations whose affiliation identification (affiliation_id) defined in Vol.7 have the following 8 values, allocation to affiliation_id=08-0B shall also be possible as reserve areas, which shall not be operated for the time being.

  Value of affiliation identification where the C-profile area for the affiliation is currently allocated:

  00, 01, 02, 03, 04, 05, 06, 07

- The reserve affiliation _id will be allocated by the governing organization described in Appendix 13.
- An initial value of the reserved C-profile area for the affiliation is a null character string.
- When the C-profile area for the affiliation has not been allocated yet, the readPersistentArray function returns null.
- Although the C-profile area for the affiliation has already been allocated, the readPersistentArray function shall return Array with the Array[0] set to null in the event when the access to the C-profile area for the affiliation was rejected by user operation or receiver setting (*).
- When writing is instructed for the first time in receivers with an implementation to reserve the memory area, it is allowed that reserving the area may be impossible due to a lack of memory.

  (*) For example, it refers to cases such as when the user was prompted by the receiver to either grant or reject access to the C-profile area for the affiliation and chose to reject the access; or when the user has configured the receiver to reject access to the C-profile area for the affiliation.

7.2.2.2 Allocation of individual operator areas in the C-profile area for the affiliation

- An affiliation-common area (8KB), which can be freely accessed by stations within the affiliation and 8 areas, which can be used exclusively by individual broadcasters shall be set in the C-profile area for the affiliation. Each operator shall be allowed to use an area of up to 2KB.
- It is necessary for receivers to acquire and release the area administrating the individual operator area by an original network unit (2KB, 32 block unit).
- In the case of a C-profile receiver, which is assumed for mobile reception, receivable broadcasting stations will change depending on the receiving location. Therefore, it shall not be possible to associate the individual operator areas in the C-profile area for the affiliation with broadcasting stations beforehand, nor shall it be possible to specify the maximum number of operators that will perform writing to the areas.
- The receiver shall allocate an individual operator area within the C-profile area for the affiliation to the relevant broadcaster in the order that each station's content wrote to the area by specifying an original network ID. When a writing operation exceeding the number of areas available on the receiver (i.e. 8) has been instructed, the writing operation shall be performed after deleting unnecessary areas by user instruction. The writing will fail if sufficient capacity could not be allocated.

7.2.3 Deletion function of the C-profile area for the affiliation

- It is necessary for receivers to have a delete function for each affiliation area and, for each individual operator area set in the affiliated area, both by instruction of the user.
- Regarding the individual operator areas, it is necessary that the user can delete by one area unit.

The guidelines of the deletion management in the C-profile area for the affiliation of the receiver are shown as follows.

- It is recommended that deletion function is able to be executed (without stopping the execution of contents) when a write function is calling with specifying original network ID in the case of no space for the individual operator area, and also is implemented as the deletion function from the menu prepared by the receiver.

- Specifically, when the writing function is called with specifying a new original network ID by an affiliated operator with all 8 individual operator areas already is used, the receiver unit shows the candidate area to delete to the user through a display window, etc., and executes the deletion of each of area according to the user's selection. When the user doesn't order the deletion, writing fails. When presenting candidates of deletion, it is

recommended to present the broadcasting station name. It is also recommended to record the presented broadcasting station name with SI information (TS name transmitted by the TS Information Descriptor of NIT) when writing is done specifying original network ID at the beginning (when securing the area).

- An initial value of the individual operator area when the deletion is executed is a null character string.
- The deletion function is executed from the menu prepared by the receiver, and the deletion of each affiliated area is executed by instruction of the user. It is recommended to present an affiliated name when presenting the candidate of the deletion of each affiliated area. The affiliation name presented is assumed to be stored in the receiver beforehand. At this time, by presenting a dialog box or another way to the viewers, it is recommended to display that serious problems may occur to a certain kind of the broadcasted services by the deleting.
- Other additional deletion functions are a matter of product planning.
- It shall be desirable to set up the function for deleting the C-profile affiliation area exclusively for the broadcasting application. If the deletion function will be shared with other applications, the viewer shall be given sufficient warning to prevent accidental erasure of information.

7.2.4 Identification of the C-profile area for the affiliation

When reading and writing information from the Multimedia service to the C-profile area for the affiliation, readPersistentArray()/ writePersistentArray() shall be executed treating each fixed length block as 1 file. Reading and writing of information from the Multimedia service to C-profile area for the affiliation shall be performed in units of fixed length blocks. The following URI shall be used for identifying the fixed length blocks.

nvram://<affiliation _id>;<original_network_id>;Cprogroup/<block number>
<affiliation _id>: Affiliation ID. Described by 2 hexadecimal digits
(0 have should be placed on the left end to make it to 2 digits when the value does not reach 2 digits)
<original_network_id>: Original network ID. Described by 4 hexadecimal digits.
<Block number>: 0 to 31

<affiliation_id>,<original_network_id> cannot be omitted

Access to this area shall fail when the affiliation ID is specified with an ID that does not belong to the broadcaster broadcasting contents currently being played. For the affiliate ID,

See Vol. 4, Chapter 9.

"0000" to "0003" can be specified for the area of original_network_id, besides the original network ID of the broadcaster currently broadcasting. When these values are specified, they are treated as a common area in the affiliation area, and they permit access from all stations of same affiliation_id. Access control by original network ID is not done.

Access to this area fails when an original network ID other than the broadcaster broadcastings contents currently playing in the area of original_network_id is specified.

7.3 Operation of keys from the Multimedia service

7.3.1 Values handled for used-key-list characteristics

The table below shows the possible combination of the <key-group> values and the keys that can be specified in the CSS property “used-key-list" to control the exclusive access of keys by a BML browser or the tuning function. The remote control keys shown in Table 7-3 are virtual names not physical keys. A key to allocate physically in the key code shown in Table 7-3 is a matter of product planning.

However, the receiver unit should offer viewers the keys virtually (software key or physical key), because contents are produced assuming the key codes shown in Table 7-3 can be sent.

Table 7-2 Values of key-groups

<key-group>	Meaning
basic	Enter key, back key
numeric-tuning	Numerical key (from 0 to 9) (Note 1)
special-1	*, # (Note 1) (Note 2)
special-2	TVlink key

(Note 1) Note that the key-group is specified for CSS, there can be no way to tune.

(Note 2) The key-group should be used in conjunction with numeric-tuning key-group.

7.3.2 Correspondence of keys, key codes and access keys

Table 7-3 shows the mapping of the characters specified as keys, key codes, and access keys that can be used by the Multimedia service.

Table 7-3 Correspondence table of keys, key codes and access keys

Key	Key code	Access key character
0 to 9	5-14	N/A
“Enter”	18	N/A
“Back”	19	‘X’
TVlink key	100	N/A
*	101	N/A
#	102	N/A

7.4 Operation of the BML version

The BML version in the BML document is major_version=12 and minor_version=0. This PI is described following the XML declaration and the DOCTYPE declaration. Refer to appendix 9 for the receiver behavier when the update of major_version and minor_version.

<?bml bml-version="12.0" ?>

7.5 Operation of character coding schemes

See ARIB STD-B24, Vol. 2, Appendix 4, “4.1. Character Coding Schemes”.

7.6 Operation range of media type and monomedia

It follows ARIB STD-B24, Vol. 2, Appendix 4, "4.2. Media Types and Operational Range of Monomedia ".

7.7 Operation of the BML elements

This section is specified for the operation of BML elements and attributes. The following modifications and restrictions are added although ARIB STD-B24 Vol. 2, Appendix 4, are referred.For the area not described here, ARIB STD-B24 is applied .

7.7.1 Declaration of XML and DOCTYPE

The XML declaration and DOCTYPE declaration in this operation are as follows.

- XML declaration
  <?xml version="1.0" encoding="Shift_JIS"?>
- DOCTYPE declaration
  <!DOCTYPE html PUBLIC "-//ARIB//DTD XHTML BML 12.0//JA"
  "http://www.arib.or.jp/B24/DTD/bml_12_0.dtd">

7.7.2 Operation of the BML elements

Operation in the basic service of elements used by BML are shown. The treatment of the element in "-" is implementation dependent. See appendix 3 for DTD of the operation.

Table 7-4 Operation of elements

Module		BML (operation)	Element
Core	Structure (required)	O	body
		O	head
		O	html
		O	title
	Text (required)	-	abbr
		-	acronym
		-	address
		-	blockquote
		O	br
		-	cite
		-	code
		-	dfn
		O	div
		-	em
		-	h1 - h6
		-	kbd
		O	p
		O	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	-	del
		-	ins
	Interaction channel	-	bdo
Form	Basic Forms	-	form
		-	input
		-	label
		-	select
		-	option
		-	textarea
	Forms	O	form
		O	input
		-	select
		-	option
		O	textarea

Module		BML (operation)	Element
		-	button
		-	fieldset
		-	label
		-	legend
		-	optgroup
Table	Basic Tables	-	caption
		-	table
		-	td
		-	th
		-	tr
	Tables	-	caption
		-	table
		-	td
		-	th
		-	tr
		-	col
		-	colgroup
		-	tbody
		-	thead
		-	tfoot
Image		O	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		O	*a&*
		-	*area&*
		O	*form&*
		O	*body&*
		-	*label&*
		O	*input&*
		-	*select&*
		O	*textarea&*
		-	*button&*
Metainformation		O	meta
Scripting		-	noscript
		O	script
Stylesheet		-	style
Style Attribute		O	
Link		O	link
Base		-	base
Name Identification (deprecated)		-	*a&*
			applet&
			form&

Module		BML (operation)	Element
			frame&
			iframe&
			img&
			map&
Legacy (deprecated)		-	basefont
		-	center
		-	font
		-	s
		-	strike
		-	u
		-	*body&*
		-	*br&*
		-	*caption&*
		-	*div&*
		-	*h1-h6&*
		-	*ht&*
		-	*img&*
		-	*input&*
		-	*legend&*
		-	*li&*
		-	*ol&*
		-	*p&*
		-	*pre&*
		-	*script&*
		-	*table&*
		-	*tr&*
		-	*th&*
		-	*td&*
		-	*ul&*
BML	BML	-	*bml:bml*
		-	*bml:bevent*
		-	*bml:beitem*
		-	*body&*
		-	*div&*
		-	*p&*
		-	*a&*
		-	*bdo&*
		-	*object&*
	Basic BML	-	*bml:bevent*
		-	*bml:beitem*
		-	*body&*
		-	*div&*
		-	*p&*
		-	*span&*
		-	*object&*
	Basic Mobile BML	O	*bml:bevent*
		O	*bml:beitem*
		O	*object&*

7.7.3 Attributes

Table 7-5 shows the operation of the attributes specified as operated in section 7.7.2 (div p br pre span a form input textarea img object meta title script link body head html bml:bevent bml:beitem).

Table 7-5 Operations for attributes of elements

Legend:

In “Operation” column, "O" means the attribute can be shownin the BML document, "Δ" means the value is fixed in the receiver units, then should not be shown in the BML document.

Element	Attribute	Operation	Restriction on operation
Common Attributes			
Core Attributes			
	id	O	Character string, maximum of 128 bytes.
	class	O	
	title	-	
I18N Attributes			
	xml:lang	-	Not operated.
Events Attributes			%Events.attrib; only operates a, input, textarea, object.
	onclick	O	
	ondbclick	-	
	onmousedown	-	
	onmouseup	-	
	onmouseover	-	
	onmousemove	-	
	onmouseout	-	
	onkeypress	-	
	onkeydown	O	
	onkeyup	O	
Style Attributes			
	style	O	
Core Modules			
Structure Module			
body	%Core.attrib;	O	Composed of %id.attrib + %class.attrib;. Definition of configuring elements are compliant with XHTML.
	%Style.attrib;	O	
head	%I18n.attrib;	-	
	profile	-	
html	xmlns	Δ	Operated with a fixed value of xmlns="http://www.w3.org/1999/xhtml" Operated with a fixed value of xmlns:bml= “http://www.arib.or.jp/bml”
	%I18n.attrib;	-	
	version	-	
title	%I18n.attrib;	-	
Text Module			
blockquote	%Common.attrib;	-	
	%Style.attrib;	-	
	cite	-	
br	%Core.attrib;	O	
	%Style.attrib;	O	
div	%Core.attrib;	O	
	%Style.attrib;	O	
h1-h6	%Common.attrib;	-	
	%Style.attrib;	-	

Element	Attribute	Operation	Restriction on operation
p	%Core.attrib;	O	
	%Style.attrib;	O	
pre	%Core.attrib;	O	Note 4)
	%Style.attrib;	O	
	xml:space	Δ	Fixed as "preserve"
span	%Core.attrib;	O	
	%Style.attrib;	O	
Text Extention Module			
hr	%Common.attrib;	-	
	%Style.attrib;	-	
Hypertext Module			
a	%Common.attrib;	O	
	%Style.attrib;	O	
	(%Events.attrib;)	O	element a operates %Events.attrib;.
	accesskey	O	
	charset	Δ	Fixed as "Shift_JIS".
	href	O	
	hreflang	-	
	rel	-	
	rev	-	
	tabindex	-	
	type	-	
List Module			
dl	%Common.attrib;	-	
	%Style.attrib;	-	
dt	%Common.attrib;	-	
	%Style.attrib;	-	
dd	%Common.attrib;	-	
	%Style.attrib;	-	
ol	%Common.attrib;	-	
	%Style.attrib;	-	
ul	%Common.attrib;	-	
	%Style.attrib;	-	
li	%Common.attrib;	-	
	%Style.attrib;	-	
Forms Module			
form	%Core.attrib;	O	
	%Style.attrib;	O	
	accept	-	
	accept-charset	-	
	action	O	
	method	O	
	enctype	Δ	Fixed as "application/x-www-form-urlencoded".

Element	Attribute	Operation	Restriction on operation
input	%Common.attrib;	O	
	%Style.attrib;	O	
	(%Events.attrib;);	O	Input element operates %Events.attrib;.
	accept	-	
	accesskey	O	
	alt	-	
	checked	-	
	disabled	-	
	maxlength	O	1 to 40. Note 1)
	name	O	
	readonly	O	
	size	-	
	src	-	
	tabindex	-	
	type	O	Use one of "text", "password", "submit".
	value	O	
label	%Common.attrib;	-	
	%Style.attrib;	-	
	for	-	
	accesskey	-	
select	%Common.attrib;	-	
	%Style.attrib;	-	
	disabled	-	
	name	-	
	size	-	
	multiple	-	
	tabindex	-	
option	%Common.attrib;	-	
	%Style.attrib;	-	
	disabled	-	
	label	-	
	selected	-	
	value	-	
textarea Note 3)	%Common.attrib;	O	Note 1) Note4)
	%Style.attrib;	O	
	(%Events.attrib;)	O	Textarea element operates %Events.attrib;.
	accesskey	O	
	cols	-	Input area is specified by CSS(width,height).
	disabled	-	
	name	O	
	readonly	O	
	rows	-	Input area is specified by CSS(width,height).
	tabindex	-	
	xml:space	Δ	Fixed as "preserve"
button	%Common.attrib;	-	
	%Style.attrib;	-	
	accesskey	-	
	disabled	-	
	name	-	
	tabindex	-	
	type	-	
	value	-	

Element	Attribute	Operation	Restriction on operation
fieldset	%Common.attrib;	-	
	%Style.attrib;	-	
legend	%Common.attrib;	-	
	%Style.attrib;	-	
	accesskey	-	
optgroup	%Common.attrib;	-	
	%Style.attrib;	-	
	disabled	-	
	label	-	
Image Module			
img	%Core.attrib;	O	
	%Style.attrib;	O	
	src	O	
	alt	O	The display is receiver dependent. optional to be presented.
	longdesc	-	
	height	-	
	width	-	
Object Module			
object	%Common.attrib;	O	
	%Style.attrib;	O	
	(%Events.attrib;)	O	Object element operates %Events.attrib;.
	archive	-	
	classid	-	
	codebase	-	
	codetype	-	
	data	O	
	declare	-	
	height	-	
	name	-	
	standby	-	
	tabindex	-	
	type	O	
	width	-	
Intrinsic Events Module			
a&	onblur	-	
	onfocus	-	
area&	onblur	-	
	onfocus	-	
frameset&	onload	-	
	onunload	-	
form&	onreset	-	
	onsubmit	O	
body&	onload	O	
	onunload	O	Note 2)
label&	onblur	-	
	onfocus	-	
input&	onfocus	-	
	onblur	-	
	onselect	-	
	onchange	O	

Element	Attribute	Operation	Restriction on operation
select &	onblur	-	
	onfocus	-	
	onchange	-	
textarea&	onfocus	-	
	onblur	-	
	onselect	-	
	onchange	-	
Metainformation Module			
meta	%I18n.attrib;	-	
	http-equiv	-	
	name	O	
	content	O	
	scheme	-	
Scripting Module			
script	id	-	
	charset	Δ	Fixed as "Shift_JIS".
	type	Δ	Fixed as "text/X-arib-ecmascript; charset=" Shift_JIS"".
	src	O	The externally referenced script must be one and be completed in syntax
	defer	-	
	xml:space	Δ	Fixed as "preserve".
Link Module			
link	%Core.attrib;	-	
	%Style.attrib;	-	
	charset	Δ	Fixed as "Shift_JIS".
	href	O	
	hreflang	-	
	type	Δ	Fixed as "text/css".
	rel	Δ	Fixed to "stylesheet".
	rev	-	
	media	Δ	Fixed as "tv".
BML module			
bml:bevent	id	O	Character string, maximum of 128 bytes.
bml:beitem	id	O	Character string, maximum of 128 bytes.
	type	O	One of following: EventMessageFired, ModuleUpdated, ModuleLocked TimerFired, DataEventChanged, MediaStopped. MainAudioStreamChanged is not operated.
	onoccur	O	
	es_ref	O	
	message_group_id	O	0 or 1
	message_id	O	
	message_version	O	
	module_ref	O	
	time_mode	O	One of following: "absolute", "origAbsolute"
	time_value	O	
	object_id	O	Only ID's of the object element that has an attribute type of "audio/X-arib-mpeg2-aac", and referes to the data transmitted in a data carousel
	subscribe	O	

Element	Attribute	Operation	Restriction on operation
object&	bml:streamstatus	O	
	bml:onfocus	O	
	bml:onblur	O	
	bml:accesskey	O	

Note 1) Characters exceeding maxlength (textarea is always 240 characters in two-byte code character) in the input and textarea are trancated. the excess part is not displayed.
One tab (0x09) or one white space character (0x20) is counted as one character. Line-feed character (0x0D0A) is counted as 1 or 2 characters however, which to use is implementation dependent.

Note 2) The expanded functions for broadcasting that can be used in the onunload event handler are only readPersistentArray(), writePersistentArray(), unlockModuleOnMemoryEx(), and unlockAllModulesOnMemory().To move quickly to the target document, it is preferable that the procedure be limited to the processes ending in a short time, like the setting to Ureg and the simple condition evaluation. etc.

Note 3) The generation of the DOM tree is the same as the operation of pre element specified in ARIB STD B24, Vol. 2, Appendix 4 5.3.2, following XHTML1.0 3.2 User Agent Conformance (http://www.w3.org/TR/2000/REC-xhtml1-20000126/#uaconf), shall be keep all the control codes except the control code of beginning and trailing (space, linefeed and tab).

Note 4) During the generation of DOM node, the tab (0x09) is kept, but the display to the screen follows CSS and only one tab character is replaced with one character of space character (0x20). In addition, in case of the textarea element, the displayed character string shall be displayed as folded at the right edge of the display area.

7.7.3.1 Restrictions on the order of elements in the head element

In the head element, the title element, meta element, link element, script element, and bml: bevent element shall appear in this order. The title element appears once, meta element, link element, and bml:bevent element appears none or once. The script element without specifying the src attribute appears none or once, and the script element with the specification of the src attribute appears none to twice.

7.7.3.2 Operation of bml:beitem element

When the attribute value is dynamically changed by DOM, first, set the subscribe attribute to false, then change the value, and after that, set the subscribe attribute to true.

7.7.3.3 Expansion of object elements

In this operation, the bml:onfocus/bml:onblur attribute are added to the object element in order that the focus can be obtained directly by the object element.

7.7.4 Entity

See ARIB STD-B24, Vol. 2, Appendix 4, “4.4.3. Entity”.

7.7.5 Operation of the BML element extended module (event)

- The maximum number of ModuleUpdated events that can have the subscribe attribute set to subscribe at the same time is 8.
- The maximum number of TimerFired events that can have the subscribe attribute set to subscribe, at the same time, is 2. It is assumed that it fires immediately when the time value of absolute playback time/ time at reception specified by TimerFired has already passed, at the moment of BML document interpretation.
- ModuleUpdated of any data carousel component can be observed regardless of which component is being presented by the contents .
- Behavier when two or more event handlers are associated with the same event is a matter of product planning. For instance, the following descriptions correspond to this.
  - Multiple bml:beitem elements which have the type attribute of "DateEventChanged" of the type attribute are described.
  - Multiple bml:beitem elements which have the type attribute of "ModuleUpdated" are described, and specify the same module in those module_ref attributes.
- The DataEventChanged event doesn't occur even if the data event is updated in an ES not being presented.
- When the ModuleUpdated event is subscribed for modules transmitted in ES's not being presented, a ModuleUpdated event is generated when the data event is updated in the ES that transmits the module. In this case, as a status value of the ModuleUpdated event, the following new values are operated.

Table 7-6 Value and meaning of the ModuleUpdated event status

Status value	Meaning
4	The data event was updated in the ES from which the module was transmitted. The module is not delivered before a data event update, and the module has been transmitted after a data event is updated.
5	The data event was updated in the ES from which the module was transmitted. The module has been delivered before a data event update, and the module has not been transmitted after a data event update.
6	The data event was updated in the ES from which the module was transmitted. The module has been delivered in both, before a data event update and after a data event update.

7.8 Operation of CSS

Table 7-7 shows the operation of CSS2 properties.

Table 7-7 Operation of CSS properties

Property	Operation
Selector	
*	O
E	O
E F	-
E > F	-
E:first-child	-
E:link	-
E:visited	-
E:focus	-
E:active	-
E:hover	-
E:lang(c)	-
E + F	-
E[foo]	-
E[foo="warning"]	-
E[lang="en"]	-
E:first-line	-
E:first-letter	-
E:before	-
E:after	-
div.warning	-
E.myclass	O
#myid	O
Assigning property values, Cascading, and Inheritance	
@import	-
!important	-
Other setting rules	
@charset	-
@font-face	-
@color-profile	-
Type of medium	
@media	O
Box model	
margin-top	-
margin-right	-
margin-bottom	-
margin-left	-
margin	O
padding-top	O
padding-right	O
padding-bottom	O
padding-left	O
padding	-
border-top-width	-
border-right-width	-
border-bottom-width	-
border-left-width	-
border-width	O
border-top-color	O
border-right-color	O
border-bottom-color	O
border-left-color	O

Property	Operation
border-color	-
border-top-style	-
border-right-style	-
border-bottom-style	-
border-left-style	-
border-style	O
border-top	-
border-right	-
border-bottom	-
border-left	-
border	-
Visual formatting model	
position	O
left	O
top	O
width	O
height	O
z-index	O
line-height	O
vertical-align	-
display	O
bottom	-
right	-
float	-
clear	-
direction	-
unicode-bidi	-
min-width	-
max-width	-
min-height	-
max-height	-
Other visual effect	
visibility	O
overflow	O
clip	-
Generated content, automatic numbering, and lists	
content	-
quotes	-
counter-reset	-
counter-increment	-
marker-offset	-
list-style-type	-
list-style-image	-
list-style-position	-
list-style	-
Paged media	
"@page"	-
size	-
marks	-
page-break-before	-
page-break-after	-
page-break-inside	-

Property	Operation
page	-
orphans	-
widows	-
Colors and backgrounds	
color	O
background	-
background-color	O
background-image	O
background-repeat	O
background-position	-
background-attachment	-
Font	
font-family	O
font-style	-
font-size	O
font-variant	-
font-weight	O
font	-
font-stretch	-
font-size-adjust	-
Text	
text-indent	-
text-align	O
text-decoration	-
text-shadow	-
letter-spacing	-
word-spacing	-
text-transform	-
white-space	O
Pseudo-class, pseudo-element	
:link	-
:visited	-
:active	-
:hover	-
:focus	-
:lang	-
:first-child	-
:first-line	-
:first-letter	-
:before	-
:after	-
Tables	
caption-side	-
border-collapse	-
border-spacing	-
table-layout	-
empty-cells	-
speak-header	-
User interface	
outline-color	-
outline-width	-
outline-style	-
outline	-
cursor	-
Aural style sheets	
volume	-

Property	Operation
speak	-
pause-before	-
pause-after	-
pause	-
cue-before	-
cue-after	-
cue	-
play-during	-
azimuth	-
elevation	-
speech-rate	-
voice-family	-
pitch	-
pitch-range	-
stress	-
richness	-
speak-punctuation	-
peak-numeral	-
Extended property	
clut	-
color-index	-
background-color-index	-
border-color-index	-
border-top-color-index	-
border-right-color-index	-
border-bottom-color-index	-
border-left-color-index	-
outline-color-index	-
pixel size	O
display-aspect-ratio	-
grayscale-color-index	-
nav-index	-
nav-up	-
nav-down	-
nav-left	-
nav-right	-
used-key-list	O
-wap-marquee-style	O
-wap-marquee-loop	O
-wap-marquee-dir	O
-wap-marquee-speed	O
-wap-accesskey	-
-wap-input-format	O
-wap-input-required	-

The ones specified as fixed values are defined as the most important rule (!important) in the default style sheet. The most important rule (!important) always overrides a normal rule. The most important rule (!important) is not operated in the BML document , then all are normal rules, the defined value is obtained as a fixed value.

7.8.1 Element applied each properties

Table 7-8 shows the combination of operated properties and applied BML elements

Table 7-8 Element that apply the CSS properties

	body	div	p	pre	br	span,a	form	input,textarea	object	img
Box model										
margin	-	Δ	Δ	Δ	-	-	Δ	Δ	Δ	Δ
padding-top	-	Δ	O	O	-	-	Δ	O	Δ	Δ
padding-right	-	Δ	O	O	-	-	Δ	O	Δ	Δ
padding-bottom	-	Δ	O	O	-	-	Δ	O	Δ	Δ
padding-left	-	Δ	O	O	-	-	Δ	O	Δ	Δ
border-width	-	O	O	O	-	-	O	O	Δ	Δ
border-top-color	-	O	O	O	-	-	O	O	-	-
border-right-color	-	O	O	O	-	-	O	O	-	-
border-left-color	-	O	O	O	-	-	O	O	-	-
border-bottom-color	-	O	O	O	-	-	O	O	-	-
border-style	-	O	O	O	-	-	O	O	Δ	Δ
Visual model										
position	-	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
left	-	O	O	O	-	-	O	O	O	O
top	-	O	O	O	-	-	O	O	O	O
width	-	O	O	O	-	-	O	O	O	O
height	-	O	O	O	-	-	O	O	O	O
z-index	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
line-height	-	-	O	O	Δ	Δ	-	O	-	-
display	Δ	Δ	O	Δ	Δ	Δ	Δ	Δ	Δ	Δ
Other visual effects										
visibility	Δ	O	O	O	-	Δ	O	O	O	O
overflow	-	Δ	Δ	Δ	-	-	Δ	Δ	Δ	Δ
Background										
color	-	-	O	O	-	O	-	O	-	-
background-color	O	O	O	O	-	O	O	O	Δ	Δ
background-image	O	-	-	-	-	-	-	-	-	-
background-repeat	Δ	-	-	-	-	-	-	-	-	-
Fonts										
font-family	-	-	Δ	Δ	-	Δ	-	Δ	-	-
font-size	-	-	O	O	-	O	-	O	-	-
font-weight	-	-	O	O	-	O	-	O	-	-
Text										

	body	div	p	pre	br	span,a	form	input,textarea	object	img
text-align	-	-	O	-	-	-	-	O	-	-
white-space	-	-	Δ	Δ	-	-	-	Δ	-	-
Extended properties										
pixel size	Δ	-	-	-	-	-	-	-	-	-
used-key-list	O	-	-	-	-	-	-	-	-	-
-wap-marquee-style	-	-	O	-	-	-	-	-	-	-
-wap-marquee-loop	-	-	O	-	-	-	-	-	-	-
-wap-marquee-dir	-	-	Δ	-	-	-	-	-	-	-
-wap-marquee-speed	-	-	O	-	-	-	-	-	-	-
-wap-input-format	-	-	-	-	-	-	-	O	-	-

7.8.2 Selector operation

The selector is used only in the contents of the default style sheet and in the CSS data transmitted by an independent resource, and only the ones shown below are operated.

- All elements Example: * { color: white }
- Type selector Example: div { color: white }
- Class selector Example: div.class { color: white }
  or .class { color: white }
- ID selector Example: #chap1 { color: white }

However, the batch assignment by combining 2 or more selectors cannot be done, except for usage other than the ones mentioned above. In addition, "," cannot be used to specify two or more selectors. Moreover, those selectors are applied only to the elements indicated as operated in Table 7-8.

7.8.3 Media type

The media type operated is only tv. Specifying the media type is made by @media rule in style sheet. Moreover, the receiver unit should ignore any specification of the properties within the @media rule that has any other type than tv.

7.8.4 Box model

The operation of the properties concerning the box model are shown below.

- margin

  This property is operated by a fixed value as provided in the following.

  Fixed value: 0

- padding-top, padding-right, padding-bottom, padding-left

  Values: <length> (integer value in the unit of px, or 0.)

  default : 0

  Inheritance : no

  For div element, object element, form element, and img element, operated as a fixed value 0.

- border-width

  This property is operated as below.

  Values: <length> (integer value in the unit of px, or 0.)

  default : 0

  Inheritance : no.

  All 4 sides are affected. For the object element and img element, operated as a fixed value 0.

- border-top-color, border-right-color, border-left-color, border-bottom-color

  These property are operated as below.

  Values: <color>|transparent | inherit

  default : transparent

  Inheritance : no

  The borders are rendered in the order of left,right,top,bottom. Each vertex is rendered again in rendering order of the borders. When these properties are not specified, these values are default (“transparent”), not the value of “color” property of the element.they are interpreted that it was not a color characteristic value of the foresaid element but the initial value (transparent). These properties use the same color unit as color property.

- border-style

  This property is operated as below.

  Values: dotted | dashed | solid | none

  default : none

  Inheritance : no

All 4 sides are affected. The display form of dotted, dashed, and solid conforms to the specification of the geometric in ARIB STD-B24, Vol. 1, Part 2, Chapter 8, 8.1. If not specified by the geometric, the form depends on the receiver unit.

Note, for img and object elements, the value is fixed as none in the operation.

7.8.5 Visual formatting model

Operation of the visual formatting model is shown as follows.

- display

This property has a different fixed value for each element as defined in the default style sheet.

div, body, input, textarea, img, object, form, and pre are fixed to block.

br, span, and a are fixed to inline.

Exceptionally, for p element, the value can be block | - wap-marquee. The default is block. And the value is not inherited.

Note 1)

meta, title, script, head, html, link, bml:bevent, and bml:beitem are fixed to none.

Note 2) The operation of p element for which -wap-marquee is specified is as follows.

- It only contains PCDATA, that should have no linefeed code (i.e. one line string).
- The maximum length of the above-mentioned PCDATA is 128 bytes. The excessive characters are trancated to display.
- The string is clipped according to the value of overflow (“hidden”), when the string is larger than the area to display specified with width and height Exceptionally, in the p element with –wap-marquee specified, clipping is not done, so the whole of the string is displayed as a line (not folded).

- position

This property has a different fixed value for each element as defined in the default style sheet. p, div, textarea, input, img, object, pre, and form are fixed to absolute.

br, span, and a are fixed to static.

It is not applied to elements other than the above.

- left, top, width, height

These properties are operated as below.

Values: <length> (integer value in the unit of px, or 0.)

default : 0

Inheritance : no

- z-index

  This property is operated as a fixed value defained as follows.

  Fixed value : auto

- line-height

  This property is operated as below..

  Values: normal | <length> (integer value in the unit of px, or 0.)| inherit

  default : normal

  Inheritance : no

  When normal is specified, the value is 1.0 times the font-size. But, for br, span, and a elements, the value is fixed as inherit in the operation. And the property can have the value “inherit” only for these elements.

7.8.6 Other visual effects

The operation of the characteristics concerning other visual effects is shown as follows.

- visibility

  This property is fixed as below in the operation.

  Values: visible | hidden | inherit

  default : inherit

  Inheritance : no

  When the visibility is set to “hidden” in the parent element, and “visible” in the child element, only the parent is hidden. the value is fixed as “inherit” for span and a elements in the operation. Also, the value is fixed as “visible” for body element.

- overflow

  This property is fixed as below in the operation.

  Fixed value : hidden

7.8.7 Colors and backgrounds

Color and background are defined as below in the operation.

- color

  This property is specified as below in the operation.

  Values: <color> | inherit

  Default : black

  Inheritance : yes

The color unit follows ARIB STD-B24 Vol. 2, Appendix 1, 4.4.1. There, only the six-digit notation of #rgb can be specified, so the notation of rgb(0.0%-100.0%,…) is prohibited. To specify by keywords, the aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow at the very least can be available

- background-color

  This property is specified as below in the operation.

  Values: <color> | transparent | inherit

  default : transparent

  Inheritance : no

  However, the values of object and img elements are fixed as transparent. Refer to “color” property for the specification of the color unit.

- background-image

  This property is specified as below in the operation.

  Values: <uri> | none

  default : none

  Inheritance : no

  The encoding of the image specified by the <uri> is jpeg, and the upper left corner of the body element is the rendering origin.

- background-repeat

  This property is fixed as below in the operation.

  Fixed value : repeat

7.8.8 Fonts

Table 7-9 defines the specification of the fonts in BML documents. One-byte character is displayed in the 1/2 width of the font size. Assignment of font type in Table 7-9 for the fonts of the receiver unit depend on the receiver unit. When a font of Gothic system is installed in the receiver unit, it is preferable to use aforesaid font.

Table 7-9 Used font

Font type	Font size
Maru gothic style	small, medium, large

The operation of the font properties is specified as below.

- font-family

  The value of the property is fixed as below in the operation.

Fixed value : "Maru gothic"

However, actual results of rendering depends on the receiver.

- font-size

This property is specified as below.

Values: small | medium | large | inherit

default : medium

Inheritance : yes

It should be noted when authoring, that there are terminals with same size of “small” and “medium”.

- font-weight

This property is specified as below.

Values: normal | bold | inherit

default : normal

Inheritance : yes

The rendering result of “bold” depends on the receiver unit.

7.8.9 Text

The properties of “Text” is specified in the operation as below.

- text-align

This property is specified as below in the operation.

Values: left | right | center

default : left

Inheritance :yes

- white-space

This property is fixed in the operation as below.

except for pre and textarea elements, that value is fixed as pre.

Fixed value : normal (pre, for pre and textarea element)

7.8.10 Default style sheets

The default style sheet is defined in appendix 4. This style sheet is assumed to be loaded virtually in every receiver unit and it does not need be transmitted.

7.8.11 length specification

The length must be specified with an integer value in the unit of pixel, e.g. "100px", etc. or “0” for 0px. For "0", DOM always returns “0px”.

7.9 extended property specification

The extended properties are specified as below in the operation.

- used-key-list

  This property is specified as below.operated as provisioned in the following. Refer to 7.3.1, for the valid key-group’s in the operation.

  default: <key-group>+ | none

  default : basic

  Inheritance : no

- resolution

  This property is fixed as below in the operation..

  Fixed value : 240x480

  Inheritance : no

- -wap-marquee-style

  This property is specified as below in the operation..

  Values: scroll | slide

  Applicable element: p element

  default : scroll

  Inheritance : no

  The animation specified by this property is repeated in “-wap-marquee-loop” times.

- -wap-marquee-loop

  This property is specified as below in the operation..

  Values: <integer>|infinite

  Applicable element: p element

  default : 1

  Inheritance : no

  The maximum value that can be specified is 16 for the finite times to loop.

  For the infinite times to loop, specify “infinite”. The receiver can be implemented with the “infinite” as an finite times to loop. Moreover, in “0”, the display stays as if it has looped in the specified times..

