ENGLISH TRANSLATION

Data Coding and Transmission Specifications for Digital Broadcasting

ARIB STANDARD

ARIB STD-B24 Version 5.1

VOLUME 2 (2/2)

Established on	June	20,	2000	Version 1.0
Revised on	June	28,	2002	Version 3.2
Revised on	February	5,	2004	Version 4.0
Revised on	May	29,	2006	Version 5.0
Revised on	March	14,	2007	Version 5.1

Association of Radio Industries and Businesses (ARIB)

General Notes to the English translation of ARIB Standards and Technical Reports

1. The copyright of this document is ascribed to the Association of Radio Industries and Businesses (ARIB).

2. 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.

3. The ARIB Standards and ARIB Technical Reports are usually written in Japanese and approved by the ARIB Standard Assembly. This document is a translation into English of the approved document for the purpose of convenience of users. If there are any discrepancies in the content, expressions, etc., between the Japanese original and this translated document, the Japanese original shall prevail.

4. The establishment, revision and abolishment of ARIB Standards and Technical Reports are approved at the ARIB Standard Assembly, which meets several times a year. Approved ARIB Standards and Technical Reports, in their original language, are made publicly available in hard copy, CDs or through web posting, generally in about one month after the date of approval. 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:

http://www.arib.or.jp/english/index.html

Contents

Preface

Volume 1 Data Coding

Part 1 Reference Model for Data Broadcasting

Part 2 Monomedia Coding

Part 3 Coding of Caption and Superimpose

Volume 2 XML-based Multimedia Coding Scheme

Appendix 1 Operational Guidelines

Appendix 2 Operational Guidelines for Implementing Basic Services

Appendix 3 Operational Guidelines for Implementing Extended Services for Fixed Receiving System

Appendix 4 Operational Guidelines for Implementing Extended Services for Portable Receiving System

Appendix 5 Operational Guidelines for Implementing Extended Services for Mobile Receiving System

Volume 3 Data Transmission Specification

Preface

ARIB (Association of Radio Industries and Businesses) establishes the "ARIB Standards" for the basic technical conditions of standard specifications related to variety of radio communication equipments, broadcasting transmission equipments, and its reception equipments using radio wave with the participation of radio communication equipment manufacturers, broadcasting equipment manufacturers, electric communication companies, service providers and other users.

"ARIB Standards" are nongovernmental standards established by combining governmental technical standards established for the purpose of effective use of frequency and to avoid interference of other users, and nongovernmental optional standards established for convenience for radio communication equipment manufacturers, broadcasting equipment manufacturers, electric communication companies, service providers and users, in order to secure appropriate quality and compatibility of radio communication equipment and broadcast equipment, etc.

This standard is established for "Data Coding and Transmission Specification for Digital Broadcasting" by the approval of the standardization committee, participated by radio communication equipment manufacturers, broadcast equipment manufacturers, electric communication companies, service providers and users irrespectively, to secure impartiality and clearness.

For data broadcasting of digital broadcasting, it is directed by the Telecommunications Technology Council on July 21, 1999 that it is desired that the most desirable multimedia coding specification in Japan at this point should be based on an XML-based specification, which is superior in many points such as "function", "contents production environment", "compatibility with other media", "data processing at terminal side", "extension ability of coding method", and "future direction of engineering development", etc., and that the detailed specifications should be standardized by the nongovernmental standardization organization with flexibility.

This standard is established as nongovernmental standard of data broadcasting specification used in Japan based on this direction, and consists of three parts: mono-media coding, multimedia coding, and data transmission specification. Compatibility with multiplex data broadcasting specification, which is already used in Japan is considered for mono-media coding. Compatibility with network usage or data broadcasting method in Europe and America is considered for multimedia coding and the coding scheme is based on XML coding specified in W3C specification adding necessary specifications for broadcasting. Each coding scheme in this standard is applied to whole broadcasting media generally and the conditions proper to broadcasting media derived from transmission methods and service requirements should be specified as operational restrictions.

Though this standard is mainly applied to BS digital broadcasting as the first step, the specification should be completed adding necessary specifications for other broadcasting media, considering trends of international standardization and new technological trends which cannot be assumed yet.

We hope that this standard will be put to practical use actively by radio communication equipment manufacturers, broadcast equipment manufacturers, electric communication companies, service providers, users, and so on.

Notice:

This standard does not describe industrial proprietary rights mandatory to this standard. However, the right proprietor of the industrial proprietary rights has expressed that "Industrial proprietary rights related to this standard, listed in the annexed table below, are possessed by the applicator shown in the list. However, execution of the right listed in the annexed table below is permitted indiscriminately, without exclusion, under appropriate condition, to the user of this standard. In the case when the user of this standard possesses the mandatory industrial proprietary rights for all or part of the contents specified in this standard, and when he asserts his rights, it is not applied."

Annexed table

Patent applicant	Name of invention	Patent number	Remarks
Matsushita Electric Industrial Co., Ltd.	情報処理装置	特開平 04-205415号	Japan
	データサーバ装置及び端末装置	特開平 06-139173号	Japan
	放送を用いて対話性を実現する送信装置、受信装置、受信方法、その受信プログラムを記録した媒体、通信システム	特開平 10-070712号	Japan
	データ入出力端末装置	特開平 10-074134号	Japan
	情報処理装置	特開平 10-083270号	Japan
	ﾃﾞｰﾀの提示を制御するﾃﾞｰﾀ提示制御装置、ﾃﾞｰﾀの提示を～情報を送信するﾃﾞｰﾀ送信装置及びﾃﾞｰﾀ～ﾃﾞｰﾀ提示制御情報編集装置	特開平 10-164530号	Japan
	ﾃﾞｼﾞﾀﾙ放送ｼｽﾃﾑ、ﾃﾞｼﾞﾀﾙ放送装置及びﾃﾞｼﾞﾀﾙ放送における受信装置	特開平 10-304325号	Japan
	ﾃﾞｼﾞﾀﾙ放送装置、受信装置、ﾃﾞｼﾞﾀﾙ放送ｼｽﾃﾑ、受信装置に適用するﾌﾟﾛｸﾞﾗﾑ記録媒体	特開平 10-313449号	Japan
	番組編集装置および番組受信装置	特願平 10-020585号	Japan
	放送局ｼｽﾃﾑ及び受信機	特願平 10-195093号	Japan
	ﾃﾞｼﾞﾀﾙ放送のための記録再生装置および方法	特願平 11-367308号	Japan
	ﾃﾞｰﾀ送受信ｼｽﾃﾑおよびその方法	特願平 11-103619号	Japan
	ﾃﾞｼﾞﾀﾙﾃﾞｰﾀ送受信ｼｽﾃﾑおよびその方法	特願平 11-124986号	Japan
	Submitted comprehensive confirmation of patents applied to the revised parts of ARIB STD-B24 Ver3.8 *5		
TOSHIBA CORPORATION	多重放送ｼｽﾃﾑとこのｼｽﾃﾑで使用される放送送信装置および放送受信装置	特開平 09-162821号	Japan

Patent applicant	Name of invention	Patent number	Remarks
	ﾃﾞｼﾞﾀﾙ放送装置及びﾃﾞｼﾞﾀﾙ放送方法、ﾃﾞｼﾞﾀﾙ放送受信装置及びﾃﾞｼﾞﾀﾙ放送受信方法、ﾃﾞｼﾞﾀﾙ放送受信ｼｽﾃﾑ*16	特許第3621682号	Japan
NHK (Japan Broadcasting Corporation)	文書情報出力装置および方法	特開平 9-244617号	Japan
	入力データの自動選択処理装置	特開平 11-328189号	Japan
	ﾏﾙﾁﾒﾃﾞｨｱ型情報ｻｰﾋﾞｽ方式およびその方式の実施に使用する装置	特開平 11-331104号	Japan
Sony Corporation *1	音声信号圧縮方法及びメモリ書き込み方法	特許第 1952835号	Japan
	オーディオ信号処理方法	特許第 3200886号	Japan
	オーディオ信号処理方法	特許第 3141853号	Japan
	信号符号化又は複合化装置、及び信号符号化又は複合化方法、並びに記録媒体	WO94/28633	Japan
	信号符号化方法及び装置、信号複合化方法及び装置、並びに記録媒体	特開平 7-168593	Japan
	符号化音声信号の複合化方法	特開平 8-63197	Japan
	音声信号の再生方法、再生装置及び伝送方法	特開平 9-6397	Japan
	音声信号の再生方法及び装置、並びに音声複合化方法及び装置、並びに音声合成方法及び装置、並びに携帯無線端末装置	特開平 9-190196	Japan
	音声符号化方法、音声複合化方法及び音声符号化複合化方法	特開平 8-69299	Japan
	音声符号化方法及び装置、音声複合化方法及び装置	特開平 9-127991	Japan
	符号化データ複合化方法及び符号化データ複合化装置	特許 2874745号	Japan
	映像信号符号化方法	特許 2877225号	Japan
	符号化データ編集方法及び符号化データ編集装置	特許 2969782号	Japan
	動画像データエンコード方法及び装置、並びに動画像データデコード方法および装置	特許 2977104号	Japan
	動きベクトル伝送方法及びその装置並びに動きベクトル複合化方法及びその装置	特許 2712645号	Japan
Mitsubishi Electric Corporation	Submitted comprehensive confirmation of patents applied to the revised parts of ARIB STD-B24 Ver3.1 *2		
	マルチメディア多重方式*3	特許第 3027815号	Japan

Patent applicant	Name of invention	Patent number	Remarks
	マルチメディア多重方式*3	特許第 3027816号	Japan
	Submitted comprehensive confirmation of patents applied to the revised parts of ARIB STD-B24 Ver4.4 *15		
Motorola Japan Ltd.	Submitted comprehensive confirmation of patents applied to the revised parts of ARIB STD-B24 Ver3.6 *4		
	Submitted comprehensive confirmation of patents applied to the revised parts of ARIB STD-B24 Ver3.8 *5		
	Submitted comprehensive confirmation of patents applied to the revised parts of ARIB STD-B24 Ver3.9 *6		
	Submitted comprehensive confirmation of patents applied to the revised parts of ARIB STD-B24 Ver4.0 *7		
	Submitted comprehensive confirmation of patents applied to the revised parts of ARIB STD-B24 Ver4.1 *9		
NTT DoCoMo, Inc. *11	動画像符号化方法、動画像複合方法、動画像符号化装置、及び動画像複合装置*11	特許第 3504256号	Japan, EPC, USA, Korea, China, Taiwan
	動画像符号化方法、動画像複合方法、動画像符号化装置、動画像複合装置、動画像符号化プログラム、及び動画像複合プログラム*11	特許第 3513148号	Japan, EPC, USA, Korea, China, Taiwan
	動画像複合方法、動画像複合装置、及び動画像複合プログラム*11	特許第 3534742号	Japan, EPC, USA, Korea, China, Taiwan
	信号符号化方法、信号複合方法、信号符号化装置、信号複合装置、信号符号化プログラム、及び、信号複合プログラム*11	特許第 3491001号	Japan, EPC, USA, Korea, China, Taiwan
	インターリーブを行うための方法および装置並びにデ・インターリーブを行うための方法および装置*13	特許第 3362051号	Japan, USA, Korea, Singapore, Australia, China

Patent applicant	Name of invention	Patent number	Remarks
	誤り保護方法および誤り保護装置*13	特許第 3457335号	Japan, USA, UK Korea, Germany, France Italy, Singapore, Australia, China
	Submitted comprehensive confirmation of patents applied to the revised parts of ARIB STD-B24 Ver4.4 *15		
Sharp Corporation *5	画像符号化装置および画像復号装置	特許第 2951861号	Japan
NEC Corporation *5	画像信号の動き補償フレーム間予測符号化・複合化方法とその装置	特許第 1890887号	Japan
	圧縮記録画像の再生方式	特許第 2119938号	Japan
	圧縮記録画像の対話型再生方式	特許第 2134585号	Japan
	適応変換符号化の方法及び装置	特許第 2778128号	Japan
	符号化方式および復号方式	特許第 2820096号	Japan
	変換符号化複合化方法及び装置	特許第 3070057号	Japan
	改良DCTの順変換計算装置および逆変換計算装置	特許第 3185214号	Japan
	適応変換符号化方式および適応変換複合方式	特許第 3255022号	Japan
Philips Japan, Ltd	Submitted comprehensive confirmation of patents applied to the revised parts of ARIB STD-B24 Ver4.0 *8		
	Submitted comprehensive confirmation of patents applied to the revised parts of ARIB STD-B24 Ver4.1 *10		
	Submitted comprehensive confirmation of patents applied to the revised parts of ARIB STD-B24 Ver4.2 *12		
Philips Electronics Japan, Ltd.	Submitted comprehensive confirmation of patents applied to the revised parts of ARIB STD-B24 Ver4.3 *14		

Note) *1: valid for the revised parts of ARIB STD-B24 Ver3.0
*2: valid for the revised parts of ARIB STD-B24 Ver3.1
*3: valid for the revised parts of ARIB STD-B24 Ver3.3
*4: valid for the revised parts of ARIB STD-B24 Ver3.6
*5: valid for the revised parts of ARIB STD-B24 Ver3.8
*6: valid for the revised parts of ARIB STD-B24 Ver3.9（accepted on October 9,2003)
*7: valid for the revised parts of ARIB STD-B24 Ver4.0（accepted on January 8,2004)
*8: valid for the revised parts of ARIB STD-B24 Ver4.0（accepted on January 29,2004)
*9: valid for the revised parts of ARIB STD-B24 Ver4.1（accepted on November 17,2004)
*10: valid for the revised parts of ARIB STD-B24 Ver4.1（accepted on December 7,2004)

*11: valid for the revised parts of ARIB STD-B24 Ver3.8（accepted on January 7,2005)
*12: valid for the revised parts of ARIB STD-B24 Ver4.2（accepted on March 14,2005)
*13: valid for the ARIB STD-B24 Ver1.0（accepted on September 26,2005)
*14: valid for the revised parts of ARIB STD-B24 Ver4.3（accepted on September 27,2005)
*15: valid for the revised parts of ARIB STD-B24 Ver4.4（accepted on March 6,2006)
*16: valid for the revised parts of ARIB STD-B24 Ver3.6（accepted on March 14,2006)

APPENDIX 1

Operational Guidelines

Contents

Chapter APPENDIX 1-1 Purpose 309

Chapter APPENDIX 1-2 Scope 310

Chapter APPENDIX 1-3 Guidelines on Coding of B-XML Documents 311
A1-3.1 XML Application Language 311
A1-3.2 Validity of XML Documents 311
A1-3.3 Document Type Definition (DTD) 311

Chapter APPENDIX 1-4 Guidelines on coding of BML documents 312
A1-4.1 Guidelines on Character Code 312
A1-4.1.1 Character coding schemes used for BML documents 312
A1-4.2 Scope Model 312
A1-4.2.1 Structure of multimedia contents 312
A1-4.2.2 Definition of scope 313
A1-4.2.3 Setting up of scope 314
A1-4.2.4 Scope of object 314
A1-4.3 Designation of id Value 314
A1-4.4 CSS-based Style Sheet 315
A1-4.4.1 Colour unit 315
A1-4.4.2 Designation of coordinates 315
A1-4.4.3 Dynamic pseudo-classes 315
A1-4.5 Inheritance Values for Style Sheet 315

Chapter APPENDIX 1-5 Guidelines on Behaviour of BML Browser 316
A1-5.1 Reference Browser Model for BML Documents 316
A1-5.1.1 Structure of reference browser 316
A1-5.1.2 Memory Control Model 316
A1-5.1.3 Flow of receiving process 317
A1-5.1.4 Interruption event model 318
A1-5.2 Structure Model for BML Documents 319
A1-5.2.1 About implementing DOM 319
A1-5.2.2 Root name for XML document 320
A1-5.2.3 Example usage of DOM 320
A1-5.3 Behaviour of BML Elements 321
A1-5.3.1 Behaviour of input element and textarea element 321
A1-5.4 Focus Control 322
A1-5.4.1 Cause of focus control 322

A1-5.4.2 Behaviour model 322
A1-5.4.3 Initial position of focus 326
A1-5.4.4 Relationship between focus control and interruption events 326

Chapter APPENDIX 1-6 Guidelines on Presenting BML Documents 328
A1-6.1 CSS 328
A1-6.1.1 Background 328
A1-6.1.2 Action for designation of colour depth 328
A1-6.1.3 Presentation for border property of inline element 328
A1-6.2 Guideline for normalStyle, focusStyle and activeStyle 329
A1-6.2.1 Inherited values of normalStyle, focusStyle and activeStyle 329
A1-6.2.2 Application, cascading, and inheritance of property values 329
A1-6.2.3 Example 329
A1-6.3 Composition of text 331
A1-6.3.1 Baseline 331
A1-6.3.2 Letter spacing 331
A1-6.3.3 Border 332
A1-6.3.4 Fold back 332
A1-6.3.5 Br element 336
A1-6.3.6 Relationship between br element and left border 336
A1-6.3.7 Relationship between position of the br element and fold back 337
A1-6.3.8 Height of line 338
A1-6.3.9 Alignment 341
A1-6.3.10 Other remarks 342
A1-6.4 Presentation of BML element 342
A1-6.4.1 Presentation of input element and textarea element 342
A1-6.4.2 Displaying a elements 342
A1-6.5 Scaling Visual Representations 342
A1-6.6 Update of modules 346
A1-6.6.1 Reference of same module by multiple objects 346
A1-6.6.2 Relationship between update and module lock 347