- -wap-marquee-dir

This property is fixed as below in the operation..

Fixed value : rtl

This property specifies the direction of the scrolling. The only rtl direction can be specified.

rtl: Scrolling from the right side to the left side

- -wap-marquee-speed

  This property is specified as below in the operation.
  Values: slow | normal | fast
  Applicable element: p element
  default : normal
  Inheritance : no
  This property specifies the scrolling speed. The actual speed for each value varies on the respective receiver unit.

- -wap-input-format

  This property is specified as below in the operation.
  Values: <format>
  Applicable element: input element and textarea element
  default : "*M"
  Inheritance : no

  - Each value of <format> is as below.

    A : Any uppercase alphabetical characters (symbols and punctuations included, numbers excluded)

    a : Any lowercase alphabetical characters (symboles and punctuation included, numbers excluded)

    N : Any numeric characters, only

    n : Numeric characters (symbols and punctuation included)

    X : Any uppercase alphabetical or numeric characters(symbols and punctuation included)

    x : Lowercase alphabetical or numeric characters (symbols and punctuation included)

    M : Any characters (set to uppercase characters as the initial value)

    m : Any characters (set to lowercase characters as the initial value)

  - With the above values, a number of the input character can be limited like "MMM" and "3M", etc. When limits are not set, "*" should be prepended to the value described above, e.g. "* M". However, it cannot be specified by a combination.
  - Refer to the ARIB STD-B24, Vol. 1, Part 2, Chapter 7, 7.3 "Shift-JIS character codes" for the characters that can be specified. The codes allocated from Row 90 to Row 94 of ARIB-STD-B5 Kanji set, tab (09) and delete character are excluded.
  - Criteria of classifying a character into a numeric, alphabetical, or symbolic character is implementation dependent.

- Receiver shall process appropriately on input of linefeed character (0x0D0A), because the “value” attribute of input elements shall not include linefeed characters although the linefeed character is not within the scope of control for the property.

7.10 Operations for procedural descriptions

7.10.1 Operations DOM

7.10.1.1 DOM Core Fundamental interfaces

Table 7-10 specifies the DOM Core Fundamental interfaces in the operation. In this operation, the mechanism that dynamically changes the tree structure of the document shall not be operated.

Table 7-10 DOM Core Fundamental interfaces

Interface	Operation
Fundamental interfaces	
DOMException	-
DOMImplementation	O
DocumentFragment	-
Document	O
Node	O
NodeList	-
NamedNodeMap	-
CharacterData	O
Attr	-
Element	O
Text	O
Comment	-
Extended interfaces	
CDATASection	O
DocumentType	-
Notation	-
Entity	-
EntityReference	-
ProcessingInstruction	-

The attributes of DOM Core Fundamental interfaces is specified in Table 7-11 in the operation.

Table 7-11 attributes and methods (DOM Core interfaces)

Interface	Attribute, method	Operation	Restrictions
DOMImplementation			
	hasFeature()	O	
Document			
	doctype	-	
	implementation	O	R

Interface	Attribute, method	Operation	Restrictions
	documentElement	O	R
	createElement()	-	
	createDocumentFragment()	-	
	createTextNode()	-	
	createComment()	-	
	createCDATASection()	-	
	createProcessingInstruction()	-	
	createAttribute()	-	
	createEntityReference()	-	
	getElementsByTagName()	-	
Node			
	nodeName	-	
	nodeValue	-	
	nodeType	-	
	parentNode	O	R
	childNodes	-	
	firstChild	O	R
	lastChild	O	R
	previousSibling	O	R
	nextSibling	O	R
	attrbutes	-	
	ownerDocument	-	
	insertBefore()	-	
	replaceChild()	-	
	removeChild()	-	
	appendChild()	-	
	hasChildNodes()	-	
	cloneNode()	-	
CharacterData			
	data	O	RW(Note)
	length	O	R(Note)
	substringData()	-	
	appendData()	-	
	insertData()	-	
	deleteData()	-	
	replaceData()	-	
Element			
	tagName	O	R
	getAttribute()	-	
	setAttribute()	-	
	removeAttribute()	-	
	getAttributeNode()	-	
	setAttributeNode()	-	
	removeAttributeNode()	-	
	getElementsByTagName()	-	
	normalize()	-	
Text			
	splitText()	-	
CDATASection			

(Note) The access to the child nodes of “script” or “textarea” elements should not be operated. only the child node of p, span, a, and pre can be written in the operation. One tab (0x09) or one white space (0x20) is counted as one character. Linefeed character (0x0D0A) is counted as 1 or 2 characters, although which to take depends on the receiver.

7.10.1.2 DOM HTML interfaces

- Operation of the interfaces applied to events

The interfaces applicable to events, belonged to the DOM HTML interfaces, is specified in the table below. Moreover, the values obtained by reading the “type” or “target” attribute in the BMLEvent interface is specifed in the latter table below according to the situation in which the access to the BMLEvent interface is done.

Table 7-12 attributes and methods (BML interfaces)

Interface name	Attribute or method name	Operation	Restriction
BMLEvent	type	O	R
	target	O	R
BMLIntrinsicEvent	keyCode	O	R
BMLBeventEvent:	status	O	R
	privateData	O	R
	esRef	O(Note 1)	R
	messageId	O	R
	messageVersion	O	R
	messageGroupId	O	R
	moduleRef	O(Note 2)	R
	object	O	R

(Note 1) The format of the value obtained by reading this attribute is /< component_tag > as the notation specified in ARIB STD-B24 Vol. 2, 9.2.

(Note 2) The format of the value obtained by reading this attribute is /<component_tag>/<module_id as the notation specified in ARIB STD-B24 Vol. 2, 9.2.

The response of the “type” attribute in the BMLEvent to the respective event is specifed in Table 7-13.

Table 7-13 Correspondence of the type attribute of the event and BMLEvent

Event	Value of type
Key specified in used-key-list is pushed down.	"keydown" (Note 3)
Key specified in used-key-list is released.	"keyup" (Note 3)
An Element is selected by pushing the “Enter” key or an access key.	"click" (Note 3)
in the focus	"focus"
out of the focus	"blur"
a document is loaded.	"load"
The advanced notice of unloading the document	"unload"
A change of the “value” attribute of an element is detected when the focus of the element blurs.	"change"
”submit” button of the form element is pushed or the “submit” method of the form element is called.	"submit"
Event message is received.	"EventMessageFired"

Event	Value of type
Module update is detected.	"ModuleUpdated"
Module is being locked.	"ModuleLocked"
Timer configured in beitem is fired.	"TimerFired"
presentation by monomedia decoder is terminated. (Note 2)	"MediaStopped"
Detected update of data_event_id.	"DataEventChanged"
The execution of a global code has been started or the function specified by executing setInterval() was invoked.	Undefined (Note 1)

Note 1) In this case, the target attribute is null.

Note 2) The monomedia that caused this event are audio/X-arib-mpeg2-aac.

Note 3) Whether the occurrence of the key event by the “Enter” key in the input or textarea element is receiver unit dependent. (Excludes input elements with a type attribute of "submit”.)

Table 7-14 execution of event handlers, and value of type attributes and target attributes

When the values are read			Value of type attribute	Value of target attribute	Occurs on receiving the broadcast?	Occurs on link status
onkeydown event handler being executed			"keydown"	Element specifying the handler	O	O
onkeyup event handler being executed			"keyup"	Element specifying the handler	O	O
onclick event handler being executed			"click"	Element specifying the handler	O	O
onsubmit event handler being executed			“submit”	Element specifying the handler	O	O
onload event handler being executed			“load”	Element specifying the handler	O	O
onunload event handler being executed			“unload”	Element specifying the handler	O	O
onchange event handler being executed			“change”	Element specifying the handler	O	O
onfocus event handler being executed			“focus”	Element specifying the handler	O	O
onblur event handler being executed			“blur”	Element specifying the handler	O	O
Onoccur event handler being executed	Type of the event	EventMessageFired	"EventMessageFired"	Element specifying the handler	O	O
		ModuleUpdated	"ModuleUpdated"		O	O
		ModuleLocked	"ModuleLocked"		O	X
		TimerFired	"TimerFired"		O	O
		DataEventChanged	"DataEventChanged"		O	X
		MediaStopped	"MediaStopped"		O	O
global code being executed			undefined	null	O	O
The procedural description specified by the call of setInterval() of the extended function for broadcasting being executed			undefined	null	O	O

- Operation of interfaces applied to the document node

  The interfaces applicable to the document node is specified in the table below, belonging to the DOM HTML interfaces or DOM BML interfaces.

Table 7-15 the interfaces applied to the document node

Interface name	Attribute or method name	Operation	Restriction
HTMLDocument	title	-	
	referrer	-	
	domain	-	
	URL	-	
	body	-	
	images	-	
	applets	-	
	links	-	
	forms	-	
	anchors	-	
	cookie	-	
	open()	-	
	close()	-	
	write()	-	
	writeln()	-	
	getElementById()	O	
	getElementsByName()	-	
BMLDocument	currentFocus	O	R
	currentEvent	O	R (Note)

(Note) The behavior when the value obtained by reading this attribute is assigned to a global variable is a receiver unit dependent.

- Operation of the interfaces applied to the element node

The operation of the interfaces applicable to the element node is specified in the table, belonging to the DOM HTML interfaces.

Table 7-16 the interfaces applied to the element node

Element by the node	Interface applied	Attribute or method	Operation	Restriction
html	HTMLElement	id	-	
		className	-	
		title	-	
		lang	-	
		dir	-	
	HTMLHtmlElement	version	-	
bml:bevent	HTMLElement	id	O	R
		className	-	
		title	-	
		lang	-	
		dir	-	
	BMLBeventElement			

Element by the node	Interface applied	Attribute or method	Operation	Restriction
bml:beitem	HTMLElement	id	O	R
		className	-	
		title	-	
		lang	-	
		dir	-	
	BMLBeitemElement	type	O	R
		esRef	O	RW
		messageGroupId	O	R
		messageId	O	RW
		messageVersion	O	RW
		moduleRef	O	RW
		timeMode	O	R
		timeValue	O	RW
		objectId	O	R
		subscribe	O	RW
body	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLBodyElement			
	BMLBodyElement	invisible	-	
		style	-	
		normalStyle	O	RW (Note 2)
		focusStyle	-	
		activeStyle	-	
head	HTMLElement	id	-	
		className	-	
		title	-	
		lang	-	
		dir	-	
	HTMLHeadElement	profile	-	
title	HTMLElement	id	-	
		className	-	
		title	-	
		lang	-	
		dir	-	
	HTMLTitleElement	text	O	R

Element by the node	Interface applied	Attribute or method	Operation	Restriction
div	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLDivElement			
	BMLDivElement	style	-	
		normalStyle	O	RW (Note 2)
		focusStyle	-	
		activeStyle	-	
		accessKey	-	
		focus()	-	
		blur()	-	
p	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLParagraphElement			
	BMLParagraphElement	style	-	
		normalStyle	O	RW (Note 2)
		focusStyle	-	
		activeStyle	-	
		accessKey	-	
		focus()	-	
		blur()	-	
pre	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLPreElement			
	BMLPreElement	style	-	
		normalStyle	O	RW (Note 2)
		focusStyle	-	
		activeStyle	-	

Element by the node	Interface applied	Attribute or method	Operation	Restriction
br	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLBRElement			
	BMLBRElement	style	-	
		normalStyle	O	RW (Note 2)
		focusStyle	-	
		activeStyle	-	
span	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLSpanElement	style	-	
		normalStyle	O	RW (Note 2)
		focusStyle	-	
		activeStyle	-	
		accessKey	-	
		focus()	-	
		blur()	-	

Element by the node	Interface applied	Attribute or method	Operation	Restriction
a	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLAnchorElement	accessKey	O	R
		charset	-	
		coords	-	
		href	O	RW
		hreflang	-	
		name	-	
		rel	-	
		rev	-	
		shape	-	
		tabIndex	-	
		target	-	
		type	-	
		blur()	-	
		focus()	-	
	BMLAnchorElement	style	-	
		normalStyle	O	RW (Note 2)
		focusStyle	-	
		activeStyle	-	
		effect	-	
form	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	

Element by the node	Interface applied	Attribute or method	Operation	Restriction
	HTMLFormElement	elements	-	
		length	-	
		name	-	
		acceptCharset	-	
		action	O	RW
		enctype	-	
		method	O	R
		submit()	O	
		reset()	-	
	BMLFormElement	style	-	
		normalStyle	O	RW (Note 2)
		focusStyle	-	
		activeStyle	-	
		accept	-	
input	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLInputElement	defaultValue	O	R
		defaultChecked	-	
		form	O	R
		accept	-	
		accessKey	O	R
		alt	-	
		checked	-	
		disabled	-	
		maxLength	O	R
		name	O	R
		readOnly	O	RW
		size	-	
		src	-	
		tabIndex	-	
		type	O	R
		useMap	-	
		value	O	RW
		blur()	-	
		focus()	-	
		select()	-	
		click()	-	
	BMLInputElement	style	-	
		normalStyle	O	RW
		focusStyle	-	
		activeStyle	-	

Element by the node	Interface applied	Attribute or method	Operation	Restriction
textarea	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLTextAreaElement	defaultValue	O	R
		form	O	R
		accessKey	O	R
		cols	-	
		disabled	-	
		name	O	R
		readOnly	O	RW
		rows	-	
		tabIndex	-	
		value	O	RW
		blur()	-	
		focus()	-	
		select()	-	
	BMLTextAreaElement	style	-	
		normalStyle	O	RW
		focusStyle	-	
		activeStyle	-	
img	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLImageElement	lowSrc	-	
		name	-	
		alt	O	R
		height	-	
		isMap	-	
		longDesc	-	
		src	O	RW (Note)
		useMap	-	
		width	-	
	BMLImageElement	style	-	
		normalStyle	O	RW (Note 2)
		focusStyle	-	
		activeStyle	-	
object	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	

Element by the node	Interface applied	Attribute or method	Operation	Restriction
	HTMLObjectElement	form	-	
		code	-	
		archive	-	
		codeBase	-	
		codeType	-	
		data	O	RW(Note)
		declare	-	
		height	-	
		name	-	
		standby	-	
		tabIndex	-	
		type	O	R
		useMap	-	
		width	-	
	BMLObjectElement	style	-	
		normalStyle	O	RW (Note 2)
		focusStyle	-	
		activeStyle	-	
		classId	-	
		accessKey	O	R
		remain	-	
		streamPosition	-	
		streamStatus	O	RW
		streamLooping	-	
		streamSpeedNumerator	-	
		streamSpeedDenominator	-	
		streamLevel	-	
		setSpeed()	-	
		movePosition()	-	
		hasAssociatedIndex()	-	
		assignToLocalEvnet()	-	
		assignToNodePlayMode()	-	
		getMainAudioStream()	-	
		setMainAudioStream()	-	
		focus()	-	
		blur()	-	

Element by the node	Interface applied	Attribute or method	Operation	Restriction
meta	HTMLElement	id	-	
		className	-	
		title	-	
		lang	-	
		dir	-	
	HTMLMetaElement	content	O	R
		httpEquiv	-	
		name	O	R
		scheme	-	
script	HTMLElement	id	-	
		className	-	
		title	-	
		lang	-	
		dir	-	
	HTMLScriptElement	text	-	
		htmlFor	-	
		event	-	
		charset	-	
		defer	-	
		src	-	
		type	-	
link	HTMLElement	id	-	
		className	-	
		title	-	
		lang	-	
		dir	-	
	HTMLLinkElement	charset	-	
		href	-	
		hreflang	-	
		type	-	
		rel	-	
		rev	-	
		target	-	
		media	-	

(Note) For the resulting display of the img/object element in which the updated area displayed by the new resource doesn't match with the origianl area displayed by the original resource when the resource to be presented is updated, see section 8.1.8.5. In the img element, the media type of the resources to be presented must be same as before when update.

(Note 2) These attributes are accessed through the BMLCSS2Properties interfaces. When these attributes are read directly, the object is returned. Moreover, it is prohibited to write directly.

7.10.1.3 BMLCSS2Properties interface

The operation of BMLCSS2Properties interface used in Table 7-17 is shown.

Table 7-17 Operation of BMLCSS2Properties interface

Attribute	Operation	Remarks
Box model		
marginTop	-	
marginRight	-	
marginBottom	-	
marginLeft	-	
margin	-	
paddingTop	O	R
paddingRight	O	R
paddingBottom	O	R
paddingLeft	O	R
padding	-	
borderTopWidth	-	
borderRightWidth	-	
borderBottomWidth	-	
borderLeftWidth	-	
borderWidth	O	R
borderTopColor	O	RW
borderRightColor	O	RW
borderBottomColor	O	RW
borderLeftColor	O	RW
borderColor	-	
borderTopStyle	-	
borderRightStyle	-	
borderBottomStyle	-	
borderLeftStyle	-	
borderStyle	O	R
borderTop	-	
borderRight	-	
borderBottom	-	
borderLeft	-	
border	-	
Reception disturbance model		
position	-	
left	O	R
top	O	R
width	O	R
height	O	R
zIndex	-	
lineHeight	-	
verticalAlign	-	
display	-	
bottom	-	
right	-	
cssFloat	-	
clear	-	
direction	-	
unicodeBidi	-	
maxHeight	-	
minHeight	-	
maxWidth	-	
minWidth	-	
Other visual effects		
visibility	O	RW
overflow	-	
clip	-	
Generation contents/ automatic number appearance/list		
content	-	
quotes	-	
counterReset	-	
counterIncrement	-	
markerOffset	-	
listStyleType	-	
listStyleImage	-	
listStylePosition	-	
listStyle	-	
Paged media		
size	-	
marks	-	
pageBreakBefore	-	
pageBreakAfter	-	
pageBreakInside	-	
page	-	
orphans	-	
widows	-	
Background		
background	-	
backgroundColor	O	RW
backgroundImage	-	
backgroundRepeat	-	
backgroundPosition	-	
backgroundAttachment	-	
Fonts		
color	O	RW
fontFamily	-	
fontStyle	-	
fontSize	-	
fontVariant	-	
fontWeight	O	RW
font	-	
fontStretch	-	
fontSizeAdjust	-	
Text		
textIndent	-	
textAlign	O	R
textDecoration	-	
textShadow	-	

Attribute	Operation	Remarks
letterSpacing	-	
wordSpacing	-	
textTransform	-	
whiteSpace	-	
Table related		
captionSide	-	
borderCollapse	-	
borderSpacing	-	
tableLayout	-	
emptyCells	-	
speakHeader	-	
User interface		
outlineColor	-	
outlineStyle	-	
outlineWidth	-	
outline	-	
cursor	-	
Voice style sheet		
volume	-	
speak	-	
pauseBefore	-	
pauseAfter	-	
pause	-	
cueBefore	-	
cueAfter	-	
cue	-	
playDuring	-	
azimuth	-	
elevation	-	
speechRate	-	
voiceFamily	-	
pitch	-	
pitchRange	-	
stress	-	
richness	-	
speakPunctuation	-	
speakNumeral	-	
BML extention		
borderColorIndex	-	
borderTopColorIndex	-	
borderRightColorIndex	-	
borderrLeftColorIndex	-	
borderBottomColorIndex	-	
backgroundColorIndex	-	
colorIndex	-	
grayscaleColorIndex	-	
outlineColorIndex	-	
clut	-	
pixel size	-	
displayAspectRatio	-	
navIndex	-	
navUp	-	
navDown	-	
navLeft	-	
navRight	-	
usedKeyList	O	RW
WapMarqueeStyle	O	R
WapMarqueeLoop	O	R
WapMarqueeDir	-	
WapMarqueeSpeed	O	R
WapAccesskey	-	
WapInputFormat	O	R(Note 1)
WapInputRequired	-	

(Note 1) When this characteristic is read, returns the character string as defined.

The value of the DOM attribute operated as mentioned above follows the operation of the CSS2 properties.

7.10.2 Operation area of built-in objects

Operation of ECMAScript built-in objects is shown in Table 7-18.

Table 7-18 Operation of ECMAScript embedded objects

Embedded object	Method and property	Operation	Remarks
(global)			
	NaN	O	
	Infinity	-	Note 1)
	eval(x)	-	
	parseInt(string, radix)	O	Note 7)
	parseFloat(string)	-	Note 1)
	escape(string)	-	Note 2)
	unescape(string)	-	Note 2)
	isNaN(number)	O	
	isFinite(number)	-	Note 1)
Object		All O	
	prototype	O	
	Object([value])	O	
	new Object([value])	O	
Object.prototype		All O	
	constructor	O	
	toString()	O	
	valueOf()	O	
Function			
	prototype	O	
	length	O	
	Function(p1, p2, . . . , pn, body)	-	
	new Function(p1, p2, . . . , pn, body)	-	
Function.prototype		All O	
	constructor	O	
	toString()	O	Note 3)
Array		All O	
	prototype	O	
	length	O	
	Array(item0, item1, . . .)	O	
	new Array(item0, item1, . . .)	O	
	new Array([len])	O	
Array.prototype		All O	
	constructor	O	
	toString()	O	
	join([separator])	O	
	reverse()	O	
	sort([comparefn])	O	
String		All O	
	prototype	O	

Embedded object	Method and property	Operation	Remarks
	length	O	
	String([value])	O	
	new String([value])	O	
	String.fromCharCode(char0[, char1, . . .])	O	
String.prototype		All O	
	constructor	O	
	toString()	O	
	valueOf()	O	
	charAt(pos)	O	
	charCodeAt(pos)	O	
	indexOf(searchString, position)	O	
	lastIndexOf(searchString, position)	O	
	split(separator)	O	
	substring(start [,end])	O	
	toLowerCase()	O	
	toUpperCase()	O	
Boolean		All O	
	prototype	O	
	Boolean([value])	O	
	new Boolean([value])	O	
Boolean.prototype		All O	
	constructor	O	
	toString()	O	
	valueOf()	O	
Number			
	prototype	O	
	MAX_VALUE	O	
	MIN_VALUE	O	
	NaN	O	
	NEGATIVE_INFINITY	-	Note 1)
	POSITIVE_INFINITY	-	Note 1)
	Number([value])	O	
	new Number([value])	O	
Number.prototype		All O	
	constructor	O	
	toString([radix])	O	
	valueOf()	O	
Math		All -	
	E	-	
	LN10	-	
	LN2	-	
	LOG2E	-	
	LOG10E	-	
	PI	-	
	SQRT1_2	-	
	SQRT2	-	
	abs(x)	-	
	acos(x)	-	
	asin(x)	-	
	atan(x)	-	
	atan2(y, x)	-	
	cos(x)	-	
	exp(x)	-	

Embedded object	Method and property	Operation	Remarks
	floor(x)	-	
	log(x)	-	
	max(x, y)	-	
	min(x, y)	-	
	pow(x, y)	-	
	random()	-	
	round(x)	-	
	sin(x)	-	
	sqrt(x)	-	
	tan(x)	-	
Date			
	prototype	O	
	Date([year , month [, date [, hours [, minutes [, seconds [, ms]]]]]])	O	
	new Date([year , month [, date [, hours [, minutes [, seconds [, ms]]]]]])	O	
	Date(value)	-	Note 4)
	new Date(value)	-	Note 4)
	Date.parse(string)	-	Note 4)
	Date.UTC([year [, month [, date [, hours [, minutes [, seconds [, ms]]]]]]])	-	Note 4)
Date.prototype			
	contructor	O	
	toString()	O	Note 3)
	valueOf()	-	Note 4)
	getTime()	-	Note 4)
	getYear()	-	Note 5)
	getFullYear()	O	
	getUTCFullYear()	O	
	getMonth()	O	
	getUTCMonth()	O	
	getDate()	O	
	getUTCDate()	O	
	getDay()	O	
	getUTCDay()	O	
	getHours()	O	
	getUTCHours()	O	
	getMinutes()	O	
	getUTCMinutes()	O	
	getSeconds()	O	
	getUTCSeconds()	O	
	getMilliseconds()	O	
	getUTCMilliseconds()	O	
	getTimezoneOffset()	O	
	setTime(time)	-	Note 4)
	setMilliseconds(ms)	O	Note 6)
	setUTCMilliseconds(ms)	O	Note 6)
	setSeconds(sec [, ms])	O	Note 6)
	setUTCSeconds(sec [, ms])	O	Note 6)
	setMinutes(min [, sec [, ms]])	O	Note 6)
	setUTCMinutes(min [, sec [, ms]])	O	Note 6)
	setHours(hour [, min [, sec [, ms]]])	O	Note 6)
	setUTCHours(hour [, min [, sec [, ms]]])	O	Note 6)
	setDate(date)	O	Note 6)

Embedded object	Method and property	Operation	Remarks
	setMonth(mon [, date])	O	Note 6)
	setUTCMonth(mon [, date])	O	Note 6)
	setFullYear(year [, mon [, date]])	O	Note 6)
	setUTCFullYear(year [, mon [, date]])	O	Note 6)
	setYear(year)	-	Note 5)
	toLocaleString()	O	Note 3)
	toUTCString()	O	Note 3)
	toGMTString()	-	Note 5)

Note 1) Prohibit to use for Float related objects.
Note 2) Prohibit to use for Unicode related objects.
Note 3) Operation is specified in ARIB STD-B24, Vol. 2, Appendix 2.
Note 4) Prohibit to use for Number related objects.
Note 5) Prohibit to use, as this is for the interchangeability with an old source code.
Note 6) Operated under restricted specifications when it is Number related objects.
Note 7) radix of parseInt() is 8, 10, 16 (same as when 0 is specified as 10).

7.10.3 Operation area of extension objects for broadcasting

Table 7-19 shows the operation of extension objects for broadcasting ECMAScript. Operation of the defined extension object for broadcasting not described in Table 7-19 is assumed to be "-".

Table 7-19 Operation of extension objects for broadcasting

Object	Method and property	Operation	Remarks
CSVTable			
	prototype	-	
	new CSVTable()	-	
CSVTable.prototype			
	constructor	-	
	close()	-	
	toString()	-	
	toNumber()	-	
	toArray()	-	
	search()	-	
BinaryTable			
	prototype	O	
	new Binarytable()	O	
BinaryTable.prototype			
	constructor	O	
	close()	O	
	toString()	O	
	toNumber()	O	
	toArray()	O	
	search()	O	

7.10.4 Operation area of Navigator pseudo-objects

Navigator pseudo-objects are not operated.

7.10.5 Operation area of browser pseudo-objects

Operated as shown in Table 7-20.

The meaning of the "Operation" column is as follows.

“O” A basic function in these provisions.

“O (*1)” An optional feature in these provisions. Therefore, when these functions are used by contents, the aforementioned function is called only when processing is possible by inspecting the processing function of the aforementioned function in the receiver unit by the getBrowserSupport() function.

“O (*2)” Installation of these functions is not necessary only when receiver units that can not use the telecommunication function from a BML browser and when the function is called, returns a failure for the return value, although it is a basic function in these provisions, as a rule.

“O (*3)” It is necessary in receiver units with a function to playback the partial TS. When these functions are used by contents, the aforementioned function is called only when processing is possible by inspecting the processing function of aforementioned function in the receiver unit by the getBrowserSupport() function.

“-” Not a basic function nor an optional function in these provisions. An error will occur in the receiver unit when the aforementioned function is called.

Table 7-20 Operation area of browser pseudo-objects

	Function	Operation	Observation
Ureg related function			
	Ureg[]	O	
Greg related function			
	Greg[]	O	Refer to 8.1.15.2.
EPG functions			
	epgGetEventStartTime()	O	
	epgGetEventDuration()	O	
	epgTune()	-	
	epgTuneToComponent()	-	
	epgTuneToDocument()	-	
	epgIsReserved()	-	
	epgReserve()	O(*1)	
	epgCancelReservation()	O(*1)	
	epgRecIsReserved()	-	
	epgRecReserve()	-	
	epgRecCancelReservation()	-	
Event group index functions			
	grpIsReserved()	-	

	Function	Operation	Observation
	grpReserve()	-	
	grpCancelReservation()	-	
	grpRecIsReserved()	-	
	grpRecReserve()	-	
	grpRecCancelReservation()	-	
	grpGetNodeEventList()	-	
	grpGetERTNodeName()	-	
	grpGetERTNodeDescription()	-	
	epgXTune()	-	
Series reservation functions			
	seriesIsReserved()	-	
	seriesReserve()	-	
	seriesCancelReservation()	-	
	seriesRecIsReserved()	-	
	seriesRecReserve()	-	
	seriesRecCancelReservation()	-	
Non-volatile memory functionss			
	readPersistentString()	-	
	readPersistentNumber()	-	
	readPersistentArray()	O	
	writePersistentString()	-	
	writePersistentNumber()	-	
	writePersistentArray()	O	
	copyPersistent()	-	
	getPersistentInfoList()	-	
	deletePersistent()	-	
	getFreeSpace()	-	
Functions for controlling access-controlled non-volatile memory areas			
	isSupportedPersistentType()	-	
	setAccessInfoOfPersistentArray()	-	
	checkAccessInfoOfPersistentArray()	-	
	writePersistentArrayWithAccessCheck()	-	
	readPersistentArrayWithAccessCheck()	-	
Interaction channel communication			
Interaction channel communication--delay calls			
	registerTransmission()	-	
	registerTransmissionStatus()	-	
	getTransmissionStatus()	-	
	setDelayedTransmissionDataOverBasic()	-	
Interaction channel communication--BASIC procedure			
	connect()	-	
	disconnect()	-	
	sendBinaryData()	-	
	receiveBinaryData()	-	
	sendTextData()	-	
	receiveTextData()	-	
Interaction channel communication--TCP/IP			
	setISPParams()	-	
	getISPParams()	-	
	connectPPP()	-	
	connectPPPWithISPParams()	-	
	disconnectPPP()	-	
	getConnectionType()	O(*2)	
	isIPConnected()	O(*2)	
	saveHttpServerFileAs()	-	
	saveHttpServerFile()	-	
	sendHttpServerFileAs()	-	

	Function	Operation	Observation
	saveFtpServerFileAs()	-	
	saveFtpServerFile()	-	
	sendFtpServerFileAs()	-	
	sendTextMail()	-	
	sendMIMEMail()	-	
	transmitTextDataOverIP()	O(*2)	
	setDelayedTransmissionData()	-	
	getTransmissionStatus()	-	
	getTransmissionResult()	-	
	setCacheResourceoverIP()	-	
Interaction channel function--Acquisition function in common state of delay calls to BASIC system procedure and IP connection			
	getDelayedTransmissionStatus()	-	
	getDelayedTransmissionResult()	-	
Interaction channel function--Function to acquire line connection state			
	getPrefixNumber()	-	
Interaction channel function--large amount of call receipt service			
	vote()	-	
Interaction channel function--Encrypted communication using CAS			
	startCASEncryption()	-	
	transmitWithCASEncryption()	-	
	endCASEncryption()	-	
Interaction channel function--Communication by secret key cryptosystem without using CAS			
	setEncryptionKey()	-	
	beginEncryption()	-	
	endEncryption()	-	
Operation control function			
	reloadActiveDocument()	O	
	getNPT()	-	
	getProgramRelativeTime()	O	
	isBeingBroadcast()	-	
	lockExecution()	-	
	unlockExecution()	-	
	lockModuleOnMemory()	-	
	unlockModuleOnMemory()	-	
	setCachePriority()	-	
	getTuningLinkageSource()	-	
	getTuningLinkageType()	-	
	getLinkSourceServiceStr()	-	
	getLinkSourceEventStr()	-	
	getIRDID()	-	
	getBrowserVersion()	O	
	getProgramID()	O	
	getActiveDocument()	O	
	lockScreen()	O	
	unlockScreen()	O	
	getBrowserSupport()	O	
	launchDocument()	O	
	launchDocumentRestricted()	-	
	quitDocument()	O	
	launchExApp()	-	
	getFreeContentsMemory()	-	
	isSupportedMedia()	-	

	Function	Operation	Observation
	detectComponent()	O	
	lockModuleOnMemoryEx()	O	
	unlockModuleOnMemoryEx()	O	
	unlockAllModulesOnMemory()	O	
	getLockedModuleInfo()	O	
	getBrowserStatus()	O	
	getResidentAppVersion()	O	
	isRootCertificateExisting()	-	
	getRootCertificateInfo()	-	
	startResidentApp()	-	
	getDataDisplayAreaSize()	-	
	setFullDataDisplayArea()	O(*1)	
Receiver unit audio control			
	playRomSound()	O	
Timer function			
	sleep()	-	
	setTimeout()	-	
	setInterval()	O	
	clearTimer()	O	
	pauseTimer()	-	
	resumeTimer()	-	
	setCurrentDateMode()	O(*3)	
External character function			
	loadDRCS()	-	
	unloadDRCS()	-	
External device control function			
	enumPeripherals()	-	
	passXMLDocToPeripheral()	-	
Other function			
	random()	O	
	subDate()	O	
	addDate()	O	
	formatNumber()	O	
Caption display control function			
	setCCStreamReference()	-	
	getCCStreamReference()	-	
	setCCDisplayStatus()	-	
	getCCDisplayStatus()	-	
	getCCLanguageStatus()	-	
Directory control function			
	saveDir()	-	
	saveDirAs()	-	
	createDir()	-	
	getParentDirName()	-	
	getDirNames()	-	
	isDirExisting()	-	
File control function			
	saveFile()	-	
	saveFileAs()	-	
	getFileNames()	-	
	isFileExisting()	-	
File input-output function			
	writeArray()	-	
	readArray()	-	
Enquiry function			
	getDirInfo()	-	
	getFileInfo()	-	
	getCarouselInfo()	-	
	getModuleInfo()	-	

	Function	Operation	Observation
	getContentSource()	-	
	getStorageInfo()	-	
Data carousel accumulation function			
	saveCarouselAs()	-	
	saveCarousel()	-	
	saveModuleAs()	-	
	saveModule()	O(*1)	Detailed specifications in this volume are T.B.D
	saveResourceAs()	-	
	saveResource()	O(*1)	Detailed specifications in this volume are T.B.D
Bookmark control function			
	writeBookmarkArray()	-	
	readBookmarkArray()	-	
	deleteBookmark()	-	
	lockBookmark()	-	
	unlockBookmark()	-	
	getBookmarkInfo()	-	
	getBookmarkInfo2()	-	
	startResidentBookmarkList()	-	
Printing related function API – Print basic function			
	getPrinterStatus()	-	
	printFile()	-	
	printTemplate()	-	
	printUri()	-	
	printStaticScreen()	-	
Printing related function API – Memory card related			
	saveImageToMemoryCard()	O(*1)	
	saveHttpServerImageToMemoryCard()	O (*1)	
	saveStaticScreenToMemoryCard()	-	
Digital terrestrial broadcasting inherent function			
	X_DPA_mailTo()	O (*1)	
	X_DPA_startResidentApp()	O	
	X_DPA_phoneTo()	O (*1)	
	X_DPA_getRcvCond ()	O (*1)	
	X_DPA_getCurPos ()	O (*1)	
	X_DPA_saveExAppFile ()	O (*1)	Detailed specifications in this volume are T.B.D
	X_DPA_startExAV()	O (*1)	
	X_DPA_stopExAV()	O (*1)	
	X_DPA_tuneWithRF()	O (*1)	
	X_DPA_writeSchInfo()	O (*1)	
	X_DPA_getComBrowserUA ()	O (*2)	
	X_DPA_writeAddressBookInfo()	O (*1)	
	X_DPA_launchDocWithLink()	O (*2)	
	X_DPA_chkAVtype()	O (*1)	
	X_DPA_getIRDID()	O	
	X_DPA_writeCproBM()	O (*2)	

7.10.5.1 Operational rule of each extended function

- Operational rule of lockScreen()

  When using lockScreen(), take into consideration that unlockScreen() is done in contents afterwards.

- Operational rule of unlockScreen()

  When using lockScreen(), take into consideration that unlockScreen() is done in contents afterwards.

- Operational rule of getActiveDocument()

  When the function is called for the BML document being changed by the URI specified by fragment, returns URI as a return value excluding the fragment specification.

- Operational rule of reloadActiveDocument()

  Does not repeat the call of this function by contents automatically when connection to communication is not possible. Re-reading is carried out assuming that the URI excluding fragment specification are specified when the aforementioned function is called for the BML document that has changed by the fragment specification.

- Operational rule of setInterval()

  It is preferable to avoid the usage when the receiver has to bear a remarkably heavy load for processing when this function is used. Moreover, the number of timers that can be set at the same time is a maximum of 4.

- Operational rule of X_DPA_launchDocWithLink()

  This function can be used only in a linked state. Moreover, using this function can change the base URI directory. Therefore, the broadcaster should operate with maximum attention when describing the URI of the link contents transition destination so that the link state is not improperly set.

  Basically, the transition within the link contents uses the href attribute of launchDocumet() and a element, and when there is a requirement to change the base URI directory, use of this function is preferable.

7.10.5.2 Operation guidelines of function that generate communication

Assume the description of contents from which a corresponding function is executed by user operation, when using function that generate communication.

7.10.6 Extended functions provided by digital terrestrial broadcasting (1)

- getResidentAppVersion() : Receiver's native application recognition function

Refer to "[Section 2] 5.9.6 Extended functions that have been added in the digital terrestrial broadcasting" for the definition of the function. In Array[1] however, returns the character string of 20 characters or less that uses byte range between 21 to 7F of one byte character of the Shift JIS code set (refer to ARIB STD-B24 Vol. 2, Appendix 4, 4.1.1) provided arbitrarily by each manufacturer. Moreover, refer to Table 7-21 for the value that can be specified for argument appName. However, ComBrowser is not specified for appName.

7.10.7 Extended function provided by digital terrestrial broadcasting (2)

- X_DPA_mailTo(): Text mail transmission by mail applications of receivers.

Syntax:

```
Number   X_DPA_mailTo(
             input String subject,
             input String body,
             input String toAddress
)
```

Argument:

subject	Title of send mail
body	Text of send mail
toAddress	Destination address of send mail

Return value:

1:	Success
NaN:	Failure

Explanation:

Delivers the address and text to the e-mail application that the receiver possesses as a receiver features, and a mail is sent by e-mail application.

The difference to sendTextMail() is text is delivered to the e-mail application without directly establishing a session by the SMTP protocol. The one that actually transmits the mail is by the e-mail application function (For instance, the transmission button is selected), and the data broadcasting contents is not involved in the transmission timing.

It is preferable that the data broadcasting contents continue while the e-mail application is in

operation, and returns to the BML browser presentation state when the e-mail application ends.

It fails when the contents of the parameters, like the format etc., of the mail address are improper.

The maximum size of each argument is as follows.

body: 500bytes

subject: 30bytes

toAddress: 50bytes

For the contents of the parameters specified that exceed the maximum size, although there is a possibility of rounding down the part that exceeds the maximum size, it is preferable to not fail as a function. Meanwhile, the transmission of mail exceeding the maximum size can be done depending on the e-mail application capability.

- X_DPA_startResidentApp(): Receiver's native application initialization function

Syntax:

```
Number X_DPA_startResidentApp(
        input String appName,
        input Number showAV,
        input String returnURI
        (, input String Ex_info)+
)
```

Argument:

appName	Receiver's native application for initialization
showAV	Flag that determines whether to continue the TV image audio playback currently being presented even after receiver's native application starts. However, it is not an error, but is considered as 1, when the values other than the following are specified. 1: Playback continuation of the TV (video and audio)is permitted 0: Playback continuation of the TV (video and audio) is prohibited
returnURI	It is URI of the component presented first, when the receiver's native application started by this function ends and the BML browser re-starts. In particular, when not specified, it is a null character string. It is not necessary for the receiver unit to work according to the specification of this argument, as this argument is only hint information for the receiver unit.

Ex_info Character string that shows supplementary information related to the initialization of receiver's native applications.

Return value:

1: Success

NaN: Failure

Explanation:

The receiver's native application specified with appName is initialized. When the receiver's native application specified with appName doesn't exist, NaN is returned.

It is assumed that contents switch by acquiring the type of communication browser installed in the receiver unit using X_DPA_getComBrowserUA() when a communication browser is started, and by changing to the appropriate URI. The operation follows the specifications in section 7.10.8, for whether to end the BML browser by starting the receiver unit built-in application specified with appName function.

The combination of assignable values for the arguments appName, showAV, and Ex_info are as follows. However, appName, showAV and Ex_info other than shown below shall require registration and they shall be managed by the governing organization described in Appendix 13.

Table 7-21 Value addressable in appName /showAV/Ex_info

appName	showAV	Ex_info
ComBrowser	Operated as defined	The fourth argument: URI character string handed over to a communication browser The fifth argument: Specification of an ideal display browser (hint information) Note 1) A browser that can display contents based on this specification in this volume: 0 A carrier specification browser : 1 HTML browser: 2 The sixth argument: A browser is displayed on all screens: 1 Note 2)
BookmarkList	1 only	None

Note 1) This describes the ideal browser to be used to display the communication contents of the transition destination.

Note 2) As a rule, the display of a data broadcasting browser is ended, when one is specified. At this time, the display size of ComBrowser is implementatiion dependent. However, it complies with the restriction of section 8.1.4.2 when leaving the display of a BML browser according to the condition of non-display of ComBrowser on all screens and indicated in section 8.1.4.2. When values other than the provisioned are specified, the operation is the

same as in the case where 1 is specified. Moreover, when omitted, it is an error.

- X_DPA_phoneTo (): Make calls by the specified number. " mobile phone native features"

Syntax:

Number X_DPA_phoneTo(
input String phone_number
)

Argument:

phone_number Phone number

Return value:

1: Success
NaN: Failure

Explanation:

Makes calls to the specified telephone number. It is preferable to continue the presentation of data even after this function is called.

The character string that can be specified for the argument is the following 1 byte character.

"0" to "9"
"#" "*"
"(" "-" ")"
"P"

Among these, three characters of "(","-",")" are used for viewability of the notation, and should be ignored without considering it an error.

When the character string including characters other than the above-mentioned are passed, dialing should not operate, and it is an error.

When "P" is specified, it is preferable to stop dialing during a fixed time of the receiver unit (pose).

When the pose is not done, "P" is ignored (posed time 0 seconds).

Handling of dialing "#" and "*", pose time of "P", restoring method to the dial operation after pose, and etc. is an implementation dependent.

Example:

When specified "(03)1234-5678P910" in the argument

Dial first "0312345678", then dial "910" after the fixed time passes.

- X_DPA_getRcvCond (): Acquires the communication state in the telecommunication lines

Syntax:

Number X_DPA_getRcvCond ()

Argument:

None

Return value:

0: No service
1: Low reception state
2: Medium reception state
3: Excellent reception state
NaN Receiver errors other than the mentioned above

Explanation:

The communication status in the telecommunication line is acquired. When 0 is returned, the communication status of no service is shown. Moreover, take it into consideration that there are receivers that don't display any service even in the communication restricted status, and this function shall return 0 in the state of the call restriction. Which status is allocated into each return value, when the division of the return value and the communication status in the receiver are different is an implementation dependent.

- X_DPA_getCurPos (): Obtain receiver geographical position.

Syntax:

Array X_DPA_getCurPos(
[,input String posInfo]
)

Argument:

posInfo: Character string that indicates positioning method of location information and return value form
"GPS": GPS measurement first
"CB" : Base station, (cell base) positioning priority

Return value:

Array [0]:
"1": In case of returning by the latitude and longitude of the world positioning system (decimal number code)
"2": In case of returning by other geodetic information
Array [1]: Character string 1 showing present location (longitude, etc.)
Array [2]: Character string 2 showing present location (latitude, etc.)

Array [3]:	Character string 1 showing supplementary information (positioning system and coordinate system, etc.)
Array [4]:	Character string 2 showing supplementary information (error information on positioning result)
null:	Failure

Explanation:

The geographical location information of the receiver is returned by using information, etc., obtained at the base station established by GPS (Global Positioning System) information using GPS satellites and mobile phones based on the location information acquiring function of receivers. It is preferable that the return value of this function returns the latitude and longitude in decimal number notation based on the world geographic coordinate system.

The positioning method is specified for argument (posInfo). When “GPS” is specified for the argument, it gives priority to information of the positioning method of GPS measurement and related to it, and returns it as a return value. When “GPS” is specified for the argument, and for receivers that do not correspond to GPS measurement, the return value of other positioning methods is returned. Similarly, it returns return values in priority to each cell base measurement information by the reception base station such as mobile phones, in case “CB” is specified and returns return values of other positioning methods that it doesn't correspond to. The positioning method when the argument is omitted is a matter of product planning and returns it by some positioning method of which the receiver unit corresponds to.

Null is returned without returning the array when the position information cannot be returned during execution of the function. When the return value returns the latitude and longitude in decimal number notation based on the world geographic coordinate system, "1" is returned to Array [0]. "2" is returned for other geographic coordinate systems. Operation (the location information acquired immediately before is returned) in which the location information cannot be acquired during execution of this function is an implementation dependent.

It is preferable that Array [1] and Array [2] return the longitude and the latitude in decimal number notation based on the world geographic coordinate system, respectively. The latitude-longitude is expressed by a one-byte character row the part of integer values in which 10^7 is multiplied by the real number of each grade. For the south latitude and the west longitude add "-" (1 byte minus) at the beginning of the character string. The format of the return value is as follows.

Array [1]: Longitude (degree)

Return value The east longitude is expressed by the character string that shows the ten digit integer.

The west longitude is expressed by the character string that shows the 11 digit integer adding "-" at the beginning.

Example of return value: "1397597250" (For east longitude 139 degrees 45 minutes 35.01 seconds)

Longitude notation range "-1800000000" (180° west) to "1800000000 (180° east)

Array [2]: Latitude (degree)

Return value A character string that shows a nine digit integer to express the North latitude.

The south latitude is expressed by a character string that shows a ten digits integer adding "-" at the beginning.

Example of return value: "356641694" (For north latitude 35 degrees 39 minutes 51.01 seconds)

Latitude notation range "-900000000" (90° south) to "900000000" (90° north)

When it is not possible to return a value by the above-mentioned return value, but by the longitude and the latitude of sexagesimal number notation, the format of the return value is as follows.

Array [1]: Longitude (ddd degrees mm minutes ss.ss seconds)

Return value East longitude "dddmmssss", west longitude "-dddmmssss"

East longitude is expressed by the character string that shows the 9-digit integer.

The west longitude is expressed by the character string that shows the 10-digit integer adding "-" at the beginning.

Example of return value : "139453501" (For east longitude 139 degrees 45 minutes 35.01 seconds)

Longitude notation range "-1800000000" (180° west) to "1800000000" (180° east)

Array [2]: Latitude (dd degrees mm minutes ss.ss seconds)

Return value North latitude "ddmmssss", south latitude "-ddmmssss"

North longitude is expressed by a character string that shows an 8-digit integer.

The south latitude is expressed by a character string that shows a 9-digit integer adding "-" at the beginning.
Example of return value : "35395101" (For north latitude 35 degrees 39 minutes 51.01seconds)

Latitude notation range "-90000000" (90° south) to "90000000" (90° north)

It is preferable in Array [3] to return the geographic coordinate system denomination and the coordinate system denomination of the format shown in the following.

"wgs84_dgree": For World Geodetic System 1984 (WGS84,ITRF) and decimal notation coordinate system

"tokyo_dgree": For Tokyo Geodetic Datum and the decimal notation coordinate system

"wgs84_dms": For World Geodetic System 1984 (WGS84,ITRF) and sexagesimal notation coordinate system

"tokyo _dms": For Tokyo Geodetic Datum and sexagesimal notation coordinate system

The return values in case of other geographic coordinate systems, coordinate systems, and the measuring methods are a matter of product planning.

It is preferable to return the value indicated in the following as an error of the positioning result in Array [4].

null: Error unclear

"1": 0m ≤ error of positioning result < 50m

"2": 50m ≤ error of positioning result < 300m

"3": 300m ≤ error of positioning result < 1km

"4": 1km ≤ error of positioning result

"5": Cell base (positioning by base station of mobile phone)

- X_DPA_saveExAppFile(): An external application is stored in the destination according to the receiver setting. (T.B.D.)

  Syntax:

  Number X_DPA_saveExAppFile(input String src_path
  [,input String type
  ,input String title
  [,input Boolean executable_flag]])

  Argument:

src_path	URI that specifies acquired external application
type	MIME type of contents
title	Title name used to display list of contents
executable_flag	Executable flag

Return value:

1:	Success
NaN:	Failure

Explanation:

It stores the external application (communication contents of another encoding method) specified with the src_path in the area of the receiver unit set beforehand. The external application is executed at once after saving when the executable_flag is set to true. The execution operation will not begin when it is not set to true.

After execution ends (When the executable_flag is true), a startup document of the channel entry component that calls this function is presented. The operation of the contents continues when an external application is loaded and saved in case of the executable_flag is false.

- X_ DPA _startExAV (): The audio and video from the communication contents is displayed.

Syntax:

```
Number X_ DPA_startExAV(
        input String avURI
)
```

Argument:

avURI	URI showing video and audio

Return value:

1:	Success
NaN:	Failure

Explanation:

The audio and video stream by way of the communication is loaded and presented by the URI specified from the document that is presented. It is preferable to maintain the operation of the data contents during presentation.

The return value is not returned when the BML browser ends when the AV playback application, etc. is started. This function can be used by the broadcasting contents and linked contents.

- X_ DPA _stopExAV (): The audio and video from the communication contents is discontinued.

Syntax:

Number X_DPA_stopExAV()

Argument:

None

Return value:

1:	Success
NaN:	Failure

Explanation:

The presentation of the audio and video stream by way of communication presented according to the function (X_DPA_startExAV()) is stopped. When the audio and video stream by way of communication ends, it is preferable to return display to the broadcasting video and audio.

The receiver that ends a BML browser when the AV playback application, etc. starts when this function is executed, only returns 1 and nothing will happen.

- X_DPA_tuneWithRF(): A channel of mobile reception is selected by specifying the reception frequency by the physical channel number.

Syntax:

```
Number X_DPA_tuneWithRF(
        input Number freqChannelNo
        , input String ServiceID
        [, input Number Mode
        , input Number GuardInterval]
)
```

Argument:

freqChannelNo	Physical channel number
ServiceID	Service ID
Mode	OFDM mode
GuardInterval	Guard interval (guard interval ratio)

Return value:

NaN

Explanation:

Function, which selects a channel of portable reception, specifies a reception frequency, by a physical channel number. This function, mainly transmits information of the actual position to the communication destination server, is assumed to be used for channel selection after loading the frequency information of the station that can be received at the actual position from server. As an example, during the viewing of a certain program, it is assumed services can

continue viewing the actual program by acquiring station information (ratio of physical channel number, mode, and guard interval) that broadcast the same program and the station that can receive from actual position, when the reception becomes unstable by the state of electrical waves, by changing the reception station with this function. It is an important function for services of mobile units in which reception during movement is assumed, and it is preferable to install this function.

When this function is called, the receiver unit does the channel selection display operation by physical channel number and service ID specified in the argument.
In the specification of physical channels only, receivers should always select channels by the service ID specified in the argument and not by the physical channel number, in order to avoid the incorrect selection of a broadcasting wave of a different broadcaster which uses the same physical channel, and to correctly select when multiple services exist within one physical channel.

The channel selection in display operation can be done at high speed by OFDM mode and the guard interval specified in the argument according to the receiver. An original system may be used for the scanning method of the mode and guard interval, and it is not necessary for receiver units to refer to this argument during channel selection operations.

Moreover, when an initial scan has already been done to the broadcasting service of physical channel specified in this function, selection can be done only by the service ID omitting the physical channel selection operation. Although it is an implementation dependent, when an initial channel is scanned, the physical channel where the scanning leakage occurs when the channel is selected by calling this function may be allocated in the remote control key.

The script following this function is not continued. Moreover, when the execution of this function fails, the continuance of the script execution that follows afterwards is not guaranteed. Do not describe this function in a global code. The operation when described is an implementation dependent.

The range of each argument is as follows.

freqChannelNo: 1 to 62 (1 to 12 VHF band selection is not indispensable.)

ServiceID : "0000" to "FFFF"

Service ID is specified in a hexadecimal character string. However, the character (string) that shows that it is a hexadecimal character string such as the "h" symbol at the end and "0x" at the beginning of the character string is not given, and to become a fixed length character string with a length of four digits, 0 is the necessary number given at the start of the character string.

Mode : 1 - 3

GuardInterval : 0 - 3

0: Guard interval ratio 1/4

1: Guard interval ratio 1/8
2: Guard interval ratio 1/16
3: Guard interval ratio 1/32

- X_DPA_writeSchInfo(): Information is written in the schedule notebook of the receiver features.

Syntax:

```
Number X_DPA_writeSchInfo (
        input Date      date,
        input String title,
        input String text,
        input Boolean sound_flag
)
```

Argument:

date	Date of schedule written
title	Title of schedule
text	Contents of schedule
sound_flag	Alarm on/off. The alarm sound format and kind is an implementation dependent.

Return value:

1:	Success
NaN:	Failure

Explanation:

The schedule management application, etc., of the receiver starts, and the information specified for the argument is handed over. The contents of the schedule might not be preserved for the character number limitation in each item of the schedule notebook in the receiver features that was exceeded. The title of the schedule is saved as the title of the schedule notebook items, though which column to allocate it is an implementation dependent. The response to the alarm sound and tone setting is an implementation dependent although the setting in which the alarm sounds at a specified time and date is possible when sound_flag is true. Installation of the function does not need to be registered within the schedule notebook prepared in the receiver itself which users write normally, although the function is specified assuming that the program is notified, and can be a separate installation.

Maximum value of each argument is as below

title:	40bytes
text:	80bytes

Operation on the schedule management application initiation and a method of information storage are implementation dependent.

- X_ DPA _getComBrowserUA (): Information to identify if a communication contents browser is acquired.

  Syntax:

  Array X_DPA_getComBrowserUA ()

  Argument:

  None

  Return value:

  Array [0][0]: Character string that shows the manufacturer ID (1)

  Array [0][1]: Character string of User-Agent Header of a browser (1)

  Array [1][0]: Character string that shows the manufacturer ID (2)

  Array [1][1]: Character string of User-Agent Header of a browser (2)

  * * *

  Explanation:

  Information to identify a communication browser installed in the receiver is acquired. When multiple numbers of browsers are installed in the receiver, all browser information is returned to the array in two dimensions. Returns the values which show the manufacturer ID administered in ARIB in Array [n][0]. The return ID of the carriers as a manufacturer ID when an aforesaid communication browser connects to a proprietary network of the carrier such as mobile phones. Contents use this function to identify the carriers. Character string of hexadecimal number notation is returned in Array [n][0], however, 0 is applied at the beginning of character strings as a mandatory number padding to make it 2 digits, although without giving a character (string) that shows it is a hexadecimal number string such as “0x” at the beginning and “h” at the end.

  A character string of 127 characters or less is returned by one byte characters in Array [n][1]. The User-Agent Header character string in carriers browsers for the contents on the internet returns character strings based on the specifications of each carrier.

  The User-Agent Header character string in Browser for the C-profile contents on the internets returns character strings provided by 8.3.14.

- X_ DPA _writeAddressBookInfo (): Information is written in the address book of the receiver features.

  Syntax:

  Number X_DPA_writeAddressBookInfo (

  input String name,

  input String kana,

  input String tel,

input String mail

)

Argument:

name	Registered name
kana	Hiragana printed to name (2 byte katakana)
tel	Registered telephone number
mail	Registered mail address

Return value:

1:	Success
NaN:	Failure

Explanation:

The address book and the telephone book management function, etc. in the receiverstart, and information is handed over to the address book that manages information in arguments (name, kana, tel, mail) as a single-unit. Coexistence with address books that already exist as a receiver features is not necessary.

It is an implementation dependent for whether an item that does not correspond is registered when the argument information on the function is mixed together in an existing address book.

- X_DPA_launchDocWithLink (): Base URI directory is switched, and the BML document is presented while linked.

Syntax:

Number X_DPA_launchDocWithLink (

input String documentName

)

Argument:

documentName Character string that specifies BML document

Return value:

1:	Success
NaN:	Failure

Explanation:

The BML document specified by the documentName is opened and presented on the screen. It is specified only when it is changed to the BML document offered by a server from the BML document offered by a server in a linked status. The host name or the directory name specified by this function is set to the base URI directory (Refer to section 8.3.7 for the definition of base URI directory). The base URI directory before it changes is invalid. The base

URI directory is assumed to be set the same as when changing while linked from the broadcasting contents in the a element and launchDocument() function.

The API whose operation is permitted in linked status, and the information shared by NVRAM becomes possible in the BML document of the changed destination with this function.

- The script following X_DPA_launchDocWithLink() is not continued.
- Neither load nor the unload event occur when X_DPA_launchDocWithLink() is executed in a global code.
- The continuance of the script execution that continues behind, when X_DPA_launchDocWithLink() fails, is not guaranteed.
- Receiver operation fails, and presents the broadcasting contents according to section 8.3.11.4. when X_DPA_launchDocWithLink() is used in the broadcasting reception status.

- X_DPA_chkAVtype (): The type of audio and video from the communication contents is confirmed.

Syntax:

```
Number X_DPA_chkAVtype (
        input String MIME_Type
        [,input String mediaName]
)
```

Argument:

MIME_Type	MIME type of image audio
mediaName	Service name of image audio

Return value:

1:	Success
NaN:	Failure

Explanation:

When there is an audio and video stream presentation function by way of communication, success is returned for the contents type specified by MIME_Type and mediaName. The MIME type is specified for MIME_Type. The service name of the image voice that cannot be specified by MIME_Type alone can be specified for the mediaName. For example, even when the MIME_Type is “video/3gpp”, a detailed specification might be different according to the service, and a mediaName specification may be necessary.

- X_DPA_getIRDID (): The identifier to specify the receiver or the viewer is acquired.

Syntax:

```
String X_DPA_getIRDID (
```

input Number type

)

Argument:

type Type of identifier that demands acquisition

Return value:

Unique identifier: Success

null: Failure

Explanation:

Whether it succeeded in acquiring the receiver unit specified by type, the user unique identifier or identifier is returned as a return value. The format of the identifier is an implementation dependent. When the unique identifier specified for the argument cannot be returned, null is returned as the return value.

The following are specified for argument (type).

1: Receiver unique identifier

Identifier of each receiver that specifies the manufacturer's serial number, terminal number, etc. Return values are some ID (character string) by which the manufacturer or each carrier can uniquely identify the receiver.

2: Televiewer-unique identifier

Identifier which specifies the viewer individually like user ID and telephone number, etc. ID (character string), which uniquely identifies the user of user.

3: Receiver unique identifier or televiewer-unique identifier

Either a receiver unique identifier or a televiewer-unique identifier is returned.

When a receiver unique identifier can be returned, it is given priority.

- X_DPA_writeCproBM (): Registration of TVlink.

Syntax:

Number X_DPA_writeCproBM(

input String title,

input String dstURI,

input String outline,

input Number CproBMtype

[,input Date expire]

)

Argument:

title	Title of TVlink
dstURI	URI link destination of TVlink
outline	Explanation of TVlink
CproBMtype	Type of TVlink
expire	Expiration date of TVlink

Return value:

1:	Success
-1:	The NVRAM area of a data size necessary for registration cannot be reserved.
-2:	As it does not support the contents presentation in the specified format of argument CproBMtype, registration has not been done.
NaN:	Other failure

Explanation:

It is a function to register information on the TVlink. Since loading is done only by the receiver's native application, the format in the receiver is an implementation dependent.

Argument title is a character string displayed in the title column of the TVlink list. The title is 40 bytes or less.

Argument dstURI is the URI of the communication contents in the destination link. URI in the destination link is 60 bytes or less.

Argument outline is an explanation of the communication contents in the destination link. The TVlink outline is 130 bytes or less. A null character string is specified when there is no TVlink outline on what should be registered.

Argument CproBMtype shows the receiver operation when registered information is used. For example, the receiver unit can select a browser that can display the communication contents specified with destination link URI by using this information. The TVlink type is specified with a numerical value from 0 to 255. The TVlink type that the receiver can present can be confirmed by using the (getBrowserSupport()) function. It is preferable to confirm the possibility of presentation of the TVlink type specified before registration by this function, if necessary.

Argument expire specifies the valid final date of TVlink information in year, month, day, hour and minute. When the expiration date is omitted, the TVlink is assumed to be always effective.

<Specification in this volume of receiver behaviour>

Receiver behaviour is an implementation dependent when an argument title is specified exceeding 40 bytes. However, it is preferable that the receiver can register 40 bytes when it

has exceeded 40 bytes, and does not store the 41st byte or more. Moreover, it is preferable not store the first byte of a two byte character when this is the 40th byte.

When argument dstURI has exceeded 60 bytes, the receiver unit assumes the execution of this function is a failure, and returns the return value NaN.

The receiver behaviour is an implementation dependent when the argument outline specified exceeds 130 bytes. However, it is preferable that 130 bytes are recorded, and the receiver does not record the 131st byte or more when it has exceeded 130 bytes. Moreover, it is preferable not to store the first byte of a two byte character when this is the 130th byte. When the TVlink type is memo (CproBMtype=0) or C-profile linked contents (CproBMtype=1), the registration of the TVlink outline to the receiver is necessary.

It is preferable to record the TVlink outline to improve user operation though the registration is optional for other types.

Argument CproBMtype does not need to be registered in the receiver unit, which the operation of the TVlink list application can be concluded without registration, of this information.

Registration of the information given by an argument expire is an implementation dependent.

Receiver with a user interface function that selects and eliminates TVlinks already recorded and maintained in an area when the NVRAM data size area necessary for a registration can not be reserved during this function execution, not executes deletion as a result of the user using this function, and when it fails to register, it returns NaN as the return value.

7.10.8 Execution operation of the extended function

It is assumed that the processes of X_DPA_startResidentApp(), X_DPA_mailTo(), X_DPA_phoneTo(), X_DPA_startExAV(), X_DPA_writeSchInfo(), X_DPA_writeAddressBookInfo () are asynchronously processed among the extended functions specified by 7.10.7. It is preferable that BML browsers continue the execution of script after the extended functions are executed. However, it is preferable to present the default component again after ending the function execution that is called by the extended function when it is impossible to continue.

7.11 Restrictions in BML document descriptions

7.11.1 Restrictions of the script element

Script elements in which the src attribute is not specified may appear 0 or 1 time, and to

which the src attribute is specified may appear between 0 to 2 times. When the content of the script element is described, all of the content is enclosed with “<![CDATA[“ and “]]>” and it is one CDATA section.

The argument is not written in the function used as an event handler. That is, setting of the event handler is followed by the syntax which becomes attr = "func();" (Where attr is an event attribute name and func is a function name).

Operation of automatic semicolon insertion is not mandatory. (ECMA-262 p.20)

7.11.2 Positioning and restrictions

This section provides operation related to positionings for the allocation of elements in the body element.

It shall add a style sheet to specify the position in the BML document. Elements included in the document should be presented properly to users, although there is no uniqueness among various platforms in presenting each element.

The constraints defined below are only applicable to elements that will occupy the 2-dimensional space when placed within the body element, and by no means they prohibit other elements from being included in the body element.

Moreover, in the sentence on the restriction conditions, it is simply recorded as, "The position is specified" for the size and position of the box specified by absolute positioning. The method of specifying the position is the relative coordinates from the parent element (equivalent to the containing block of CSS2) is as follows:

left: Distance (number of pixels) from the left end of the box of the parent element;
top: Distance (number of pixels) from the top of the box of the parent element;
width: Width (number of pixels);
height: Height (number of pixels);
visibility: "visible" or "hidden" or "inherit";

As the positioning is operated by a fixed value (i.e. the specified value is “absolute”), Therefore it is not specified in the style part.

Furthermore, the following terms are used in in the sentence below.

boxed element: a) elements (input, textarea) that have two dimensional expansion, b) element (img, object) that have intrinsic dimension. c) div element, d) form element, e) p element, f) pre element is generically called ‘boxed element’.

normal flow element: Elements (br span a) other than boxed elements are called ‘normal flow elements’ in elements that can be used in operation.

Note) html head body meta script link title bml:bevent bml:beitem elements do not belong to either.

Restriction 1:	A position of boxed element must be always specified.
Restriction 2:	A position of normal flow element must not be specified.
Restriction 3:	The child element of body element must be either div, form, p, or pre elements.
Restriction 4:	Only boxed elements can be involved in the div element and form element. A nest positioning becomes possible by having div element nested.
Restriction 5:	The p element and pre element can have only the element of normal flow and the sequence of the text as the child element.
Restriction 6:	Nesting of the span elements and a element must not be done.

The condition provided by DTD is overwritten by the above-mentioned restriction condition.

(Example)

```
<p style="...box specification...">
        abcde<span style="...">fghij</span>
</p>
```

In CSS2, although absolute positioning and normal flow can be specified in each element, the basic idea of the above restriction conditions is to completely separate the arrangement of the elements of normal flow that are not position specified and elements specified by absolute positioning.

7.11.3 Vertical writing

The specification of the direction of writing by the BML document is not operated. Vertical writing is not operated.

7.11.4 Restrictions of element positioning

- A visibility property shall not set in visible at the same time as multiple p elements that specify -wap-marquee properties.
- For the p element, that -wap-marquee property is specified and the object element, which refers to animation GIF, both visibility properties shall not be set in visible simultaneously.
- Object elements with which the animation GIF figure is associated can present up to four at the same time within the range of the restriction in section 5.2.2.11.

Elements may be allocated in the back of the rendering area of the elements, and the background-image of the body element that includes the element can be specified. However, object elements associated with visible animation GIF figures and p elements for which

marquee is specified, cannot be placed in the background. In other words, there must be no overlapping area among animation GIF figures with visibility set in visible and p element for which marquee is specified.

When other elements are overlapped in front of the object element with which the animation GIF figure is associated and the p element for which marquee is specified, it is not possible to display both at the same time. That is, visibility should be “hidden” in either the elements or all elements placed in front of them.

Table 7-22 Restriction of visibility when elements are allocated on top of each other

Element			Rear					
			animation GIF		marquee		Other elements	
		visibility	visible	hidden	visible	hidden	visible	hidden
Front	animation GIF	visible	-	O	-	O	O	O
		hidden	O	O	O	O	O	O
	marquee	visible	-	O	-	O	O	O
		hidden	O	O	O	O	O	O
	Other elements	visible	-	O	-	O	O	O
		hidden	O	O	O	O	O	O

O *** Operated - *** Not operated

7.11.5 Operation related to object elements

Note) The presentation behaviour of individual monomedia presented by object elements is specified in section 7.12.3.

7.11.5.1 Operation related to the presentation operation of animation GIF

This section specifies for the interpretation of attributes when animation GIF is specified for object elements and operation related to screen presentation. For the operation of the presentation behaviour of animation GIF related to streamstatus, See 7.12.6.2.

- The data attribute can be changed only when animation GIF image has not been played.
- Whether to stop the rewriting of the presentation frame of animation GIF during which a screen update is prohibited by execution of lockScreen() or to continue playback is implementation dependent.

7.11.6 Operation related to nested elements

Neither the span element nor the a element become nested. In fact, other elements are not included in the span element and a element. However, as an exception the br element can be included.

7.11.7 Operation related to the p element

The interpretation when -wap-marquee property is specified in the p element and the operation related to screen presentation is provided below.

- The contents of p element can be rewritten only when the visibility is hidden.
- Whether to stop the modification of a display position in p element contents during which a screen update is prohibited by execution of lockScreen() or to continue playback is implementation dependent.
- When visibility changes from hidden in visible, playback is from the initialization status.
- The child element of p element applied is only one PCDATA node. Neither span, a, nor br elements are applicable. The maximum length of the PCDATA is 128 bytes. When exceeding this, the character displayed is rounded down.
- Do not set the visibility property of multiple p elements for which the wap-marquee property is specified to visible at the same time.
- For the p element,that the -wap-marquee property is specified and the object element, which refers to animation GIF, both visibility properties shall not be set in visible simultaneously.
- Transparency cannot be specified for the background color of p elements for which this property is specified.

7.12 Operational guidelines related to presentation

7.12.1 Operational guidelines of object presentation

The presentation order of objects is the appearance order of the elements. In other word, the element that appears first in the BML document is presented at the inner part (position far from viewer) of the screen. The relationship between the structure and presentation of DOM objects is shown in Figure 7-1. DOM objects generate a tree structure in the order of element presentation. The presentation is done in order of 1, 2, 4, 5, 7, 8, 9, 3, and 6 when the structure of object 1 to object 9 becomes as shown in Figure 7-1.

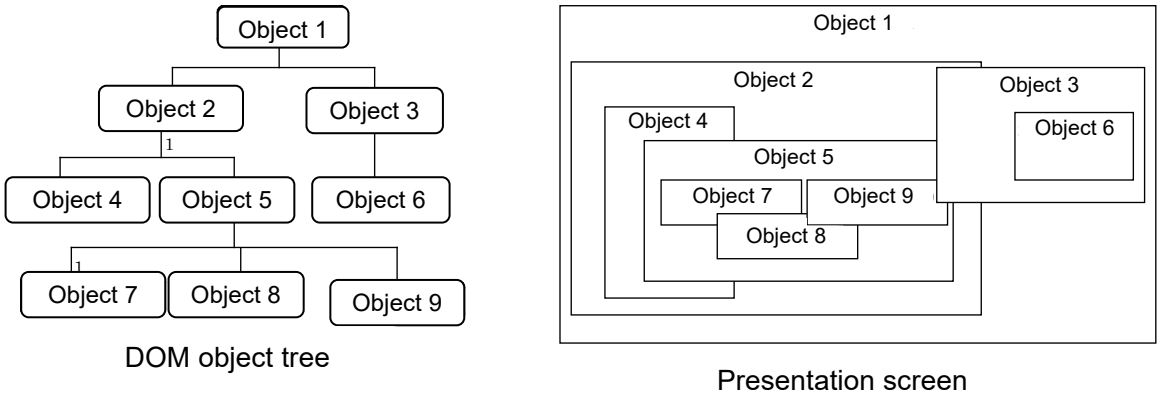


Figure 7-7-1 Structure and presentation of DOM objects

7.12.2 Presentation plane

Rendering of the objects described in 7.12.1 is done in a virtual plane for a BML browser shown in section 3.2.1. The virtual plane has a logical Cartesian coordinate system where the upper left corner is the starting point (0,0), and the coordinates are in units of integers. A size of 1 pixel of each vertical and horizontal corresponds to 1px in the BML contents. Therefore, the horizontal width of this virtual plane becomes 240 pixels and the vertical width becomes 480 pixels.

Moreover, each pixel has color information of RGB in each 8 bits. However, the bit depth of color information is implementation-dependent. For this detail, see section 3.2.2.

7.12.3 Operation of monomedia presentation

The following shows operation of monomedia presentation using an object elements and img elements.

7.12.3.1 Positioning by object elements and img elements

Operations of the top, left, width, and height in the CSS2 properties applied to object elements and img elements are shown below.

■ top property, left property

The value that the top property and the left property may take according to type attribute (media type of the monomania specified as the src attribute in case of the img attribute) of object elements is shown in Table 7-23.

Table 7-23 The value available for top/left properties

type attribute (Media type)	
image/jpeg	Arbitrary in both x and y coordinates
image/gif (case of GIF)	Arbitrary in both x and y coordinates
image/gif (case of animation GIF)	Arbitrary in both x and y coordinates

■ width property, height property

The formula for computation of the width characteristic and height characteristic is shown in Table 7-24. The screen presentation when the value, in which the width property and height property of object elements are not filled in with the relationships of this table, is implementation-dependent.

Table 7-24 width/height property

type attribute (Media type)	width	height
image/jpeg	W	H
image/gif (case of GIF)	W	H
image/gif (case of animation GIF)	W	H

Note) For W and H, refer to Table 7-25

■ Vertical and horizontal pixel numbers

W (Image horizontal pixel number), H (Image vertical pixel number) will take the values of Table 7-25.