Chapter APPENDIX 1-7 Guidelines on XSLT 349
A1-7.1 Style Sheet for XML Documents 349
A1-7.2 Structure of XSL Document 349

Chapter APPENDIX 1-8 Guidelines on Procedural Description 350
A1-8.1 Guidelines on Delayed Call Origination 350
A1-8.1.1 Function executed on call origination 350
A1-8.1.2 Handling text strings used for datagram 350
A1-8.1.3 Executable procedures when performing delayed call origination 350
A1-8.2 Guidelines on Extended Functions for Storing 350
A1-8.2.1 Relationship between source and destination 350
A1-8.2.2 Extended APIs for Storing 352
A1-8.2.3 Sample behaviours of file/directory management functions 354

A1-8.2.4 Sample behaviours for copying information in data carousel to another location 356
A1-8.2.5 Sample behaviours for copying file on HTTP/FTP server to another location 358
A1-8.3 Guidelines on Extended Functions for Reserving Records or Storing Information 359
A1-8.4 Guidelines on Bookmarks 359
A1-8.4.1 Identification of Bookmark Area 359
A1-8.4.2 NVRAM operations in bookmark services 359
A1-8.4.3 Operational Guidelines on Bookmark Service 374
A1-8.5 Guidelines on Other Extended Functions 379
A1-8.5.1 Operation of operational control functions 379
A1-8.5.2 Launch of external applications 380
A1-8.5.3 Receiver resident application 382

Chapter APPENDIX 1-9 Guidelines on Contents Transmission and Namespace 384
A1-9.1 List of Resources in Multipart Format Modules 384
A1-9.2 Guidelines on Resources Obtained via Transmission Channel using IP Packet 384
A1-9.3 Namespace Available to BML documents 384
A1-9.3.1 Namespace available to attributes of elements or CSS properties specified in BML document 385
A1-9.3.2 Extended Objects for Broadcasting and Extended Functions for Broadcasting for URI Specification 386
A1-9.4 Guidelines on Root Certificate 400
A1-9.4.1 Format of module containing root certificate 400
A1-9.4.2 Generic root certificate storage area 401
A1-9.4.3 Generic root certificate contained in receiver 402
A1-9.4.4 Guidelines on receiver's behaviour for updating generic root certificate 403

Chapter APPENDIX 1-10 Guidelines on Security of Content 405
A1-10.1 Guidelines on Security Classes 405
A1-10.2 Guidelines on Referred Content 405

Chapter APPENDIX 1-11 Guidelines on Functions Specific to Broadcasters and Others 406
A1-11.1 Guidelines on Ensured Uniqueness of Identified Extended Function 406
A1-11.1.1 Guidelines on ensured uniqueness of identified extended function 406
A1-11.1.2 Guidelines on function beyond operational standard regulation specific to media type 407
A1-11.2 Guidelines on Uniqueness of Namespace for Pre-downloaded Data Services 414

Chapter APPENDIX 1-12 Guidelines on Functions for printing 415
A1-12.1 Reference model for functions for printing 415
A1-12.1.1 Transmitting file for printing obtained from broadcast stream to printer 415
A1-12.1.2 Transmitting no file for printing from receiver to printer 416
A1-12.1.3 Capturing and transmitting image 416
A1-12.2 Operational Sequence Model 416
A1-12.3 Operation of Functions for printing and Sample Behaviour 418
A1-12.3.1 Operation of getPrinterStatus() and Sample Behaviour 418

A1-12.3.2 Operation of printFile() and Sample Behaviour 419
A1-12.3.3 Operation of printTemplate() and Sample Behaviour 419
A1-12.3.4 Operation of printUri() and Sample Behaviour 421
A1-12.3.5 Operation of printStaticScreen() and Sample Behaviour 422
A1-12.3.6 Operation of saveImageToMemoryCard(), saveStaticScreenToMemoryCard(), and saveHttpServerImageToMemoryCard() 422
A1-12.4 Sample Behaviours of Functions for Printing 422

Chapter APPENDIX 1-13 Operation relating to Coding of XHTML document for Printing 424
A1-13.1 Coding of Document for Printing 424
A1-13.2 Character Coding of Printing Document 424
A1-13.2.1 EUC-JP 424
A1-13.2.2 Shift_JIS 424
A1-13.2.3 Additional Signs 425
A1-13.2.4 External characters 425
A1-13.2.5 Operated Character Reference 425
A1-13.3 Operated Media Types and Monomedia 426
A1-13.4 Bitmap 426
A1-13.4.1 PNG 426
A1-13.4.2 JPEG 426
A1-13.5 XHTML Document for Printing 426
A1-13.5.1 XML declaration 426
A1-13.5.2 Document type declaration 427
A1-13.5.3 Namespace for XHTML document to be printed 427
A1-13.6 Elements for XHTML Document to Be Printed 427
A1-13.6.1 Entities 427
A1-13.6.2 Operational guidelines on elements 427
A1-13.6.3 Operational guidelines on attributes 428
A1-13.7 Operated CSS-based Style Sheets 428
A1-13.7.1 Media type 428
A1-13.7.2 Box model 428
A1-13.7.3 Visual formatting model 430
A1-13.7.4 Visual effects 431
A1-13.7.5 Generated content, automatic numbering, lists 432
A1-13.7.6 Colours and Backgrounds 432
A1-13.7.7 Fonts 433
A1-13.7.8 Text 434
A1-13.7.9 Tables 435
A1-13.7.10 Paged media 435
A1-13.7.11 Page formatting 436
A1-13.7.12 Guidelines on page layout 436
A1-13.7.13 Operational guidelines on page controlling 437
A1-13.7.14 Default style sheet 437
A1-13.7.15 Operational guidelines on length 438

A1-13.7.16 Considerations for preparing XHTML document to be printed 438
A1-13.8 Namespace for Content to Be Printed 438
A1-13.8.1 Obtaining XHTML document to be printed through HTTP schema 438
A1-13.8.2 Obtaining XHTML document to be printed through ARIB-DC schema 438
A1-13.9 Japanese Hyphenation 439

Annex A Default Style Sheet Applied to XHTML Document to Be Printed 441

Informative Explanation 444
1 Using Box Model 444
2 Example Behaviour for the Properties which describe Remote control Operations 444
3 Guidelines on Time-specified Call Functions using BASIC-based Protocols 444
4 Sample Behaviours of Interaction channel Functions for IP Connection 445
5 Bookmark Services 459
6 Sample Usage of Function for Communicating Information with External Device 464

[BLANK]

Chapter APPENDIX 1-1 Purpose

The Appendix 1 contains the guidelines developed to explicitly describe things to be taken in account to successfully operate and utilize the XML-based multimedia coding scheme specification responsible for the data broadcasting scheme, part of the digital data broadcasting scheme specified as the standard in Japan.

Chapter APPENDIX 1-2 Scope

The guidelines contained in this appendix are applied to a multimedia service that uses XML-based multimedia coding for data broadcasting carried out as part of digital data broadcasting.

Chapter APPENDIX 1-3 Guidelines on Coding of B-XML Documents

A1-3.1 XML Application Language

The XML is used for encoding various types of data used in applications. An XML application language is defined for each application by setting up a document type definition (DTD) for XML. The data for each application is encoded by using the defined XML application language. The B-XML does not specify an XML application language for any specific application. An XML application language must be specified for each application by authors of XML documents (content) and developers responsible for processing the XML documents.

A1-3.2 Validity of XML Documents

An XML document can be any well-formed or valid document. A Well-formed document does not require a document type declaration, but a valid document must have a document type declaration. A valid document must undergo the verification of its validity upon being sent out.

A1-3.3 Document Type Definition (DTD)

Any valid document requires a document type definition (DTD). For the DTD, an external subset and/or an internal subset may be used. However, default attribute values must not be defined in an external subset to prevent different receivers from producing different results of processing because some XML processors that are mounted to receivers verify the validity and others do not perform the validation. Default attribute values must be defined in an internal subset or by an application, which uses a DOM API.

Chapter APPENDIX 1-4 Guidelines on coding of BML documents

A1-4.1 Guidelines on Character Code

A1-4.1.1 Character coding schemes used for BML documents

It is recommended that two or more code systems must not exist together throughout a data broadcasting content.

The BML document uses one of the following character encoding schemes:

- EUC-JP defined as a subset of eight-bit character code system
- Shift-JIS
- JIS X 0221

Note that not more than one scheme must be used in any single document and any external ECMAScript files and CSS files handled by the document.

A1-4.2 Scope Model

The following lists guidelines on the scope with which the effective range of various settings and behaviours for the multimedia contents using a BML document is determined.

A1-4.2.1 Structure of multimedia contents

The following lists the units making up multimedia contents:

1) BML document: A unit making up one screen.

2) Document group: This unit consists of one or more BML documents. It is desirable that one document group must always include one start-up document.

3) Content: This unit consists of one or more document groups. It is desirable that a content must always include one start-up document group to be launched at first.

4) Content group: This unit consists of one or more contents.

Figure 4-1 shows the relationships among the elements of the multimedia content explained above:

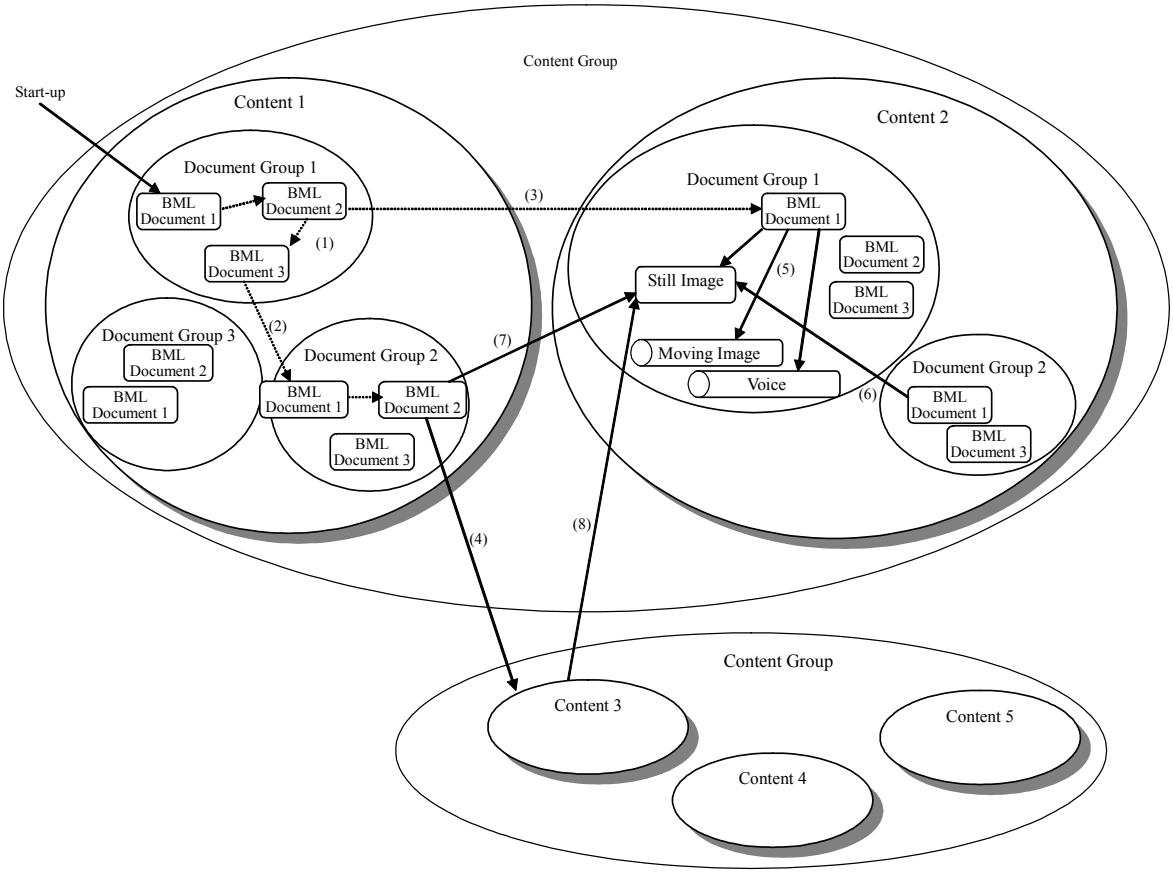


Figure 4-1 Structure Model of Multimedia Contents by BML

The following document switchover operations occur when presenting multimedia contents. The item numbers in the following list correspond to those in Figure 4-1.

(1) Document switchover within a document group.

(2) Document switchover between document groups.

(3) Document switchover between contents.

(4) Document switchover between content group.

The same data is shared with more than one document in the following reference relationships:

(5) Reference within a document group.

(6) Reference between document groups.

(7) Reference between contents.

(8) Reference between content group.

A1-4.2.2 Definition of scope

Each element listed above has its corresponding scope as follows. Table 4-1 defines the scope number for each scope and introduces virtual identifiers corresponding to each scope:

Table 4-1 Scope number and Scope identifier

Scope number	Scope	Identifier
1	Within a BML document	Document scope identifier
2	Within a document group	Document group scope identifier
3	Within a content	Content scope identifier
4	Within a content group	Content group scope identifier

These identifiers are constant within each scope. If an identifier changes, it means that the multimedia content shifts out of the scope indicated by the changed identifier. However, if an identifier for a wider scope (a scope with a larger scope number) is changed, it means that the multimedia content falls outside the corresponding scope.

A1-4.2.3 Setting up of scope

A document, document group, content, and content group in the structure model of multimedia contents must be mapped to identifiers and namespaces in transmission systems. It is also necessary to define in operation how a document scope identifier, document group scope identifier, content scope identifier, and content group scope identifier are mapped to identifiers in transmission systems.

A1-4.2.4 Scope of object

Objects used in a normal BML document are valid within the document. However, it is desirable that sharing the data listed in the following table with different documents must be defined in operation:

Table 4-2 Objects within a BML document

Type of Object	How to Maintain	How to Access
Mono-media data	Stored in content memory	object element
Table data	Stored in content memory	CSVTable( ), BinaryTable( )
External character data	Stored in content memory	loadDRCS()
Permanent storage data	Stored in memory such as NVRAM	readPersistent* ( ), writePersistent* ( )
Delayed call originating data	Stored in memory such as NVRAM	Registered functions
Stream	remain attribute in object element	object element
CLUT data	Stored in content memory	clut property

Information that is stored in a content memory, as listed above, by using lockModuleOnMemory( ) is shared by documents of a document group. Information contained in a content memory by using lockModuleOnMemoryEx() is shared by documents both within and outside of a document group.

A1-4.3 Designation of id Value

This section lists characters that can be used for the id value of an element. They are limited to the range of characters that can be used for variables in ECMAScript because, in the future, designation of the id value directly by scripts may be put into practice.

"a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z",
"A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z",

Note that following characters can be used for the second or later characters:

"0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "_"

A1-4.4 CSS-based Style Sheet

A1-4.4.1 Colour unit

- Colour must be described in six-digit representation when specified in the form of #rgb.
- As the depth of display colours in the form of rgb(0.0%-100.0%,.....) depends on the implementation of a receiver, it is desirable that its usage range must be predefined separately in operation.
- When specifying index colour, if a CLUT area does not exist at a position of an element or the index number is invalid, drawing dose not occur.

A1-4.4.2 Designation of coordinates

For designation of coordinates values and size in px as a unit, coordinates values defined by coordinates of text graphics plane are used.

A1-4.4.3 Dynamic pseudo-classes

Dynamic pseudo-classes are applied to the following cases:

- :focus class

  when a focus is coming over with an operation of a remote control or other triggers.

- :active class

  when the Enter key or access key on a remote control is pressed.

A1-4.5 Inheritance Values for Style Sheet

When "inherit" is specified as a value for a style sheet, a computed value based on the parent is used unless the concerned element must inherit a value other than computed values, as defined in CSS2. For example, line-height must inherit a value other than a computed value.

Chapter APPENDIX 1-5 Guidelines on Behaviour of BML Browser

A1-5.1 Reference Browser Model for BML Documents

This section describes the guidelines on each function based on the reference browser model of BML documents.

A1-5.1.1 Structure of reference browser

Figure 5-1 shows a reference browser model. The browser consists of a document process system and script process system. The document process system generates DOM objects based on the results of parsing a document, and manages the CSS2 properties that control the display style. The script process system executes scripts that are part of a BML document.