Table 7-25 Values applicable to W and H

type attribute (Media type)	W	H
image/jpeg	16 - 240	16 - 320
image/gif (case of GIF)	2- 240	2- 320
image/gif (case of animation GIF)	2- 240	2- 240

7.12.4 Guidelineson clipping

Clipping can be done to the child element by the parent element because it is an operation done according to a fixed value in which the overflow property is specified as hidden in these guidelines. When the content overflows a rectangular area of a specified element (object 1), clipping is done as in the left of Figure 7-2. In addition, when clipping the child element (object 3) by the parent element (object 2), as shown this figure on the right, it becomes non-display excluding the rectangular area specified by the parent element.

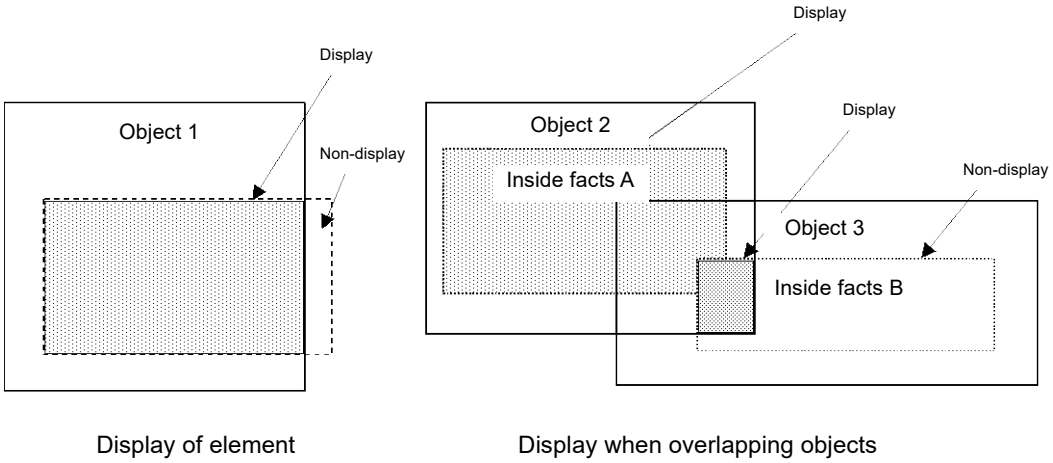


Figure 7-2 Example of a clipping behaviour

When jpeg images, GIF images, and animation GIF images are referenced by object elements or img elements, the clipping by the parent div elements is undefined. It should not be described in the contents specifying that the monomedia image specified by the object/img element overflows the div element of parents. Moreover, when the “-wap-marquee” property is specified in the p element, the clipping by the parent div element is undefined.

These rules are similar for clipping the second generation or higher and should not described the ancestor elements to perform clipping in the contents.

7.12.5 Range of values where CSS2 properties is applicable

The value range that is applicable “<length>” as the value, among CSS2 properties, is shown in Table 7-26 .

Table 7-26 Range of values where CSS2 properties can be taken

Property	Range of values
top	-480px to 480px
left	-240px to 240px
width	0 to 240px
height	0 to 480px
border-width	0 to 240px
padding-top,bottom	0 to 480px
padding-right,left	0 to 240px
line-height	normal or 30px - 420px

7.12.6 Provision of box models

The margin property is fixed to 0. Therefore, this model is operated only by the borders and padding shown in Figure 7-3. Moreover, the arbitrary boxed elements shown by 7.11.2 shall satisfy the following, when (x0,y0) is the relative position of the border area upper left corner of this element to the body element upper left corner and when “w” and “h” are width and height

respectively before clipping taking into account the contents area, the padding area, and the border area of this element.

x0 ≥ -240, y0 ≥ -480, x0 + w ≤ 480, y0 + h ≤ 960, 0≤w≤240, 0≤h≤480

where , w = width + padding-right + padding-left + border-width * 2 and h = height + padding-top + padding-bottom + border-width * 2

When the value of CSS2 properties that do not satisfy the above-mentioned conditions, the presentation is implementation- dependent.

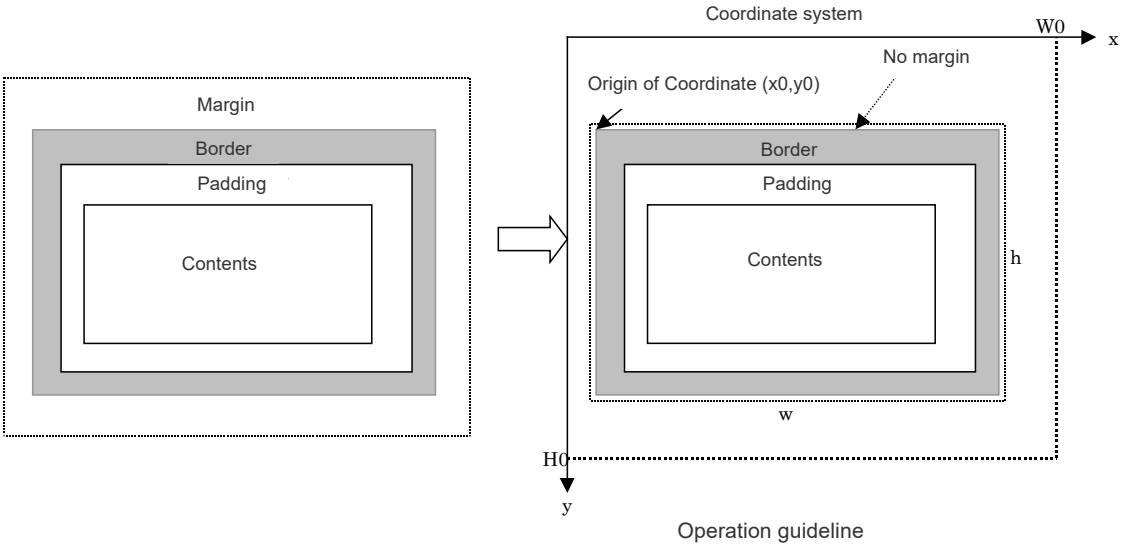


Figure 7-3 Presentation of the box model

7.12.6.1 Attribute operation related to stream presentation behaviour

Table 7-27 shows the operation of the streamstatus attribute of object elements

Table 7-27 Attribute operation related to stream presentation

Type attribute	Streamstatus attribute
audio/X-arib-mpeg2-aac	play,stop Note 2) Note 4)
image/jpeg	play Note 1)
image/gif	play,stop,pause Note 3)

Note 1) The initial value of the streamstatus attribute of the object element that refers to media of the type attribute which only “play” is taken, shall be “play”.

Note 2) The initial value of the streamstatus attribute of the object element that refers to media of the type attribute which “stop” can be taken, shall be “stop”. The MediaStopped event doesn't occur in the initial status. Moreover, the dynamic changes of the type attribute and the dynamic changes of the scheme by changes in the data attribute in audio cannot be done in the object element.

Note 3)The initial value of the streamstatus attribute shall be “play”. Moreover, the assignment of the streamstatus attribute and the change by script cannot be done when it is not animation GIF. In addition, the switching of GIF and animation GIF cannot be done by a change in the data attribute.

Note 4)When the playback ends, the streamstatus attribute becomes stop.

7.12.6.2 Presentation behaviour of animation GIF and operation of the streamstatus attribute

The interpretation of the attribute when animation GIF is specified for the object element and operation of the screen presentation is explained.

- The operation of animation GIF according to the initial value of the streamstatus attribute is shown in Table 7-28.

Table 7-28 Operation of initial value of the streamstatus attribute

Initial value of Streamstatus attribute	Behaviour
play	Start playback from the first frame.
stop	The frame shall not be displayed. The playback frame position is reset at the beginning.
pause	Display the first frame.

- The presentation status of animation GIF when the streamstatus attribute is changed by using DOM API is shown in Table 7-29.

- **Table 7-29 Access to the streamstatus attribute**

Value of original streamstatus attribute	Set to play	Set to stop	Set to pause
play	-	• Stop playback and do not show the frame.	• Stop playback and display any one frame. Which frame to display is implementation dependent.
stop	• Start playback from first frame.	-	• First frame shall be displayed.
pause	• Start playback. Which frame to display first is implementation dependent.	• Any Frame shall not display.	-

8 Guidelines for browser's behaviors

8.1 Guidelines for presentation

The guidelines for the presentation function of C-profile receivers are shown below.

8.1.1 Principle of presentation in C-profile receivers

8.1.1.1 Principle of simultaneous presentation of video, audio and multimedia data

In C-profile receivers, video, audio and data are unified as broadcasting contents, and in principle shall not be presented separately. However, the following cases are exceptions, provided that "8.1.3 Operation during the startup of the TV reception function and tuning" is followed accordingly.

- If audio is presented by itself by user operation
- If BML browser is displayed in full screen by user operation
- If it applies to "8.1.8.2 Exceptions in multimedia data display"

8.1.2 Presentation of video and audio

8.1.2.1 Presentation and control of broadcasting video and broadcasting audio

Presentation and control of broadcasting video and broadcasting audio shall be conducted by receiver features. No control of presentation including scaling from multimedia shall be conducted.

8.1.2.2 Display position and display size of video

The display position and display size to display video in the receiver are Implementation dependent. The receiver does not necessarily need to display the video transmitted in QVGA in the same pixel size. Specifically, display is permissible with a reduced video pixel size and frame rate if the processing performance is insufficient. Also, the display may be enlarged in receivers with a pixel size level of QVGA or higher.

8.1.2.3 Optimizing video display size

With receivers whose display device pixel size is not 320 dots in the horizontal direction, it is recommended scaling the video depending on the receiver features according to the pixel size of the display device.

8.1.2.4 Display of QVGA 16:9 video

When displaying QVGA (16:9) video, the area within the display device that the video possesses is 60 dots smaller in the vertical direction in an imaginary plane compared to the area possessed by QVGA (4:3) video. It is recommended setting 60 dots of area space as the visible area for data broadcasting rather than leaving it empty.

8.1.2.5 Changing the video display size

Receivers may switch from several display sizes to display video.

8.1.3 Operation during the startup of the TV reception function and tuning

8.1.3.1 Principle of display status when the TV reception function startups and during tuning

When the TV reception function starts up and immediately after tuning using the procedure below, only video, audio, and data from broadcasting is presented on the receiver's display device and no other contents outside of broadcasting or applications shall be displayed (such as e-mail screen, etc.). Contents and applications that were displayed before tuning shall not be displayed. Also, this principle may be skipped only when users operate tuning related buttons during operation of the TV reception function.

- When the function (X_DPA_tuneWithRF()) for tuning is called from data broadcasting contents or C-profile linked contents.
- When channels are selected using receiver features other than BML browser. (Tuning by communication browser's "a" element URL expansion, etc., tuning by API's of java applications, tuning using API's to call the tuning function implemented in receivers, etc.)
- Tuning from a TVlink list

8.1.3.2 Principle of BML browser key acquisition

During operation of the TV reception function, BML browsers shall always be able to acquire the enter key and back key in principle. In the same manner, focus transfer and scrolling shall be useable by receiver features. However, if other applications are simultaneously displayed and the application contains the focus, this may not apply. Focus transfer and scrolling implementation procedures (keys to use, etc.) are implementation dependent, but it is preferable to use the up/down keys.

8.1.4 Prohibition of mixed display and simultaneous display

8.1.4.1 Princilple of mixed display prohibition

It is prohibited to possess mixed display functions in receiver. This “mixed display” means displaying contents provided by multiple different suppliers which may cause viewers to misunderstand that those contents are provided by same supplier such as cases where certain contents suppliers intentionally display contents provided by different suppliers at the same time by getting one set of contents to be related to other contents or cases where one set of contents affects the other display contents, etc. The function, which enables such display, is called a mixed display function. However, with C-profile linked contents in a link state, 2 contents of TV and communication can be regarded as being supplied by the same supplier, and thus is not a mixed display.

8.1.4.2 Simultaneous display of broadcasting screens and screens other than broadcasting (browsers for the contents on the internet, applications, etc.)

- When starting up contents not supplied by the broadcasting company while presenting broadcasting contents, it is recommended to turn off the broadcasting screen and present the contents concerned on a full screen in order to avoid mixed display.
- While simultaneous display of broadcasting screens and screens other than broadcasting is not encouraged, when conducting simultaneous display according to implementation dependent, leave BML browser on as shown in 8.1.8.1 and display the following at the minimum. Also, same manner of handling is required for cases where caption display is conducted in place of BML browser by user operation, etc. Even if these measures are taken, keep in mind that conducting simultaneous display has a risk of becoming mixed display, and thus consideration in implementation dependent shall be taken.
  (i) When starting up contents other than broadcasting during TV presentation, in principle, notify the viewers by with a dialog box, etc. that contents unrelated to broadcasting will be displayed.
  (ii) Presenting options for the viewers to choose when starting contents other than those broadcast during TV presentation is recommended, unless the choice between simultaneously display or switching is specified by API(X_DPA_startResidentApp()).
  (iii) Always display so that it is clear to everyone that contents unrelated to broadcasting are being displayed during simultaneous display.
  (iv) Broadcasting contents shall always be displayed on the forefront.
  (v) It is desirable that a broadcasting station name and program name are displayed on the display broadcasting content video. It is also desirable that a video display position can be moved through operation from the user.

8.1.4.3 Various operations during simultaneous display

- Even when simultaneous display of BML browsers and browsers for the contents on the internet is conducted, C-profile linked contents shall be displayed on BML browser.
- When an e-mail screen is started when receiving e-mail and is simultaneously displayed, a simultaneous display function is recommended to limit e-mail from addresses stored in the address books of terminals, even while being displayed under the above restrictions.

8.1.4.4 Operations of broadcasting content during simultaneous display

- Even during simultaneous display, BML browsers shall continue to operate and be able to acquire each event from broadcasting.
- Even when browsers other than broadcasting and applications are simultaneously displayed, BML browser shall be able to acquire key entries according to operation of the viewers.

8.1.5 Prohibition of simultaneous display while displaying a part of broadcasting content (video audio only, audio only or BML browsers)

8.1.5.1 Principle of prohibition of simultaneous display while displaying a part of the broadcasting contents

- As a rule, when only a part of the broadcasting content (video or BML browser) is being presented on the screen, the partial content shall be displayed on its own and must not be displayed simultaneously with any content and/or application provided by other operators.
- To start presentation of contents or applications supplied by another company during presentation of a part of the broadcasting contents, the broadcasting screen shall be turned off and switching to a full screen shall be conducted, excluding the exceptions given in the following clause. To continue to receive broadcasting and to present audio in the background is implementation dependent.

8.1.5.2 Exceptions to simultaneous display prohibition when presenting only a part of the broadcasting contents

The following screens may be simultaneously displayed as exceptions even when presenting only a part of broadcasting content.

- Screens below in e-mail application
    - e-mail composition screen
    - Transmission related screen
    - e-mail reception notification excluding subject line

  - e-mail sender, subject line and content of e-mail from senders whose address entry is in the address book of the receiver
- Screen which is displayed by a receiver specific application which was installed in the receiver at the time of purchase and does not conduct communication. (Including when the receiver specific application is updated using communication in order for debugging, etc.)
- To present streamed videos acquired via communication using the X_DPA_startExAV() function according to the specifications in “8.1.7.6 Display of videos acquired from communication”
- Message, etc. provided in 8.1.6

8.1.6 Simultaneous display of messages presented by receiver systems

Messages displayed by receiver systems for the convenience of viewers or to seek consent from viewers may be simultaneously displayed. (Same for cases where a part of the broadcasting contents is presented)

- Program table and information using SI information acquired from broadcasting
- Receiver’s system status display
- Receiver’s set up screen
- Operation support information for users
- Incoming telephone call notification and caller number notification (for compound terminals with telephones)
- Supplementary information for transmission such as displaying the entered phone number, etc. when users make phone calls (for compound terminals with telephones)
- Display of user interfaces to display contents for permission and to obtain the approval of users when the permission of users is required to for communication connection
- TVlink list screen
- Alert of operations which may cause inconvenience to the user

8.1.7 Specifications regarding communication contents presentation

8.1.7.1 Contents presented in BML browsers

BML browsers are browsers implemented based on these specifications, and present data broadcasting contents and C-profile linked contents

8.1.7.2 Activation of communication content other than C-profile linked content

If unlinked communication content will be activated from data broadcasting or C-profile linked content, the content shall be presented by activating another receiver-specific communication

browser using X_DPA_startResidentApp(). Specification of a receiver specific communication browser is implementation dependent, and the choice of the browser to implement in the receiver is free. However, fulfilling the Browser for the C-profile contents on the internet specifications provided in chapter 7 is recommended.

8.1.7.3 Specifications to display receiver specific browsers for the contents on the internet

To startup receiver specific browsers for the contents on the internet using X_DPA_startResidentApp(), display strictly according to the specifications of mixed display prohibition provided in 8.1.4. Also, if full screen view is specified by the argument of this function, the receiver shall turn off the broadcasting screen. (Follow the receiver setting if full screen view is not specified by the argument of this function)

Even if non-linked communication contents are displayed by using receiver specific browsers for the contents on the internet by receiver features and not the function, strictly follow the principle of mixed display prohibition. From the view point of mixed display prohibition, turning off the BML browser and video, and switching full screen to communication browser is recommended (generic term for Browser for the C-profile contents on the carriers' proprietary browsers).

8.1.7.4 Common use of BML browsers and Browser for the C-profile contents on the internets

BML browsers and receiver specific browsers for the contents on the internet are logical concepts and they do not limit the implementation procedures of receivers. For example, it is possible to implement a browser which has BML browser functions and Browser for the C-profile contents on the internet functions and use them by logically switching the operations. However, in that case, the principle of mixed display prohibition shall strictly be followed to present non-linked communication contents and to conduct display.

Moreover, whether or not to implement a C-profile communication browser to a receiver already incorporating a browser other than a C-profile communication browser as its specific communication browser shall be implementation dependent.

8.1.7.5 Browser to present with TVlink

When presenting communication contents based on the TVlink information (URI) recorded by broadcasting contents, the contents shall not be presented in a link state due to the principle of mixed display prohibition, regardless of the channel that recorded TVlink and the channel currently being shown. Therefore, presenting communication contents from TVlink information,

for instance, on receiver specific communication browser is assumed. (Excluding CproBMtype=1 TVlink)

8.1.7.6 Display of videos acquired from communication (optional)

To display streamed videos acquired via communication by X_DPA_startExAV(), as a general rule, stop to present broadcasting video and audio, and display this movie in a broadcasting video display area. However, it is recommended to continue with presentation by data broadcasting.

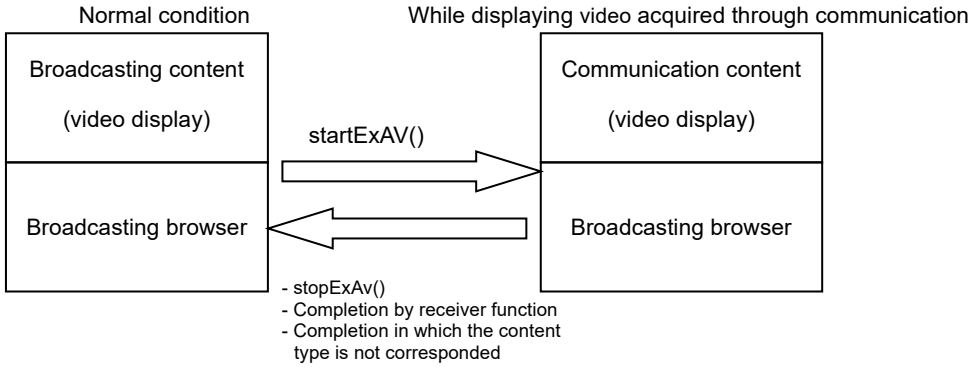
In this case, the receiver shall consider that the viewer can identify that the video was acquired via communication.

Completion of communication videos may be conducted by X_DPA_stopExAV() and by using the completion function of communication video in receiver features if it is implemented. When the communication video ends, returning the display to broadcasting video is recommended.

The transition in which data broadcasting presentation continues when displaying video that the receiver acquired via communication is shown in A of ”Figure 8-1 Display of video acquired through communication”. Data broadcasting continues to be shown while displaying streamed video acquired through communication.

If it is impossible to continue presenting data broadcasting while displaying video that receiver acquired through communication, BML browser can be stopped once as shown in B of ”Figure 8-1 Display of video acquired through communication”. However, it is recommended to quickly re-open the broadcasting contents once the display of video acquired through communication stops.

A. If data broadcasting is continued while displaying video acquired through communication



B. If data broadcasting presentation cannot continue while displaying video acquired through communication

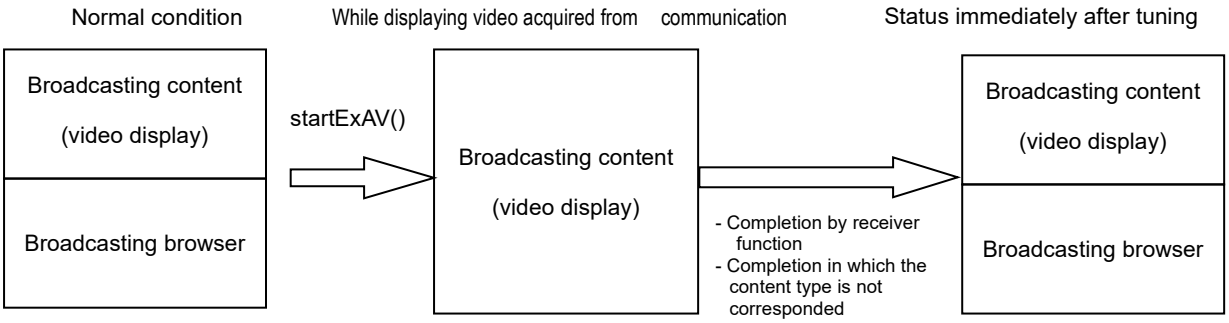


Figure 8-1 Display of video acquired through communication

8.1.7.7 Permission during communication generation and personal information dispatch

Permission from the viewer shall be acquired for BML browser to communicate. Permission acquisition is supposed to be conducted using receiver features, and the timing to acquire the permission is implementation dependent. However, some content may frequently access a single server within short duration, so it is necessary to implement keeping in mind that the service may break if permission is issued every time communication is generated. (Especially if transmitTextDataOverIP() function is called) As a specific implementation procedure, it is ideal to acquire permission at once when communication is generated after tuning, and not display the permission screen until another tuning or TV function completion is conducted.

Also, when the contents transmit private information, the broadcasting company (on the contents side) shall acquire user permission without fail independent from whether the receiver features has acquired permission or not.

8.1.8 Presentation of multimedia data

8.1.8.1 Multimedia data display area

In the multimedia data display area (visible area for data broadcasting), it is recommended to be able to display contents of 125 dots in the Y axis direction over an imaginary plane (6 lines in normal size characters). Leave enough area space to display contents of at least 50 dots in the Y axis direction over the imaginary plane (2 lines in normal size characters) when the data area is temporarily reduced for reasons such as simultaneous display, etc.

8.1.8.2 Exceptions in multimedia data display

As exceptions to the principle of simultaneous display of video, audio and multimedia data, it does not matter if the display of BML browser is not carried out even while video is being displayed is limited to the following cases:

- Cases where displaying video on a full screen, etc., when the visible area for data broadcasting specified in the previous clause cannot be placed in the screen after placing the video area in the display device.
- When the video is displayed in a direction which is not the original display direction with receivers that possess a function to rotate video 90 degrees from its original display direction in order to display video larger by using vertical display devices in a horizontal direction. (*)
- Cases where the data broadcasting browser can be immediately displayed with a simple operation, such as one action instructed by the user. However, also in this case, a data broadcasting area shall be displayed immediately after activating the broadcasting screen function in accordance with the specifications provided in 8.1.3.1.
- If a data carousel ES does not exist in the PMT, there is no need to take BML browser display area. However, if a data ES does exist in the PMT, a BML browser display area needs to be kept, independent from the fact a data service is conducted or not. If the data service is not temporarily conducted, an empty carousel shall be played in principle.
- For receivers which cannot secure other di
- splay areas for caption other than the video display area and visible area for data broadcasting etc., caption may be displayed in the visible area for data broadcasting.

* However, it is recommended to be able to display as much data broadcasting as possible even in such cases. Especially with terminals whose original display direction is in the vertical direction, it is preferable to display data broadcasting in the area below the video when video in a 16:9 aspect is displayed.

8.1.8.3 Optimizing presentation size

It is recommended to display the size of multimedia data (characters and images) on display by scaling to appropriate size depending on the pixel size of the display device, size, presentation size of the video, regardless of the pixel size of mono-media to transmit.

Contents shall be rendered on an imaginary plane of 240 pixels in the horizontal direction and 480 pixels in the vertical directions according to the next clause “Assumptions in contents creation”. Either scale the rendering results in the imaginary plane following receiver specific presentation procedures, keeping the vertical/horizontal ratio, or possess a function to review arbitrary parts of the imaginary plane by using the scrolling representation.

8.1.8.4 Changing the size of BML browser display area

The purpose of the function (setFullDataDisplayArea()), that enlarges BML browser display area, is to (temporarily) increase the number of displayed characters in a vertical direction according to the contents. Receivers may choose BML browser display area enlarged by these functions on their own, unit by unit, depending on the pixel size of display device. It is implementation dependent whether or not to change the video display area to BML browser display area by implementing these functions.

8.1.8.5 Image size possible for display

GIF, animation GIF, JPEG images of 240 dots in the horizontal direction and 320 dots in the vertical direction (240 dots for animation GIF) or smaller shall be able to display. They may not be displayed depending on the receiver if an image exceeding this restriction is used. Also, receivers may display images by scrolling depending on the vertical size of the display area even if the image size is within the restrictions.

If a still image resource size is within the specified size of “width” and “height” of the “img/object” element, it shall be displayed in the same size. In other cases, perform the clipping process or the reduction process keeping aspect ratio to display.

If display by the clipping process is performed, place the top left corner of the still image resource in the top left corner of the “img/object” element.

Displays within the “img/object” element range but outside of still image resource rendering range shall be processed as “transparent”. Also, display accordingly for cases where replacement with a different size still image by DOM is generated. Also, the display of animation GIF whose size is different from the “object” element size shall be implementation dependent.

8.1.8.6 Character fonts and possible number of characters for display

For “medium” sized characters, the font shall be of a dot number which enables 12 characters or more to be displayed in the horizontal direction. For example, it shall be a 20 or less dot font for receivers with display devices of 240 dots in the horizontal direction.

For “large” sized characters, the font shall be of the dot number which enables 8 characters or more to be displayed in the horizontal direction. For example, it shall be a 30 or less dot font for receivers with display device of 240 dots in the horizontal direction.

For “small” sized characters, the font shall be a dot number which enables 15 characters or more to be displayed in the horizontal direction. For example, it shall be a 16 or less dot font for receivers with display device of 240 dots in the horizontal direction. Also, “small” sized character fonts may not be able to be implemented due to the display device pixel size, and implementation may not be possible in such case.

8.1.8.7 Position of CSS and contents

Contents shall be positioned in the imaginary plane specified by the CSS. For receivers that cannot display as specified by the CSS due to the pixel size of the display device or aspect, the receiver shall optimize the display.

8.1.9 Guidelines for contents creation

Below are the guidelines to be followed when broadcasters, etc. create contents etc.

8.1.9.1 Pixel size assumed in contents creation

Pixel size for the imaginary plane for BML browsers assumed by contents creators shall be 240 pixels in the horizontal direction and 480 pixels in the vertical direction.

8.1.9.2 Character size controls and layout to be assumed in contents

For “medium” size characters, a 20 dot font is assumed, and a maximum of 12 characters are placed in the horizontal direction. For “large” size characters, a 30 dot font is assumed, and a maximum of 8 characters are placed in the horizontal direction, and for “small” size characters a 16 dot font is assumed and a maximum of 15 characters are placed in the horizontal direction.

If a “small” size font is specified, some receivers may display them in “medium”. Therefore, contents shall be considered to avoid collapse of presentation in such cases.

Table 8-1 Character size controls assumed in contents

Size	Dots	Number of displayed characters in the horizontal direction
large	30	8 characters maximum
medium	20	12 characters maximum
small	16	15 characters maximum

8.1.9.3 Size of images used

The maximum size for GIF, animation GIF, JPEG images is 240 dots in the horizontal direction and 320 dots in the vertical direction (240 dots for animation GIF). However, contents shall be created that assume display with scrolling in the vertical direction.

If the still image resource size is smaller than the “width” and “height” specification of ”img/object” element and concerned element’s “visibility” property is also “visible”, it is recommended not to layer elements that match all of the following conditions in the “width” and “height” property specified range of the concerned element. (Strongly suggested)

1) It is not “body” element.
2) It appears before the “img/object” element concerned in the document.
3) The “visibility” property value is “visible”.
4) It possesses display contents.

   For example, the “div” element whose “border-style” property value is none and “background-color” property value is “transparent” is an element which does not posses display contents.

When these elements are layered and the “data” attribution of the “img/object” element concerned is re-written as a different size mono-media by DOM API, it is possible it will take some time before display depending on the receiver. Therefore, this point shall be kept in mind when authoring contents.

8.1.9.4 CSS

Always perform positioning specification by the CSS. Display is not performed accordingly if the CSS is not written.

8.1.9.5 Acquiring and releasing “*”, “#” and number keys

Once a document acquires a number key, “*” or “#” key by the “used-key-list”, one-touch tuning is disabled until it is released explicitly or the document transits. Therefore, the use of such keys shall be kept to a minimum. Also, if these keys are acquired due to necessity, release them as soon as entry is completed.

8.1.9.6 Notes on communication charges generated and information transmission

The permission for BML browsers to generate communication is assumed to be performed by the receiver, and thus it is unnecessary to reacquire permission on the contents side.

Also, when contents transmit private information, independent from whether the receiver features acquires permission or not, user permission shall always be acquired by the broadcaster (on the contents side).

Information acquired by getProgramID() and getActiveDocument() shall not be transmitted in connection to information that can identify the individual.

8.1.9.7 Permission when acquiring position information

The permission when acquiring position information is assumed to be performed by the receiver depending on necessity, and thus it is unnecessary to reacquire permission on the contents side. However, follow 8.1.9.6 when transmitting the position information acquired to interactive servers.

8.1.9.8 Permission when acquiring receiver-unique identifiers or televiewer-unique identifiers

The permission when acquiring receiver-unique identifiers or televiewer-unique identifiers is assumed to be performed by the receiver depending on necessity, and thus it is unnecessary to reacquire permission on the contents side. However, follow 8.1.9.6 when transmitting the receiver-unique identifier or the televiewer-unique identifier acquired to servers.

8.1.10 Display position for caption

It is recommended to display caption in a position that does not layer the video display area or data broadcasting area in display devices. However, for receiver that cannot secure display areas outside of the video display area and the data broadcasting area, caption may be displayed in the data broadcasting area by user operation. Also, if the display of caption in the video display area does not hinder the viewing the video, caption may be displayed over the video display area.

8.1.11 Guidelines for DOM operation

Refer to ARIB STD-B24 Vol. 2 Appendix 4 “5.3. Guidelines on behaviors of DOM”. However, ignore the comments and do not generate the comment node.

8.1.12 Guidelines for external characters operation

External characters are not operated in C-profile.

8.1.13 Guidelines for ECMA script implementation

Refer below. In BML documents in which multiple script elements are recorded, the following restrictions are applied for the status where all scripts (scripts recorded in the resources specified by the src attribute of script elements, as well as scripts recorded inside script elements which do not possess the src attribute) are read.

- ARIB STD-B24 Vol. 2 Appendix 4 “5.4.1. Operation of script working environment”
- ARIB STD-B24 Vol. 2 Appendix 4 “5.4.2. Data type”, ”5.4.3. Effects on basic objects caused by data type restrictions” and “5.4.4. Operation rule of implementation-dependent portion”

8.1.14 Guidelines for operation of extended objects for broadcasting

8.1.14.1 Operational specifications for BinaryTable objects

- BinaryTable object size shall be 128 KB maximum.
- Number of lines shall be 2048 maximum.
- Maximum record length shall be 512 Bytes.
- Maximum field number shall be 32.
- Maximum comparison condition number shall be 4.

8.1.15 Guidelines for browser pseudo-object operations

Operations of browser pseudo-object are provided below. Follow,

- “Appendix 1 Operational guidelines”
- “Appendix 2 Operation guidelines for implementing basic services”

in ARIB STD-B24 Vol. 2 “XML-based multimedia coding scheme” for those not specifically recorded.

8.1.15.1 Operations of Ureg

Ureg values shall be valid within contents groups. Ureg values are initialized at browser startup and contents groups change (at tuning including media change).

Independent from ARIB STD-B24 Vol. 2 Appendix 2 “5.6.1. Operation of Ureg”, receivers shall not write the contents group identifier in Ureg[0] (in which the service_id is changed to hexadecimal characters in a "0xXXXX" format).

Ureg values shall be initialized at browser startup and contents group changes (at tuning including media change) by receiver.

The first value read after initialization is empty string.

8.1.15.2 Operations of Greg

Greg implementation is in principle necessary but implementation of the following manner may be conducted for receivers which do not receive other broadcasting media (media including data broadcasting) but only receive C-profile as exceptions:

- Writing value in Greg: Does not memorize the value. However, it does not generate an error.
- Reading Greg value: Returns empty string.

Follow ARIB STD-B24 Vol. 2 “7.6.16 Greg pseudo object properties”. Greg values are initialized at the startup of the broadcasting receiving function, and the value is always maintained while the broadcasting receiving function is valid. The first value to be read after initialization is empty string.

While broadcasting reception function is valid, the Greg value will be maintained even while presenting non-linked contents that are not supported by Greg on browsers other than BML

browsers. In the same manner, it is ideal to maintain the Greg value while presenting media that does not support Greg.

If the Greg value cannot be maintained while presenting other media, etc. for some reason, receivers shall initialize the Greg value when presenting media that supports Greg for the first time after that.

8.1.15.3 Operation of EPG functions

In order to correspond to EPG functions, L-EIT reception is needed by the receiver. However, functions operated in C-profile are epgGetEventStartTime() and epgGetEventDuration(), so compatibility with 8-bit character codes is not necessary.

(1) epgGetEventStartTime()

Specifiable event by event_ref shall only be the event currently presented, and the start_time value shall be acquired by referring to L-EIT.

(2) epgGetEventDuration()

Specifiable event by event_ref shall only be the event currently presented, and duration value shall be acquired by referring to L-EIT.

(3) epgReserve()

Always specify startTime.

If there is another viewing reservation overlapping with the viewing reservation time by this function, the process shall be implementation dependent.

8.1.15.4 Interaction channel communication – Operation of TCP/IP

Refer to 8.3.6.3.

8.1.15.5 Operation of the operational control function

Follow STD-B24 for operations of functions not especially relevant to this clause.

Refer to 8.3.6.4 for utilization from communication contents.

(1) Operation of lockModuleOnMemoryEx() and unlockModuleOnMemoryEx()

Not only the module of the component currently being presented but also components that are not currently being presented can be locked by lockModuleOnMemoryEx(). This function is operated as follows:

- All ES (0x80 0x8B) modules that transmit Cprofile data carousels, independent from whether it is being presented or not, are always lockable using lockModuleOnMemoryEx(). 0x80 0x8B modules are simultaneously instructed to lock by lockModuleOnMemoryEx() from time to time, and thus it is necessary for receivers to be able to acquire 2ES modules at the same time. Operation to acquire a module at a time from 2 ES is not permitted. However, this

excludes cases where the total of the module size acquired at the same time exceeds 256KB.

- Always reserve 256KB of available space in receivers to acquire modules from the carousel, and do not perform the module fixation process if 256KB memory space cannot be secured by implementing lockModuleOnMemoryEx() function.
- Receivers shall always reserve the available capacity specified by the argument which specifies the available capacity of this function. Do not perform the module fixation process if the memory area concerned cannot be secured by implementation. The capacity specified by the argument shall not include the previously mentioned 256KB space. Interpret this as 0B if the argument is omitted.
- If it is judged by the PMT that the module specified by lockModuleOnMemoryEx() does not exist (such as the ES to transmit the concerned module not being included in the PMT, etc.), lockModuleOnMemoryEx() ends with a return value of -3. If the module specified by lockModuleOnMemoryEx() does not exist but cannot be judged by the PMT (such as an existing ES but not the module, etc.), lockModuleOnMemoryEx() ends with a return value of 1. And as soon as it is judged that the module concerned does not indeed exist, the ModuleLocked event is occured at status=-2.
- Even if the version of the module fixed in the memory is updated, the receiver does not automatically reacquire the module. The processes to detect module update, release the lock, and re-fix module to the memory shall be described in BML document.