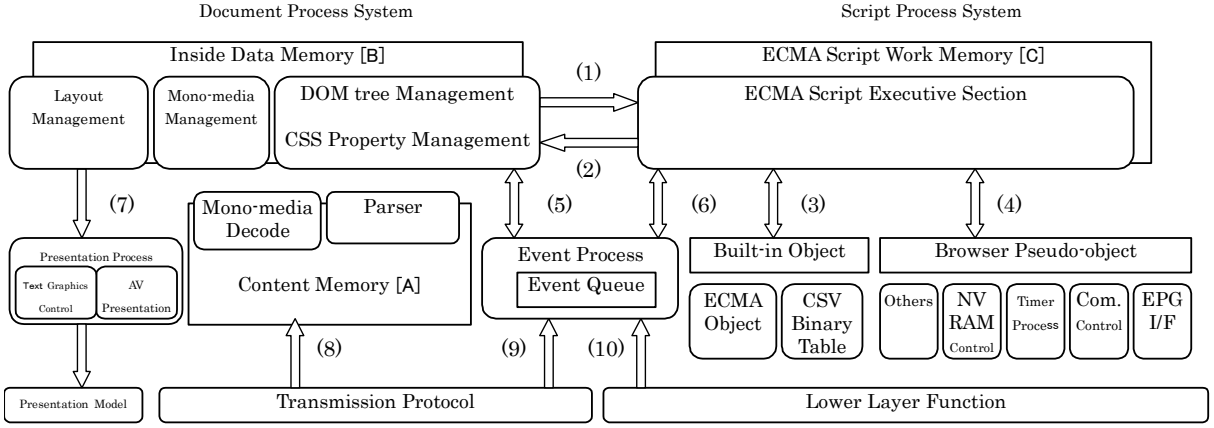


Figure 5-1 BML Reference Browser Model

The following explains the meaning of symbols (1) through (10) shown in Figure 5-1:

(1) Calls a script from a BML document.

(2) Calls a DOM-API from a script.

(3) Calls a built-in object.

(4) Calls a browser pseudo-object.

(5) Processes an interruption event for a BML document.

(6) Processes an interruption event for a script.

(7) Presentation process.

(8) Receives contents.

(9) An external interruption event from transmission system.

(10) An external interruption event generated in a receiver.

A1-5.1.2 Memory Control Model

The reference browser model manages these three types of memory:

[A] Content memory

The module in which a content is stored is held.

[B] Internal data memory

The following inside data which are required for presentation of a BML document are held.

- DOM tree and CSS properties made from the module by parser.
- Mono-media data (decoded data or coded data)
- Other data such as the layout data made by the form process.

[C] ECMAScript work memory

ECMAScript source code or compiled data which is required for execution of ECMAScript are held.

To ensure conformance, the contents of item [B] and item [C] given above must be defined separately in operation. Similarly, the contents of item [A] must be defined in detail in operation. At that time, since item [A] requires guidelines on actions to control contents and map transmission system, the following explains the actions.

The content memory is the memory area which stores the following modules received from carousels.

- The module that was received as a BML document to be presented newly.
- The module that is specified to fix to the memory by lockModuleOnMemory() and lockModuleOnMemoryEx(). To share the module is the purpose to fix it.
- The cache module that was read ahead based on the priority set up by setCachePriority().

The following lists the guidelines on controlling the content memory:

- The available capacity of the whole content memory should be managed in any time.
- If a guideline on cache priority within the service currently being broadcast is indicated by setCachePriority( ) in a document being executed, it is desirable to perform the read-ahead caching by receiving and caching modules sequentially in a descending order of priority concurrently with presenting operation.
- At the time of the lockModuleOnMemory() API or the lockModuleOnMemoryEx()API execution, fixing a target module to a content module must take priority over a caching process.

A1-5.1.3 Flow of receiving process

The following outlines how this reference browser acts when it receives and presents a BML document:

[1] A module is taken out by putting data carousels on the sieve, and stored in the content memory.

[2] DOM tree and CSS properties are made in the internal data memory by parsing process.

[3] The ECMAScript code which is described in the script element of the BML document is compiled, and object data are made in the ECMAScript work memory.

[4] Monomedia data (PNG/MNG etc.) which are referred by the BML document are decoded and stored in the internal data memory as much as possible.

[5] Layout data are made from form process using DOM tree and CSS properties stored in the internal data memory.

[6] The focus is processed to “nav-index:0”, the focus event is queued in the synchronous queue, and the global codes of ECMAScript that were compiled at [3] are executed. The synchronous events that are occurred in the execution of global codes are added to the synchronous queue behind the focus event of “nav-index:0”. Then the interruption events in the synchronous event queue are processed. Note that when the focus is processed without involving nva-

index[A.T.1], the initial position of the focus depends on the implementation of a receiver. A focus event related to the initial position is queued in the synchronous queue. Any synchronous event that occurred during the execution of global codes of ECMAScript is appended to the focus event that is related to the initial position. Then, the synchronous queue is processed.

[7] The BML document is presented based on the layout data on the screen.

[8] A load event occurs, and an event handler which corresponds to the onload attribute of the body element is executed.

[9] It goes into waiting state of the interruption event stored in the interruption event queue.

The event handler mentioned in above description means a user-defined function that is triggered by an interruption event and the global code means any script outside of the function declarations.

Next, the behaviour when a transition to another BML document occurs by selecting an anchor or executing launchDocument() and other functions is shown below.

[1] Starts the acquisition of the module which includes the BML document of the transition target.

[2] During the module acquisition, all interruption events are destroyed, and only an unload event is occurred. An event handler which corresponds to the onunload attribute of the body element is executed. Further the unload interruption event occurs not only at the transition of the documents but also at the switch of service when epgTune() , epgTuneToComponent(), or epgTuneToDocument() is executed or broadcasting station to be viewed is selected at the receiver.

[3] After the storage of the module to the content memory is completed, the data stored in the internal data memory are destroyed.

[4] The receiving processes previously mentioned are carried out from the process [2].

A1-5.1.4 Interruption event model

The BML document browser is driven by interruption event. For this reason, each interruption event is corresponded with a respective event handler that is a user-defined function.

Figure 5-2 shows an interruption event driven model of the BML document browser.

It operates on two queues: one is a synchronous event queue and the other is an asynchronous event queue. Only focus, blur, click, change, and submit events enter the synchronous queue and other events enter the asynchronous queue. Further, when the input element or[A.T.2] the textarea element loses focus, and the value attribute of the said element has been changed, the change event occurs before the blur event. The submit event is queued in the synchronous event queue, following the click event.

- Asynchronous event queue

  The interruption events enter an asynchronous event queue of a FIFO type in the order they are generated. If the interruption event that enters the queue has the corresponding event handler, it triggers the event handler. Taking out the next interruption event from the event queue is postponed until the current running event handler completes. This means that the script is not reentrant. On the other hand, external interruption events can enter the event queue asynchronously to their event handler. Further, if the synchronous event queue is not empty, the control is transferred to the synchronous event queue.

- Synchronous event queue

  A keydown event, for example, is retrieved from the asynchronous event queue. Then, the interruption events from the synchronous event queue starts to be retrieved immediately after the onkeydown script is processed. If a script that is attached to the onkeydown event does not exist,

control is transferred to the synchronous event queue immediately after the keydown event is retrieved. The focus events, the blur events, the click events, the change events, and the submit events that are generated by the scripts launched by the synchronous event queue enter the synchronous event queue one after another. The interrupt events which have entered the synchronous event queue are processed in the order of their arrivals. When the synchronous event queue becomes empty, control is transferred to the asynchronous event queue.

Based on the interruption event model described above, the scripts called by an intrinsic event or a bevent element operate synchronously. When an interruption event is generated, it enters the event queue temporarily. If the corresponding ECMAScript function (event handler) does not exist, the interruption event is destroyed. If the function exists, it is called and has been completed before the next interruption event is addressed.

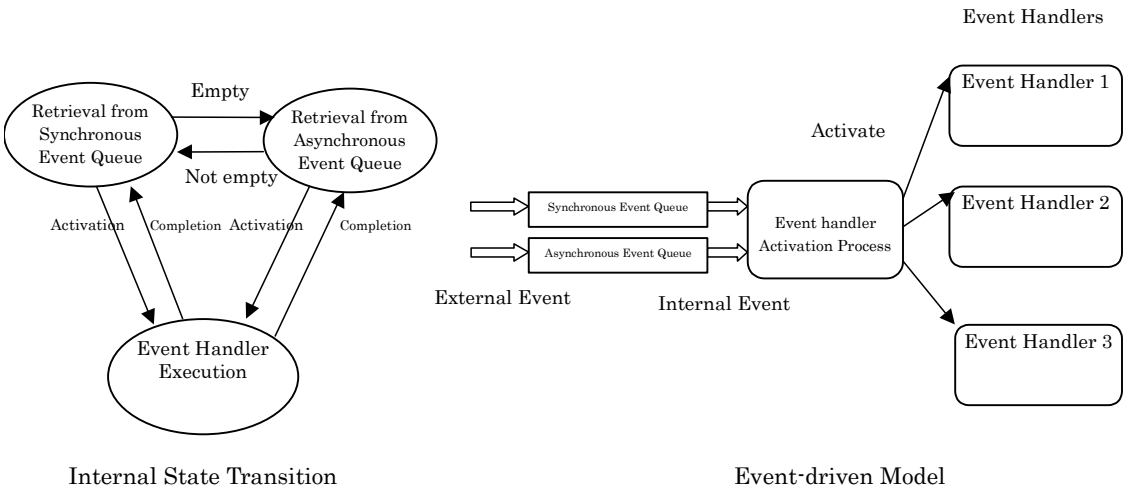


Figure 5-2 Interruption Event Model

Calling DOM from an ECMAScript must be of a synchronous type. If the DOM attribute is set up from the ECMAScript, the script proceeds to the next process when all the processes to be done caused by the attribute setup completes. If a method in DOM is called, control proceeds to execution of the next script when the method completes.

A1-5.2 Structure Model for BML Documents

A1-5.2.1 About implementing DOM

The definition of the DOM standard consists of the next two parts: the Core DOM and HTML DOM. Further, the Core DOM is divided into the following two parts: fundamental interfaces and extended interfaces.

The fundamental interfaces of the Core DOM represent basic functions for an XML document, and work as the base of HTML DOM. It is desirable that DOM must be implemented by implementing all of the fundamental Core interfaces before implementing the HTML section and the extended Core interfaces.

As DOM is not case sensitive, however, it must be case sensitive for the use of BML, in order to be consistent with XML. And, the style description is not case sensitive according to CSS2 specifications, however, if the property value is read using DOM, lower case letters must be returned.

In addition, if the className attribute of an element is picked up by scripts using DOM and the class attribute is not specified, then an initial value is returned if it exists, otherwise the return value is indefinite for the case that the attribute which does not exist is accessed.

A1-5.2.2 Root name for XML document

To access an XML document through DOM, an access to the Document object which is the root of the tree structure of the whole document is required. The name representing the root Document object must be fixed to "document".

Figure 5-3 shows a tree structure generated from loading and parsing an XML document.

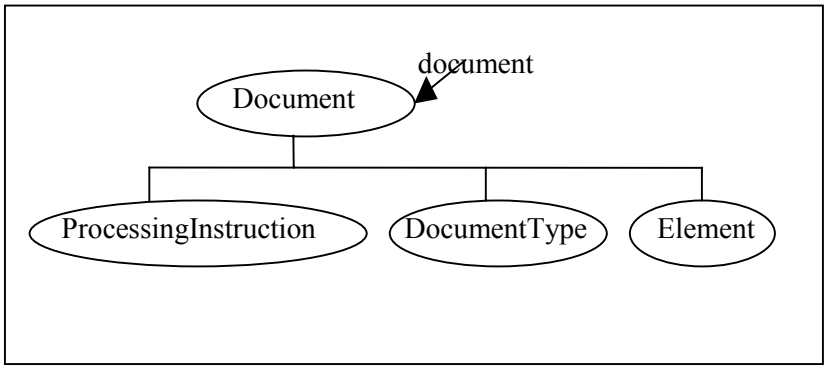
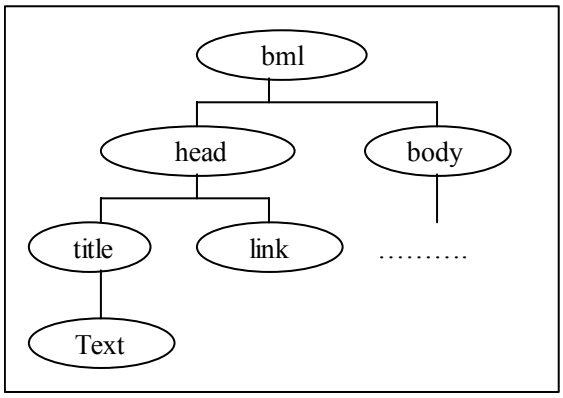


Figure 5-3 Tree Structure of an XML Document

A1-5.2.3 Example usage of DOM

This section shows examples of the Core DOM use. Figure 5-4 indicates an example of a BML document that is equivalent to the Element node in Figure 5-3, and its tree structure.




```
<bml>
<head>
 <title>A study of DOM</title>
 <link  id=”ID1” rel="stylesheet"
         type="text/css">
</head>
<body>
  ... document body...
</body>
</bml>
```

Figure 5-4 Example of a BML Document Tree Structure

The following two examples describe how to take out the text that is enclosed by the <title> tag from the DOM tree structure of the document in Figure 5-4 and to assign it to variable T1. Note that it assumes a DOM implementation to be already created on memory.

[Example 1] Example of describing a Core DOM

// The following document.documentElement indicates the object of the BML root element.

// By walking the tree structure, this example references the value of the child of the node

//<title> that is the first child of the node (<head>) that is the first child of the object.

```
var testNode = document.documentElement;
var T1 = testNode.firstChild.firstChild.firstChild.nodeValue;
```

[Example 2] Example of describing a Core DOM

```
// The following searches for elements with the specified tag name "title", retrieves
// a <title> node with the requested order number from a NodeList derived from the
// search, and references the value of the child of the node.
// * Since there is only one <title> tag, this example does not specify the order number.
var testNodeList = document.getElementsByTagName("title");
var i;
for(i = 0;i < testNodeList.length;i++)
    var T1 = testNodeList.item(i). firstChild.nodeValue;
```

[Example 3] Example of describing an HTML DOM

```
//Assuming that the id value is uniquely specified within a document,
//the following searches for tags with the id value.
//It finds the <link> tag with the id value, ”ID1,” and references the value of the child of the node.
var linkNode = document.getElementById("ID1");
var L1 = linkNode.firstChild.nodeValue;
```

A1-5.3 Behaviour of BML Elements

A1-5.3.1 Behaviour of input element and textarea element

A1-5.3.1.1 maxlength attribute

If the input data exceeds the maxlength, the exceeded portion is cut off. If it exceeds the frame, the exceeded portion is not displayed. The data up to the number of the maxlength value can be obtained.

For example, if maxlength=”40” and the length of the display frame (width of the input element) is smaller than 40 characters, its presentation depends on an implementation. The width of the display frame should be larger than what is expected based on maxlength numbers of characters.

A1-5.3.1.2 value attribute

If the value attribute is specified, the input data is added to the last character.

For example, if value=”abc” is set, “abc” is presented in the input area as an initial state. When a user puts “1”, the “1” is presented after “abc”. That is, “abc1” is presented.

A1-5.3.1.3 Cursor in input element and textarea element

Even if the focus moves to the input element, the focus movement (nav-up, nav-down, nav-left, and nav-right) is applied to the control with an arrow key of a remote control.

It depends on receivers whether the decision is made when the Enter key is pressed.

The focus on input is not released automatically with the Enter key. If the input must be terminated, the focus must be moved implicitly by describing focus() or blur() in the script specified by the

onclick event (if the content has been changed, the script for the onchange event is executed when the focus disappears.). And, the onchange event occurs before the onblur event. The onchange event occurs Under an operational environment that does not specify the focus movement, how a focus is moved may depend on the implementation of a receiver, in spite of the aforementioned specification.

In addition, if readOnly = true, the focus can be moved. However, if disabled = true, the focus is not allowed to be moved. If the disabled attribute for the input element which has been focused is changed into disabled = true, the focus disappears. The process when disabled = true takes precedence over that when readOnly=true.

A1-5.3.1.4 line-height property of input element

The line-height property of the input element specifies the height of the input frame. It is desirable that the value of the height property and that of the line-height property are same. However, if different values are specified for them, the behaviour depends on an implementation.

A1-5.4 Focus Control

The sections 5.4.1 to 5.4.4 describe how a focus must be controlled. Note that in spite of the following descriptions, under a certain operational environment, the properties for navigation (nav-up, nav-down, nav-left, and nav-right) and applications of styles may depend on the implementation of a receiver.

A1-5.4.1 Cause of focus control

The following methods control the movement or disappearance of a focus within BML documents:

- Access key
- “nav-index”
- “focus( )” method and “blur( )”
- The value of the visibility property of the focused element is changed from "visible" to "hidden".
- The disabled attribute of the focused input element is set up to be true.

An access key is prioritized when an access key and nav-index are specified at the same time.

The number of elements on which focus is set is one at most. And, if the focus( ) method is executed on an element while another element has been focused, the focus is moved. However, any focus( ) cannot be used in the event handler activated by onfocus. This means that reentrance of scripts is not allowed. In the same way, any blur( ) cannot be used in the event handler activated by onblur.