- If transition to another document within the same service occurs before the completion of locking after executing lockModuleOnMemoryEx(), the locking operation will continue. In this case, the ModuleLocked event will be occured once this locking operation is complete in principle, but it may not happen depending on the document transition timing. If the data event of the ES, the lock object, is updated before the completion of locking after executing lockModuleOnMemoryEx(),the ModuleLocked event generates at status=-1, and the locking operation is not performed. If the lock object module is updated before the completion of locking after executing lockModuleOnMemoryEx(), the receiver locks the updated module.
- The procedure using lockModuleOnMemory() to update the presentation when updating the module that the object is referring to is described in ARIB STD-B24 Vol. 2 Appendix 1 “6.6.2 Relationship between update and module lock”, but the same effect can be acquired by using lockModuleOnMemoryEx() also.
- The procedure using lockModuleOnMemory() to update the presentation when updating the binary table that the object is referring to is described in ARIB STD-B24 Vol. 2 Appendix 2 “5.5.2.2 Behavior of BinaryTable”, but the same effect can be acquired by using lockModuleOnMemoryEx() also.
- The procedure using lockModuleOnMemory() to share JPEG images and binary tables within documents is described in ARIB STD-B24 Vol. 2 Appendix 2 “5.6.7 Operation of operational control functions”, but the same effect can be acquired by using lockModuleOnMemoryEx() also. It is described in Appendix 2 that sharing of resource documents is only valid within the document group, but it shall be valid within the contents group in this specification.
- In broadcasting reception status, the existence of modules can be confirmed by locking the module which includes the document at the transition destination by previously using lockModuleOnMemoryEx() before the transition to another BML document of another ES.
- If the ES that was transmitting the module locked by lockModuleOnMemoryEx() is deleted, it is implementation dependent on whether or not the module lock continues. It is recommended to release the lock of such modules from the contents explicitly. Transition to such modules cannot be performed.
- The module locked by lockModuleOnMemoryEx() can be unlocked by unlockModuleOnMemoryEx() independent from whether the module is included in the carousel being transmitted at the moment.
  (Hereafter, refer to 8.3.7 for the meaning of link state)
- In the following cases, the lock by lockModuleOnMemoryEx( ) is unlocked.

  - When it is explicitly released by execution of unlockModuleOnMemoryEx() or unlockAllModulesOnMemory().
  - When the service tuning in presentation is finished.
  - When the data event in presentation is finished (includes deletion of the component in presentation)
  - When the entry component is vanished due to a PMT update
  - When transition to the entry component due to execution of quitDocument()
- In the following cases, the lock by lockModuleOnMemoryEx() continues.
  - When the data event is updated in components not in presentation
  - When the locked module version is updated.
  - When the ES that was transmitting the locked module becomes an empty carousel.
  - When transition to another component in the same service as the document in presentation is performed.
  - When transition to link state.
- lockModuleOnMemoryEx() is fixable (including own components) within the service where the BML document in presentation is transmitted. Therefore, it stays fixed in memory unless the service tuning switches or there is a data event update of the carousel, where the BML document in presentation is transmitted, is extracted.

  Table 8-2 is a summary of conditions to release lockModuleOnMemoryEx.

Table 8-2 Release conditions of lockModuleOnMemoryEx

	Module locked by using lockModuleOnMemoryEx
Data event update in components in presentation	Unlock
Data event update in components not in presentation	No change
Update date of version of locked modules	No change
Tuning operation	Unlock
Transition to another component in the same service as the document in presentation	No change
Transition to entry component to quitDocument() (including cases where quitDocument() was executed during entry component viewing)	Unlock
Transition to link state	No change
When another application was started up by using X_DPA_startResidentApp(), and also when BML browser continues to be presented	No change

Operation example 1

A typical operation example of lockModuleOnMemoryEx() is shown in Figure 8-2. Services in this example consist of 2 components of component A and component B. The rhombus symbols over the straight lines which stand for components show the points where data events are updated in the component. The explanation below is according to the figure.

(a) Component A is presented initially. At time t1, component B module is locked by lockModuleOnMemoryEx().

(b) At time t2, it is moves to the component B module locked in (a). At this point, the lock on the module previously locked continues.

(c) At time t3, the component A module is locked by lockModuleOnMemoryEx().

(d) At time t4, it is moved to a module of component A locked in (c). The 2 modules locked previously continue to be locked.

(e) At time t5, data event of component A in presentation is updated, thus releasing all the locks on the modules previously locked by lockModuleOnMemoryEx().

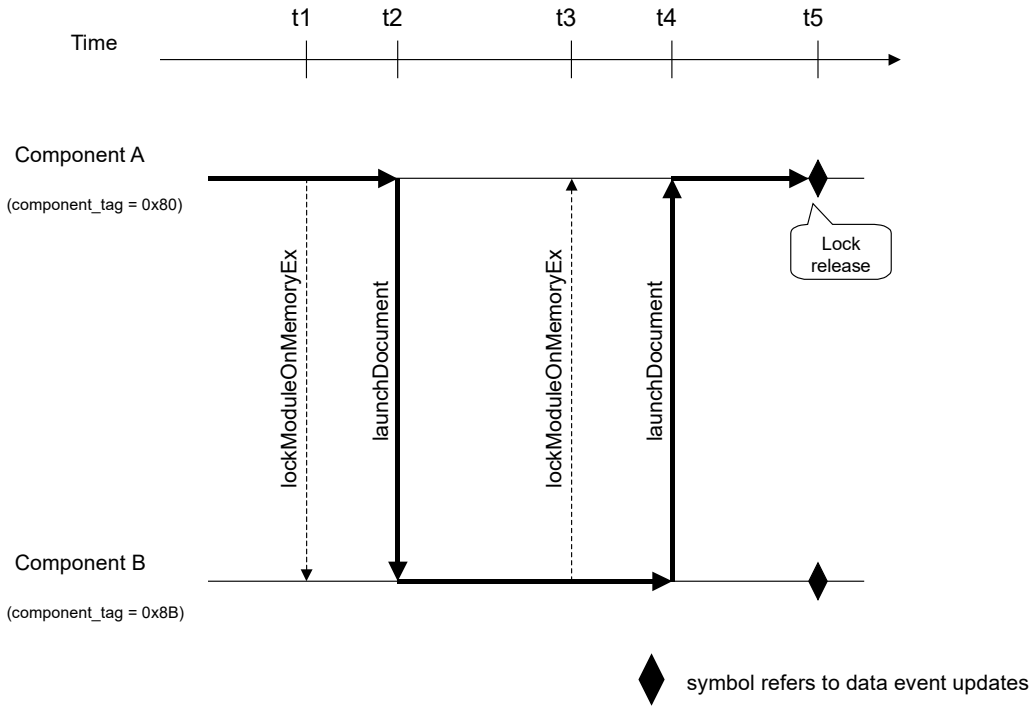


Figure 8-2 Example 1 of lockModuleOnMemoryEx operations

Operation example 2

A complicated operation example of lockModuleOnMemoryEx() is shown in Figure 8-3. As in operation example 1, service consists of 2 components of component A and component B. The explanation below is according to the figure.

(a) Component A is presented initially. At time t1, the presented module is locked by lockModuleOnMemoryEx().

(b) At time t2, the component B module is locked by lockModuleOnMemoryEx().

(c) At time t3, the module locked in (b) is moved to the component B module. At this point, the locks on the 2 modules previously locked continue.

(d) At time t4, the data event of component A is updated, but it does not affect the module already locked.

(e) At time t5, it is moved to previously presented module of component A. A new data event is already transmitted in component A, but the document presented by this transition is the document that was transmitted at the time of the lock in (a).

(f) At time t6, the component B data event is updated, but it does not affect the module already locked.

(g) At time t7, the data event of component A in presentation is updated, thus unlocking all the locks on modules previously locked by lockModuleOnMemoryEx().

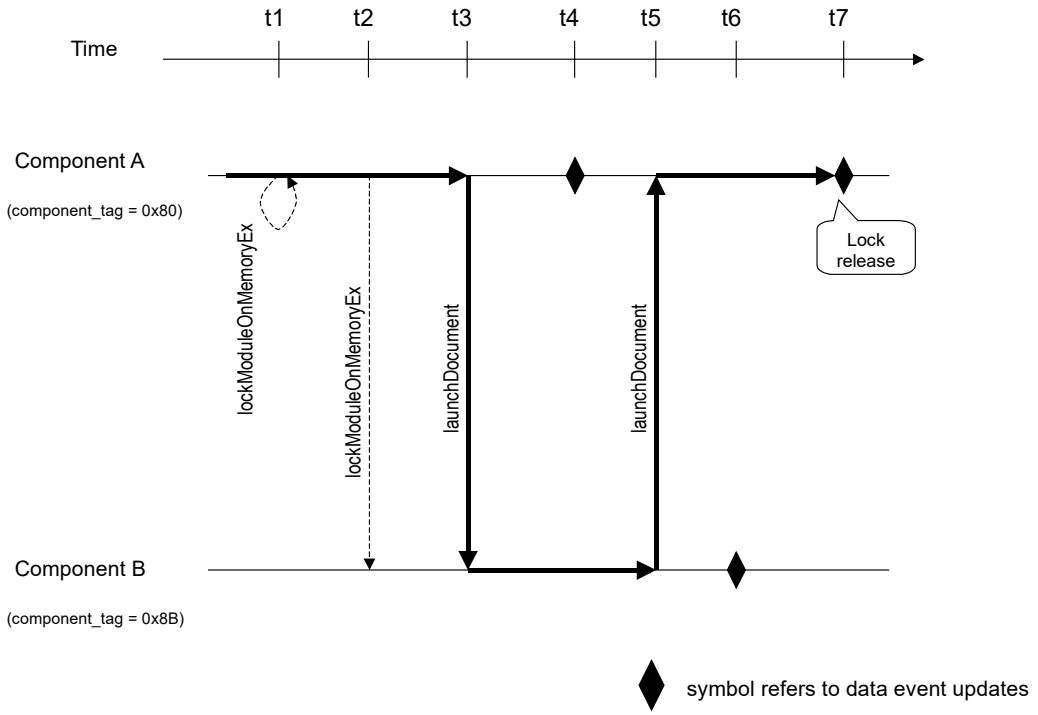


Figure 8-3 Example 2 of lockModuleOnMemoryEx operations

(2) Operation of launchDocument ()

Process of the second argument's transition effect based on the specification of transitionStyle is implementation dependent.

(3) Operation of launchDocumentRestricted()

This is not operated in C-profile.

(4) Operation of quitDocument()

Refer to "8.3.6.4 Operational control functions"

(5) Operation of getLockedModuleInfo()

The module name Array[n][0] (module name) of getLockedModuleInfo function's return value is a String object, and it is always stored in the:

/<component_tag>/<moduleName>

format.

Return value Array[n][1] and Array[n][2] are Number types.

If there is not even a single fixed or fixation-in-progress module, array with 0 length is returned, but receivers that return null are permitted as well.

When authoring contents, pay attention to the fact that one of the return value returns.

(6) Operation of detectComponent()

The component specified as the argument of detectComponent() is limited to the data component included in the service being presented. Follow the specifications in 8.2.3 to use the broadcasting contents, and use abbreviations starting with "/" for URI's that specify components. Follow the specifications in 8.3.10.4 to use in communication contents, and use absolute URI's starting with "arib-dc://-1.-1.-1" for URI's that specify components.

(7) Operation of getProgramRelativeTime()

Returns the starting time (start_time) of programs recorded in L-EIT and current time acquired by receivers per second.

(8) Operation of getActiveDocument()

Return abbreviated URI values (relative path starting with "/").

(9) Operation of getBrowserStatus()

Refer to Appendix 6 for argument operations.

(10) Operation of getResidentAppVersion()

This is necessary for function to acquire a TVlink list application version. In the return values from receivers with TVlink list implementation calling this function, specifying BookmarkList as appName, Array[4] returns empty strings.

ComBrowser is not specified in appName. If the TVlink list function is not implemented in receivers that do not possess communication functions, the return value after calling this function, specifying BookmarkList as appName, is null.

If the specified appName is not compatible, null is returned as the return value. (Note)

(Note) There is no appName specifications of operations other than BookmarkList in current operations because ComBrowser is not supposed to be specified, but these specifications are performed in case appName is added to specifications in the future.

(11) Operations of startResidentApp()

This is not operated in C-profile

(12) Operations of setFullDataDisplayArea() (option)

When receivers are specified full screen view (mode=1), the maximum area possible for data display (position and size of the area are up to receiver) is given to data display. At this point, it is not necessary for video to be displayed. However, it is recommended for audio to continue.

This function is an option but implementation is recommended.

(13) Operations of getBrowserSupport()

Arguments are operated as follows.

- "DPACpro" is specified in sProvider. Do not specify "ARIB".
- The corresponding relationships between the value of additional info, when the function name is "APIGroup", and the extended function are shown in the table below. Extended functions where the additionalinfo is blank are not operated.

Table 8-3 Value of additional info

	Function	Additional info
Ureg related functions		
	Ureg[]	Misc.Basic
Greg related functions		
	Greg[]	Misc.Basic
EPG functions		
	epgGetEventStartTime()	EPG.Basic
	epgGetEventDuration()	EPG.Basic
	epgTune()	
	epgTuneToComponent()	
	epgTuneToDocument()	
	epgIsReserved()	
	epgReserve()	EPG.Ext
	epgCancelReservation()	EPG.Ext
	epgRecIsReserved()	
	epgRecReserve()	
	epgRecCancelReservation()	
Event group index functions		
	grpIsReserved()	
	grpReserve()	
	grpCancelReservation()	
	grpRecIsReserved()	
	grpRecReserve()	
	grpRecCancelReservation()	
	grpGetNodeEventList()	
	grpGetERTNodeName()	
	grpGetERTNodeDescription()	
	epgXTune()	
Series reservation functions		
	seriesIsReserved()	
	seriesReserve()	
	seriesCancelReservation()	
	seriesRecIsReserved()	
	seriesRecReserve()	
	seriesRecCancelReservation()	
Permanent memory		
	readPersistentString()	
	readPersistentNumber()	
	readPersistentArray()	Persistent.Basic
	writePersistentString()	
	writePersistentNumber()	
	writePersistentArray()	Persistent.Basic
	copyPersistent()	
	getPersistentInfoList()	
	deletePersistent()	
	getFreeSpace()	
Functions for controlling access-controlled non-volatile memory areas		
	isSupportedPersistentType()	
	setAccessInfoOfPersistentArray()	
	checkAccessInfoOfPersistentArray()	
	writePersistentArrayWithAccessCheck()	
	readPersistentArrayWithAccessCheck()	
Interaction channel communication		

	Function	Additional info
Interaction channel communication– Delayed calls		
	registerTransmission()	
	registerTransmissionStatus()	
	getTransmissionStatus()	
	setDelayedTransmissionDataOverBasic()	
Interaction channel communication – BASIC procedures		
	connect()	
	disconnect()	
	sendBinaryData()	
	receiveBinaryData()	
	sendTextData()	
	receiveTextData()	
Interaction channel communication - TCP/IP		
	setISPParams()	
	getISPParams()	
	connectPPP()	
	connectPPPWithISPParams()	
	disconnectPPP()	
	getConnectionType()	Com.IP.GetType
	isIPConnected()	Com.IP
	saveHttpServerFileAs()	
	saveHttpServerFile()	
	sendHttpServerFileAs()	
	saveFtpServerFileAs()	
	saveFtpServerFile()	
	sendFtpServerFileAs()	
	sendTextMail()	
	sendMIMEMail()	
	transmitTextDataOverIP()	Com.IP.Transmit
	setDelayedTransmissionData()	
	getTransmissionStatus()	
	getTransmissionResult()	
	setCacheResourceoverIP()	
Interaction channel communication – Acquisition function in delayed call status common in BASIC procedures and IP connection		
	getDelayedTransmissionStatus()	
	getDelayedTransmissionResult()	
Interaction channel communication – Function to acquire line connection status		
	getPrefixNumber()	
Interaction channel communication – Mass calls reception service		
	vote( )	
Interaction channel communication – Encrypted communication using CAS		
	startCASEncryption()	
	transmitWithCASEncryption()	
	endCASEncryption()	
Interaction channel communication – Communication by encrypted code not using CAS		
	setEncryptionKey()	
	beginEncryption()	
	endEncryption()	

	Function	Additional info
Operational control function		
	reloadActiveDocument()	Ctrl.Basic
	getNPT()	
	getProgramRelativeTime()	Ctrl.Basic
	isBeingBroadcast()	
	lockExecution()	
	unlockExecution()	
	lockModuleOnMemory()	
	unlockModuleOnMemory()	
	setCachePriority()	
	getTuningLinkageSource()	
	getTuningLinkageType()	
	getLinkSourceServiceStr()	
	getLinkSourceEventStr()	
	getIRDID( )	
	getBrowserVersion()	Ctrl.Basic
	getProgramID()	Ctrl.Basic
	getActiveDocument()	Ctrl.Basic
	lockScreen()	Ctrl.Basic
	unlockScreen()	Ctrl.Basic
	getBrowserSupport()	Ctrl.Basic
	launchDocument()	Ctrl.Basic
	launchDocumentRestricted()	
	quitDocument()	Ctrl.Basic
	launchExApp()	
	getFreeContentsMemory()	
	isSupportedMedia()	
	detectComponent()	Ctrl.Basic
	lockModuleOnMemoryEx()	Ctrl.Basic
	unlockModuleOnMemoryEx()	Ctrl.Basic
	unlockAllModulesOnMemory()	Ctrl.Basic
	getLockedModuleInfo()	Ctrl.Basic
	getBrowserStatus()	Ctrl.Basic
	getResidentAppVersion()	Ctrl.RAVersion
	isRootCertificateExisting()	
	getRootCertificateInfo()	
	startResidentApp()	
	getDataDisplayAreaSize()	
	setFullDataDisplayArea()	Ctrl.MobileDisplay
Receiver sound control		
	playRomSound()	RomSound.Basic
Timer functions		
	sleep()	
	setTimeout()	
	setInterval()	Timer.Basic
	clearTimer()	Timer.Basic
	pauseTimer()	
	resumeTimer()	
	setCurrentDateMode()	Timer.DateMode
External character functions		
	loadDRCS()	
	unloadDRCS()	
External device control functions		
	enumPeripherals()	
	passXMLDocToPeripheral()	

	Function	Additional info
Other functions		
	random()	Misc.Basic
	subDate()	Misc.Basic
	addDate()	Misc.Basic
	formatNumber()	Misc.Basic
Caption display control functions		
	setCCStreamReference()	
	getCCStreamReference()	
	setCCDisplayStatus()	
	getCCDisplayStatus()	
	getCCLanguageStatus()	
Directory operation functions		
	saveDir()	
	saveDirAs()	
	createDir( )	
	getParentDirName()	
	getDirNames()	
	isDirExisting()	
File operation functions		
	saveFile()	
	saveFileAs()	
	getFileNames()	
	isFileExisting()	
File input/output functions		
	writeArray()	
	readArray()	
Inquiry functions		
	getDirInfo()	
	getFileInfo()	
	getCarouselInfo()	
	getModuleInfo()	
	getContentSource()	
	getStorageInfo()	
Data carousel storage functions		
	saveCarouselAs()	
	saveCarousel()	
	saveModuleAs()	
	saveModule()	Storage.Ext
	saveResourceAs()	
	saveResource()	Storage.Ext
Bookmark control functions		
	writeBookmarkArray()	
	readBookmarkArray()	
	deleteBookmark()	
	lockBookmark()	
	unlockBookmark()	
	getBookmarkInfo()	
	getBookmarkInfo2()	
	startResidentBookmarkList()	

	Function	Additional info
Print related functions API – Print basic functions		
	getPrinterStatus()	
	printFile()	
	printTemplate()	
	printUri()	
	printStaticScreen()	
Print related function API – Memory card related		
	saveImageToMemoryCard()	Print.MemoryCard1
	saveHttpServerImageToMemoryCard()	Print.MemoryCard2
	saveStaticScreenToMemoryCard()	
Digital terrestrial television broadcasting specific functions		
	X_DPA_mailTo()	Xdpa.mailTo
	X_DPA_startResidentApp()	Xdpa.RAStart
	X_DPA_phoneTo()	Xdpa.phoneTo
	X_DPA_getRcvCond ()	Xdpa.RcvCond
	X_DPA_getCurPos ()	Xdpa.CurPos
	X_DPA_saveExAppFile()	Xdpa.saveExApp
	X_DPA_startExAV()	Xdpa.startExAv
	X_DPA_stopExAV()	Xdpa.stopExAv
	X_DPA_tuneWithRF()	Xdpa.tuneRF
	X_DPA_writeSchInfo()	Xdpa.SchInfo
	X_DPA_getComBrowserUA ()	Xdpa.ComBrowserUA
	X_DPA_writeAddressBookInfo()	Xdpa.AddressBook
	X_DPA_launchDocWithLink()	Xdpa.launchWithL
	X_DPA_chkAVtype()	Xdpa.chkAV
	X_DPA_getIRDID()	Xdpa.getIRDID
	X_DPA_writeCproBM()	Xdpa.CproBM

Below are specifiable value combinations for arguments of getBrowserSupport().

functionname	additionalinfo	Operations of getBrowserSupport()
ResidentApp	"ComBrowser"	If a communication browser is installed in the receiver as a receiver's native application, return 1
	"Bookmark"	If a TV link list function is installed in the receiver as a receiver's native application, return 1
	"JapaneseInput"	If a character input function (refer to 3.6) is installed in the receiver as a receiver's native application, return 1
WriteCproBM	"BMtype02"	If the receiver can write TVlink of CproBMtype=2, return 1
	"BMtype03"	If the receiver can write TVlink of CproBMtype=3, return 1
	"BMtype04"	If the receiver can write TVlink of CproBMtype=4, return 1
OSDPixel size	"240x480"	If the receiver possesses 240 x 480 coordinate area as the imaginary plane for data broadcasting, return 1
BMLversion	"major number. minor number"	If the browser supports playback of BML documents in the version specified by additionalinfo, return 1. If Additionalinfo is omitted, "12.0" is considered to be specified.
MediaDecoder	"scheme name" ,"media type" Refer to ARIB STD-B24 Vol. 2 (1/2) Annex C for scheme name and media type. If the "media type" is "audio/X-arib-mpeg2-aac", specify "sampling frequency" (Hz unit) as the third additionalinfo. (If omitted, "48000" is considered to be specified)	If the browser possesses a decoding function of mono-media that is identified by media type and scheme name, return 1. If the media type is "audio/X-arib-mpeg2-aac", return 1 only when it corresponds to specified sampling frequency.
APIGroup	Extended function group specification Refer to table 8-3 for correspondence to extended function for each broadcaster and extended function group specification.	If it is possible for the browser to implement all of extended function group that was specified by argument's extended function group specification, return 1.
TransmissionProtocol	"application", "HTTP" [, HTTP version]	Return 1 if it possesses a interaction channel communication function using HTTP, whose version is the version specified in [HTTP version], in the priority connection line's connection type used by the browser. It is interpreted as 1.0 if the version number is omitted.
	"application", "TLS" [, TLS version]	Return 1 if it possesses a secure interaction channel communication function by TLS, in the priority connection line's connection type used by the browser. It is interpreted as 1.0 if the version number is omitted.
Storage	"cachesize", size	Return 1 if the browser possesses content memory specified by size. (Note) Size unit shall be 1024 bytes. For example, specify "512" if 512Kbytes.
BookmarkButton		Return 1 if the receiver possesses a TVlink button.
AudioFile	filesize	Return 1 if the audio content in the file type that is the size (in byte unit) specified by filesize is available for playback.

(14) Operation of lockScreen()

When lockScreen() is started up, the presentation screen changes from DOM and the API is stopped until unlockScreen() is started up. However, when the event handler ends by lockScreen() without starting up unlockScreen(), a presentation screen change is performed immediately after it ends.

If multiple values are set within the same attribute or multiple method callings are performed, only the last change shall be valid.

If lockScreen() is called multiple times before unlockScreen() is called, calls after the first one will normally be ignored. Nesting of the lock process is not performed. It is implementation dependent whether or not to stop the rewriting of the presentation frame of animation GIF during the operation of lockScreen().

Presentation and operation of object elements whose type attribute starts with audio are not influenced by the screen lock status.

(15) Operation of unlockScreen()

If unlockScreen() is called without lockScreen() being called, it will normally be ignored. Even when lockScreen() is being called multiple times, the lock can be released by starting up unlockScreen() once.

(16) Operation of setInterval()

The first argument, with the same description as the calling of functions as the event handler, is controlled operationally only in "func();" type. The maximum timer number that can be set simultaneously is 4, and the setting shall be in 100msec units. Depending on the number of interrupting seconds, it is permitted that accuracy corresponds to each receiver.

8.1.15.6 Non-volatile memory functions and operation of the data carousel storage function

(1) Operation of writePersistentArray()

Period, as an omittable argument, does not require operation. Lifetime control of the written data is not performed, and it is always overwritable. Specification of structure is the same as BinaryTable objects, but the following restrictions apply.

Useable field type	Corresponding data type	Operation restriction
B	Boolean	Error if byte aligning does not match
I	Number	Cut off upper bit if it is less than 32 bit. Error if value over 32 bit is set.
U	Number	Cut off upper bit if less than 32 bit. Error if a value over 32 bit is set. Error if a negative value is written.
S	String	Cut off if it exceeds the specified character length. Add space(0 x 20) if it is too short. The variable length string is specified as S:1V.
P	--	

If the actual data and BinaryTable data type are different, utilize the type conversion specifications in ARIB STD-B24 Vol. 2 Appendix 4 5.4.2.2, but return an error if it is not appropriate. Also, if the array data is not sufficient for the field number specified by structure, it is an error. Operations are not guaranteed with arrays of two or more dimensions.

(2) Operation of readPersistentArray()

Operation of structure shall be the same as the writePersistentArray(). Operations are not guaranteed if reading was performed with a structure that is not the structure used in the writePersistentArray().

For operation of the data carousel storage function, T.B.D.

8.1.15.7 Operation of functions related to TVlinks

When recording C-profile contents (CproBMtype=1) as a TVlink, there are cases in which broadcasting is not receivable when calling the concerned link contents. Therefore, consider the following possibilities when authoring contents.

1. Time handled with ECMA Script's Date object is not accurate, or error is returned.
2. Possibility that event EventMessageFired does not occure.
3. Possibility that event ModuleUpdated does not occure.
4. Possibility that the extended function returns an error because broadcasting is not receivable.

8.1.15.8 Operation of special function for terrestrials

(1) Operation of X_DPA_mailTo() (optional)

Delivers the address and contents to the argument and transmits e-mail. The application that the receiver possesses is what actually transmits the e-mail, and transmission operation and permission procedure are implementation dependent.

(2) Operation of X_DPA_phoneTo () (optional)

This calls the telephone number specified by the argument. Follow "8.1.4 Prohibition of mixed display and simultaneous display" when this function is called, and pay extra attention to mixed display when using the TV telephone function. The calling operation and permission procedure are implementation dependent.

(3) Operation of X_DPA_ getRcvCond () (optional)

This acquires communication status in the communication line. The layout procedure and permission procedure, when the return value status of this function is different from the

classification of the radio receive condition within the communication function, are implementation dependent.

(4) Operation of X_DPA_ getCurPos () (optional)

This returns the current location information of the receiver based on the receiver's location information acquisition function, using Global Positioning System geodetic information using GPS satellites and information acquired from base stations established for mobile phones, etc. It is recommended to return the latitude and longitude of decimal number writing based on the global positioning system for the return value of this function. Permission, etc. when calling this function is implementation dependent.

(5) Operation of X_DPA_saveExAppFile() (T.B.D.)

(6) Operation of X_DPA_startExAV () (optional)

To display streamed video acquired via communication using this function, stop the presentation of video and audio of broadcasting, in principle, and display the video via communication in the display area of the broadcasting video. However, it is recommended to continue to present data broadcasting.

Receivers shall consider the viewers so that they can identify that it is video acquired via communication.

The return value is not returned if BML browser stops when starting up the AV playback application, etc. This function is useable in broadcasting contents and contents in link state.

Follow "8.1.7.6 Display of videos acquired from communication" when this function is called.

(7) Operation of X_DPA_stopExAV () (optional)

This stops the presentation of video sound streamed via communication presented by the (X_ DPA _startExAV ()) function. It is recommended to return the display back to broadcasting video and audio when the video sound stream via communication is complete. With receivers that stop BML browser when starting AV playback applications, etc., nothing other than returning NaN is performed if this function is implemented.

(8) Operation of X_DPA_tuneWithRF() (optional)

When the service in current tuning is re-selected by this function, BML browser performs the same operation as tuning. Only broadcasting contents shall be presented on the screen immediately after tuning, according to "8.1.3 Operation during the startup of the TV reception function and tuning". It is recommended to continue presenting the video section without sudden cut-off's, noise generation, etc.

The operation when tuning fails upon the calling of this function is implementation dependent.

(9) Operation of X_DPA_writeSchInfo() (optional)

This starts up the schedule management application within receivers and delivers the specified information to the argument. Some schedule contents, which exceeds the maximum character number restriction in each item of schedule in the receiver features, may not be saved. It is recommended to save the schedule title as the schedule book's item title, but it is implementation dependent to choose which column to allot. An alarm can be set to ring at specified time and date, but the alarm sound correspondence and sound color settings are implementation dependent. The function is defined, supposing to notify programs. However, this function's implementation does not need to be registered among the schedule book which the user normally writes in, and can be implemented on its' own.

The operation upon schedule management application startup and information saving procedure, etc., are implementation dependent.

(10) Operation of X_DPA_writeAddressBookInfo() (optional)

The startup of the address book within receivers and the address book control function, and transmission information such as the name, telephone number, e-mail address, etc. are specified in the function to the address book. Mixed existence with address books that already exist as receiver features is not necessary.

For mixing the existence of the information of concerned functions in existing address books, is implementation dependent for whether or not to register items that do not match.

(11) Operation of X_DPA_startResidentApp ()

This is mandatory for functions to startup TVlink list. Other receiver's native applications startup is optional. Also, for receivers that do not possess the communication function, implementation of the TVlink list function is not necessary. If this function is called, NaN is returned as the return value.

(12) Operation of X_DPA_getComBrowserUA () (optional)

If multiple browsers are installed in a receiver, all browser information returns to 2 dimensional array. To connect to a proprietary network of the carrier such as the communication browser, mobile phone, etc. concerned, the communication company's ID shall be returned as the maker ID. Contents utilize this function in order to identify communication companies.

(13) Operation of X_DPA_launchDocWithLink ()

Unlike the launchDocment() function, the base URI directory is changed if used in link state. This changes the base URI directory to the URI directory specified in the function, and transmits it.

(14) Operation of X_DPA_chkAVtype () (optional)

This is a function that confirms the type of video sound from communication contents. The receiver confirms whether it corresponds to the video sound stream type that was specified as the function, before the (X_DPA_startExAV ()) function implementation.

(15) Operation of X_DPA_getIRDID ()

This is a function to acquire the ID that identifies the receiver and viewer. It can acquire the receiver-unique identifier and the viewer specific identifier by argument specification. The type of identifier, permission for function call, etc., are implementation dependent.

8.1.15.9 Operation of print functions (optional)

Printing function

Refer to and based on ARIB STD-B24 Vol. 2 “7.6.17 Functions for printing” and Vol. 2 Appendix 1 “Guideliines on functions for printing” for functions and specifications regarding printing.

Extended API group

Printing functions are receiver implementation options. Therefore, when printing related functions are utilized in contents, examine whether printing related processing is available in the receiver by getBrowserSupport(), and call the printing related function only when the processing are available.

Printing related functions are divided into the following groups:

A) Function group to print using printers

- getPrinterStatus()
- printUri()

B) Function group to store data for printing in receiver’s internal memory, memory card, etc.

- saveImageToMemoryCard()
- saveHttpServerImageToMemoryCard()
- saveStaticScreenToMemoryCard()

Receivers that support printing functions shall extract each function implementation status by specifying “APIGroup” in the function name of getBrowserSupport().

API in A) is not operated in C-profile.

API in B) is useable even when the printing device is offline. However, saveStaticScreenToMemoryCard() is not operated. The following procedures are options to print the saved data within a receiver or memory card.

- Use the printing function by outputting the printing data saved within a receiver or memory card that was implemented as receiver features to the printer when the printer is on-line.
- Also, in some cases, a memory card can be utilized as the bridge media to deliver data to printing devices.

Table 8-3 provides the extended function group specification to specify as additionalinfo when functionname is "APIGroup".

Printing data format

For still image files (JPEG) for printing, in case they are to be shared by both display and printing in the BML, and there is a case only for printing. For each case, operate as follows.

	Still image file	Operation
Shared by both display and printing	JPEG	Refer to 5.2.1
Printing by itself	JPEG	ISO/IEC10918-1 base line, JFIF (Jpeg File Interchange Format) and Exif Maximum pixel size 640x480 Maximum file size 256KBytes Sampling 4:2:0 or 4:2:2 Pixel aspect ratio 1:1 (square pixels)

- Still image file format specifiable by saveHttpServerImageToMemoryCard()

  The still image file format that can be specified by this function is JPEG, specified in the above table. Each file name extension is “jpg”. The maximum size for the still image files is specified in the above table.

Supplemental items regarding each printing related API

- URI specified by saveHttpServerImageToMemoryCard()

  For the URI, which is specified by saveHttpServerImageToMemoryCard(), only resources that are described with http:// or https:// and are 256 Bytes or less are specifiable.

Presentation by receiver

Even when implementing the store functions (saveImageToMemoryCard() and saveHttpServerImageToMemoryCard()) of printing data on memory card, the BML contents shall present a reminder message for memory card insertion, and processing overlapping file names, etc. The storage directory name within the receiver or memory card is implementation

dependent.

Even during the display of messages etc., by the receiver within the printing function, the process from document cancellation to re-presentation will be executed if there is a data event update.

In the following cases, the receiver system shall delete the message/interface and discontinue the process.

- When a data event update is occured in the currently viewed ES.
- When tuning

When http:// (or https://) is specified in saveHttpServerImageToMemoryCard function, the receiver acquires printing data within the function via communication. Therefore, there may be cases where it takes some time from the function call until the return. If an event is generated while the receiver is acquiring the printing data via communication, the event will be added to the interrupting queue. However, it will not be implemented until it is returned from the function, so be sure to consider this point.

8.1.16 Built-in objects

Time handled by the Date object shall be time corrected by the TOT or another procedure.

The Date object shall be able to acquire values down to figures of 0.01 seconds. It is recommended to avoid implementation that returns figures 0.1 seconds and 0.01 seconds in absolute value. (Note) Even though the accuracy of the Date object's absolute value is assumed to be about the accuracy level of TOT, it is recommended that the accuracy in the difference between 2 Date objects is plus/minus 0.1 seconds or less. It is only necessary for editing by TOT, etc. to be performed at the time of tuning.

Time handled by the ECMA Script Date object shall be the time JST (UTC + 9hours) that does not include summer daylight savings time offset in the calculation. If local time with summer daylight savings time is offset, consideration is necessary in order to display the current time etc., and local time shall be acquired by adding the time offset by using addDate(), etc., to the time acquired the Date() function in the contents.

Time handled by the Date object within the communication contents display shall follow the time information which the broadcasting contents were referring to before it changed to communication contents.

- (Note) The reason that acquisition of a figure 0.01 seconds is necessary is (like when acquiring the time difference from answering start time and time the button is pushed in a quiz game, for example) because the application to acquire the relative time difference between 2 points which are relatively close is assumed.

8.1.17 Other restrictions

Refer to ARIB STD-B24 Vol. 2 Appendix 4 “5.7. Other restrictions”

8.2 Transmission, reference and name space of content

8.2.1 Scope mapping to the transmission system

Follow ARIB STD-B24 Vol. 2 Appendix4 “6.1. Scope mapping to the transmission system”. Refer to 8.3.7 for communication contents.

8.2.2 Restrictions when a mono-media is referred across a different media

In C-profile, reference of mono-media, etc. is not performed other than in the following cases. Refer also to 8.3.9 for communication contents.

- Tuning by functions for tuning
- Event message reference when the reference origin is C-profile linked contents and the reference destination is broadcasting contents
- To subscribe the event in th URI beginning with "arib-dc://-1.-1.-1" in C-profile linked contents (refer to table 7-14)

8.2.3 Operations of name space

Follow ARIB STD-B24 Vol. 2 Appendix 4 “6.3 Namespace”. Refer to 8.3.10 for C-profile linked contents.

The following specifications in this volume are added for broadcasting contents.

- Reference for other services shall only be available in the following extended functions for broadcasting.

  X_DPA_tuneWithRF(), epgReserve() and epgCancelReservation()
- Other than the following cases, refer to the abbreviated form (ARIB STD-B24 Vol. 2 9.2) for the name space description when specifying broadcasting contents within BML documents.
  - Reference to other services
  - Argument of the extended function for broadcasting whose argument is an event specification
  - Reference of an event message transmitted by broadcasting from C-profile linked contents
- Always omit event_id, excluding argument for the extended function for broadcasting whose argument is an event specification

- As described in ARIB STD-B24 Vol. 2 “9.2.5.1 Identification of currently selected broadcasting service on receiver”, currently selected service is considered to be specified if "arib://-1.-1.-1" was specified as the service name. Here, currently selected service refers to the service currently being received if it is in broadcasting reception status.
- For the “href” attribute of the “a” element and launchDocument(), the BML document that is transmitted with the component which was included in the same service as the BML document being presented or C-profile linked contents placed in a server can be specified. For the “action” attribute for “form” element, C-profile linked contents placed in servers is specifiable. For X_DPA_launchDocWithLink(),C-profile linked contents placed in servers is specifiable.
- Operation of #fragment is available, according to ARIB STD-B24 Vol. 2 Appendix 4 “6.3 Namespace”, but restrictions specified in 8.2.3.4 are established.

8.2.3.1 Restriction for the number of resources which are managed by receiver

- The maximum number of resources (including both broadcasting contents and C-profile linked contents) simultaneously securable in receiver’s content memory is 256. For the restriction, it should keep the total number of resources (which possess specific name spaces) in a single data event period within 256. However, if the resource number can be limited within the limitation above at one time when authoring, the total number of resources in a data event period may be 256 or more.
- If, against the above restriction, fixation of resources that exceed the above number were specified due to lockModuleOnMemoryEx(), etc., the receiver may not perform this.
  Also, resources refer to the following two here.
  - Resource directly mapped in modules
  - Resource stored in modules in HTTP/1.1’s entity format

8.2.3.2 Name space regarding multiple ES module locks

- Regardless of ES being presented, all data carousel modules within the same service can be specified as an argument of lockModuleOnMemoryEx().

8.2.3.3 Name space regarding multiple ES module version watch

- Modules that watch module version updates (module_ref whose type attribute is “bml:beitem” element of ModuleUpdated) is also specifiable for any module in a data carousel of C-Profile within the same service.

8.2.3.4 Reception operation and contents guidelines in #fragment operations

- It is recommended to present documents without reloading if only #fragment is used in the “href” attribute of the “a” element and launchDocument() and transition within the same

document is specified.
(ex: browser.launchDocument("#top", "cut");)

- However, it is recommended to reload in cases like below, even if transition within the same document was specified.
(ex: browser.launchDocument("startup.bml#top", "cut");))
- Follow the specification and present the document if transition to another document was specified using #fragment by means of specification of X_DPA_launchDocWithLink() and the “href” attribute of the “a” element and launchDocument().
- Presentation that includes elements specified by #fragment if transitioned by URI specifications that use #fragment., the presentation is implementation dependent.

8.2.4 Reference guidelines of contents transmitted by components different from the BML document being presented

- In C-profile, resources which are transmitted by components different from BML document being presented can be referred to. In such occasions, conduct locking by lockModuleOnMemoryEx().
- With resources that will be required immediately after BML document presentation starts, such as the JPEG referred to as the background-image of body, it is necessary to pay attention to make sure locking completely by lockModuleOnMemoryEx() is performed before BML document presentation upon creating contents.
- Specifications below are specially established for reference of ECMAScript and CSS, that are transmitted as independent resources.
  - When referring to an ECMAScript and CSS from the BML document of a component different from the component in which they are transmitted, perform locking of ECMAScript and CSS by lockModuleOnMemoryEx(), and perform transition to the BML document after the completion of locking. Operation when document transition was performed without completing locking is implementation dependent.
  - When referring to the CSS and ECMAScript that are transmitted in components different from BML document being presented, there are cases that CSS and ECMAScript are not referable to from the vanished ES, etc. In that case, failures as above can be avoided by transmitting CSS data and ECMAScript data in a data entry component.
  - Even when referring to CSS and ECMAScript transmitted in the BML document being presented from the BML document, it is recommended to lock ECMAScript and CSS prior to the BML document transition concerned and perform transition after the completion of locking.

8.3 Operation of C-profile communicating content

8.3.1 Guidelines regarding presentation for C-profile communicating contents

Same as 8.1 ”Guidelines for presentation”

8.3.2 Guidelines for operations of external fonts in C-profile communicating contents

Same as 8.1.12 “Guidelines for external characters operation”

8.3.3 Operation of DOM in C-profile communicating contents

Same as 8.1.11 “Guidelines for DOM operation”

8.3.4 Operation of ECMAScript implementation script in C-profile communicating contents

Same as 8.1.13 “Guidelines for ECMA script implementation”

8.3.5 External objects for broadcasting in C-profile communicating contents

Same as 8.1.14 “Guidelines for operation of extended objects for broadcasting”

8.3.6 Operation of browser pseudo-objects in C-profile communicating contents

The behaviourof the extended function for broadcasting which is in browser pseudo-objects in C-profile communicating contents differs depending on the status (data broadcasting reception status and link status) of the receiver. Refer to 8.3.7 for the basic concept of data broadcasting reception status and link status. Actual function operation is specified from Table 8-4 to Table 8-15. The meanings of “O”, “O(*1)”, “O(*2)”, “O(*3)”, “X” and “-” are specified below.

“O”	Basic function in this volume.
“O(*1)”	Optional function in this volume. Therefore, if these functions are used in contents, inspect the availability of this function in receiver by the getBrowserSupport() function. Only when it is available for processing can this function be called.
“O(*2)”	Basic function in this volume in principle. However, implementation of these functions is not necessary only for receivers that cannot utilize the communication function from BML browsers. When those functions is called, the value for the failure is returned as the return value.
“O(*3)”	Necessary for receiver that possess a function to playback the partial TS. If these functions are used in contents, inspect the availability of processing this function in receivers by the getBrowserSupport() function. Only when it is available for processing can this function be called.
“X”	Implementation is prohibited in broadcasting status, link status and browser for the C-profile contents on the internets. If the function is

called, receiver performs a failure behaviour. Refer to 8.3.11.4 for specifications in this volume of failure behaviours.

"-" Neither basic functions nor optional functions in this volume. If the function is called, an error is ocurred in the receiver.

8.3.6.1 "Ureg" function

Operations of "Ureg" functions in C-profile linked contents are specified in Table 8-4. Refer to 8.1.15.1 and 8.1.15.2 for operations in data broadcasting reception status.

Table 8-4 Behaviours of the Ureg function in C-profile linked contents

	Link status
Ureg[ ]	O
Greg[ ]	O

8.3.6.2 EPG functions

Operations of EPG functions in C-profile linked contents are specified in Table 8-5. Refer to 8.1.15.3 for behaviours in data broadcasting reception status. epgReserve() and epgCancelReservation() are options.

Table 8-5 Behaviour of EPG functions in C-profile links

	Link status
epgGetEventStartTime()	O
epgGetEventDuration()	O
epgReserve()	O(*1)
epgCancelReservation()	O(*1)

8.3.6.3 Interaction channel function -TCP/IP

Operations of TCP/IP are specified in Table 8-6. Details regarding behaviour of each function are described below.

Table 8-6 Behaviours of TCP/IP interaction channel communication in C-profile linked contents

	Link status
getConnectionType()	O(*2)
isIPConnected()	O(*2)
transmitTextDataOverIP()	O(*2)

- getConnectionType() function behaviours

This function is only utilized to acquire hintn in order to estimate communication speed from information on what kind of line connection is made.

This is used for confirmation of preferred line type in ARIB STD-B24 Vol. 2 Attachment 1

Informative Explanation 4. However, provisions of getConnectionType( ) do not grasp all of the line types and thus cases that do not fit in the sequence are in Informative Explanation 4. Therefore, note that it is only used as hint.

Refer to ARIB STD-B24 Vol. 2 Attachment 3: 5.6.5.2 for details on the return value of the function. In environments where the receiver is connected by using a line type not described in Appendix 3, this function returns NaN. This function returns NaN in receiver that do not possess a communication function.

Also, return the fixed value “300” (mobile phones (if type cannot be differentiated)) in mobile phones.

- isIPConnected() function behaviour

Refer to ARIB STD-B24 Vol. 2 Attachment 3: 5.6.5.2 for details on the return value of the function. Using this function, receiver judge if resources on the internet are acquirable and returns the value. This function returns NaN in receiver that do not possess the communication function.

Also, return the fixed value “1” (IP connection is established by automatic connection) in mobile phones.

- transmitTextDataOverIP() function behaviour

Operate in link status.

This is a function that assumes telegram transmission and reception is performed in the communication network. It is mainly utilized from broadcasting contents. The restrictions below are established for the arguments of this function.

- Scheme for URI as argument are only 2 types: "http" and "https".
- The maximum length of the string for argument text is 4096Bytes.
- Character code of text data to transmit shall be a fixed operation of only "Shift_JIS". Therefore, specify "Shift_JIS" for the third argument charset.

When receiver transmit text to servers, they transmit to the URI specified by the argument using the POST method. If there is no text to transmit, specify empty string (“”) for the argument.

Receiver specify "application/x-www-form-urlencoded" as Content-Type within the requirement message, and "Denbun" for name attribute. Also, follow the URI encoding provisions specified in RFC2396 (2.4.1 Escaped Encoding) for text encoding to encode. Upon this, Japanese strings should be encoded as Shift-JIS character code. When an empty string (“”) is specified in the text, specify 7 in Content-Length to consider a string of "Denbun=".

Figure 8-4 provides the requirement message upon transmitting text “transmit telegram 20060101” to the server.

```
POST http://localhost/test.cgi HTTP/1.1
…header omitted…
Content-Type: application/x-www-form-urlencoded
Content-Length: 85

Denbun=%74%72%61%6E%73%6D%69%74%20%74%65%6C%65%67%72%61%6D%20
%32%30%30%36%30%31%30%31
```

Figure 8-4 An example of requirement messages which transmit TextDataOverIP( )

Servers having processed the requirements may add text data in addition to the response status if there is any text data to be delivered to the receiver. In that case, specify Content-Type as "text/plain", and Charset as "Shift_JIS".

Figure 8-5 specifies an example of a response message which contains the text data “Text will be returned” in the entity-body.

```
HTTP/1.1 200 OK
…header omitted…
Content-Length: 22
Content-Type: text/plain; Charset=Shift_JIS

Text will be returned.
```

Figure 8-5 An example of respose message that transmitTextDataOverIP( ) receives

Upon this, the maximum entity-body size that receiver receives shall be 4096Bytes. If a receiver receives text that exceeds the limit, text reception process after 4097Bytes is implementation dependent. The letter code of text data shall be a fixed operation of "Shift_JIS" only.

Table 8-7 specifies detailed contents regarding return values.

Table 8-7 Return value of transmitTextDataOverIP() (Array[0])

Return value		Meaning	Detailed Content
Array[0]	1	Success	Argument text was successfully transmitted to the internet resource specified by argument uri and received its response.
	-1	Wrong parameter	Wrong argument given to the function
	-2	Line was disconnected during transmission	Physical line was disconnected during data transmission and reception
	-3	Time-out	Data transmission process and reception process did not complete within certain period of time

Return value		Meaning	Detailed Content
	-300	Failure upon automatic connection	An error occurs upon automatic connection and connection to the Internet resource specified by argument uri was not performed. This error value is returned when error generates during physical line connection process, PPP and TCP process.
	-400	Failure upon DNS name conversion	Host name specified by argument uri was not able to convert to IP address
	-500	Failure upon TLS process	Failed in TLS process performed upon HTTPS use
	NaN	Failure by other cause	Condition of restrictions of automatic retransmission was detected.

8.3.6.4 Operational control functions

Table 8-8 specified the operation of function for operational control function in C-profile linked contents. Then, details for each function operation is provided. Refer to 8.1.15.5 for function operations of the data broadcasting reception status.

Table 8-8 Operations of C-profile linked contents operational control function

	Link status
reloadActiveDocument()	O
getProgramRelativeTime()	O
getBrowserVersion()	O
getProgramID()	O
getActiveDocument()	O
lockScreen()	O
unlockScreen()	O
getBrowserSupport()	O
launchDocument()	O
quitDocument()	O
detectComponent()	O
lockModuleOnMemoryEx()	X
unlockModuleOnMemoryEx()	O
unlockAllModulesOnMemory()	O
getLockedModuleInfo()	O
getBrowserStatus()	O
getResidentAppVersion()	O(*2)
setFullDataDisplayArea()	O(*1)
saveImageToMemoryCard()	O(*1)
saveHttpServerImageToMemoryCard()	O(*1)

- reloadActiveDocument() function behaviour

When this function is called, the receiver should re-acquire BML document being presented and mono-media from the server. If already acquired BML documents and mono-media exist

within a receiver, the receiver may display without change. However, such cache function shall be implementation dependent. When the receiver is playing C-profile linked contents that cannot specify the resource name, such as when "/" was specified at the end of URI or when ?query was specified, perform to re-acquire the currently presented C-profile linked content by re-transmitting the URI that specified the concerned BML document to the server.

For example, the operation when C-profile linked contents transitioned by launchDocument("http://localhost/hoge/", "cut") calls the reloadActiveDocument function shall be the same as when launchDocument function is called.

- getProgramRelativeTime() function behaviour

  Behaviour in link status shall be the same as the data broadcasting reception status.

- getBrowserVersion() function behaviour

  Behaviour in link status shall be the same as the data broadcasting reception status.

- getProgramID() function behaviour

  Behaviour in link status shall be the same as data broadcasting reception status.

- getActiveDocument() function behaviour

  This operates in link status. Return value shall be returned in a format that starts with "/" (abs_path specified in RFC1808), omitting scheme and host name.

  When receiver is playing C-profile linked content that cannot specify the resource name, such as when "/" was specified at the end of URI or when ?query was specified, return value excluding scheme and host name of URI that specified the concerned BML document.

  For example, the return value is “/hoge/foo?query" when getActiveDocument function was called by C-profile link content transitioned by launchDocument("http://localhost/hoge/foo?query", "cut").

- lockScreen() function behaviour

  Behaviour in link status shall be the same as the data broadcasting reception status.

- unlockScreen() function behaviour

  Behaviour in link status shall be the same as the data broadcasting reception status.

- getBrowserSupport() function behaviour

  Behaviour in link status shall be the same as the data broadcasting reception status.

- launchDocument() function behaviour

  This is operated in link status. Restrictions below are established in link status.

  - Restriction in link status

- When transitioning to broadcasting contents, specify the absolute URI that starts with "arib-dc: ". Refer to 8.3.10 for details.
- If outside of the base URI directory scope, the data broadcasting browser should be in a failed operation and transitioned to broadcasting contents. Refer to 8.3.11.3 for details.

- quitDocument() function behaviour

Behaviour of the quitDocument() function differ depending on the data broadcasting reception status and link status. Refer to Table 8-9. Refer to 8.3.11.3 for status transition.

Table 8-9 behaviourof quitDocument()

Status	Behaviour of quitDocument()
Data broadcasting transmission/reception status	When quitDocument() is executed, the presentation of currently presented broadcasting contents ends, and it is executed step 3 or later for “Receiver operation at the beginning of data broadcasting” described in 4.1.5.1, and is transitted to the startup document in the entry component. Upon this, all modules locked by lockModuleOnMemoryEx function are released.
Link status	When quitDocument() is executed, the presentation of the currently presented C-profile linked contents ends, and it is executed step 3 or later for “Receiver operation at the beginning of data broadcasting” described in 4.1.5.1, and is transitted to startup document in entry component. Upon this, all modules locked by lockModuleOnMemoryEx function are released. If connection was performed at this time, the receiver may disconnect the connection, following 8.3.12.2. For example, the receiver unit may make a judgment whether to disconnect or not, depending on the status such as communicating with other than BML browser, etc. when quitDocument() is specified.

- detectComponent() function behaviour

Behaviour in link status is the same as data broadcasting reception status. Refer to 8.1.15.

- lockModuleOnMemoryEx() function behaviour

This does not operate in link status.

- unlockModuleOnMemoryEx() function behaviour

Behaviour in link status is the same as data broadcasting reception status. Refer to 8.1.15. Use of this function in link status can be thought of as releasing the broadcasting contents module locked status by lockModuleOnMemoryEx()in broadcasting reception after its transition to link status.

- unlockAllModulesOnMemory() function operation

Behaviour in link status is the same as data broadcasting reception status. Refer to 8.1.15. Use of this factor in link status can be thought of as releasing the broadcasting contents module

locked by lockModuleOnMemoryEx()in broadcasting reception status after its transition to link status.

- getLockedModuleInfo() function behaviour

Behaviour in link status is the same as data broadcasting reception status. Refer to 8.1.15. Use of this function in link status can be thought of as releasing the broadcasting contents module locked by lockModuleOnMemoryEx() in broadcasting reception status after its transition to link status.

- getBrowserStatus() function behaviour

This operates in link status. Behaviour in link status is the same as the data broadcasting reception status.

- getResidentAppVersion () function behaviour

This operates in link status. Behaviour in link status is the same as the data broadcasting reception status.

- setFullDataDisplayArea () function behaviour

setFullDataDisplayArea() is an option. This operates in link status. Behaviour in link status is the same as the broadcasting reception status.

- saveImageToMemoryCard() behaviour

saveImageToMemoryCard() is an option. This operates in link status. Behaviour in link status is the same as the broadcasting reception status. Refer to 8.1.15.9 for details.

- saveHttpServerImageToMemoryCard() behaviour

saveHttpServerImageToMemoryCard() is an option. This operates in link status. Behaviour in link status is the same as the broadcasting reception status. Refer to 8.1.15.9 for details.

8.3.6.5 Receiver sound control

Table 8-10 specified functions behaviour of receiver sound control in C-profile linked contents.

Table 8-10 Behaviour of receiver sound control in C-profile linked contents

	Link status
playRomSound()	O

This is operated in link status. Behaviour is the same as the data broadcasting reception status. Refer to 8.1.15 for behaviour of the data broadcasting reception status.

Behaviour when the timing of operation sound (ex: button operation sounds on mobile phones) and that of sound by playRomSound() are overlapping, is implementation dependent. Also, the sound quarity ringing by playRomSound() is not provided. Therefore, output of sound according to receiver features is permitted.

8.3.6.6 Timer functions

Table 8-11 specifies the behaviour for timer functions in C-profile linked contents. Refer to 5.12.6 for behaviours in the data broadcasting reception status.

Table 8-11 Behaviour of timer functions in C-profile linked contents

	Link status
setInterval()	O
clearTimer()	O
setCurrentDateMode()	O(*3)

- setInterval() function behaviour

This operates in link status. behaviourin link status is the same as the data broadcasting reception status.

- clearTimer() function behaviour

This operates in link status. behaviourin link status is the same as the data broadcasting reception status.

- setCurrentDateMode() function behaviour

setCurrentDateMode() shall be an option. This operates in link status. behaviourin link status is the same as the data broadcasting reception status.

8.3.6.7 Other functions

Table 8-12 specified function behaviour of other functions in C-profile linked contents. Refer to 8.1.15 for the data broadcasting reception status.

Table 8-12 Behaviourof other functions in C-profile linked contents

	Link status
random()	O
subDate()	O
addDate()	O
formatNumber()	O

“Other functions” operate in link status. Behaviours in link status are the same as the data broadcasting reception status.

8.3.6.8 Data carousel storage functions (T.B.D.)

Table 8-13 specified the behaviour of the data carousel storage functions in C-profile linked contents.

Table 8-13 Operation of data carousel storage functions in C-profile linked contents

	Link status
saveModule()	O (*1)
saveResource()	O (*1)

saveModule() and saveResource() are optional. The “Data carousel storage function” operates in link status. Behaviours in link status are the same as the data broadcasting reception status.

8.3.6.9 Digital terrestrial broadcasting specific functions

Table 8-14 specified behaviour of digital terrestrial broadcasting specific functions in communication contents.

Table 8-14 Behaviours of digital terrestrial broadcasting specific functions

	Link status
X_DPA_mailTo()	O(*1)
X_DPA_startResidentApp()	O
X_DPA_phoneTo()	O(*1)
X_DPA_getRcvCond ()	O(*1)

	Link status
X_DPA_getCurPos ()	O(*1)
X_DPA_saveExAppFile()	O(*1)
X_DPA_startExAV()	O(*1)
X_DPA_stopExAV()	O(*1)
X_DPA_tuneWithRF()	O(*1)
X_DPA_writeSchInfo()	O(*1)
X_DPA_writeAddressBookInfo()	X
X_DPA_getComBrowserUA ()	O(*2)
X_DPA_launchDocWithLink()	O(*2)
X_DPA_chkAVtype()	O(*1)
X_DPA_getIRDID()	O
X_DPA_writeCproBM()	O(*2)

- Behaviour of X_DPA_mailTo()

X_DPA_mailTo() is an option. Operation in link status is the same as the data broadcasting reception status. Refer to 8.1.15.

- Behaviour of X_DPA_startResidentApp()

Operation in link status is the same as the data broadcasting reception status. Refer to 8.1.15.

- Behaviour of X_DPA_phoneTo ()

X_DPA_phoneTo () is an option. Behaviour in link status is the same as the data broadcasting reception status. Refer to 8.1.15.

- Behaviour of X_DPA_getRcvCond ()

X_DPA_getRcvCond () is an option. Behaviour in link status is the same as the data broadcasting reception status. Refer to 8.1.15.

- Behaviour of X_DPA_getCurPos ()

X_DPA_getCurPos () is an option. Behaviour in link status is the same as the data broadcasting reception status. Refer to 8.1.15.

- Behaviour of X_DPA_saveExAppFile()

X_DPA_saveExAppFile() is an option. Behaviours in link status shall be the same as data broadcasting reception status. Refer to 8.1.15.

- Behaviour of X_DPA_startExAV ()

X_DPA_startExAV () is an option. Behaviour in link status is the same as the data broadcasting reception status. Refer to 8.1.15.

- Behaviour of X_DPA_stopExAV ()

X_DPA_stopExAV () is an option. Behaviour in link status is the same as the data broadcasting reception status. Refer to 8.1.15.

- Behaviour of X_DPA_tuneWithRF()

Behaviour in link status is the same as the data broadcasting reception status. Refer to 8.1.15.

- Behaviour of X_DPA_ writeSchInfo()

X_DPA_ writeSchInfo() is an option. Behaviour in link status is the same as the data broadcasting reception status. Refer to 8.1.15.

- Behaviour of X_DPA_writeAddressBookInfo()

This does not operate in link status.

- Behaviour of X_DPA_getComBrowserUA()

Behaviour in link status is the same as the data broadcasting reception status. Refer to 8.1.15.

- Behaviour of X_DPA_getIRDID()

Behaviour in link status is the same as the data broadcasting reception status.

- Behaviour of X_DPA_launchDocWithLink()

This is mandate for receiver units that possess communication functions. However, this function is allowed to use only in link status. If this function would be used in broadcasting reception status, the receiver shall fail and shall present broadcasting contents following 8.3.11.4.

- Behaviour of X_DPA_chkAVtype()

Behaviour in link status is the same as the data broadcasting reception status.

- Behaviour of X_DPA_writeCproBM()

Behaviour in link status is the same as the data broadcasting reception status.

8.3.6.10 Non-volatile memory functions

Table 8-15 specified behaviour of non-volatile memory functions in communication contents. Refer to 8.1.15 for operations in data broadcasting reception status.

Table 8-15 Behaviour of non-volatile memory functionss in communication contents

	Link status
readPersisitentArray()	O
wirtePersisitentArray()	O

The NVRAM area that can be used in link status is C-profile area for the affiliation. Refer to 7.2.4. for the name space. If a name space other than that is specified, the receiver will perform a failure behaviour.

8.3.7 Communication contents scope mapping

BML browser has 2 types of status, which are the “broadcasting reception status” and “link status”. These statuses mean condition for behaviour of receiver. From the contents point of view, there are 2 types of contents of “broadcasting contents”, which is content that receiver can refer to and present in broadcasting reception status, and “C-profile linked contents”, which are contents that receiver can acquire in link status. These 2 contents together are called “broadcaster’s contents”. Here, we will explain the “link status” of the above.

- Link status

- A status in which the BML document specified in the URI under the base URI directory is being presented shall be referred to as link status. Refer to "Description of the base URI directory" in this section for the concept of the base URI directory.
- Host name or directory name specified by X_DPA_launchDocWithLink() is set as the URI base directory. Status presenting the BML document described by the URI which is under the URI of this base URI directory shall be link status.
- Refer to 8.3.9 for references.

C-profile linked contents behave as one of the contents of the contents group. Communication contents in this status shall behave as the contents group in which data events do not exist.

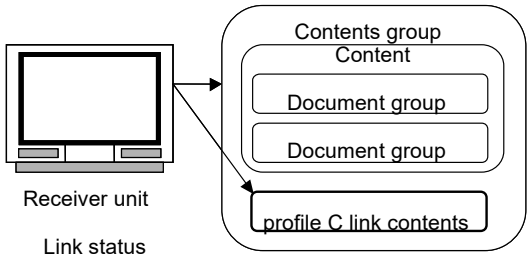


Figure 8-6 Link status

- Restrictions of C-profile linked contents
- In link status, the extended functions for broadcasting that can be utilized are restricted.
- The BML document of broadcasting contents and BML document in C-profile linked contents have different name spaces. Scheme of “arib:” or “arib-dc:” should always be specified in order to specify broadcasting contents from C-profile linked contents. Scheme of "http: " or "https: " should be specified in order to specify C-profile linked contents from broadcasting contents. Be aware of the difference in abbreviation handling as well. Refer to 8.3.10 for operation of name spaces in C-profile linked contents.
- In C-profile linked contents, there is no concept of a startup document, unlike broadcasting contents. However, receiver may specify the URI on a directory level, without specifying resources of C-profile linked contents directly. In that case, C-profile linked contents of specifications in this volume which follow the setting of servers is returned. (ex: index.bml, etc.)
- Because there is no data event in C-profile linked contents, receivable event messages are only “ event_msg_group_id=1 ”. If the concerned id receives “ event_msg_group_id=0 ”, it is ignored. Refer to 4.3 for operations of event_msg_group_id.
- If transitioned to C-profile linked contents located under the URI (base URI directory hereafter) directory specified by the broadcasting contents, the link status should be kept. However, if another URI was specified by use of the “href” attribute of the “a” element, launchDocument() or the “action” attribute of the “form” element, etc., BML browser will fail and will present broadcasting contents following 8.3.11.4. Refer to "Description of the base URI directory" in this section for the concept of the base URI directory.

- After transition from C-profile linked contents to link status by X_DPA_launchDocWithLink(), if URI under the base URI directory which was specified by X_DPA_launchDocWithLink() is specified by following method, the “href” attribute of the “a“element, launchDocument() or the “action” attribute of the “form” element, etc., the link status is kept. Also, if other than base URI directory was specified, receiver unit will fail and will present broadcasting contents following 8.3.11.4.
- With receivers that possess closed caption presentation functions, closed caption should be able to be presented if closed caption are included in the broadcasting service they belong to, even in link status.

* If the receiver unit receives response in the 300’s and is redirect-specified, the link status is kept as long as the redirect destination is below the URI directory.
* Also, follow separate provisions for communication protocols regarding mobile phones.

● Description of the base URI directory

The base URI directory is utilized as a C-profile link content document group identifier in link status. Base URI directory refers to the specified host name and directory name by: launchDocument(), the “href” attribute of the “a” element and the “action” attribute of the “form” element from broadcasting contents or by X_DPA_launchDocWithLink() from C-profile linked contents. For example, the identifier that refers to the C-profile linked contents base URI directory specified by launchDocument("http://localhost/hoge/index.bml", "cut") is "//localhost/hoge/".

Specifications below are established for judgment of base URI directories.

- Identifier that describes base URI directories that do not include port numbers. For example, base URI directory of C-profile linked contents which is specified by launchDocument("http://localhost:10080/hoge/index.bml", "cut") from data broadcasting BML document is ”//localhost/hoge/”. Therefore, even in C-profile link content transition between different port numbers, link status is kept as long as the base URI directory matches.
- Name space whose URI was encoded should be handled as the same level as name space that is not encoded. For example, the base URI directory for "http://localhost/%7Ehoge/index.bml" and "http://localhost/~hoge/ test.bml" should be handled as matching directories.
- Upper case letters and lower case letters are not differentiated with host names. Upper case letters and lower case letters are differentiated with directory names.
- The base URI directory to be set is one. Also, execution of X_DPA_launchDocWithLink() will be the operation that changes the base URI directory.

Resources directly under the base URI directory and resources stored in directories under are all considered to be within the document group of C-profile linked contents, and the link status is kept. If the URI that does not match the base URI directory was specified, data broadcasting browser will fail and will present broadcasting contents following 8.3.11.4. Refer to 8.3.12 for status transition details.

● Link status due to X_DPA_launchDocWithLink() transition

If transitioned using the X_DPA_launchDocWithLink() function from link status, the base URI directory before transition shall be invalid. Set the host name or directory name specified by this function in the URI directory, and the receiver shall be in link status.

8.3.8 Guidelines for C-profile linked contents transmission

The following shows the operational guidelines for the transmission of C-profile linked contents. Refer to section 3.5 for the communication function of the receiver. Details of the communication protocol related to mobile phone shall follow the separately defined prescriptions.

- C-profile linked contents shall be transmitted in accordance with HTTP/1.1 defined in RFC2616.
- If "http:" has been specified in the URI, the receiver and the interactive Web server shall perform communication via HTTP/1.1 using the port specified by the URI.
- If "https:" has been specified in the URI, the interactive Web server shall perform encrypted communication based on HTTP/1.1 after having established a connection via TLS or SSL using the port specified by the URI.
- If the port number was not specified in the URI, ports 80 and 443 shall be used for "http:" and "https:" respectively.
- Refer to Appendix 11 for the details of the cache function.
- Servers should consider cases where the following media types are requested.

  text/plain, text/html,
  text/css, text/X-arib-ecmascript,
  image/jpeg, image/gif
  audio/X-arib-mpeg2-aac,
  application/X-arib-btable
  application/octet-stream(T.B.D.)

*In the case of mobile phones,follow other specification for communication protocols..

8.3.9 Resource references between C-profile linked contents and broadcasting contents

8.3.9.1 Resource references from broadcasting contents to C-profile linked contents

For resource references from broadcasting contents to C-profile linked contents, follow ARIB STD-B24 Vol. 2 Appendix 4　6.2 “Guidelines on reference across media types” and do not perform reference. Only BML document transition is possible from broadcasting contents to C-profile linked contents.

8.3.9.2 Reference from C-profile linked contents to broadcasting contents

In cases where referring from C-profile linked contents to event messages transmitted by broadcasting or where specifying resources transmitted by broadcasting upon event’s subscribe, etc., always only use absolute URI’s that start with "arib-dc://-1.-1.-1".

(ex: <bml:beitem id="em" type="EventMessageFired" es_ref="arib-dc://-1.-1.-1/89" …omission…/>)

8.3.10 Name space in C-profile linked contents

8.3.10.1 Restriction for URI

For operations of name space, follow ARIB STD-B24 Vol. 2 Appendix 4 6.3 “Namespace”. However, when presenting BML communication contents acquired from servers, the following restrictions are established for the BML document’s name space.

- Multi Byte characters such as Japanese are not utilized for the URI.
- BML document should be specified for “href” attributes of the “a” element, arguments of launchDocument() and arguments of X_DPA_launchDocWithLink () described in BML documents in broadcasting reception status and link status.

8.3.10.2 Operation of name spaces in servers

In link status, if “/” was specified at the end of the URI upon contents acquisition from servers, servers should return BML documents in C-profile linked contents, according to the settings of server.

(ex: browser.launchDocument("http://localhost/hoge/", "cut");)

Receiver behaviour is implementation dependent,when BML documents were not returned from servers due to each status.

When abbreviation of URI (relative URI) is ued, its scheme is the same as scheme of presenting contents. It is treated as relative URI from directory names located presenting contents. (ex: If the BML document URI is http://localhost/test/tmp/index.bml and "../hoge.bml" was described in the document, its URI will be http://localhost/test/hoge.bml.)

However, when using X_DPA_launchDocWithLink(), use an absolute URI for its argument.

8.3.10.3 Operation of name spaces upon transition from broacasting contents to C-profile linked contents

Reference from broadcasting contents to C-profile linked contents shall not be performed, according to specifications in 8.3.9.1. When transitioning from broadcasting contents to C-profile linked contents, use absolute URI that start with "http: " or "https: ".

8.3.10.4 Operation of name spaces upon transition from C-profile link content to broadcasting content

When transitioning from C-profile linked contents to broadcasting BML document, always only use absolute URI that start with “arib-dc://-1.-1.-1”.

(ex: browser.launchDocument("arib-dc://-1.-1.-1/80/0000/startup.bml", "cut");)

8.3.11 Guidelines for operations of C-profile linked contents

8.3.11.1 Note on operations of C-profile linked contents

Service provider(including broadcaster) that supplies C-profile linked contents should consider that it is extremely difficult for them to present BML content to viewers in the same quality level as data broadcasting, due to the property of communication system used in C-profile linked contents distribution. Communication system properties are listed below.

- There is no guarantee that the transmission bandwidth is invariable.
- There are differences in useable transmission bandwidth depending on the receiver.
- Steady response to viewers cannot be guaranteed because the variable communication path can be selected.
- There is a possibility that data transmission is blocked due to proxy properties in the route.

Furthermore, there is a risk that response quality may greatly decrease due to heavy loads on servers. Note that due to these elements, presentation of C-profile linked contents distributed in communication systems is not invariable for all viewers and the receiver's presentation may not perform correctly depending on the network situation.

Due to above reasons, it is recommended for BML contents fitting in items below to be distributed via data broadcasting rather than via communication.

- BML contents recognized execution timing such as video/audio sync as critical factor.
- BML contents recognized presenting all elements properly as critical factor.

8.3.11.2 Guidelines for description of C-profile linked contents

Notes of description of C-profile linked contents are below.

- C-profile linked contents are described in profiles according to operations in 8.1.
- Note that there is difference between broadcasting contents and C-profile linked contents regarding the name space. Refer to 8.3.10 for details.
- It is recommended not to specify the port number for C-profile linked contents which are assumed to operate in link status.
- Always add "/" at the end when specifying C-profile linked contents by directory specifications. (ex: "http://localhost" ×: incorrect, "http://localhost/"O: correct)
- #fragment can be added to the URI that specifies C-profile linked contents. Refer to 8.2.3.4 regarding guidelines and behaviour of receiver for contents using #fragment. (ex: "http://localhost/test.bml#fragment")

- Following documents and files should be described in Shift-JIS, BML documents to compose C-profile linked contents, binary tables, ECMAScript files related in the “src” attribute of “script” elements and CSS files related in “link” elements.
- The BML Version described in C-profile linked contents is ”12.0”, as broadcasting contents.
  (ex: <?bml bml-version ="12.0" ?>)
- Extension of BML documents that compose C-profile linked contents should be bml. (ex: index.bml)
- By putting Mono-media in one page under the same directory, it can reduce receiver unit’s pass search process
- BML document should be specified for “href” attributes of the “a “element, arguments of launchDocument() and arguments of X_DPA_launchDocWithLink ()described in C-profile linked contents.
- Within C-profile linked contents, links to other BML documents or a procedure to quit it should always be described. This is because receiver units will have no procedure of transition to another status other than tuning, just in case receiver transit to a BML document with no link.
- Resources (mono-media such as JPEG, GIF, animation GIF and MPEG-2 AAC) in C-profile linked contents can be specified separately from the scope of the base URI directory.
- Because available functions are different in link status, it is different behaviour even in the same contents.
- Total amount of C-profile linked contents resources referred to in BML documents at the same time shall not exceed 256KBytes , considering that the area reserved for receiver’s module acquisition is 256KBytes.
- If the X_DPA_launchDocWithLink function is used, a host name or directory name described by the argument of its function is set as the URI directory, and goes in link status. Therefore, as specified in 7.10.5.1, available functions in link status can be used and can access to NVRAM, etc. Therefore, for contents description ,it should pay close attention to such issues.