A1-5.4.2 Behaviour model

In order to explain the behaviour model in this clause, the following three process units are introduced:

- The following process is “focus transition(X,Y)”. However, if the X element and the Y element are same, the process described below is not executed.
  1. The style designated by :focus pseudo-class of X element is released.
  2. The style designated by :focus pseudo- class of Y element is applied.
  3. The interrupt event blur of X element is occurred.
  4. The interrupt event focus of Y element is occurred.
- The following process is “focus generation(X)”.

1. The style designated by :focus pseudo-class of X element is applied.
2. The interrupt event focus of X element is occurred.

Note: After the style designated by :focus is applied, effect of the focusStyle attribute which is changed by an interruption event handler launched by the onfocus function is reflected on the screen.

- The following process is “focus disappearance(X)”.

1. The style designated by :focus pseudo-class of X element is released.
2. The blur event of X element occurs.

This process is occurred when blur( ) is executed or the visibility property becomes to hidden.

A1-5.4.2.1 Moving of focus

Figure 5-5 shows how a focus acts when it moves from the current focused element A to element B by pressing one of the arrow keys.

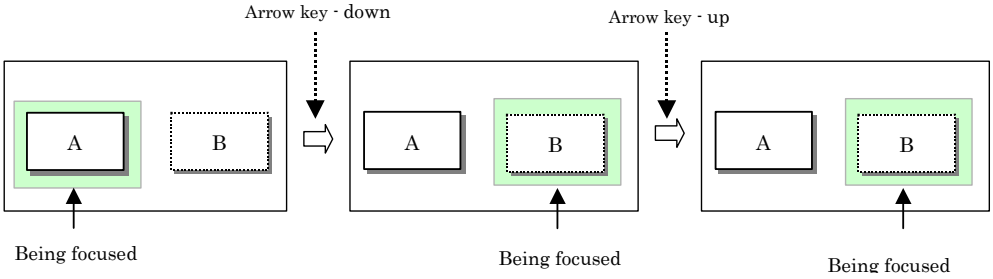


Figure 5-5 Example of the Focus Movement.

The behaviour of the example in Figure 5-5 is described below.

[1] Pressing the key causes a keydown event for element A and the script specified in the onkeydown is executed.

[2] The element which focus exists in is made an A' element temporarily as a result of [1].

- It is A'=A so far as focus isn't moved during the process of [1].
- Focus transition (A, A') is carried out when focus is moved during the process of [1].

[3] When an A' element does not exist, the following processes are not carried out.

[4] A navigation related nav-* properties is applied to the A' element. Consider the element that focus is applicable newly to be a B element.

[5] A focus transition (A', B) is carried out.

[6] Releasing the key causes a keyup event to be occurred on the element B and the script specified in the onkeyup is executed.

However,

- Any interruption events (interruption events related to Intrinsic Event module, those of broadcasting services, and those caused by setTimeout() and setInterval()) which have occurred during the process [1] through [5] are not processed until the process [1] through [5] is completed.

- The action in item [6] is delayed until all processes for the interruption events occurred during the process [1] through [5] are completed.

A1-5.4.2.2 Behaviour when pressing Enter key

Figure 5-6 shows how a focus acts when the Enter key is pressed, while a focus is on the element A.

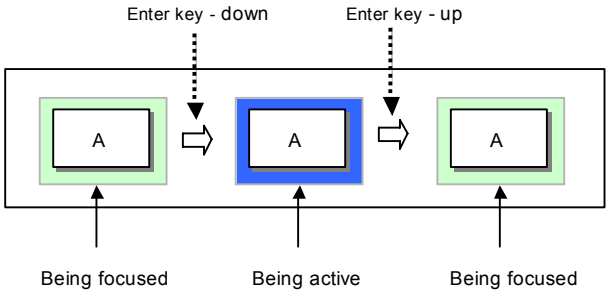


Figure 5-6 A Movement when Enter Key is pushed down under the Focused Condition

[1] Pressing the Enter key causes a keydown event for element A and the script specified by the onkeydown is executed.

[2] The element which focus exists in is made an A' element temporarily as a result of [1].

- It is A'=A so far as focus is not moved during the processes of [1].
- Focus transition (A, A') is carried out when focus is moved during the processes of [1].

[3] When an A' element does not exist, the following processes are not carried out.

[4] The style specified in the :active pseudo-class for element A’ is applied.

[5] An click event for element A’ is queued in a synchronous event queue.

[6] Releasing the key causes a keyup event on element A’ and the script specified by the onkeyup is executed.

[7] The style specified in the :active pseudo-class for element A’ is released.

However,

- Any interruption events occurred during the processes [1] through [5] (interruption events related to Intrinsic Event module, those of broadcasting services, and those caused by setTimeout() and setInterval()) are not processed until the process [1] through [5] is completed.
- The action in item [6] to [7] is delayed until all interruption events occurred during the process [1] through [5] have been processed.

A1-5.4.2.3 Behavior when pressing access key

Figure 5-7 shows how a focus acts when the accesskey that corresponds to element B is pressed while the focus is on element A.

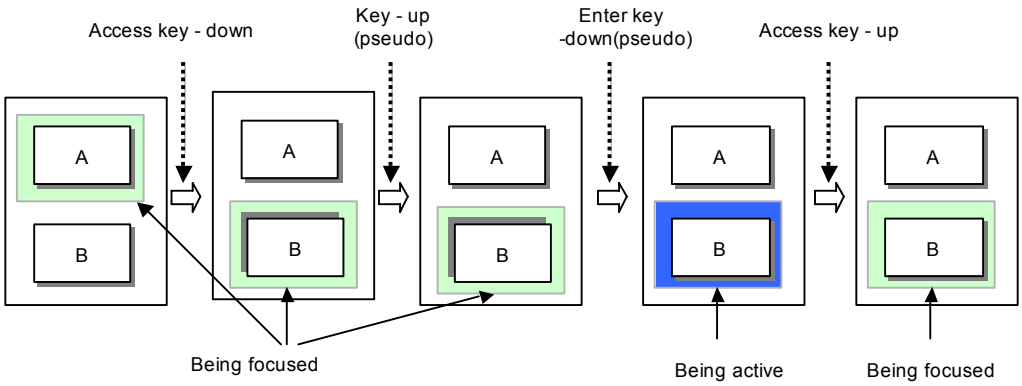


Figure 5-7 A Movement when the Access Key is pushed down

In the case of the example depicted in Figure 5-7, the access key works as follows:

[1] Pressing the key causes a keydown event for element A to be raised and triggers the script specified in the onkeydown.

[2] The element which focus exists in is made an A' element temporarily as a result of [1].

- It is A'=A so far as focus isn't moved while it processes of [1].
- Focus transition (A, A') is carried out when focus is moved while it processes of [1].

[3] When an A' element does not exist, a focus disappearance (A) is carried out, and focus generation (B) is carried out. Then the processes from the item [5] to the end are carried out.

[4] A focus transition (A', B) is carried out.

[5] An interruption event is taken out of a synchronous event queue and the corresponding script is executed.

[6] A keyup event is pseudoly raised against element B and the script specified by the onkeyup is triggered.

[7] A keydown event is pseudoly raised against element B and the script specified by the onkeydown is triggered.

[8] The element which focus exists in is made an B' element temporarily as a result of [7].

- It is B'=B so far as focus is not moved during the processes of [7].
- Focus transition (B, B') is carried out when focus is moved during the processes of [7].

[9] The style specified in the :active pseudo-class for element B’ is applied.

[10] A click event for element B’ is queued in a synchronous event queue.

[11] Releasing the key causes a keyup event to be raised against element B’ and triggers the script specified by the onkeyup.

[12] The style specified in the :active pseudo-class for element B’ is released.

However,

- Pressing the access key acts as if a focus is moved and then the Enter key is selected. And the key code of the access key is returned for the keyCode that is a DOM's attribute in items [1] and [6]. The key code of the Enter key is returned for the keyCode that is a DOM's attribute in items [7] and [11].
- Any interruption event except events mentioned during the process [1] through [10] (an interruption event related to Intrinsic Event module, an interruption event related to broadcasting

services, and an interruption event caused by setTimeout() or setInterval()) is not processed until the process [1] through [10] is completed.

The action in item [11] to [12] is delayed until all events raised during the process [1] through [10] have been processed.

When accesskey is specified to the element whose visibility attribute is “hidden”, even if the accesskey is pressed down no focus change is occurred nor no onclick event which must follow this action occurs.

A1-5.4.3 Initial position of focus

The focus is initially positioned at an element whose nav-index value is set to 0. If there is no element whose nav-index value is 0, no element is focused until focus( ) is executed.

However, in a profile where a focus is not controlled by using nav-index , an initial position of a focus depends on an implementation.

A1-5.4.4 Relationship between focus control and interruption events

An example that focus is moved from the A element to the B element is described. Here, after the focus has been moved to the A1 element by the focus() method in the script of the A element launched by the keydown event, it is further moved to the A2 element by another focus() method.

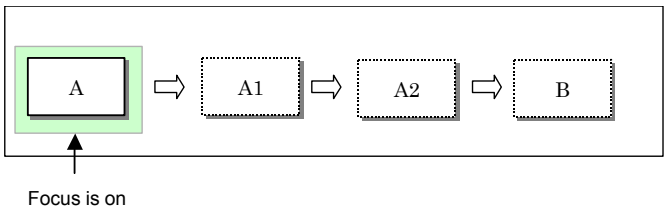


Figure 5-8 Example of the focus movement

The behaviour sequence of the above example is shown below.

[1] Focus is on A, and a button is pressed.

[2] The keydown event occurs on A and it is queued in the asynchronous event queue.

[3] The keydown event is retrieved from the asynchronous queue, and the onkeydown script is executed.

[4] The focus() function to move the focus to A1 in the script is executed.

[5] The style specified in the :focus pseudo class for A is released.

[6] The blur event occurs on A and is queued in the synchronous queue.

[7] The style specified in the :focus pseudo class for A1 is applied.

[8] The focus event occurs on A1 and is stored in the synchronous event queue.

The above occurrences of [5] to[8] are the focus transition (A, A1).

[9] The focus() function in the script to move focus to A2 is executed.

[10] The style specified in the :focus pseudo class for A1 is released.

[11] The blur event occurs on A1 and is stored in the synchronous event queue.

[12] The style specified in the :focus pseudo class for A2 is applied.

[13] The focus event occurs on A2 and is stored in the synchronous event queue.

The above occurrences of [10] to[13] are the focus transition (A1, A2).

[14] The execution of the onkeydown script is finished.

[15] The interruption events are retrieved from the synchronous queue and the corresponding scripts are executed.

[16] Until the synchronous queue becomes empty, the interruption events are retrieved and the corresponding scripts are executed.

[17] The navigation property for the focus transition from A to B is applied.

[18] The style specified in the :focus pseudo class for A2 is released.

[19] The blur event occurs on A2 and is queued in the synchronous event queue.

[20] The style specified in the :focus pseudo class for B is applied.

[21] The focus event occurs on B and is stored in the synchronous event queue.

The above occurrences of [18] to [21] are the focus transition (A2, B).

[22] The interruption events are retrieved from the synchronous queue and the corresponding scripts are executed.

[23] Until the synchronous queue becomes empty, the interruption events are retrieved and the corresponding scripts are executed.

[24] The keyup event occurs on B and it is stored in the asynchronous event queue.

Asynchronous events including keyup events can be stored in the asynchronous event queue in parallel with behaviours of the above sequence. Any retrieval of an event from the synchronous event queue does not occur until the corresponding script has been executed.

With regard to the change of the control process from on the synchronous event queue to on the asynchronous event queue, the control of the asynchronous event queue is started if the synchronous event queue is empty at the end of the execution of event handlers. Conversely, if the synchronous event queue is not empty, event handlers corresponding to the synchronous events in the synchronous event queue are executed one after another. The control process is transferred to on the asynchronous event queue when the synchronous event queue becomes empty.

Chapter APPENDIX 1-6 Guidelines on Presenting BML Documents

A1-6.1 CSS

A1-6.1.1 Background

The initial value for the background-color property must be transparent so that the underlying elements at z-index see through. If any underlying elements do not exist, a black (or black as specified in the keyword) background is assumed to exist in processing. Note that when a default style sheet contains any other value for the background-color property than the aforementioned value, the aforementioned value is overwritten by a value specified in the default style sheet.

A1-6.1.2 Action for designation of colour depth

It is desirable that colour depth must be restricted in the grayscale-color-index property by operation, because how much colour depth representation is actually possible depends on the functions of a receiver.

A1-6.1.3 Presentation for border property of inline element

When an a element or span element is displayed across more than one line, its border is displayed as shown in Figure 6-1 per CSS2 9.4.2:

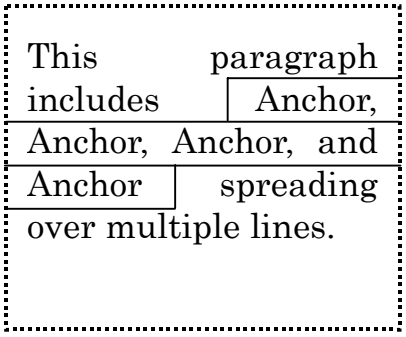


Figure 6-1 Example of Displaying Border Property for Inline Element (1)

And, the border is displayed as shown in Figure 6-2 if a new line is started at any other position than the end of a line by inserting a linefeed (or a br element) or other operations.

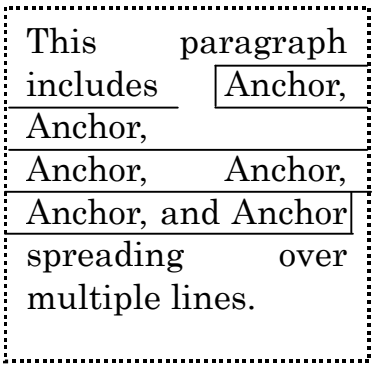


Figure 6-2 Example of Displaying Border Property for Inline Element (2)

A1-6.2 Guideline for normalStyle, focusStyle and activeStyle

This section specifies the guideline for the application of normalStyle, focusStyle, and activeStyle defined in ARIB STD-B24 Volume 2 Section 7.1.

A1-6.2.1 Inherited values of normalStyle, focusStyle and activeStyle

CSS property values which are held in the normalStyle, focusStyle, and activeStyle are the computed values.

When a final value is determined by combining different properties such as combining border-style:none with border-width:10px, values of the concerned attributes, instead of computed values based on the combined properties, are returned. In the case of combining border-style:none with border-width:10px, 10px is obtained from borderWidth and returned.

A1-6.2.2 Application, cascading, and inheritance of property values

Cascading process is performed with the following process. Every time when values are written in the normalStyle, focusStyle, and activeStyle by DOM interface, the declaration which specifies the written value is added after the whole declaration which has been declared up to that time. The declaration which is added during this process must follow Table 6-1 with regard to the origin, the specificity, and the order specified.

Table 6-1 Cascading process for normalStyle, focusStyle, and activeStyle

Type	Application	
origin	Handled as the description with author style sheets	

specificity	normalStyle	Same as specified with #myid.
		specificity=100

	focusStyle	Same as specified with #myid:focus.
		specificity=110

	activeStyle	Same as specified with #myid:active.
		specificity=110

order specified	Applied with priority as being specified at the later position than the style attribute of the element.	

Further, the old declaration must always be hidden at the second or later write process, as the new write-in is added to the later part than the declaration which is added at the previous write-in. Therefore even if the declaration to be hidden is deleted, the same result can be obtained. That is, the number of the declaration to be kept may be one at most for the implementation.

Moreover, if the value is inherited from the parent element in the process of deciding the computed value, the value applied to the parent element at that time is inherited.

In other cases, it will be processed according to the specification of CSS2 chapter 6 as far as there is no particular description.

A1-6.2.3 Example

```
<head>
    .....
    <style>
        p:focus {background-color-index:inherit;}
```

```
        #p1 {background-color-index:inherit;}
        #div1 {background-color-index:3;}
    </style>
    .....
</head>
<body>
    <div id=”div1”>
        <p id=”p1”>  the first p element  </p>
        <p id=”p2”>  the second p element  </p>
    </div>
    .....
</body>
```

When the normalStyle, focusStyle, and activeStyle is specified, the application of the CSS properties to the above document must follow Table 6-2.