8.3.11.3 Guidelines for receiver unit status transition operations

Receiver can be in the following 2 status as receiving BML contents.

(1) “Data broadcasting reception status” in which data broadcasting, video and audio are being received and played.

(2) “Link status” in which C-profile linked contents, video and audio are being received and played together.

Refer to 8.3.7 for link status. Receiver transitions betweenthese 2 statuses by the specification of contents or the operation of users. Figure 8-7 specified a situation of status transition.

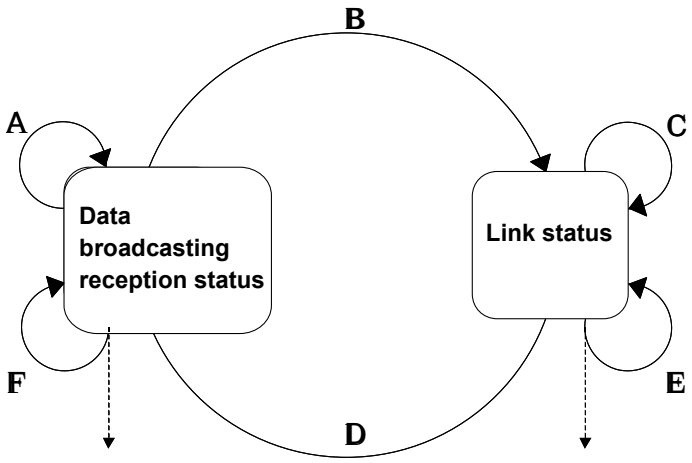


Figure 8-7 Status transition of receivers in C-profile linked contents reception

- Description of transition A (from data broadcasting reception status to data broadcasting reception status)

Transition A means a transition from data broadcasting reception status to data broadcasting reception status. Refer to chapter 4 for details of operations. In this transition, the URI scheme that can be specified for the “href” attribute of the “a” element and launchDocument() is "arib-dc: ".

- Description of transition B (from data broadcasting reception status to link status)

Transition from data broadcasting reception status to link status is possible by the specification of launchDocument( ), the “href” attribute of the “a” element or “action” attribute of the “form” element. There are 2 types of URI schemes to specify C-profile linked contents: "http: " and "https: ". Refer to 8.3.10 for details of name spaces in C-profile linked contents.

(ex: browser.launchDocument("http://localhost/hoge/index.bml", "cut");)

C-profile linked contents can specify the directory without specifying the resources. When the directory is specified, a server returns the C-profile linked contents that follow the setting. “index.bml”, for example. When specifying the directory, make sure to add “/” at the end to indicate that it is a directory.

(ex: browser.launchDocument("http://localhost/hoge/", "cut");)

BML browser fails and will present broadcasting contents following 8.3.11.4, if it is specified a document whose extension is “html” or “htm” by following method in data broadcasting status , the “href” attribute of the “a“ element and launchDocument() and the “action” attribute of the “form” element.

- Description of transition C (from link status to link status)

The scope of the document group in C-profile linked contents is the range specified under the first accessed URI’s host address and directory (base URI directory hereafter) or the range specified under the host name and directory name specified by X_DPA_launchDocWithLink() in link status. It does not depend on the scheme (protocol) specification. (ex: //localhost/hoge/)

When link status is kept by means of the base URI directory, the link status is kept even after transitioning from "http://localhost/hoge/index.bml" to “http://localhost/hoge/foo/test.bml” by using the “href” attribute of the “a” element or launchDocument(). Also, in case of a transition to “https://localhost/hoge/foo/test.bml”, the link status is kept in the same way.

However, if something other than the base URI directory was specified in the link status by the “href” attribute of the “a” element, the “action” attribute of the “form” element, and launchDocument(), BML browser will fail and will present broadcasting contents following 8.3.11.4. If a document whose extension is “html” or “htm” is specified by the “href” attribute of the “a” element, the “action” attribute of the “form” element, and launchDocument(), BML browser will fail and will present broadcasting contents following 8.3.11.4.

Note that when the link is kept by the base URI directory, the link status is still kept after transitioning from "http://localhost/hoge/index.bml" to “http://localhost/hoge/foo/test.bml” by utilizing X_DPA_launchDocWithLink(), but the base URI directory will be set as "//localhost/hoge/foo/".

It will be in link status when transitioned by X_DPA_launchDocWithLink(), even in cases where the host address is changed as in the transition from "http://localhost/hoge/index.bml" to “http://localhostdpa/dpa/foo/test.bml”, for instance. The base URI directory in that case is set as ”//localhostdpa/dpa/foo/.”

- Description of transition D (from link status to data broadcasting reception status)

Transition from link status to data broadcasting reception status can be performed by using the “href” attribute of the “a” element or launchDocument( ). Operation is an implementation dependent in cases where there is no BML document of transition destination or where the BML document of transition destination is not acquirable due to reception status.

Always specify "arib-dc:" for the URI scheme to specify broadcasting contents. Note that only currently presented services such as "arib-dc://-1.-1.-1/80/0000/startup.bml" can be specified when transitioning using launchDocument().

(ex: browser.launchDocument("arib-dc://-1.-1.-1/80/0000/startup.bml","cut");)

Also, by using quitDocument(), the transition from link status to data broadcasting reception status is possible.

By tuning of the user’s action, transition from link status to the data broadcasting reception status can also be executed. .

By using X_DPA_tuneWithRF(), transition from link status to the data broadcasting reception status is also executed.. However, an entry component is presented after transition in this case.

Also, if "arib-dc:" and "arib:" are specified by X_DPA_launchDocWithLink(), BML browser will fail and will present broadcasting contents following 8.3.11.4.

- Transition E (transition from link status to broadcasting reception status and other non-link status)

If X_DPA_startResidentApp() is executed in link status, the receiver’s native application specified by argument will startup. When continuing presentation of BML browser such as simultaneously displaying with communication browser, the link status is kept and presentation of C-profile linked contents will continue. Refer to 8.1.15.8 for details of X_DPA_startResidentApp().

- Transition F (broadcasting reception status to broadcasting reception status and non-link status)

If X_DPA_startResidentApp() is executed in broadcasting reception status, the receiver’s native application specified by argument will startup. When continuing presentation of BML browser, as simultaneously displaying with communication browser, the broadcasting reception status is kept and presentation of broadcasting content will continue. Refer to 8.1.15.8 for details of X_DPA_startResidentApp().

8.3.11.4 Guidelines for failure behaviour of receiver units

If the restricted function in status specified in 8.3.11.3 is used, the receiver will perform an operation failure.

Also, if a URI, which does not match the base URI directory, is specified by the “href” attribute of the “a” element and launchDocument() and the “action” attribute of the “form” element in link status, BML browser will fail and the receiver will display an error message.

In sections specified as “operation failure” and “failure behaviour” of BML browsers in 8.3.6 etc., receivers shall perform the following processes in order.

(1) BML browser ends once.

(2) An error message is displayed.

(3) Step 3 and later described in 4.1.5.1 “Receiver operation at the beginning of data broadcasting” are executed for the service which is tuned currently.

- Guidelines for receiver upon error response reception

The following guidelines are below when receiver requests C-profile contents and resources to consist of those from servers, and an error response[*1] is returned:

- Receiver behaviour upon reception of error responses in BML documents is an implementation dependent.
- Presentation of receiver upon reception of error response regarding resources that consist of BML documents is implementation dependent.

*1: If status code was in the 400’s or 500’s. Refer to RFC2616 for details.

8.3.12 Guidelines for connection and disconnection

The specification of this section do not apply to receivers which are always on Internet access. .

8.3.12.1 Guidelines for receiver upon automatic connection

If the receiver does not perform IP connection and the following events are occurred, the receiver performs an automatic connection to the communications system. Although in the case of mobile phone receivers should perform automatic connection to communications system when the following events are occurred, receivers may follow other specifications.

- If transition to a BML document on a server in link status is performed
  (transition by means of the “action” attribute of the “form” element, the “href” attribute of the “a” element, launchDocument(), and X_DPA_launchDocWithLink())
- If transmitTextDataOverIP() is used

8.3.12.2 Guidelines for receiver upon disconnection

Upon disconnection of communication by the quitDocument() function, the receiver shall completely disconnect the connection before the startup document is displayed. Within the period between calling of quitDocument() function and presentation of the startup document, the timing of disconnection is an implementation dependent.

Upon disconnection of communication by tuning and use of the X_DPA_tuneWithRF() function, the receiver shall completely disconnect the connection before tuning is complete. Within the period between tuning or calling of X_DPA_tuneWithRF() function and completion of tuning, the timing of disconnection is implementation dependent.

8.3.13 Operation of certificates

In C-profile, the root certificate is not transmitted from the broadcasting station to the receiver. Instead, the root certificate already embedded and incorporated on the receiver shall be used. Contact the governing organization described in Appendix 13 regarding the root certificate to be operated by broadcasters.

- The hash function for signatures in the server certificate and intermediate certificate shall be limited to either SHA-1 or SHA-2.
- The receiver shall be implemented such that the user cannot voluntarily add a root certificate to the receiver. Alternatively, the receiver shall fail in TLS/SSL communications using a server certificate linked to a root certificate not operated by the broadcaster.
- Incorporation of update functions for the root certificate for when the certificate expires and for the root certificate information operated by the broadcaster to the receiver shall be a matter of product planning. However, it shall be desirable that the receiver incorporates both of these update functions.

8.3.14 Operation of User-Agent for identification of BML browsers and a browser for the C-profile contents on internet

As one of the method within servers to judge browsers, use of User-Agent of an HTTP Request header can be expected. Therefore, BML browsers and a browser for the C-profile contents on internet in receivers that have connectable features shall implement User-Agent

By using “User-Agent” as a procedure to judge receiver’s models within servers, broadcasters can get receiver’s various information and judge receiver’s models, and then can specify appropriate contents corresponding to the receiver. .

The following specifies information and the format setting for User-Agent. Also, in the case of mobile phones, if there are other specifications, they may follow those. .

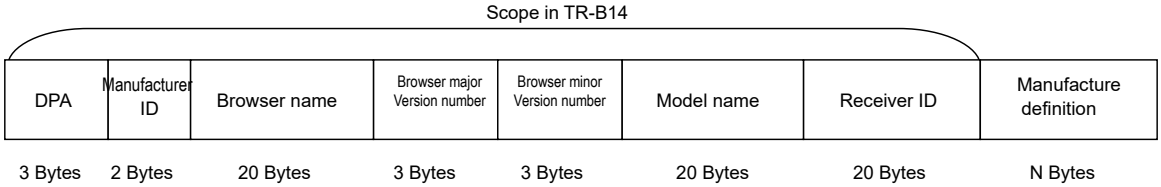


Figure 8-8 User-Agent format specified in TR-B14

- DPA: stands for request from browser specified in TR-B14. String that is a fixed length and “DPA”. When using User-Agent in BML browsers and browser for the C-profile contents on the internets, this shall always be added to the front.
- Manufacturer ID: stands for manufacturer’s ID managed by ARIB. Maximum size is 2 Bytes. This shall not be omitted in the specified format.
- Browser name: stands for browser’s name. Return string that is a maximum of 20 characters, using "0"-"9" and "A"-"Z" which are specified for each manufacturer. This shall not be omitted from the specified format.
- Browser major version number: stands for browser’s major version number. Return string that is described with a decimal number that is maximum of 3 digit. This shall not be omitted in the specified format
- Browser minor version number: stands for browser’s minor version number. Return string that is described with decimal number that is maximum of 3 digit. This shall not be omitted in the record.
- Model name: stands for receiver’s model name. Maximum size is 20 Bytes. This shall not be omitted from the specified format.
- Receiver unit ID: stands for receiver’s ID. Maximum size is 20 Bytes. This may be omitted from the specified format. However, it is recommended for the receiver to get a user permission before the fact in order to set the receiver’s ID.
- Manufacturer definition area: area which is defined and used by manufacturers and carriers.

- 1 For “manufacturer ID”, a hexadecimal string is returned, but it shall be 2 digit without adding the string implying it is a hexadecimal string, such as "0x" in front or "h" at the end. If it is less than 2 digit, “0” is added as necessary padding.
- 2 If “browser name” and ”model name” are less than the maximum Byte size, stuff with space (0x20).
- 3 Stuff with space (0x20) in case of omitting items that are allowed to be omitted.
- 4 For “browser major version number” and “browser minor version number”, if they are less than 3 digit, add “0” as padding.
- 5 Use “/” as a partition for each item.
- 6 “Manufacture ID”, “browser name”, “browser major version number” and “browser minor version number” shall be the same as the return value upon execution by getBrowserVersion() .
- 7 For “model name”, if it is mobile phones, it is recommended to set the model name that is set within User-Agent of carriers' proprietary browser. Set the model name that is managed by each manufacturer in the case of receivers other than mobile phones.

(Usage example)

User-Agent : DPA/1C/AAABBBCCC7 /023/007/DPA-H90001 /receiver unit ID/Manufacturer definition area…

8.3.15 Operation of X_DPA_getComBrowserUA()

By using information acquired by X_DPA_getComBrowserUA() for judgment of browser’s models, X_DPA_startResidentApp() can be used in changing the transition URI.

8.3.16 Content transition and browser startup

Figure 8-9 specified the contents transition and browser startup. Also, Figure 8-9 simply specified ”transition” of the contents and ”startup” of the browser. Refer to specifications in 8.1.4 - 8.1.7 for detailed information regarding startup and presentation of each browser and application.

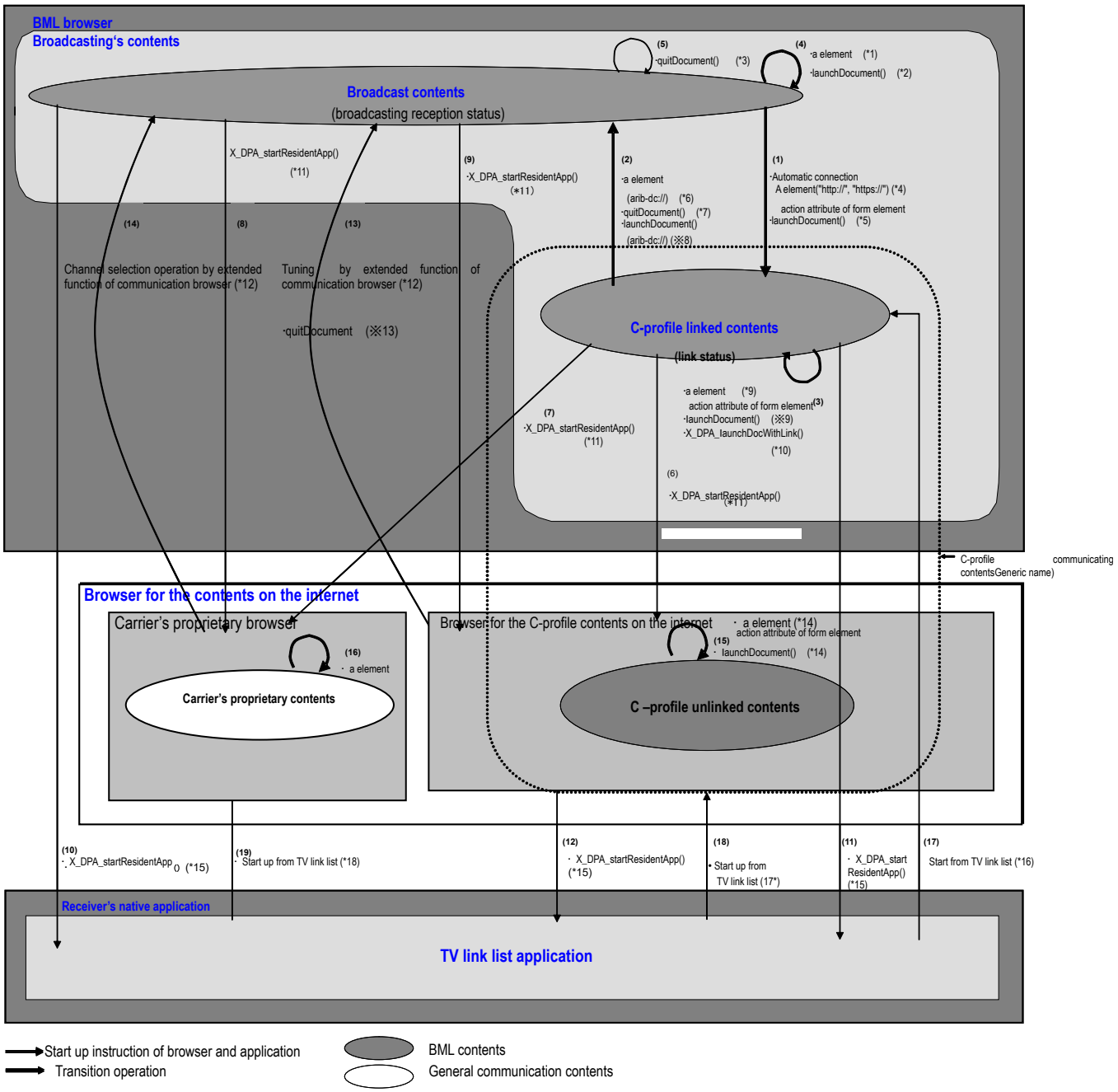


Figure 8-9 Transition and startup of each browser

*1: When broadcast contents are specified by the “href” attribute of the “a” element from broadcast contents. Use abbreviation (refer to ARIB STD-B24 Vol. 2: 9.2) for the name space description to specify broadcasting contents within BML documents.

*2: Transition from broadcasting contents to broadcasting contents by use of launchDocument(). Use abbreviation (refer to ARIB STD-B24 Vol. 2: 9.2) for the name space description to specify broadcasting contents within BML documents.

*3: Refer to 8.3.6.4 for quitDocument(). In this case, however, finish BML browser once and present entry component after restarting.

*4: Specify the URI of C-profile linked contents by the “href” attribute of the “a” element and the “action” attribute of the “form” element. Only "http: " and "https: " are specifiable for the URI scheme.

*5: Specify the URI of C-profile linked contents. It will be in link status if transitioned by this funcion. Only "http: " and "https: " are specifiable in the URI scheme.

*6: Specify only the URI of broadcasting contents by the “href” attribute of the “a” element. Only specify absolute URI that start with “arib-dc://-1.-1.-1" for the URI scheme.

*7: Refer to 8.3.6.4 for quitDocument(). In this case, however, finish data broadcasting browser once and present the entry component after restarting. However, upon this, the connection may be disconnected if already connected. Also, upon specification of quitDocument(), receivers may judge whether or not to disconnect the connection , depending on situations such as communicating with something other than BML browser, etc.,.

*8: Specify only the URI of broadcasting contents. Only specify absolute URI that start with “arib-dc://-1.-1.-1" for the URI scheme.

*9: Specify the URI of C-profile linked contents by the “href” attribute of the “a” element, launchDocument() and the “action” attribute of the “form” element. However, the URI in this case shall be within the base URI directory.

*10: Specify the URI of C-profile linked contents. If transitioned by X_DPA_launchDocWithLink(), the host name or directory name specified by X_DPA_launchDocWithLink() is set as the base URI directory and becomes a link status. Only "http: " and "https: " are specifiable for the URI scheme.

*11: X_DPA_startResidentApp() is executed in broadcasting reception status and link status, and after that a browser for the contents on the internet is started up. Perform presentation according to 8.1.4 to 8.1.7 upon this.

*12: Refers to behaviour when the tuning is specified by the extended function of a browser for the contents on the internet.

*13: If quitDocument() is used, finish the Browser for the C-profile contents on the internet once and present the entry component after restarting, not transitioning. After the browser is finished, the channel to tune shall be the last channel. However, connection may be disconnected if already connected upon this. Also, upon specification of quitDocument() , receivers may judge whether or not to disconnect the connection, depending on situations such as communicating with something other than BML browser, etc.,.

*14: Specify the URI of the C-profile unlinked contents by the “href” attribute of the “a” element, launchDocument() and the “action” attribute of the “form” element. Only "http: " and "https: " are specifiable for the URI scheme. URI of broadcasting contents cannot be specified.

*15: In broadcasting contents, C-profile linked contents and C-profile unlinked contents ,when X_DPA_startResidentApp() is executed by specifying "BookmarkList" for argument “appName”, TVlink list applications as one of the receiver’s native applications starts up.

*16: If TVlink set as CproBMtype=1 is selected by users in the TVlink list application as one of the receiver’s native applications, present C-profile linked contents according to 3.7.7.

*17: If TVlink set as CproBMtype=2 is selected by users in the TVlink list application as one of receiver’s native applications, present C-profile unlinked contents according to 3.7.7

*18: If TVlink set as CproBMtype=3 or CproBMtype=4 is selected by users in the TVlink list application as one of the receiver’s native applications, present the communication company specification contents according to 3.7.7

9 Functions with which C-profile basic receiver unit that only receive video and audio

This chapter specifies functions with which C-profile basic receiver unit that only receive and present video and audio.

9.1 Configuration of the receiver unit

Below are specifications of each processor from the viewpoint of hardware configuration necessary for reception of video and audio data.

9.1.1 Hardware configuration

Figure 9-1 specifies the hardware configuration of receiver units.

Digital broadcasting signals that enter into C-profile basic receiver units are transformed into a transport stream by a tuner and demodulator. The transport stream, by transport stream decoding process, is then demultiplexed into video and. Video stream is output to a video decoding process, and audio stream is output to an audio decoding process. With the above processes, C-profile basic receiver units perform playback of video and audio in a stream format.

From the viewpoint of such hardware processing operations, the following specifications are necessary.

(1) Receivable data by the transport decoder
(2) Playback audio data of the stream system
(3) Playback video data of the stream system
(4) Presentation of video

(1) for the TS decoding function, and (2), (3) and (4) for presentation functions, are especially specified.

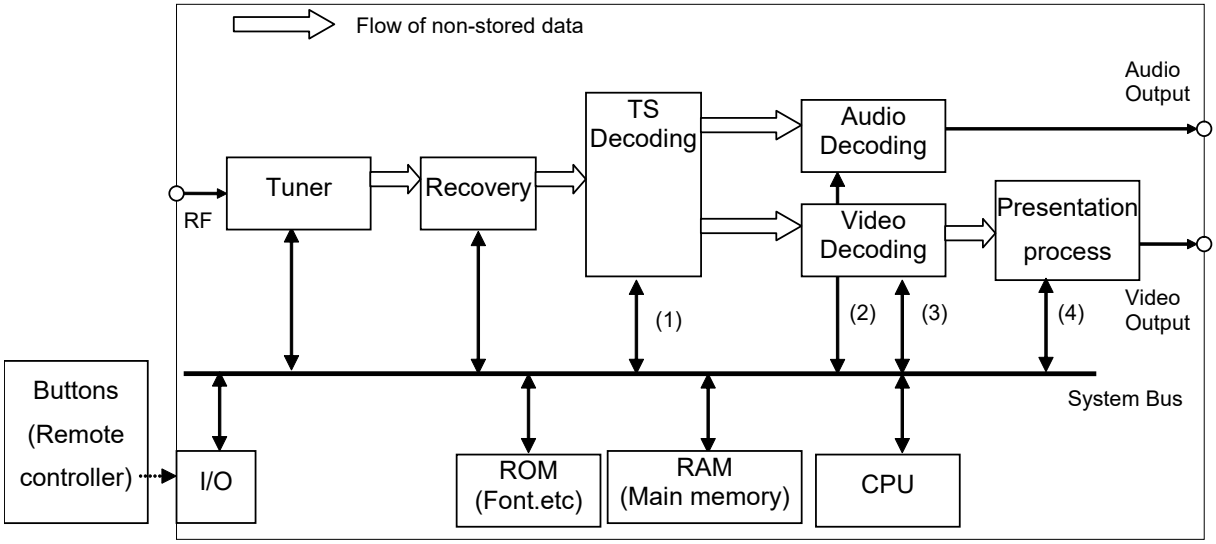


Figure 9-1 Hardware configuration of a C-profile basic receiver unit (video data/audio data code)

9.2 Presentation functions

9.2.1 Video presentation function

Table 9-1 indecates specifications for each coding formula of visual mono-media, whose presentation is desired in C-profile basic receiver unit that only receive video and audio.

Table 9-1 Visual mono-media whose presentation is desired

Coding formula		Contents of specification	
Video coding	H.264|MPEG-4 AVC	Transmission method	Video PES; stream format identifier = 0x1B
		Video size	QVGA(320x240 (4:3), 320x180 (16:9))
		Scaling	Implementation dependent

9.2.2 Audio playback function

Table 9-2 indecates specifications for coding formula of audio mono-media, whose presentation is desired in C-profile basic receiver units that only receive video and audio. Operation details for the coding method refer to Chapter 5.

Table 9-2 Audio mono-media whose presentation is desired

Coding method	Contents of specification	
MPEG-2 AAC	Transmission method	Audio PES; stream format identifier = 0x0F
	Sampling frequency	24kHz,48kHz

9.3 Guidelines for reception operation

1. If PMT 2nd loop includes components of component_tag=0x81, it shall be considered as a low-frame-rate and low-resolution picture (H.264|MPEG-4 AVC) stream and played back. Also, if it includes component_tag=0x83, it shall be considered as a MPEG-2 AAC (sampling frequency=24kHz) sound stream and played back. In the same manner, if it includes component_tag=0x85, it shall be considered as a MPEG-2 AAC (sampling frequency=48kHz) sound stream and played back.
2. It is not necessary to follow data (data carousel / event message) changes other than the video / audio stream.

9.4 Guidelines for presentation

In this section, guidelines for video and audio presentation in C-profile basic receiver units that only receive video and audio are specified.

9.4.1 Presentation of video / audio

9.4.1.1 Display position and display size of video

- Display position and display size of video placed within display devices of receiver units are implementation dependent.
- The receiver units do not necessarily need to display video transmitted by QVGA in the same pixel size. Specifically, they are allowed to display video with less pixel size and frame rate, if the receiver unit's performance or processing capability are insufficient. On the other hand, they are allowed to display enlarged video for receiver units whose pixel size is above QVGA.

9.4.1.2 Optimizing video display size

With receiver units whose display device possesses pixel size other than 320 dots in the horizontal direction, it is recommended to display video by scaling it according to display device's pixel size.

9.4.1.3 Display of QVGA 16:9 video

When displaying QVGA (16:9) video, areas within the display device that the video possess is 60 dots smaller in the vertical direction compared to QVGA (4:3) video. With receiver units with no multimedia presentation function, the display of contents, applications etc. supplied by broadcaster in this area should not be performed.

9.4.2 Presentation upon TV reception function startup and tuning

With C-profile basic receiver units that only receive video and audio, only video and audio originated in broadcasting should be presented and contents or applications other than broadcasting may not be displayed on display devices of receiver units upon TV reception function startup. The contents and applications that were displayed right before tuning should be invisible.

9.4.3 Restriction of mixed display of non-broadcasting screen and broadcasting screen and simultaneous display

9.4.3.1 Principle of mixed display prohibition

It is prohibited to possess a mixed display function in receiver units. This “mixed display” refers to such cases where a content supplier, to another content supplied by another supplier, purposely simultaneously displays their own content in relation to the other content, or such cases where different contents by multiple suppliers are displayed by influencing the display, etc. in order to mislead viewers into thinking that they are the same contents. The function which enables such displays is called mixed display function.

9.4.3.2 Prohibition of simultaneous display on receiver units that only receive video and audio

With receiver units that only receive video and audio, broadcasting contents and other contents supplied by other providers should not be simultaneously displayed. They should only display broadcasting contents. To display contents supplied by other providers, the broadcasting screen should be turned off and a full-screen switch should be performed.

However, the following screens may be simultaneously displayed as exceptions.

- The following screens for e-mail applications
  - e-mail composition screen
  - e-mail send screen
  - e-mail reception notification excluding subject line

- e-mail sender, subject line and content of e-mail from senders whose address entry is in the address book of receiver unit

- Screens which are displayed by receiver unit specific applications which were implemented in the receiver unit at the time of purchase and do not execute communication. (Including when receiver unit specific application was updated using communication in order to execute bug fix, etc.) For the messages, etc. refer to section 9.4.3.3.

9.4.3.3 Simultaneous display of messages displayed by receiver unit

The messages displayed by receiver unit as the following examples may be simultaneously displayed with broadcasting screens.

- System status display for receiver units
- Set-up screen of receiver units
- Operation support information for users
- Reception notification of sound communication and transmitter phone number notification
- Supplemental information for transmission, such as display of entered phone number upon user calling operations
- Display of approval request displayed when user permission is necessary upon connecting to communication and user interface for user to approve.
- Alert for operations that may generate disadvantages for the user
- Other messages displayed by receiver units not by communication with a third person

9.4.3.4 Operation of broadcasting contents upon simulaneous display

- It is recommended for video and audio of broadcasting to continue presenting, even while displaying contents whose simultaneous display is indicated in section 9.4.3.3.

Appendix 1 Regards for NVRAM access

Refer to Appendix-6 of [Section 2]

Appendix 2　Module compression format

Refer to Appendix-2 of [Section 2]

Appendix 3 DTD for operation scope checking for basic service

```
<!-- ======== Broadcast Markup Language (BML) for Mobile DTD [OPERATABLE]
======== -->
<!ENTITY %      ContentType      "CDATA" >
<!ENTITY %      Charset          "CDATA" >
<!ENTITY %      Character        "CDATA" >
<!ENTITY %      LinkTypes        "NMTOKEN" >
<!ENTITY %      Number           "CDATA" >
<!ENTITY %      URI              "CDATA" >
<!ENTITY %      Script           "CDATA" >
<!ENTITY %      StyleSheet       "CDATA" >
<!ENTITY %      Text             "CDATA" >

<!-- ================ Character mnemonic entities
========================== -->

<!ENTITY quot    "&#34;" ><!-- double quote -->

<!ENTITY % Events.attrib
"onclick         %Script;         #IMPLIED
onkeydown       %Script;         #IMPLIED
onkeyup         %Script;         #IMPLIED"
>
<!ATTLIST form
onsubmit        %Script;         #IMPLIED
>
<!ATTLIST input
onchange        %Script;         #IMPLIED
>
<!ATTLIST body
onload          %Script;         #IMPLIED
onunload        %Script;         #IMPLIED
>
<!ATTLIST object
bml:onfocus         %Script;        #IMPLIED
bml:onblur          %Script;        #IMPLIED
```

```
bml: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 % Blkstruct.class "p | div | pre" >
<!ENTITY % Block.class
"%Blkstruct.class;"
>
<!ENTITY % Boxed.mix
"%Block.class;
| img
| object
| form
| input
| textarea"
>

<!ENTITY % BlkNoForm.mix
"%Block.class;
| img
| object
| input
```

```
| textarea"
>

<!ENTITY % Br.content "EMPTY" >
<!ELEMENT br %Br.content; >
<!ATTLIST br
%Core.attrib;
>
<!ENTITY % Span.content "( #PCDATA | br )*" >
<!ELEMENT span %Span.content; >
<!ATTLIST span
%Core.attrib;
>
<!ENTITY % Div.content "( %Boxed.mix; )*" >
<!ELEMENT div %Div.content; >
<!ATTLIST div
%Core.attrib;
>
<!ENTITY % P.content "( #PCDATA | %Inline.class; )*" >
<!ELEMENT p %P.content; >
<!ATTLIST p
%Core.attrib;
>

<!ENTITY % Pre.content "( #PCDATA | %Inline.class; )*" >
<!ELEMENT pre %Pre.content; >
<!ATTLIST pre
%Core.attrib;
>

<!ENTITY % Script.content "( #PCDATA )" >
<!ELEMENT script %Script.content; >
<!ATTLIST script
src             %URI;           #IMPLIED
>
```

```
<!ENTITY % A.content "( #PCDATA    | br )*" >
<!ELEMENT a %A.content; >
<!ATTLIST a
%Common.attrib;
href                %URI;                #IMPLIED
accesskey        %Character;          #IMPLIED
>

<!ENTITY % Img.content "EMPTY" >
<!ELEMENT img %Img.content; >
<!ATTLIST img
%Core.attrib;
src          %URI;             #REQUIRED
alt           %Text;           #REQUIRED
>

<!ENTITY % Object.content "EMPTY" >
<!ELEMENT object %Object.content; >
<!ATTLIST object
%Common.attrib;
data           %URI;                              #IMPLIED
type           %ContentType;                      #REQUIRED
bml:streamstatus ( stop | play | pause )          #IMPLIED
>

<!ENTITY % Form.content "( %BlkNoForm.mix; )+" >
<!ELEMENT form %Form.content; >
<!ATTLIST form
%Core.attrib;
action          %URI;           #REQUIRED
method          (get|post)          'get'
>

<!ENTITY % InputType.class
"( text | password | submit )"
>
```

```
<!ENTITY % Input.content "EMPTY" >
<!ELEMENT input %Input.content; >
<!ATTLIST input
%Common.attrib;
type          %InputType.class;      'text'
name          CDATA                  #IMPLIED
readonly      (readonly)             #IMPLIED
value         CDATA                  #IMPLIED
maxlength     %Number;               '40'
accesskey     %Character;            #IMPLIED
>

<!ENTITY % Textarea.content "( #PCDATA )">
<!ELEMENT textarea %Textarea.content; >
<!ATTLIST textarea
%Common.attrib;
accesskey     %Character;            #IMPLIED
name          CDATA                  #IMPLIED
readonly      (readonly)             #IMPLIED
>

<!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 % Link.content "EMPTY" >
<!ELEMENT link %Link.content; >
<!ATTLIST link
href          %URI;          #IMPLIED
>
```

```
<!ENTITY % Head.content "( title, meta?, link?,    script*, bml:bevent? )" >
<!ELEMENT head %Head.content; >

<!ENTITY % Body.content
"( div | form | p | pre )+"
>
<!ELEMENT body %Body.content; >
<!ATTLIST body
%Core.attrib;
>
<!ENTITY % Html.content "( head, body )" >
<!ELEMENT html %Html.content; >

<!ENTITY % Bevent.content "( bml:beitem )+" >
<!ELEMENT bml:bevent %Bevent.content; >
<!ATTLIST bml:bevent id ID #IMPLIED>
<!ENTITY % BMLEventType
"(EventMessageFired|ModuleUpdated|ModuleLocked|TimerFired
|DataEventChanged|MediaStopped|MainAudioStreamChanged)"
>
<!ENTITY % BMLTimeMode
"(absolute|origAbsolute)"
>
<!ENTITY % Beitem.content "EMPTY" >
<!ELEMENT bml:beitem %Beitem.content; >
<!ATTLIST bml:beitem
Id                  ID                  #REQUIRED
type                %BMLEventType;      #REQUIRED
onoccur             %Script;            #REQUIRED
es_ref              %URI;               #IMPLIED
message_group_id    (0 | 1)             '0'
message_id          %Number;            #IMPLIED
message_version     %Number;            #IMPLIED
module_ref          %URI;               #IMPLIED
time_mode           %BMLTimeMode;       #IMPLIED
time_value          CDATA               #IMPLIED
```

```
object_id          CDATA              #IMPLIED
subscribe          (subscribe)        #IMPLIED
>

<!-- End of BML DTD -->
```

Appendix 4 Default style sheet

```
@media tv {
/* margin */
div, p, pre, form, input, textarea, object, img { margin: 0 !important }
/* padding */
div, form, object, img { padding-top: 0 !important; padding-right: 0 !important; padding-bottom: 0 !important; padding-left: 0 !important }
/* border */
div, p, pre, form, input, textarea { border-width: 0; border-top-color: transparent; border-right-color: transparent; border-bottom-color: transparent; border-left-color: transparent; }
object, img { border-width: 0 !important; border-style: none !important }
/* display */
html, head, title, meta, script, link, bevent, beitem { display: none !important }
body, div, pre, form, input, textarea, object, img { display: block !important }
p { display: block }
br, span, a { display: inline !important }
/* position */
div, p, pre, form, input, textarea, object, img { position: absolute !important }
br, span, a { position: static !important }
/* top, left, width, height */
div, p, pre, form, input, textarea, object, img { top: 0; left: 0; width: 0; height: 0 }
/* z-index */
body, div, p, pre, br, span, a, form, input, textarea, object, img { z-index: auto !important }
/* line-height */
br, span, a { line-height: inherit !important }
/* visibility */
body { visibility: visible !important }
span, a { visibility: inherit !important }
/* overflow */
div, p, pre, form, input, textarea, object, img { overflow: hidden !important }
/* color */
p, pre, input, textarea { color: black }
span, a { color: inherit }
/* background-color */
object, img { background-color: transparent !important }
body { background-color: white }
/* background-repeat */
body { background-repeat: repeat !important }
/* font-family */
p, pre, span, a, input, textarea { font-family: "round gothic " !important }
/* text-align */
p, input, textarea { text-align: left }
/* white-space */
p, input { white-space: normal !important }
pre, textarea { white-space: pre !important }
/* pixel size */
body { pixel size: 240x480 !important }
/* marquee */
p { -wap-marquee-loop: 1; -wap-marquee-dir: rtl !important }
}
```

Appendix 5 Operation scope for browser pseudo-object in Browser for the C-profile contents on the internets

Operated as Table Appendix 5-1.