Table 6-2 Application of CSS properties with the specification of normalStyle, focusStyle, and activeStyle

id		Initial status (Note 1)	Behaviour 1 (Note 2)	Behaviour 2 (Note 3)	Behaviour 2’ (Note 4)	Behaviour 3 (Note 5)
div1	normalStyle	3	“4”	4	4	“6”
	activeStyle		“1”	1	1	1
	focusStyle	3	4	“5”	5	5
p1	normalStyle	(inherit) (Note 6)	4 (inherit)	4 (inherit)	5 (inherit)	6 (inherit)
	activeStyle	3 (inherit)	4 (inherit)	4 (inherit)	5 (inherit)	6 (inherit)
	focusStyle	3 (inherit)	4 (inherit)	4 (inherit)	5 (inherit)	6 (inherit)
p2	normalStyle	(default) (Note 7)	8 (default)	8 (default)	8 (default)	8 (default)
	activeStyle	8 (default)	8 (default)	8 (default)	8 (default)	8 (default)
	focusStyle	3 (inherit)	4 (inherit)	4 (inherit)	5 (inherit)	6 (inherit)

Note 1: “Initial status” is a value of DOM resulted by the initial behaviour.
Note 2: “Behaviour 1 ”: It is the case where
normalStyle.backgroundColorIndex is set to 4 and
activeStyle.backgroundColorIndex is set to 1
for the div1 which is not in the status being focused.
Note 3: “Behaviour 2”: It is the case where it is in the status of “Behaviour 1” and
focusStyle.backgroundColorIndex of div1 is set to 5.
Note 4: “Behaviour 2’”: It is the case where it is in the status of “Behaviour 2” and
The focus is on the div.
Note 5: “Behaviour 3”: It is the case where it is in the status of “Behaviour 2” and
normalStyle.backgroundColorIndex of div is set to 6.
Note 6: (inherit) means that the specified value is inherited. The node must hold that it inherits the value of the parent property.
Note 7: (default) means that the value which follows the default style sheet is set.

A1-6.3 Composition of text

This section specifies how composition of text must be implemented. Note that under an operational environment that does not require uniqueness of presentation what this section contains may be ignored.

A1-6.3.1 Baseline

The position of the baseline is always the lowest part of the fonts. A baseline applied to be a single line of characters must be a straight line, even if characters are of different sizes. The height of the line is affected by the font sizes.

Figure 6-3 shows an example. The font sizes of “apple” and “APPLE” are different, but the baseline is displayed with alignment.

Figure 6-3 Handling of the baseline

A1-6.3.2 Letter spacing

The letter space is not added to before the first character of the text of an element and after the last character. And if it has a child element, the letter space of the parent element is added between the character of the parent element just before the child element and the first character of the child element. Moreover, the letter space of the parent element is added between the last character of the child element and the character of the parent element just after the child element.

An example is described below.

Example:

applegrapepear

Figure 6-4 shows the document structure and the result of the presentation. The letter space is not inserted before the p element "apple". But the letters spaces of the p element are inserted just before and after the span element "grape". And, the letter space is not inserted just after the p element "pear".

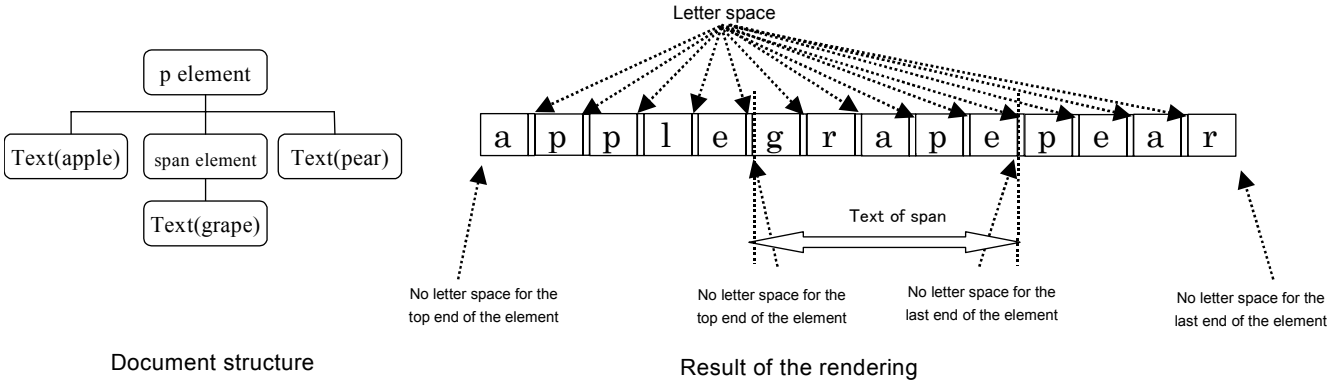


Figure 6-4 Example of processing letter spacing

A1-6.3.3 Border

A right border and a left border are differently handled from a top border and a bottom border.

In the inline box, the right and left border of the child element does not cut into the border of the parent element. However, the top and bottom border of the child element happens to cut into the border of the parent element. Figure 6-5 shows an example. Figure 6-5(a) shows an example of the right border. The border of the child element is added to the next part of the letter space which exists at the right side of the character “B” of the parent element in the inline box. The right border of the parent element will be displayed at the next part of the right border of the child element. Figure 6-5 (b) shows an example of the left border. The border of the child element comes after the border of the parent element. The top and bottom borders of the child element happen to cut into the border of the parent element in the inline box in order to present the baseline with alignment. The border which hangs the padding is clipped as shown in Figure 6-5(c).

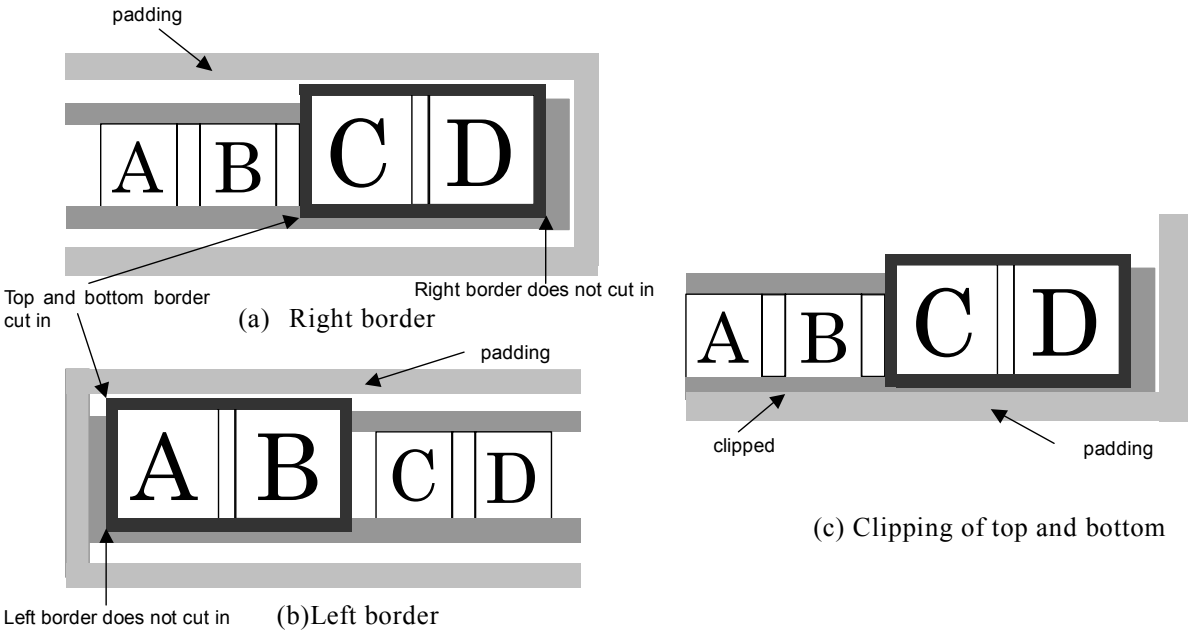


Figure 6-5 Example of Border

A1-6.3.4 Fold back

The fold back occurs depending on the positional relationship of text and padding. The position of the fold back is called a cutting point below. The cutting point locates at the right end of the font in principle. However, if the fold back occurs at the first character of the text included in an element or in the middle of the series of borders, it locates at the right end of the border. For the convenience of the explanation, it is described before and after fold back separately below.

A1-6.3.4.1 Relationship to letter spacing

The letter space is not forwarded to the next line. The following is an example. Figure 6-6 shows an example when the text with letter spaces hangs the padding and is folded back. As shown in the left part of the figure, when the position of the padding hangs the letter space, the cutting point is the right side of the character “D” and the fold back occurs. In this case, the letter space is not added before the first character which is forwarded to the next line. When the position of the padding hangs the character “E”, the cutting point is the right side of the character “D”. In this case, the letter space is added neither after the character “D” nor before the character “E”.

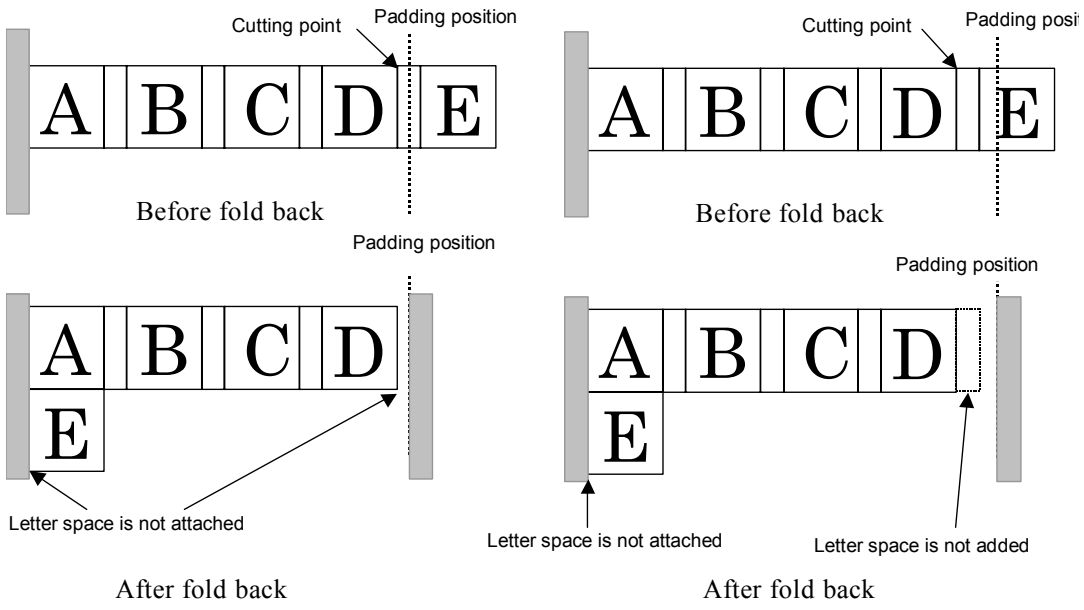


Figure 6-6 Relationship between letter spacing and fold back

A1-6.3.4.2 Relationship between display effects

The display effect means the border, background and so on, which are added to the element. The display effect does not appear at the cutting point.

Figure 6-7 shows the relationship between border and the fold back. As the cutting point occurs at the right side of the character “D”, the display effect occurs up to the character “D” and does not occur in the right from here. Therefore, the border is drawn to the right end of the character “D”.

The letter space is not added to the character “E” which is forwarded to the next line and the border is drawn from the left end of the character “E”.

The process of the background is based on the above process.

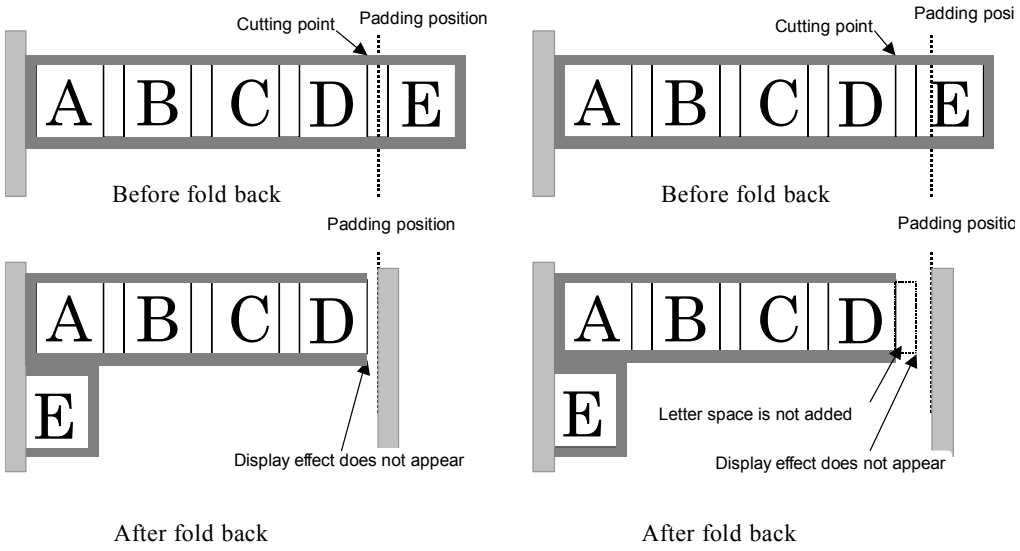


Figure 6-7 Relationship between border and fold back

A1-6.3.4.3 Relationship to child element

Figure 6-8 shows the case when the right border of the child element hangs the padding position of the parent element. As the cutting point is the right side of the character “E”, the border of the child element is forwarded to the next line. The display effect does not occur at the track of the forwarded border.

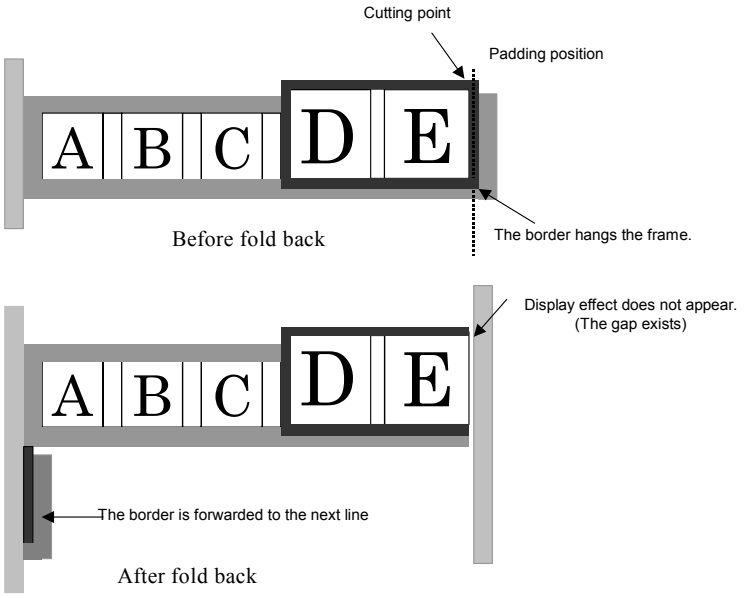


Figure 6-8 Relationship to the right border of the child element

An example of the case when the left border of the child element hangs the padding position is shown below.

When the border of the starting position of the child element hangs the padding position, the border of the child element is forwarded to the next line, as the right side of the character “C” of the element is the cutting point.

And, when the character of the child element hangs the padding, the border of the child element is not forwarded and the string after the character “D” is located in the next line, as the cutting point is the left side of the character “D” of the child element. This is as same as the case when the width of the border is specified with 0px. In this case, it is assumed that the width of the border is 0 and the right letter space of the character “C” is left.

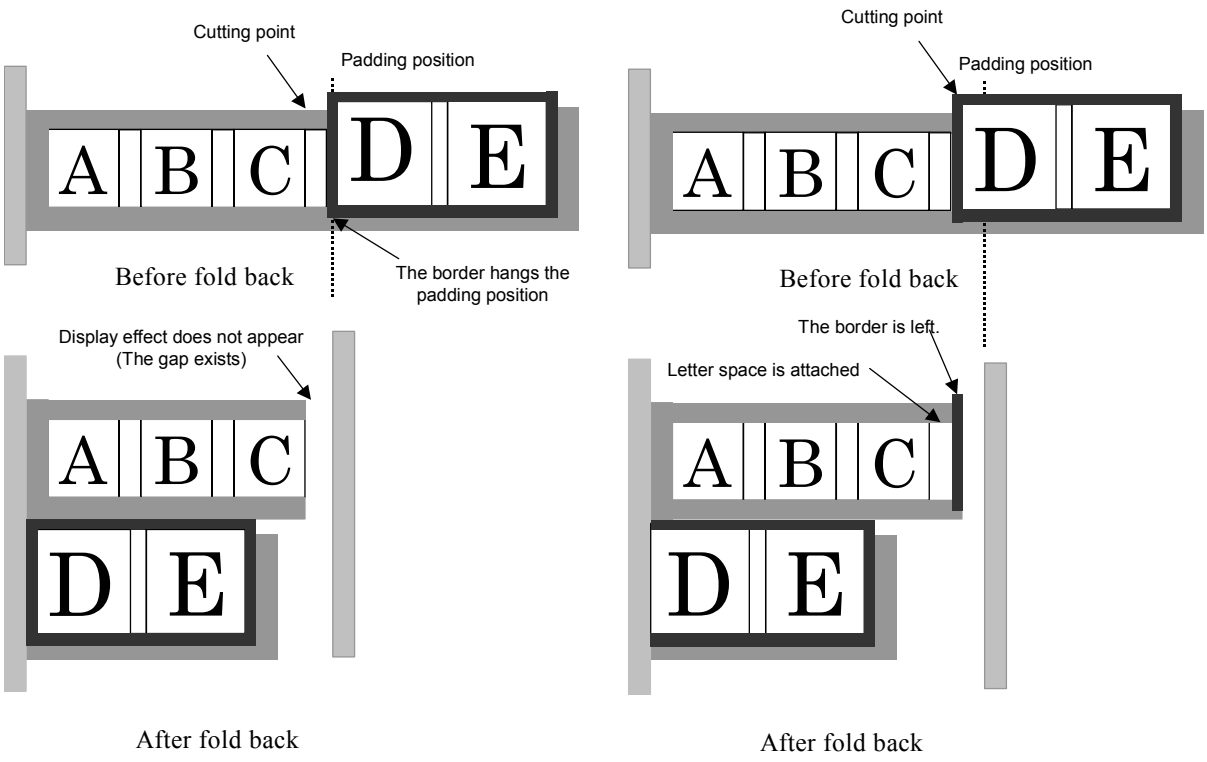


Figure 6-9 Relationship to the left border of the child element

An example of the case when multiple borders are added is shown below. When multiple borders are added, the cutting point is the right side of the border which does not overlap the padding position.