Descriptions in the “operation” column are as follows.

“O” Basic function in this specification.

“O(*1)” Optional function in this specification. Therefore, when utilizing these functions in contents, inspect the availability of the concerned function in the receiver by the getBrowserSupport() function, and call the concerned function only when it is available.

“O(*2)” Basic function in this specification in principle. However, with receivers which cannot utilize communication functions in BML browsers, implementation of these functions is not necessary. If the function is called, failure is returned as the return value.

“O(*3)” Necessary for receivers that is available the partial TS playback function. When utilizing these functions in contents, inspect availability of the concerned function in receivers by the getBrowserSupport() function, and call the concerned function only when it is available.

“X” Execution is prohibited in broadcasting status, link status and by Browser for the C-profile contents on the internets.
If the function is called, failure is returned as the return value. Refer to 8.3.11.4 for specification of failure behaviours.

“-” Neither a basic function nor an optional function in this specification. When the function is called, the receiver will occur an error.

Table Appendix 5-1 Operation scope of browser pseudo-objects

	Function	Operations in Browser for the C-profile contents on the internet	Notes			
			Others	Operations in broadcasting status	Operations in link status	(Note in original table)
Ureg function						
	Ureg[]	X		O	O	
Greg function						
	Greg[]	O		O	O	Greg lifetime shall be the time period while reception function is operating

	Function	Operations in Browser for the C-profile contents on the internet	Notes			
			Others	Operations in broadcasting status	Operations in link status	(Note in original table)
EPG functions						
	epgGetEventStartTime()	O		O	O	
	epgGetEventDuration()	O		O	O	
	epgTune()	-		-	-	
	epgTuneToComponent()	-		-	-	
	epgTuneToDocument()	-		-	-	
	epgIsReserved()	-		-	-	
	epgReserve()	O(*1)		O(*1)	O(*1)	
	epgCancelReservation()	O(*1)		O(*1)	O(*1)	
	epgRecIsReserved()	-		-	-	
	epgRecReserve()	-		-	-	
	epgRecCancelReservation()	-		-	-	
program group index functions						
	grpIsReserved()	-		-	-	
	grpReserve()	-		-	-	
	grpCancelReservation()	-		-	-	
	grpRecIsReserved()	-		-	-	
	grpRecReserve()	-		-	-	
	grpRecCancelReservation()	-		-	-	
	grpGetNodeEventList()	-		-	-	
	grpGetERTNodeName()	-		-	-	
	grpGetERTNodeDescription()	-		-	-	
	epgXTune()	-		-	-	
Series reservation functions						
	seriesIsReserved()	-		-	-	
	seriesReserve()	-		-	-	
	seriesCancelReservation()	-		-	-	
	seriesRecIsReserved()	-		-	-	
	seriesRecReserve()	-		-	-	
	seriesRecCancelReservation()	-		-	-	
Non-volatile memory functionss						
	readPersistentString()	-		-	-	
	readPersistentNumber()	-		-	-	
	readPersistentArray()	X		O	O	
	writePersistentString()	-		-	-	
	writePersistentNumber()	-		-	-	
	writePersistentArray()	X		O	O	
	copyPersistent()	-		-	-	
	getPersistentInfoList()	-		-	-	
	deletePersistent()	-		-	-	
	getFreeSpace()	-		-	-	
Functions for controlling access-controlled areas						
	isSupportedPersistentType()	-		-	-	
	setAccessInfoOfPersistentArray()	-		-	-	
	checkAccessInfoOfPersistentArray()	-		-	-	
	writePersistentArrayWithAccessCheck()	-		-	-	

	Function	Operations in Browser for the C-profile contents on the internet	Notes			
			Others	Operations in broadcasting status	Operations in link status	(Note in original table)
	readPersistentArrayWithAccessCheck()	-		-	-	
Interaction channel functions						
Interaction channel functions - Delayed call functions						
	registerTransmission()	-		-	-	
	registerTransmissionStatus()	-		-	-	
	getTransmissionStatus()	-		-	-	
	setDelayedTransmissionDataOverBasic()	-		-	-	
Interaction channel functions - Communication Functions assuming BASIC procedure						
	connect()	-		-	-	
	disconnect()	-		-	-	
	sendBinaryData()	-		-	-	
	receiveBinaryData()	-		-	-	
	sendTextData()	-		-	-	
	receiveTextData()	-		-	-	
Interaction channel functions - Communication functions assuming TCP/IP						
	setISPParams()	-		-	-	
	getISPParams()	-		-	-	
	connectPPP()	-		-	-	
	connectPPPWithISPParams()	-		-	-	
	disconnectPPP()	-		-	-	
	getConnectionType()	O(*2)		O(*2)	O(*2)	
	isIPConnected()	O(*2)		O(*2)	O(*2)	
	saveHttpServerFileAs()	-		-	-	
	saveHttpServerFile()	-		-	-	
	sendHttpServerFileAs()	-		-	-	
	saveFtpServerFileAs()	-		-	-	
	saveFtpServerFile()	-		-	-	
	sendFtpServerFileAs()	-		-	-	
	sendTextMail()	-		-	-	
	sendMIMEMail()	-		-	-	
	transmitTextDataOverIP()	O(*2)	User approval shall be acquired by the receiver prior to implementation. Transmission shall not be implemented without approval.	O(*2)	O(*2)	
	setDelayedTransmissionData()	-		-	-	
	getTransmissionStatus()	-		-	-	

	Function	Operations in Browser for the C-profile contents on the internet	Notes			
			Others	Operations in broadcasting status	Operations in link status	(Note in original table)
	getTransmissionResult()	-		-	-	
	setCacheResourceoverIP( )	-		-	-	
Interaction channel functions - Deleyed call functions applicable to BASIC procedure and IP connections						
	getDelayedTransmissionStatus()	-		-	-	
	getDelayedTransmissionResult()	-		-	-	
Interaction channel functions - Function for obtaining line connection status						
	getPrefixNumber()	-		-	-	
Interaction channel functions - Communication functions using the mass call reception service						
	vote()	-		--	-	
Interaction channel functions - Functions for encrypted communication using CAS						
	startCASEncryption()	-		-	-	
	transmitWithCASEncryption()	-		-	-	
	endCASEncryption()	-		-	-	
Interaction channel functions - Functions for communication with public key encryption not using CAS						
	setEncryptionKey()	-		-	-	
	beginEncryption()	-		-	-	
	endEncryption()	-		-	-	
Operational control functions						
	reloadActiveDocument()	O		O	O	
	getNPT()	-		-	-	
	getProgramRelativeTime()	X		O	O	
	isBeingBroadcast()	-		-	-	
	lockExecution()	-		-	-	
	unlockExecution()	-		-	-	
	lockModuleOnMemory()	-		-	-	
	unlockModuleOnMemory()	-		-	-	
	setCachePriority()	-		-	-	
	getTuningLinkageSource()	-		-	-	
	getTuningLinkageType()	-		-	-	
	getLinkSourceServiceStr()	-		-	-	
	getLinkSourceEventStr()	-		-	-	
	getIRDID()	-		-	-	
	getBrowserVersion()	O		O	O	
	getProgramID()	X		O	O	
	getActiveDocument()	O		O	O	
	lockScreen()	O		O	O	
	unlockScreen()	O		O	O	

	Function	Operations in Browser for the C-profile contents on the internet	Notes			
			Others	Operations in broadcasting status	Operations in link status	(Note in original table)
	getBrowserSupport()	O		O	O	
	launchDocument()	O	Only those name spaces starting with http:// and https:// can be specified.	O	O	
	launchDocumentRestricted()	-		-	-	
	quitDocument()	O		O	O	
	launchExApp()	-		-	-	
	getFreeContentsMemory()	-		-	-	
	isSupportedMedia()	-		-	-	
	detectComponent()	X		O	O	
	lockModuleOnMemoryEx()	X		O	X	
	unlockModuleOnMemoryEx()	X		O	O	
	unlockAllModulesOnMemory()	X		O	O	
	getLockedModuleInfo()	X		O	O	
	getBrowserStatus()	O		O	O	
	getResidentAppVersion()	O		O	O	
	isRootCertificateExisting()	-		-	-	
	getRootCertificateInfo()	-		-	-	
	startResidentApp()	-		-	-	
	getDataDisplayAreaSize()	-		-	-	
	setFullDataDisplayArea()	X		O(*1)	O(*1)	
Receiver sound control						
	playRomSound()	O		O	O	
Timer functions						
	sleep()	-		-	-	
	setTimeout()	-		-	-	
	setInterval()	O		O	O	
	clearTimer()	O		O	O	
	pauseTimer()	-		-	-	
	resumeTimer()	-		-	-	
	setCurrentDateMode()	X		O(*3)	O(*3)	
External character functions						
	loadDRCS()	-		-	-	
	unloadDRCS()	-		-	-	
Functions for controlling external devices						
	enumPeripherals()	-		-	-	
	passXMLDocToPeripheral()	-		-	-	
Other functions						
	random()	O		O	O	
	subDate()	O		O	O	
	addDate()	O		O	O	
	formatNumber()	O		O	O	
Subtitle presentation control functions						
	setCCStreamReference()	-		-	-	
	getCCStreamReference()	-		-	-	

	Function	Operations in Browser for the C-profile contents on the internet	Notes			
			Others	Operations in broadcasting status	Operations in link status	(Note in original table)
	setCCDisplayStatus()	-		-	-	
	getCCDisplayStatus()	-		-	-	
	getCCLanguageStatus()	-		-	-	
Directory management functions						
	saveDir()	-		-	-	
	saveDirAs()	-		-	-	
	createDir()	-		-	-	
	getParentDirName()	-		-	-	
	getDirNames()	-		-	-	
	isDirExisting()	-		-	-	
File management functions						
	saveFile()	-		-	-	
	saveFileAs()	-		-	-	
	getFileNames()	-		-	-	
	isFileExisting()	-		-	-	
File input/output functions						
	writeArray()	-		-	-	
	readArray()	-		-	-	
Inquiry functions						
	getDirInfo()	-		-	-	
	getFileInfo()	-		-	-	
	getCarouselInfo()	-		-	-	
	getModuleInfo()	-		-	-	
	getContentSource()	-		-	-	
	getStorageInfo()	-		-	-	
Data carousel storage functions						
	saveCarouselAs()	-		-	-	
	saveCarousel()	-		-	-	
	saveModuleAs()	-		-	-	
	saveModule()	X		O(*1)	O(*1)	
	saveResourceAs()	-		-	-	
	saveResource()	X		O(*1)	O(*1)	
Functions for controlling bookmark area						
	writeBookmarkArray()	-		-	-	
	readBookmarkArray()	-		-	-	
	deleteBookmark()	-		-	-	
	lockBookmark()	-		-	-	
	unlockBookmark()	-		-	-	
	getBookmarkInfo()	-		-	-	
	getBookmarkInfo2()	-		-	-	
	startResidentBookmarkList()	-		-	-	
Printing functions API - printing basic functions						
	getPrinterStatus()	-		-	-	
	printFile()	-		-	-	
	printTemplate()	-		-	-	
	printUri()	-		-	-	
	printStaticScreen()	-		-	-	

	Function	Operations in Browser for the C-profile contents on the internet	Notes			
			Others	Operations in broadcasting status	Operations in link status	(Note in original table)
Printing functions API - memory card functions						
	saveImageToMemoryCard ()	X		O(*1)	O(*1)	
	saveHttpServerImageToMemoryCard()	X		O(*1)	O(*1)	
	saveStaticScreenToMemoryCard()	-		-	-	
Digital terrestrial broadcasting specific functions						
	X_DPA_mailTo()	X		O(*1)	O(*1)	
	X_DPA_startResidentApp( )	O		O	O	
	X_DPA_phoneTo()	X		O(*1)	O(*1)	
	X_DPA_getRcvCond ()	X		O(*1)	O(*1)	
	X_DPA_getCurPos ()	X		O(*1)	O(*1)	
	X_DPA_saveExAppFile()	X		O(*1)	O(*1)	
	X_DPA_startExAV()	X		O(*1)	O(*1)	
	X_DPA_stopExAV()	X		O(*1)	O(*1)	
	X_DPA_tuneWithRF()	O(*1)	Finish browser after implementation	O(*1)	O(*1)	
	X_DPA_writeSchInfo()	X		O(*1)	O(*1)	
	X_DPA_getComBrowserUA ()	O(*2)		O(*2)	O(*2)	
	X_DPA_writeAddressBookInfo()	X		O(*1)	X	
	X_DPA_launchDocWithLink()	X		X	O(*2)	
	X_DPA_chkAVtype()	X		O(*1)	O(*1)	
	X_DPA_getIRDID()	X		O	O	
	X_DPA_writeCproBM()	X		O(*2)	O(*2)	

Appendix 6 Operation of the getBrowserStatus() argument

The table below shows functions available for inspections in combination of “statusname” and “additionalinfo” when “sProvider” is "DPA".

Table Appendix 6-1 String specified as getBrowserStatus() arguments

statusname	additionalinfo	Operations of getBrowserStatus()
IRDState	One of the following "Broadcast" "Link" "UnLink"	If browser is in the status specified by “additionalinfo”, return 1. "Broadcast": data broadcasting reception status "Link": link status "UnLink": status other than data broadcasting status and link status

Appendix 7 Communication restrictions for carriers

With mobile phones, there are times when the carriers has communication restrictions that are not specified in the this specification, based on agreements with the broadcaster. Specifically, limited domains available for communication from broadcasting contents per broadcaster affiliation to one, and limited communication from contents presented in BML browsers to be URI's under this domain (including sub-domains). (However, in cases like presenting contents by starting up browsers for the contents on the internet by X_DPA_startResidentApp(), if another application, started up from a BML browser API, performs communication, this domain restriction does not apply.)

A communication restriction is implemented as receiver features processing, using tables (correspondence table of domain name and affiliation ID that specifies affiliation) pre-stored in mobile phones. Therefore, when mobile phones communicate from contents presented in BML browsers, communication with servers outside of the domain specified by the table will fail, even if the communication destination is under the base URI. Also, as a result of this communication restriction, NVRAM areas for the affiliation will no longer be accessed from contents other than contents under the specified domain.

Such communication restrictions are not necessary for receivers other than mobile phones. However, broadcasters cannot avoid arbitrary receivers performing similar communication restrictions. If the communication destination will be limited, a table showing the correspondence between affiliation ID for identifying the affiliation and domain names can be acquired from the governing organization shown in Appendix 13.

Also, only with operations of this restriction, the direct specification of IP addresses as URI host names will result in a communication failure as servers outside of the domain.

Appendix 8 Arrangement of elementary stream and empty carousel operations in the PMT

In C-profile, the visible area for data broadcasting is defined in an independent area. Therefore, unlike in profile A, allocation and release of the area is assumed in C-profile. That is to say, specifications specified in Table Appendix 8-1 can be assumed as operation example of receivers here. In this example, receivers control the allocation and release of the display area by description of default components (component_tag=0x80) in the PMT.

In consideration of this example, when data broadcasting contents are not transmitted for a certain period of time, such as programs, it is effective for the receiver to delete the default component in the PMT, if this is to avoid securing unnecessary visible area for data broadcastings for the receiver, and to avoid letting the receiver to continue to receive data.

However, if the description of the default component in PMT changes in a short period of time, it may inconvenience users due to the blink of the visible area for data broadcasting. Transmission of empty carousels can be assumed as a countermeasure for this blink in the visible area for data broadcasting, if the time period is too short to not transmit data broadcasting contents that should be presented. Here, we need to be careful since there is a possibility that it will cause receivers to secure unnecessary visible area for data broadcastings, if empty carousels are transmitted for a long period of time.

Considering the above points, whether there is description of the default component in the PMT or not, and the transmission of empty carousel recommend to be operated with great care.

Also, this appendix is described in order to specify general reference information upon designing and operation of transmission systems. This appendix do not specify specific restrictions for receiver specifications, transmission system specifications and transmission operations.

Table Appendix 8-1 Relationship examples for reception conditions and the visible area for data broadcasting

Item	Reception condition			Presentation status			
	description for default component in PMT	Carousel status	Other causes	Presentation of broadcasting contents		Presentation of link contents	
				Presentation area	Presentation contents	Presentation area	Presentation contents
Upon tuning							
	No	-	-	No	-	-	-
	Yes	Empty	-	Yes	Black screen or built-int wallpaper	-	
	Yes	Not empty	-	Yes	Presentation of startup document	-	
Data event update							
(data_event_id change)	Yes	Not empty -> not empty	-	Yes	Present startup document of data event after update	Yes	Continue presenting document in presentation
	Yes	Not empty -> empty	-	Yes	Black screen or built-in wallpaper	Yes	Continue presenting document in presentation
	Yes	Empty -> empty	-	Yes	Black screen or built-in wallpaper	Yes	Continue presenting document in presentation
	Yes	Empty -> not empty	-	Yes	Present data event startup document after update	Yes	Continue presenting document in presentation
Upon PMT update							
(Change in description for default component)	Yes -> No	-	-	No	-	Yes	Continue presenting document in presentation
	No -> Yes	-	-	Yes(*1)	Startup document	Yes	Continue presenting document in presentation
Change in broadcasting reception status							
(RF level, PMT reception, DII reception, etc.)	-	-	Good -> Ng	No change	No change	No change	Continue presenting document in presentation
	-	-	Bad -> Ng	No change	Present latest content (*2)	No change	Continue presenting document in presentation

- : Cause or symptom not assumed

(*1) Operations worth on 0x80 existence upon tuning

(*2) If a data event update and PMT update is detected after recovering the reception status, standardize on normal process.

Mandatory reloading or startup document transition is not assumed.

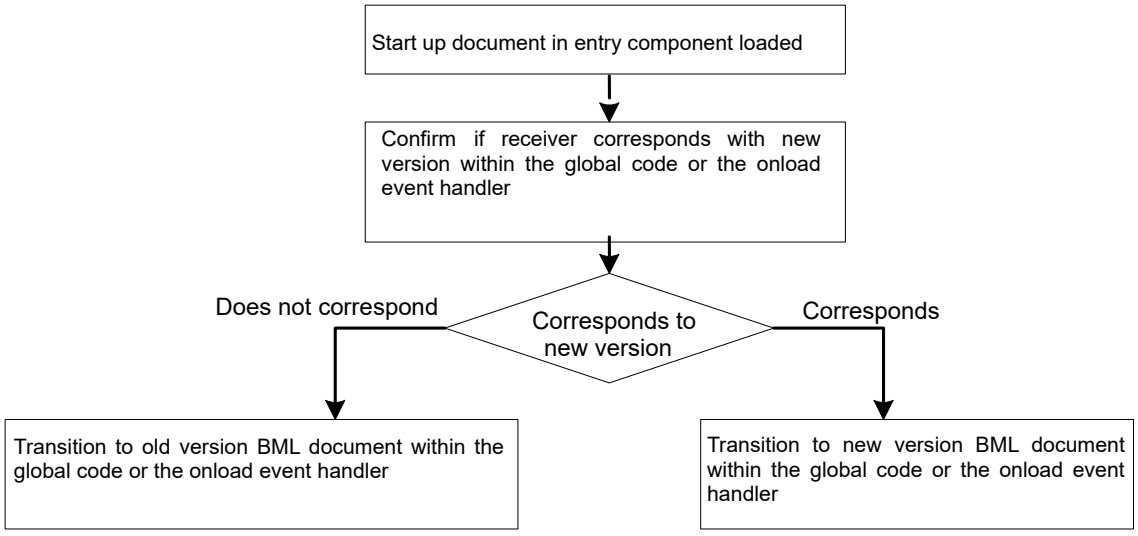
Appendix 9 Version update of the BML version

Because BML updates are assumed along the development of services, assumed examples for receiver behaviors and operations regarding BML version updates are defined below. Behavior necessary for current version receivers(major version=12, minor version=0) and operation of version numbers are specified based on the assumptions.

1. Assumed examples of contents operations when the different versions are transmitted

◆ If the “bml_major_version” of Data Component Descriptor in PMT is 12:

A. If the startup document is shared



B. If the startup document is not shared within the same component

- Standardize the new version startup document with a name that is different from "startup.bml”.
- Transmit multiple startup documents corresponding to each version to the same component

◆ If the “bml_major_version” of Data Component Descriptors in PMT are 12 and the other number:

C. When transmitting startup documents in another component

- Specify a new version entry component by component tag value other than 0x80.
- Transmit to the same TS with an arrangement of elementary stream that includes entry components corresponding to each version.

2. Behaviors necessary for current version receivers (major version = 12, minor version = 0)

The following specifies receiver behaviors in order to avoid occurring abnormal behaviors when assuming transmission of multiple version contents upon version updates specified in A-C above.

- An error shall not be occurred even if a BML document, whose version's operation does not correspond with the receiver, is included in components in presentation.
- An error shall not be occurred even if a resource, whose media type's operation does not correspond with the receiver, is included in components in presentation.
- An error shall not be occurred if a BML document, whose minor_version only is different, is presented. Operations of non-supported functions shall be implementation dependent.
- Behavior upon presentation of a BML document, whose major_version is different, is implementation dependent.
- Behavior upon presentation of resources of a media type which is not supported by receiver is implementation dependent.
- Startup "startup.bml" first and start contents presentation (in case of B).
- Start the startup document from components whose component tag value is 0x80 and start contents presentation (in case of C)
- Even if it is an ES with a undefined component tag value, it shall not be an error (in case of C).
- Behavior upon reception other than "bml_major_vesion" of Data Component Descriptor in PMT is implementation dependent (if major_vesion=12 is not included)

3. Operation of version numbers

- Minor version updates shall be limited to functions that can be implemented without failure behaviour even in older version receivers.
- Major version updates for functions that cannot be implemented without a failure behaviour even in older version receivers.
- The implementation shall be discussed with the governing organization described in Appendix 13.

Appendix 10 Guideline concerning the operation of personal information in NVRAM and communications in data broadcasting

1. Definition of personal information in data broadcasting

Under these guidelines, personal information shall mean "information about a living individual which can identify the specific individual by name, date of birth or other description contained in such information (including such information as will allow easy reference to other information and will thereby enable the identification of the specific individual)" as prescribed in Article 2 (1) of the "Act on the Protection of Personal Information".

2. Handling of information stored in NVRAM in data broadcasting

- Information writing/reading shall be performed exclusively by contents supplied by broadcasters.
- Personal information shall be handled upon user agreement.
- Stored information belongs to the user.
- Data such as “points” used for prizes/games/premium exchanges, etc. shall be handled carefully, from the viewpoint of user profit security.

3. Handling of personal information in TCP/IP communication

- TLS/SSL shall be used when handling personal information in Interactive Data Broadcasting Services and/or C-profile linked contents.

4. Guideline concerning the operation of personal information in data broadcasting content

- Utilization scope and objectives of personal information shall clearly be presented.
- Personal information shall be utilized under the permission of the viewer.

5. Guidelines for privacy policy (including “FAQ” and “help”, etc.) by data broadcasting contents

- “privacy policy” shall be specified per broadcaster, and shall be released by a standard procedure that can be acquired by viewers.
- It is recommended to describe the following items in “privacy policy”
  (1) Registered information is in receiver’s memory.
  (2) Personal information in receiver’s memory shall be managed in viewer’s responsibility, and needs to be erased by the viewers themselves upon transfer or disposal of the receiver.
  (3) Personal information shall be utilized only in the utilization scope and objectives specified by broadcater.
  (4) A disclaimer is necessary for data deletion.

Appendix 11 Guideline concerning the cache function and URI history

The following shows the guideline concerning the cache function of the receiver, storage of the URI history and transitions based on the URI history. In the case of mobile phone, separately defined prescriptions shall be followed if they exist.

1. Cache function in the receiver unit

If the receiver will implement a function for cacheing C-profile linked content via HTTP protocol, the receiver shall operate by referencing the following HTTP headers and fields.

- Operation of Cache-Control
  - In the event when the receiver receives a response message with "no-store" specified in the Cache-Control header from the interactive Web server, the receiver must not cache the file specified in the HTTP session.
  - In the event when the receiver receives a response message with "no-cache" specified in the Cache-Control header from the interactive Web server, the receiver shall operate in the same manner as when "no-store" has been specified.
  - The operation of the receiver when it receives values other than "no-cache" and "no-store" shall be implementation dependent.

- Operation of Pragma

In the event when the receiver receives a response message with "no-cache" specified in the Pragma field from the interactive Web server, the receiver shall operate in the same manner as when the above "Cache-Control:no-cache" has been specified.

- Operation of other headers

In addition to the cache function, the receiver shall also support "If modified since" of the request header as a means for reducing the communication transactions. The date/time format shall be the fixed length subset defined in RFC1123; and the receiver shall be able to interpret this format.

- Operation of cache-clear

  Cached content shall always be cleared if any of the following operations occurs:

  - When the channel has been switched
  - When the BML browser has terminated
  - In the case of a transition from C-profile linked content to broadcasting content

    * However, a receiver supporting "If modified since" shall not clear the cache if a transition is made from C-profile linked content to broadcasting content.

2. Storage of URI history by the receiver function and transitions based on the URI history

This document does not assume the receiver to have the function of recording the URI of displayed documents as URI history and allowing transitions to be made directly from URI history via user operation; and it has been created based on the premise that the broadcasted content does not have such a function either. For this reason, unexpected transitions may potentially occur on a receiver implementing the relevant function, causing the presentation of some content to fail. Therefore, it is not recommendable to implement such type of function. If such type of function will be implemented based on product planning, the following restrictions must at least be observed.

- Restrictions on transitions based on the URI history

  Transitions based on the URI history shall be limited to within C-profile linked content and must not be performed for data broadcasting content.

- Restriction regarding the storage of URI history

  URI history shall be cleared if any of the following operations occurs:

  - When the channel has been switched
  - When the BML browser is terminated
  - In the case of a transition from C-profile link content to broadcasting content

Appendix 12 Details of the communication function

The details of the communication function are as follows.

In the case of mobile phone, separately defined prescriptions shall be followed if they exist.

1. Operation of methods

- If "http:" has been specified in the URI, the receiver and the interactive Web server shall perform communication based on HTTP/1.1 using the port specified by the URI.
- If "https:" is specified in the URI, the receiver and the interactive Web server shall perform encrypted communication based on HTTP/1.1 after establishing a connection via TLS1.0, SSL3.0, or TLS1.2 using the port specified by the URI.
  See Vol. 6 for the implementation of TLS and SSL on receivers.
- If the port number was not specified in the URI, ports 80 and 443 shall be used for "http:" and "https:" respectively.

2. Operation of HTTP/1.1

- The fixed length subset defined in RFC1123 shall be operated for the date/time formats.
  - It shall be desirable that the interactive Web server transmits dates to the receiver only in the date/time format of the fixed length subset defined in RFC1123.
  - The receiver shall be able to interpret the fixed length subset defined in RFC1123 as the date/time formats. In the event when a date in RFC1036 or ANSI C format is received, it shall be desirable that they be interpreted but they may also be ignored.

    Examples:

    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
- Only the "Shift_JIS" character set shall be operated in request messages, response messages and entities. The operation of the receiver when it receives other character sets shall be implementation dependent.
- "identity" shall be operated for content coding (content-coding). "deflate" shall be operated as an option. Operations of the receiver not supporting "deflate" when it receives "deflate" and when it receives any other values shall be implementation dependent.
- "chunked" shall be operated if transfer coding (transfer-coding) will be specified. The operation of the receiver when it receives other values shall be implementation dependent.
- If the browser name and version will be specified, the product token shall be consistent with the return value of getBrowserVersion()
- Operation of quality values shall be implementation dependent.

- "ja" shall be operated for the language tag. The operation of the receiver when it receives other language tags shall be implementation dependent.
- The receiver shall implement "Keep-Alive".

3. Security

To protect personal information, the receiver shall implement the TLS and SSL protocols for encrypted communication.

(1) Cipher Suite

- The receiver shall incorporate the cipher suite described in Table Appendix 12-1. However, incorporation of cipher suites other than below shall be implementation dependent.

Table Appendix 12-1 List of cipher suites to be supported by the receiver

Cipher Suite	Key exchange	Data encryption	Hash	Remarks
TLS_RSA_WITH_DES_CBC_SHA	RSA	DES	SHA	*1
TLS_RSA_WITH_3DES_EDE_CBC_SHA	RSA	Triple-DES	SHA	
SSL_RSA_WITH_DES_CBC_SHA	RSA	DES	SHA	*1
SSL_RSA_WITH_3DES_EDE_CBC_SHA	RSA	Triple-DES	SHA	*1
TLS_RSA_WITH_AES_128_CBC_SHA	RSA	AES_128	SHA	*2
TLS_RSA_WITH_AES_256_CBC_SHA	RSA	AES_256	SHA	*2
TLS_RSA_WITH_AES_128_CBC_SHA256	RSA	AES_128	SHA256	*2
TLS_RSA_WITH_AES_256_CBC_SHA256	RSA	AES_256	SHA256	*2
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA	DHE_RSA	Triple-DES	SHA	*2
TLS_DHE_RSA_WITH_AES_128_CBC_SHA	DHE_RSA	AES_128	SHA	*2
TLS_DHE_RSA_WITH_AES_256_CBC_SHA	DHE_RSA	AES_256	SHA	*2
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	DHE_RSA	AES_128	SHA256	*2
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256	DHE_RSA	AES_256	SHA256	*2
TLS_RSA_WITH_AES_128_GCM_SHA256	RSA	AES_128	SHA256	*2
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	DHE_RSA	AES_128	SHA256	*2
TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	ECDHE_ECDSA	Triple-DES	SHA	*3
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	ECDHE_ECDSA	AES_128	SHA	*3
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	ECDHE_ECDSA	AES_256	SHA	*3
				*3
TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	ECDHE_RSA	Triple-DES	SHA	*3
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	ECDHE_RSA	AES_128	SHA	*3
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	ECDHE_RSA	AES_256	SHA	*3
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	ECDHE_ECDSA	AES_128	SHA256	*3
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	ECDHE_ECDSA	AES_256	SHA384	*3
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	ECDHE_RSA	AES_128	SHA256	*3
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	ECDHE_RSA	AES_256	SHA384	*3
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	ECDHE_ECDSA	AES_128	SHA256	*3
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	ECDHE_ECDSA	AES_256	SHA384	*3
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	ECDHE_RSA	AES_128	SHA256	*3
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDHE_RSA	AES_256	SHA384	*3

*1: Cipher Suite that is not required to be implemented in receivers that only support TLS1.2.

*2: Cipher Suite that is required to be implemented in receivers that support TLS1.2.

*3: Cipher Suite that is required to be implemented in receivers that support ECC encryption (optional).

(2) Authentication function

- The receiver shall support server authentication in order to prevent "phishing" in various services.

(3) Certificate verification items

- The certificate of the interactive Web server shall have less than 5 hierarchical levels. The receiver shall be able to verify at least 5 certificates.

- The certificate shall comply with RFC3280, and it shall be verified according to the verification content shown in Table appendix 12-2. However, signature verification shall not be required for root certificates.

Table Appendix 12-2 Certificate verification content

Verification item	Root Certificate	Intermediate certificate	Server certificate
Basic Certificate Field			
Signature (signatureValue)	-	Verification shall be required.	Verification shall be required.
Version (version)	-	The version shall be 3.	The version shall be 3.
Issuer (issuer's DN)	-	It shall be a non-empty distinguished name (DN) matching the Subject of the certificate in the one higher level.	It shall be a non-empty distinguished name (DN) matching the Subject of the certificate in the one higher level.
Expiration period (validity)	It shall contain the current time.	It shall contain the current time.	It shall contain the current time.
CN (CommonName) of Subject (Subject's DN)	-	-	It must match the FQDN of the access server.
Certificate Extensions -> They shall be confirmed when the following fields exist and are critical.			
Basic constraints (Basic Constraints)	—	【Conditions】 If "Basic Constraints" is critical: ・ "cA" must be true. ・ If "pathLenConstraint" is included, its value shall meet the depth restriction of the certificate chain.	—
Key application (KeyUsage)	—	—	【Conditions】 If "KeyUsage" is critical: Either "keyEncipherment" or "keyAgreement" shall be included.
Extension key application (ExtendedKeyUsage)	—	—	【Conditions】 If "ExtendedKeyUsage" is critical: "id-kp-serverAuth" shall be included.

* "-" indicates that verification is not required.

(4) In the event of TLS and/or SSL error

- The receiver shall display an alert message in the event when a TLS or SSL error occurs. The message display method shall be implementation dependent, but the message examples in Table Appendix 12-3 shall desirably be followed.

- The receiver shall not connect to the interactive Web server in the event when a TLS or SSL error occurs.

Table Appendix 12-3 Examples of alert messages to be displayed when a TLS or SSL error occurs

	Cause	Message example
1	If a root certificate is not set on the receiver side	"No root certificate is set on the receiver side. Connection cannot be established because the safety of the connection destination cannot be verified."
2	If a root certificate is set on the receiver side, but cannot be verified against the server certificate at the connection destination	"Connection cannot be established because the safety of the connection destination cannot be verified using the currently set root certificate."
3	If the root certificate is expired	"Connection cannot be established because the safety of the connection destination cannot be verified using the currently set root certificate."
4	In the case of an authentication processing error due to network failure, etc. (i.e. a timeout process)	"Connection could not be established within the set time limit."
5	If the certificate at the connection destination is expired	"The certificate at the connection destination is expired. Connection cannot be established because the safety of the connection destination cannot be verified."
6	In the case of mismatch CommonName of the server certificate does not match	"The certificate at the connection destination does not contain the title of the page to be displayed. Connection cannot be established because the safety of the connection destination cannot be verified."
7	If the certificate is disabled on the receiver side	"The root certificate for authenticating the connection destination is currently disabled on the receiver side. Please enable the root certificate and try connecting again."
8	If the certificate is tampered with	"Connection will be aborted because falsified certificate has been detected at the connection destination."
9	In the case of an authentication error due to invalid chain, etc.	"Connection will be aborted because a problem has been detected in the certificate setting at the connection destination."

Appendix 13　Governing organization

The Association for Promotion of Advanced Broadcasting Services
(http://www.apab.or.jp/)

OPERATIONAL GUIDELINES FOR
DIGITAL TERRESTRIAL TELEVISION BROADCASTING

ARIB TECHNICAL REPORT

ARIB TR-B14 VERSION 6.7-E1
(Fascicle 3)
(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