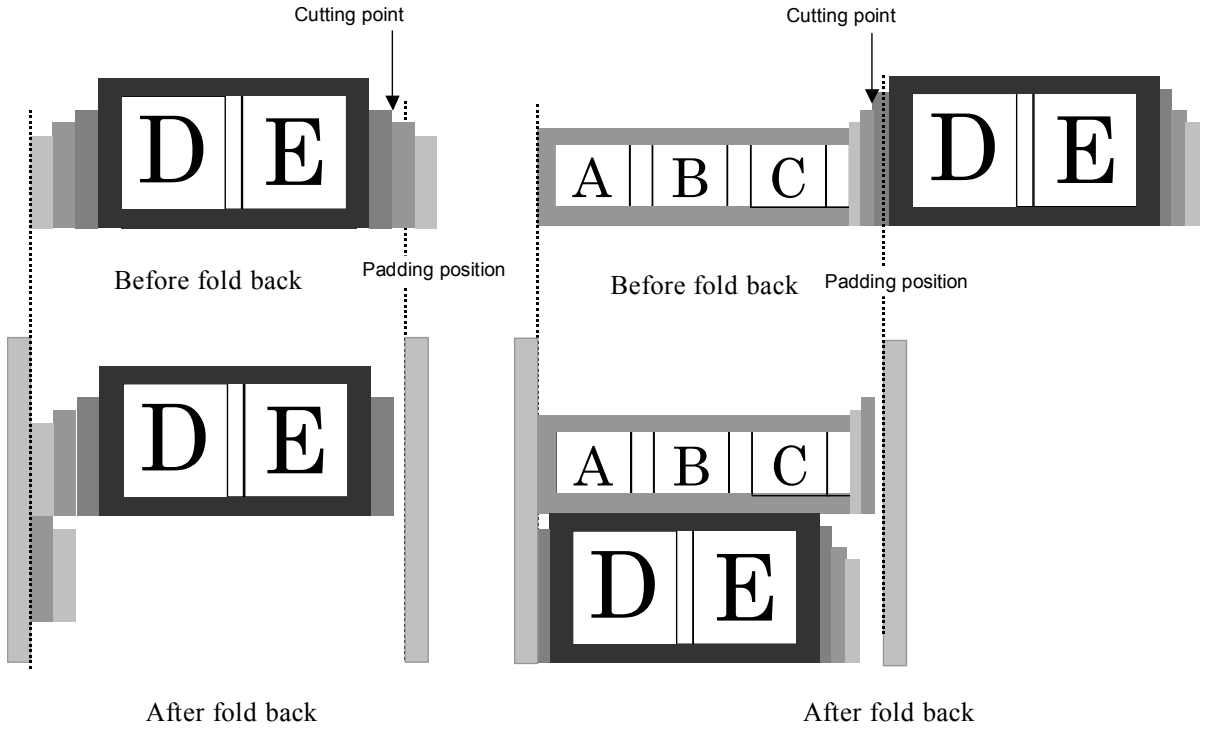


Figure 6-10 Relationship of multiple border and fold back

A1-6.3.5 Br element

The br element is processed as it has no size. When the br element causes the fold back, the cutting point is the right side of the character just before the br element.

A1-6.3.5.1 Relationship between br element and fold back

Figure 6-11 shows the relationship between the br element and the fold back. When the padding is located between the letter space of the character “C” just before the br element and the character “D” just after the br element, the cutting point is as same as that caused by the br element. When the padding is located at the letter space of the character “C”, the cutting point is the right side of the character “C” and the character “D” is forwarded to the next line.

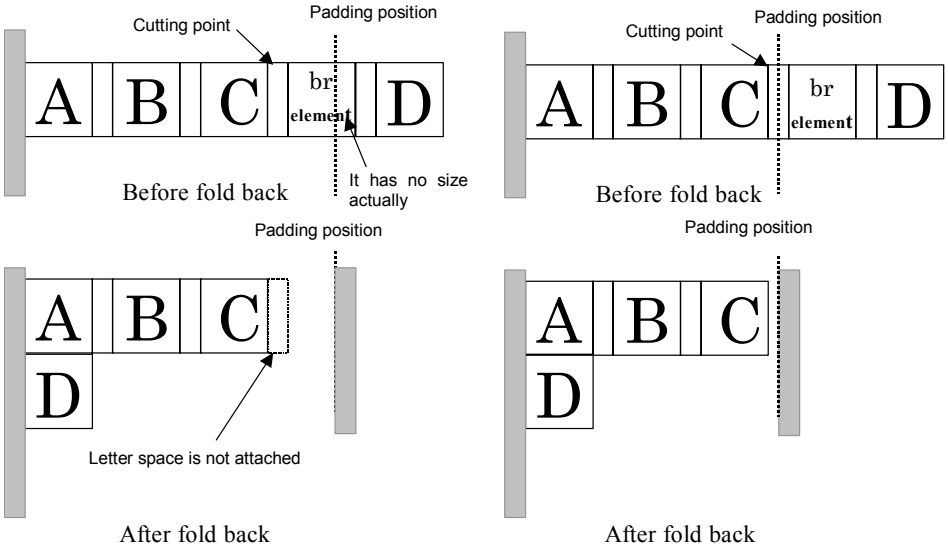


Figure 6-11 Relationship between the br element and the cutting point

A1-6.3.6 Relationship between br element and left border

When the br element appears at first in the child element of an element and overlaps the padding position, the left border of the child element is forwarded to the next line. Figure 6-12 shows an example.

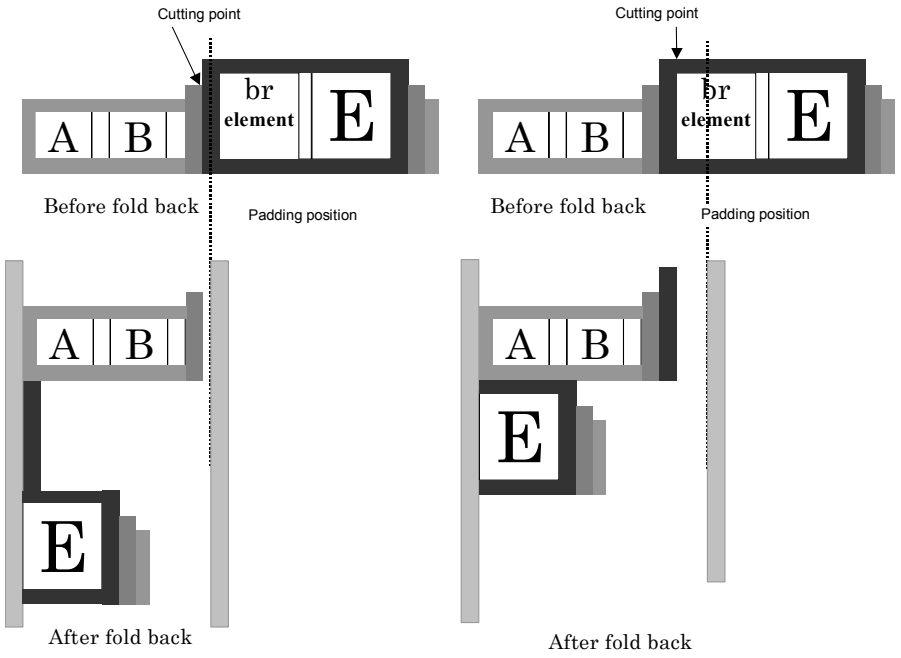


Figure 6-12 Relationship between the br element and the left border

A1-6.3.7 Relationship between position of the br element and fold back

When the br element appears at the first position of a parent element, it is folded back leaving the border. And, when the br element occurs in succession, it is forwarded to the next line by the specified value with the line–height property depending on the number of the br elements. Figure 6-13 shows an example.

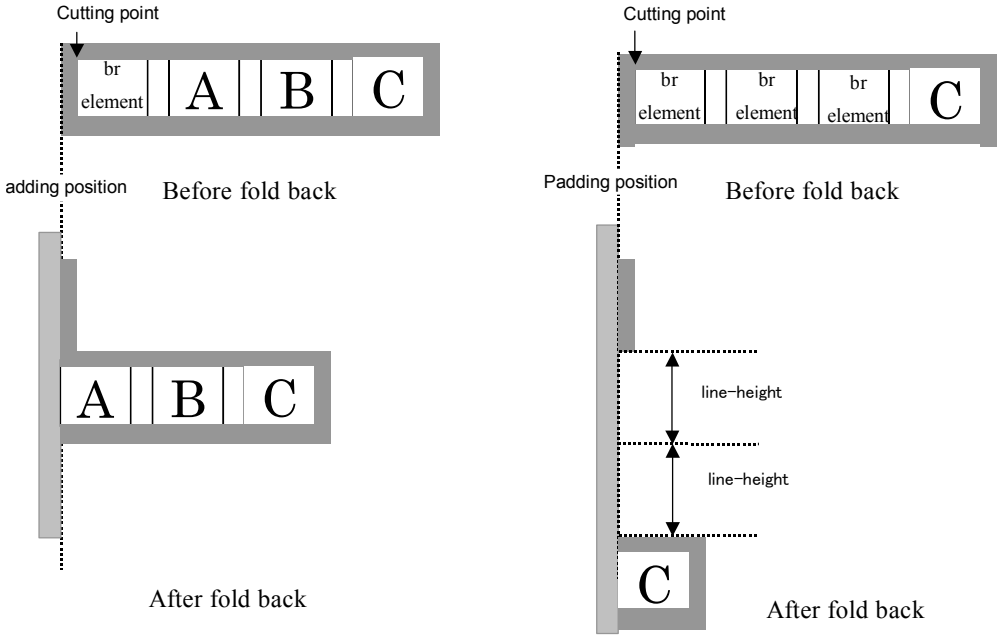


Figure 6-13 Relationship between position of br element and fold back

A1-6.3.7.1 Relationship between br element and display effect

When the br element causes the fold back, the display effect does not occur at the cutting point either. Neither the letter space is forwarded to the next line. Figure 6-14 shows an example.

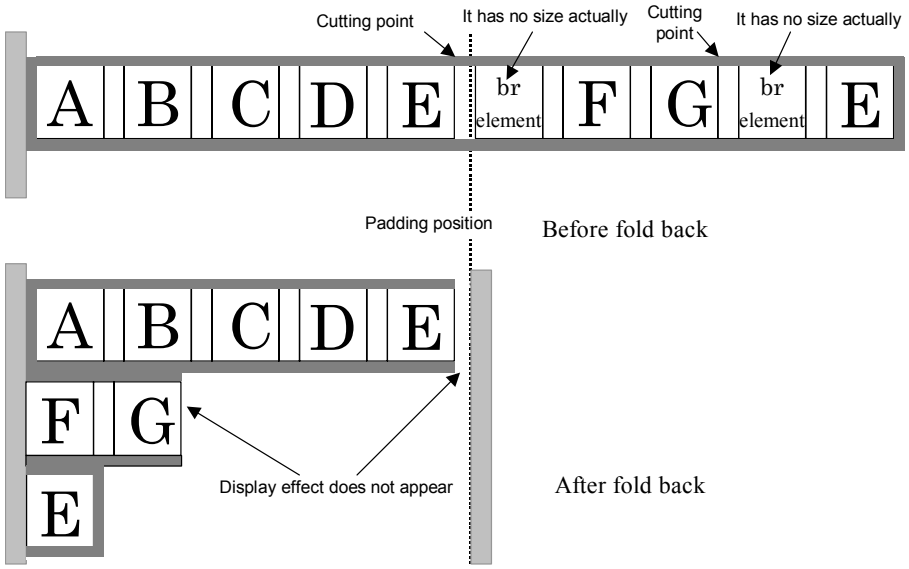


Figure 6-14 Relationship between br element and display effect

A1-6.3.8 Height of line

A line-height value specified for the inline element must be equal to or greater than the value of “font-size+2*border-width”. A line-height value specified for the p element must be the value of “font-size <= line-height”. If one of these conditions fails, the presentation depends on an implementation. If the line-height value is “normal”, it results the equal size of the font-size property. Thus, when the border must be added, the line-height value must be specified implicitly.

A1-6.3.8.1 Relationship between line-height and character positioning

The process of the character positioning with the line-height property is described below.

[1] Check the relationship among line-height, font-size, and border-width

[2] Locate the font in the center of the height specified by line-height

[3] Configure the line with the alignment of the font baselines

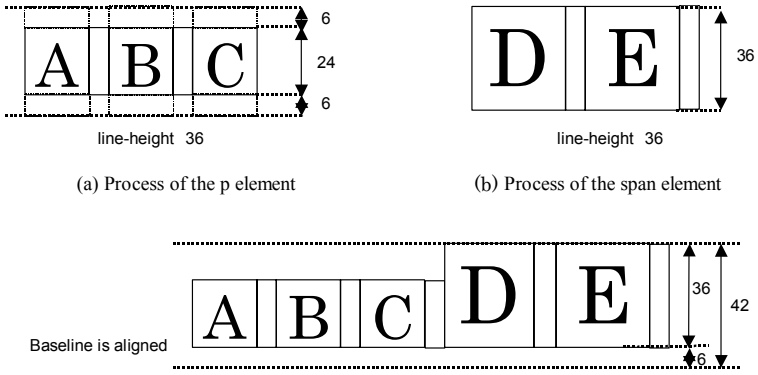
An example is described according to the above process.

Example:

<p style="line-height:36px”> ABC <span style="font-size:36px”>DE</p>

Figure 6-15 shows the result of the presentation for the example. The text “ABC” of the p element is located in the center of the 36px height specified by line-height as shown in Figure 6-15(a). The line-height of the span element, which is a child element, is fixed to “inherit” in the operation resulting 36px. The font is located in the center. In this case, it coincides with the font-size as shown in Figure 6-15(b).

After this process described above, a line is configured with the baseline alignment of each text as shown in Figure 6-15(c).



(a) Process of the p element

(b) Process of the span element

(c) Rendering result of one line

Figure 6.15 Relationship between line-height and character positioning

Further, when the line-height is an odd number, the position of the font is shifted downwards. Figure 6-16 shows an example.

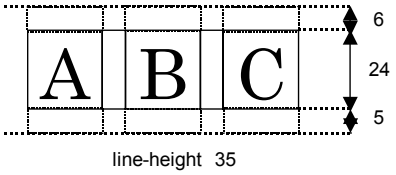


Figure 6-16 Odd number of line-height

A1-6.3.8.2 Relationship between line-height and the fold back

The character positioning is processed for every line. Thus, after deciding the cutting point, the text is divided for each line and the height of the line which is used for presentation of each line is calculated. The following is an example.

Example:

<p style="line-height:36px”> ABC <span style="font-size:36px”>DEFGHJKL</p>

Figure 6-17 shows the case when the padding position is located between the character “E” of the span element and the character “F” of the p element. It must be remarked that the heights of the line differ each other.

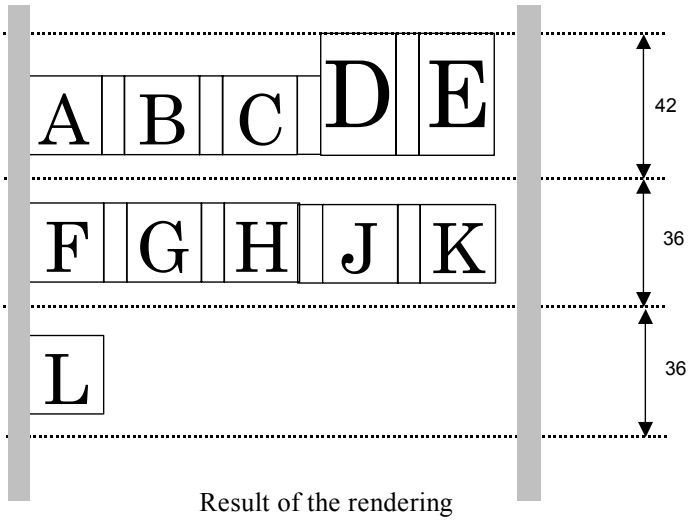


Figure 6-17 Line-height and fold back

A1-6.3.8.3 Relationship of line-height and border

Figure 6-18 shows a display example of the case when the border is added.

Example:

```
<p style="border-width:6px; border-style:solid; line-height:44px">
    ABC
    <span style="border-width:2px; border-style:solid;font-size:30px;">
        DE
    </span>
</p>
```

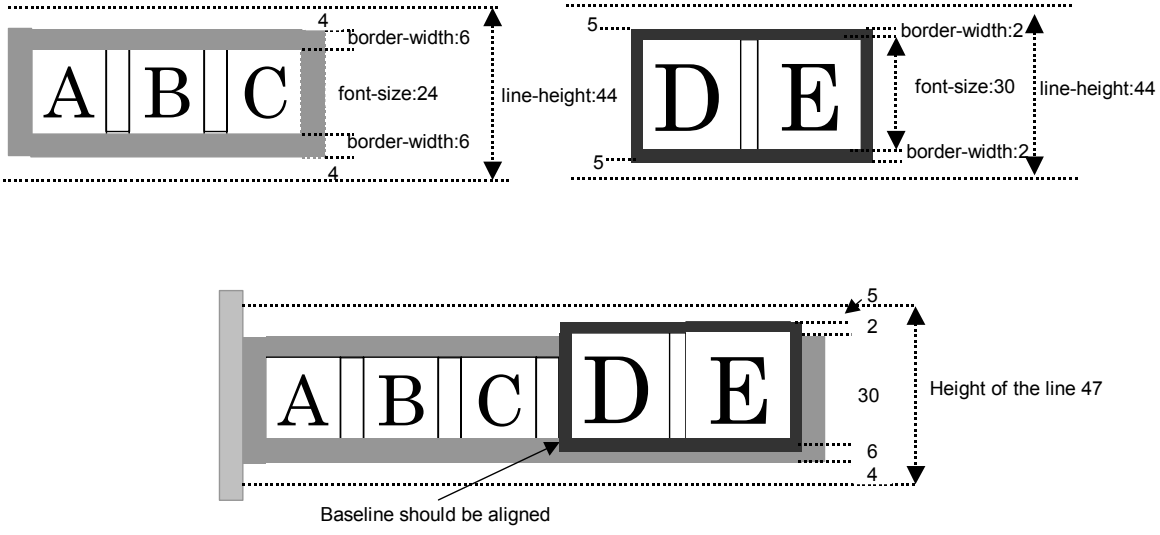


Figure 6-18 Relationship of line-height and border

A1-6.3.9 Alignment

The text is aligned left normally, however it can be specified as being aligned right and centered. Figure 6-19 shows an example. If it is centered, the middle of the length between the start position and the cutting point of one line is placed to the center of the padding. Therefore, The width of the part where no display effect occurs is divided equally to right and left. If it is aligned right, the right end of the part where no display effect occurs is filled and the part without display effect is produced at the left end.

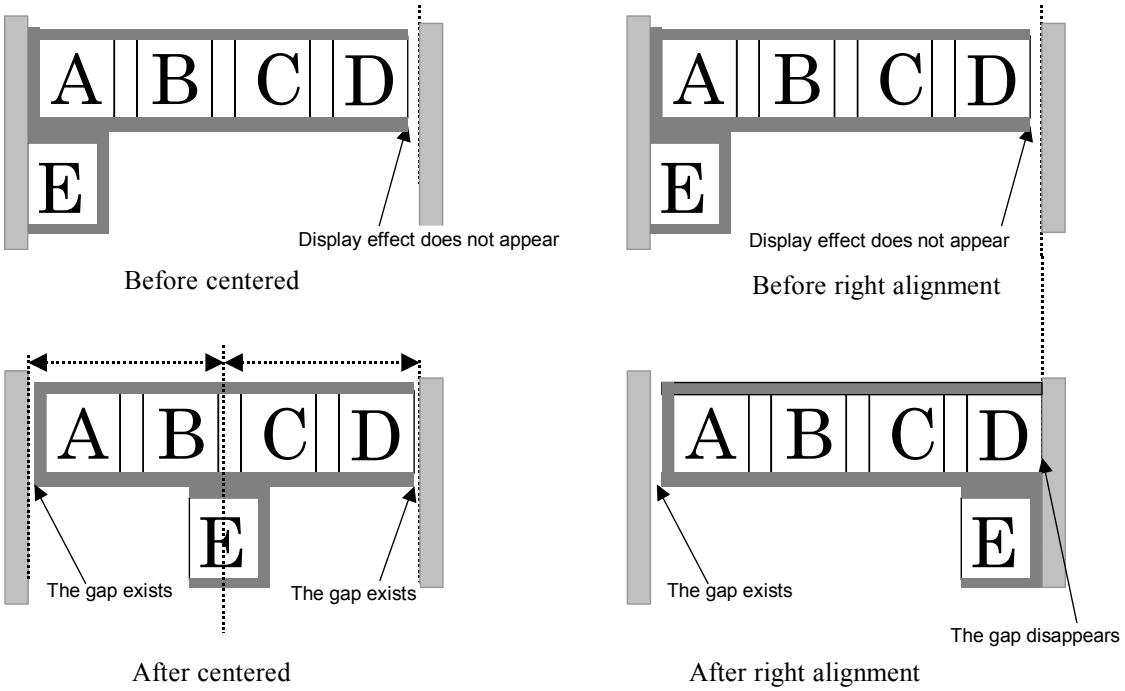


Figure 6-19 Presentation for centered and aligned right status

Further, if the width between the padding is an odd number, it is aligned left. Figure 6-20 shows an example.

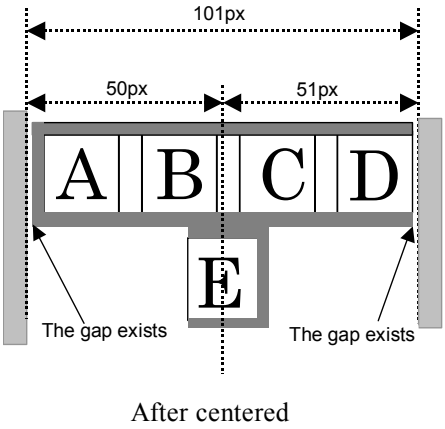


Figure 6-20 Odd number of the width between the padding

A1-6.3.10 Other remarks

Other remarks are explained in this section.

- Handling of bold

  When bold is specified for font-weight, the presentation depends on an implementation so far as the text composition is not broken.

- Interdiction process

  The interdiction process is not performed.

- Change of font-size and font-family

  The font-size property and the font-family property can be changed only separately. Thus, if both are changed sequentially, it may happen to be in the state of the combination unused for the operation temporally at the time when one property is changed. For such a case, lockScreen() must be executed so that the status of the combination unused for the operation is not presented. If it is not executed, the presentation depends on an implementation.

A1-6.4 Presentation of BML element

A1-6.4.1 Presentation of input element and textarea element

To ensure uniqueness of presentation, any input element must not involve implicit display that depends on the implementation of a browser. For example, no change in presentation caused by the disable attribute involves implicit display. Note that under an operational environment that does not require uniqueness of presentation this may be ignored.

A1-6.4.2 Displaying a elements

To meet a need, if any, to ensure uniqueness of presentation, any a element must not involve implicit display that depends on an implementation of a browser.

Historically, the HTML has an implicit rule that an a element must be underlined, but in the broadcast contents,

- As the uniqueness of presentation is regarded important, it is not desirable that the default is a presentation function that depends on the implementation of a browser.
- It is desirable that more flexible presentation should be possible because control is basically performed remotely.

For these two reasons, the rule at the beginning of this section must be established. Note that under an operational environment that does not require uniqueness of presentation this may be ignored.

A1-6.5 Scaling Visual Representations

Presentation control regarding scaling video and other visual representations is performed with the coordinates defined in ARIB STD-B24 Volume 1. Figure 6-21 shows a virtual process model when video transmitted with transmission channel is presented as one of monomedia data on the text graphics plane in the end.

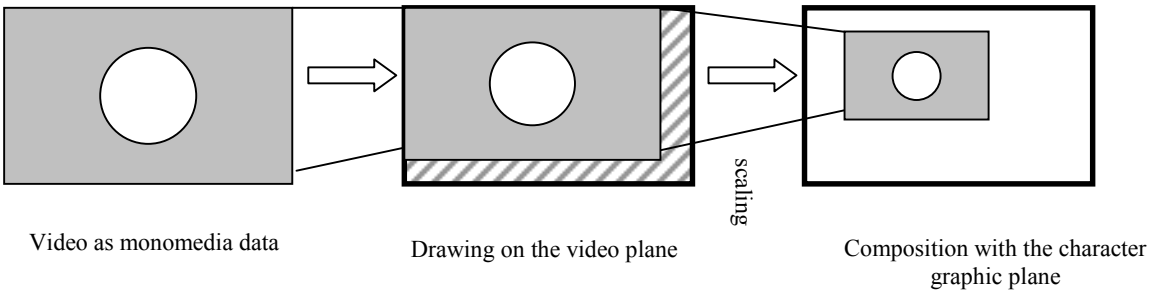


Figure 6-21 Virtual process model of video scaling

In this process model, video is rendered on the video plane at first. Then it is scaled down when it is composed with the text graphics plane.

Next the resolution of the corresponding video plane is defined in Table 6-3 with respect to the resolution and aspect ratio of the text graphics plane (they correspond to resolution property and display-aspect-ratio property) in the process model shown in the Figure 6-21.

Table 6-3 Video plane resolution corresponding to text graphics plane

Resolution of text graphics plane	Resolution of video plane	Remarks
1920 x 1080 (16:9)	1920 x 1080	
1280 x 720 (16:9)	1280 x 720	
960 x 540 (16:9)	1920 x 1080	Line and pixel of text graphics plane are drawn twice
720 x 480 (16:9)	720 x 480	
720 x 480 (4:3)	720 x 480	

The drawing process of various video formats in the above mentioned virtual model is described as examples below. Video formats of 1920x1080, 960x540 and 720x480 are used for this explanation as representative resolutions. In the drawing process on the video plane below, the vertical resolution is preserved exclusively at the receiving with conversion and the horizontal resolution is changed, if necessary, so that the aspect ratio may not change. In addition, the width and height values (in bold) of video in the figure of composition/drawing of text graphics plane in each table are to be specified for the object element. For your information, this Appendix neither restricts the combination of text graphics plane and video plane as described below nor limits the drawing process within the described items.

Table 6-4 shows the presentation process when various video data are presented with the same or the 1/2 scaled down size for the 1920x1080 text graphics plane.

Table 6-4 Video presentation in case of 1920 x 1080 text graphics plane

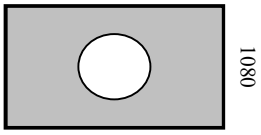
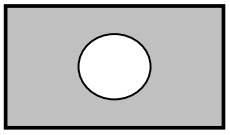
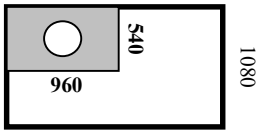
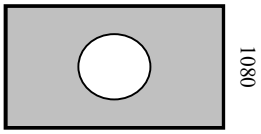
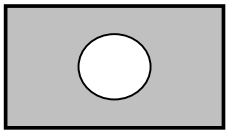
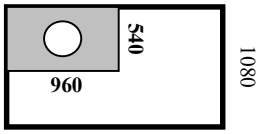
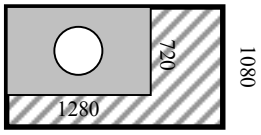
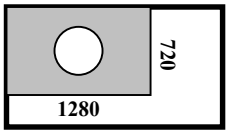
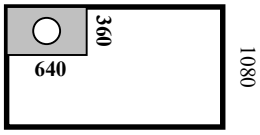
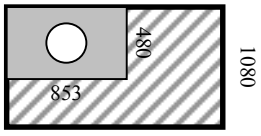
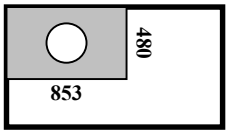
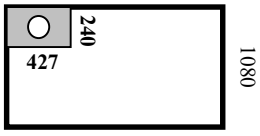
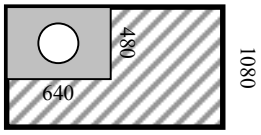
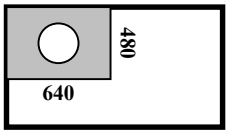
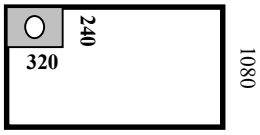
Video format	Drawing on the video plane	Composition and drawing on the text graphics plane	
		Composition in case of same size (128/128)	Composition in case of 1/2 (64/128)size
1920x1080/16:9 1440x1080/16:9			
1280x720/16:9 (1080i down converted)			
1280x720/16:9			
720x480/16:9			
720x480/4:3			

Table 6-5 shows the presentation process when various video data are presented with the same or the 1/2 scaled down size for the 960x540 text graphics plane.

Table 6-5 Video presentation in case of 960 x 540 text graphics plane

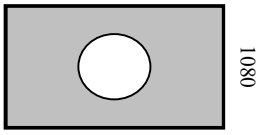
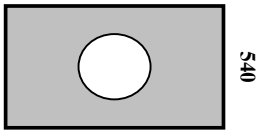
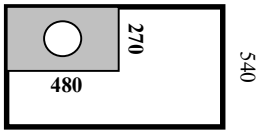
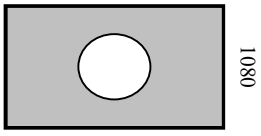
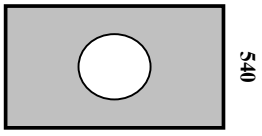
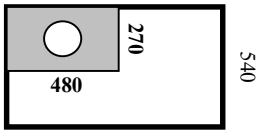
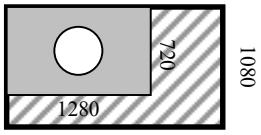
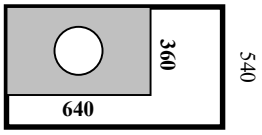
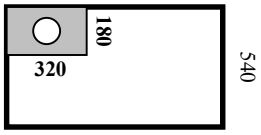
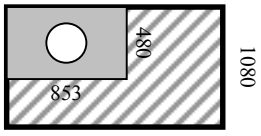
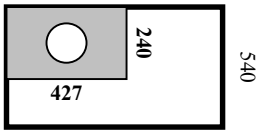
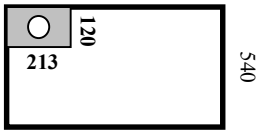
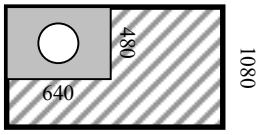
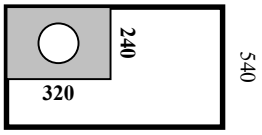
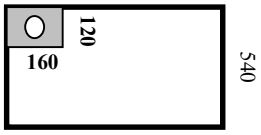
Video format	Drawing on the video plane	Composition and drawing on the text graphics plane	
		Composition in case of same size (128/128)	Composition in case of 1/2 (64/128)size
1920x1080/16:9 1440x1080/16:9			
1280x720/16:9 (1080i down converted)			
1280x720/16:9			
720x480/16:9			
720x480/4:3			

Table 6-6 shows the presentation process when various video data are presented with the same or the 1/2 scaled down size for the 720x480 (16:9) text graphics plane.

Table 6-6 Video presentation in case of 720 x 480 (16:9) text graphics plane

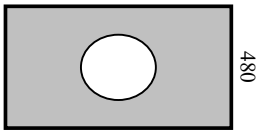
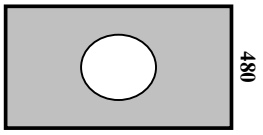
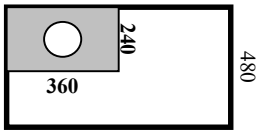
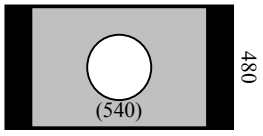
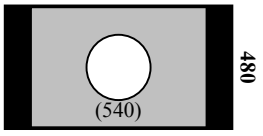
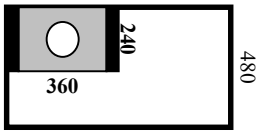
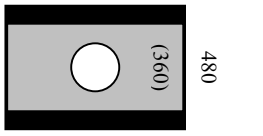
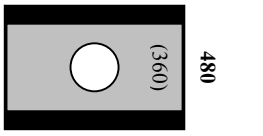
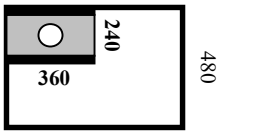
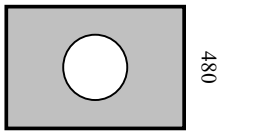
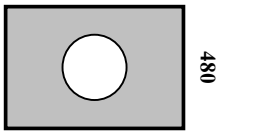
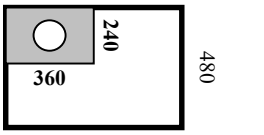
Video format	Drawing on the video plane	Composition and drawing on the text graphics plane	
		Composition in case of same size (128/128)	Composition in case of 1/2 (64/128)size
720x480/16:9 544x480/16:9			
720x480/4:3 544x480/4:3 (side panel)			

Table 6-7 shows the presentation process when various video data are presented with the same or the 1/2 scaled down size for the 720x480 (4:3) text graphics plane.

Table 6-7 Video presentation in case of 720 x 480 (4:3) text graphics plane

Video format	Drawing on the video plane	Drawing on the video plane	
		Composition in case of same size (128/128)	Composition in case of 1/2 (64/128)size
720x480/16:9 544x480/16:9 (letter box)			
720x480/4:3 544x480/4:3			

A1-6.6 Update of modules

This section describes updates of modules related to monomedia data.

A1-6.6.1 Reference of same module by multiple objects

When a module which an object is referencing is updated, the result is reflected on the presentation by updating the data attribute after completing the version-up.

When multiple objects are referencing the same module, the update is reflected on the presentation by updating the data attributes of all objects after completing the version-up as shown in Figure 6-22. If the data attributes of all objects referencing the same module are not updated, the presentation depends on an implementation. Therefore the content must be so produced that the objects which are expected to be displayed differently must refer the different modules or resources explicitly.

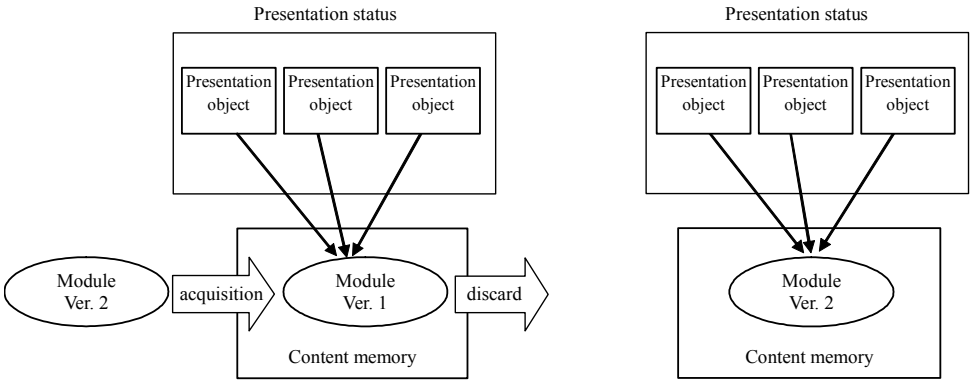


Figure 6-22 Reference of same module by multiple objects

A1-6.6.2 Relationship between update and module lock

When the data attribute is changed without locking the modules, the presentation depends on implementations because the update of the presentation depends on the timing and the status whether the module in the transmission stream is updated or not. Therefore it is desirable to take the following procedure if the presentation must be updated as the modules are updated:

- Changes the data attribute after detecting the module update
- Confirms that the module of the previous version is not locked when the data attribute is changed

Moreover it is desirable to take the following procedure if the presentation must not be updated as the modules are updated:

- Does not Change the data attribute
- Locks the modules at the previous change of the data attribute or at the reception if the data attribute may change

The example in the case when the module of new version is displayed securely resulting from the update of the module is described below.

[1] Waits for the update of the module by setting the subscribe attribute of the beitem element.

[2] Event handler A is launched by detecting the update of the module.

[3] The module is unlocked in the event handler A.

[4] Executes LockModuleOnMemory() in the event handler A and waits for the modules.

[5] An interrupt event with the module being locked occurs and the event handler B is launched.

[6] The data attribute is changed in the event handler B and the module of the new version is presented.

The process described above is shown in Figure 6-23.

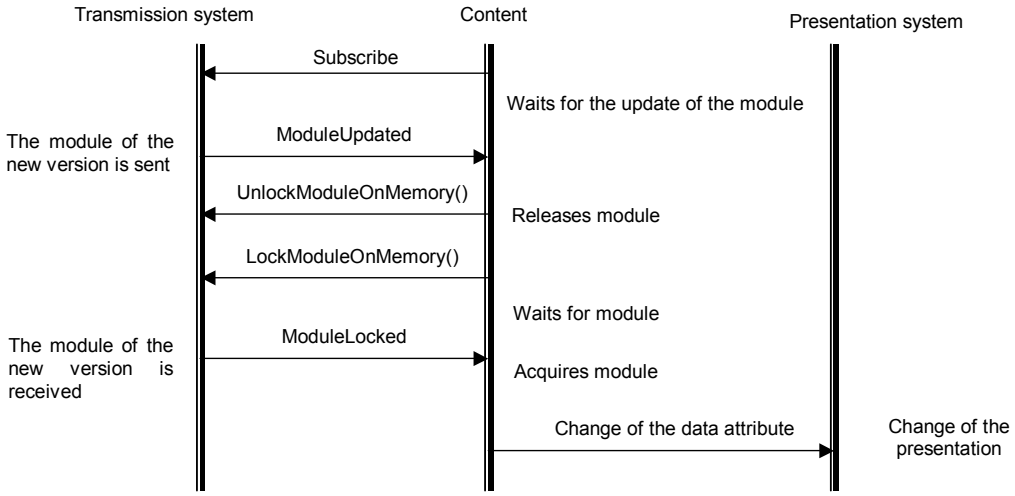


Figure 6-23 Update of the module and the presentation timing

Chapter APPENDIX 1-7 Guidelines on XSLT

A1-7.1 Style Sheet for XML Documents

The XSL and CSS are style sheets that define how XML documents are displayed, but only the XSL style sheet must be used as a style sheet in B-XML because the CSS does not have formatting objects of input system such as buttons or text boxes.

However, the CSS can be used as a style sheet for BML documents to be generated in an XSL document because the CSS is used as a style sheet for BML documents.

A1-7.2 Structure of XSL Document

The XSLT language, a Tree Transformation language for XSL must be applied to an XSL document for the purpose of operating B-XML to describe rules that convert an XML document into a BML document. The XSL document must be an XML document that is defined in B-XML.

Chapter APPENDIX 1-8 Guidelines on Procedural Description

A1-8.1 Guidelines on Delayed Call Origination

A1-8.1.1 Function executed on call origination

In the execution of an interaction channel telecommunication process registered for delayed origination, the startConnection() function that is defined in the script file given by the argument pgm_uri of registerTransmission() is executed. The script file is a text file that includes scripts written in character-set defined in an operational standard regulation, and at least the startConnection() function must be defined in the script file.

A1-8.1.2 Handling text strings used for datagram

When registerTransmission() has two or more strdata arguments, the arguments are passed to the same number of arguments of the startConnection() function.

A1-8.1.3 Executable procedures when performing delayed call origination

A script executed in delayed call origination does not use any DOM API

Table 8-1 lists the broadcast extended functions that are available in to a script executed in delayed call origination:

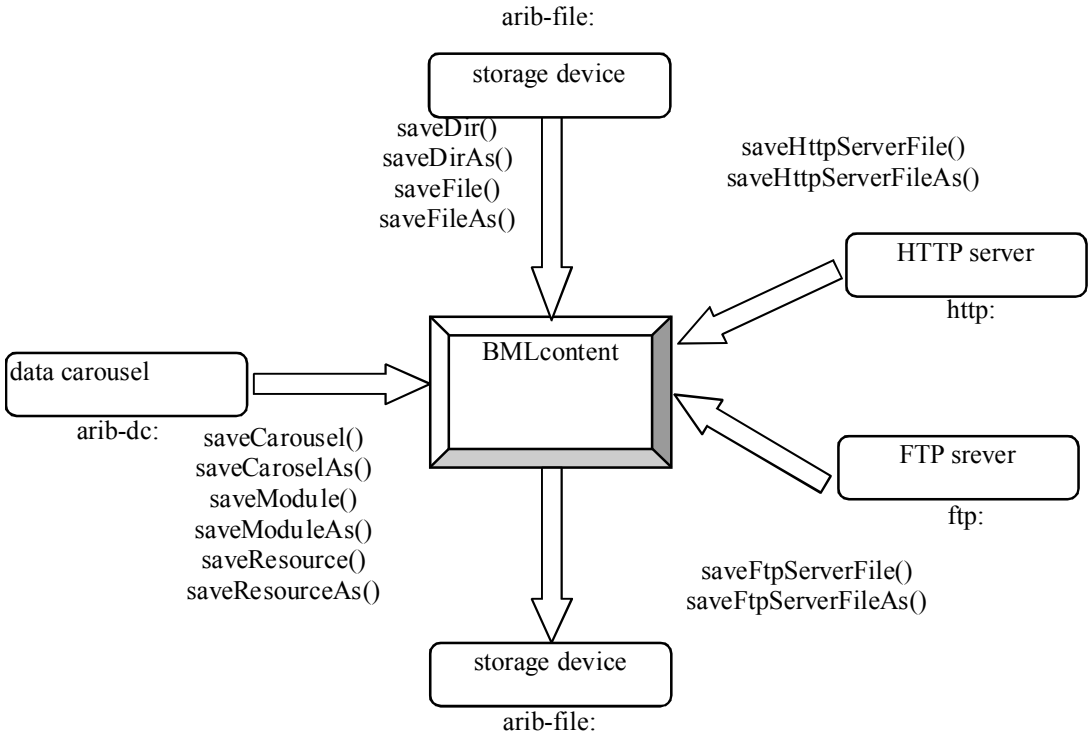
Table 8-1 Broadcast Extended Functions Available for Delayed Call Origination

Type of Broadcast Extended Function	Available for Delayed Call Origination ?
EPG related functions	No
Program group index related functions	No
Caption display control functions	No
Permanent storage functions	Yes
Interaction channel functions	Yes
Action control functions	Only getIRDID() and getBrowserSupport() are available.
Receiver audio control	No
Timer functions	Only sleep() is available
External character functions	No
Other functions	Yes
Ureg pseudo-object property	No

A1-8.2 Guidelines on Extended Functions for Storing

A1-8.2.1 Relationship between source and destination

The following diagram shows how a saveXXX function acts on sources and destinations to store a content.



Behavior of saveXXX functions

The applicable URIs are described below.

- When URI is specified with arib-dc:, the URI represents:
  - [1] data carousel to be transmitted
  - [2] modules to be transmitted in a multipart format
  - [3] modules to be transmitted in a format other than multipart
  - [4] resources that are contained in a module to be transmitted in a multipart format and that contains no “/”.
  - [5] resources that are contained in a module to be transmitted in a multipart format and that contains “/”.
- When URI is specified with http:, https:, or ftp:, the URI represents:
  - [6] directories located on a server
  - [7] files located on a server
  - [8] directories to be created on a server
  - [9] files to be created on a server
- When URI is specified with arib-file:, the URI represents:
  - [10] directories located in a storage device
  - [11] files located in a storage device
  - [12] directories to be created in a storage device
  - [13] files to be created in a storage device
- When URI is specified not using the above descriptions, the URI represents:
  - [14] a file or directory that is not a file nor directory listed above.

A1-8.2.2 Extended APIs for Storing

This section briefly explains how each extended API for storing specifies URI.

When the dest_path argument contains an existing file including [7] and [11] above, the file is allowed to be overwritten.

Directory Management Functions

Number	Function	src_path	dest_path	Description
1	saveDirAs	[10]	[10][12]	Stores a directory by specifying a destination.
2	saveDir	[10]	-	Stores a directory without specifying a destination.
3	createDir	-	[12]	Creates a directory.
4	getParentDirName	-	[10][11]	Obtains a name of a parent directory.
5	getDirNames	-	[10]	Obtains an Array object representing directory names.
6	isDirExisting	-	[10]	Verifies whether or not a directory exists.

File Management Functions

Number	Function	src_path	dest_path	Description
7	saveFileAs	[11]	[10][11] [13]	Stores a file by specifying a destination.
8	saveFile	[11]	-	Stores a file without specifying a destination.
9	getFileNames	-	[10]	Obtains an Array object representing file names.
10	isFileExisting	-	[11]	Verifies whether or not a file exists.

File Input/Output Functions

Number	Function	src_path	dest_path	Description
11	writeArray	-	[11][13]	Writes information in an Array object to a logical storage.
12	readArray	-	[11]	Obtains the content of a file that exists in a logical storage as an Array object.

Inquiry Functions

Number	Function	src_path	dest_path	Description
13	getDirInfo	-	[10]	Obtains information about a directory.
14	getFileInfo	-	[11]	Obtains information about a file.
15	getCarouselInfo	-	[1]	Obtains information about a carousel.
16	getModuleInfo	-	[2][3]	Obtains information about a module.
17	getContentSource	-	-	Verifies whether a content was invoked from a broadcasting service or invoked from a storage device.
18	getStorageInfo	-	-	Obtains information about a storage device.

Data Carousel Storage Functions

Number	Function	src_path	dest_path	Description
19	saveCarouselAs	[1]	[10][12]	Stores a carousel by specifying a destination.
20	saveCarousel	[1]	-	Stores a carousel without specifying a destination
21	saveModuleAs	[2][3]	[10][11] [12][13]	Stores a module by specifying a destination.
22	saveModule	[2][3]	-	Stores a module without specifying a destination.
23	saveResourceAs	[4][5]	[10][11] [12][13]	Stores a resource by specifying a destination.
24	saveResource	[4][5]	-	Stores a resource without specifying a destination.

Storing functions for HTTP server files

Number	Function	src_path	dest_path	Description
25	saveHttpServerFileAs	[7]	[10][11] [13]	Stores a file on a HTTP server by specifying a destination.
26	saveHttpServerFile	[7]	-	Stores a file on a HTTP server without specifying a destination.
27	sendHttpServerFileAs	[11]	[6][7][9]	Send a file to a HTTP server.

Storing functions for FTP server files

Number	Function	src_path	dest_path	Description
28	saveFtpServerFileAs	[7]	[10][11] [13]	Stores a file on a FTP server by specifying a destination.
29	saveFtpServerFile	[7]	-	Stores a file on a FTP server without specifying a destination.
30	sendFtpServerFileAs	[11]	[6][7][9]	Send a file to a FTP server.

Storing functions for Mail server files

Number	Function	src_path	dest_path	Description
31	sendTextMail	-	-	Sends a mail.

A1-8.2.3 Sample behaviours of file/directory management functions

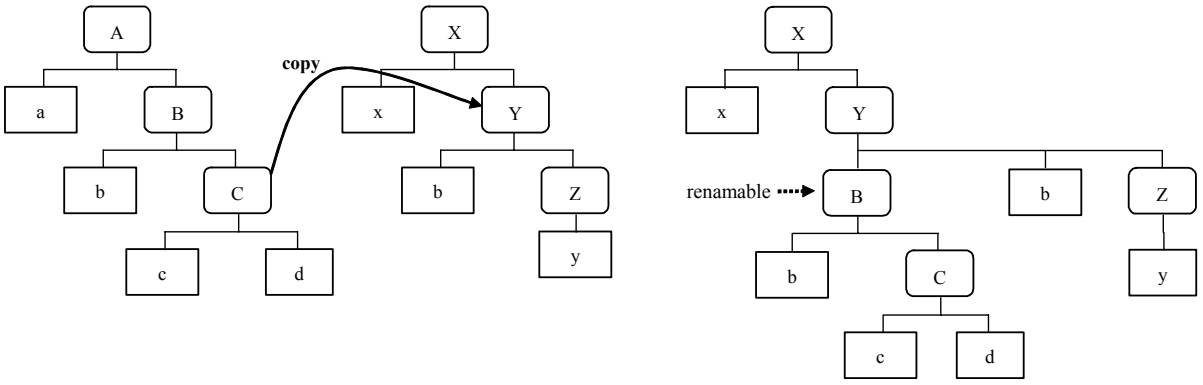
The following summarizes behaviours of file/directory management functions.

Source	Specified Destination	Destination's existence	Result	Remarks
File	Directory	Exists	The source file is stored under the specified directory.	See the following diagram
		N/A	N/A	
	File	Exists	The destination file is overwritten.	
		Does not exist	A new file is created as a destination.	
Directory	Directory	Exists	The directory is overwritten.	See the following diagram
		Does not exist	A new directory is created as a destination	See the following diagram
	File	Exists	Failure	
		N/A	N/A	

Note that whether or not an end user has to be informed of an update resulting from an overwritten file/directory has to be noticed must be specified in an operational guideline.

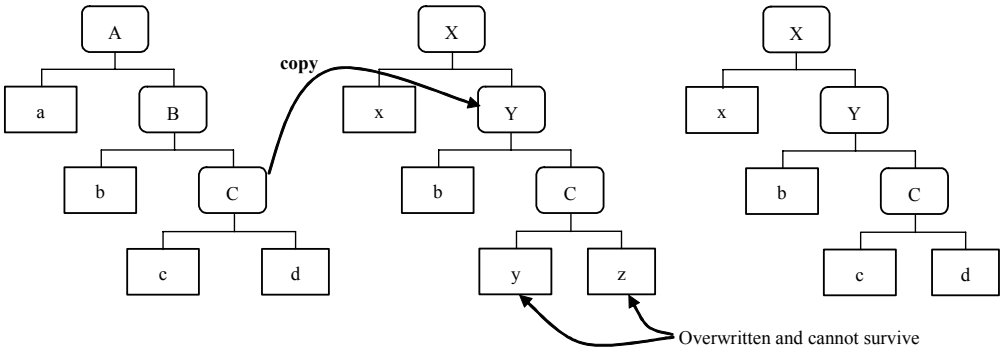
- The following diagram illustrates a sample behaviour of saveDirAs(arib-file://A/B, arib-file://X/Y).

When a source directory and a destination directory are specified, the source is copied to a location immediately under the specified directory.



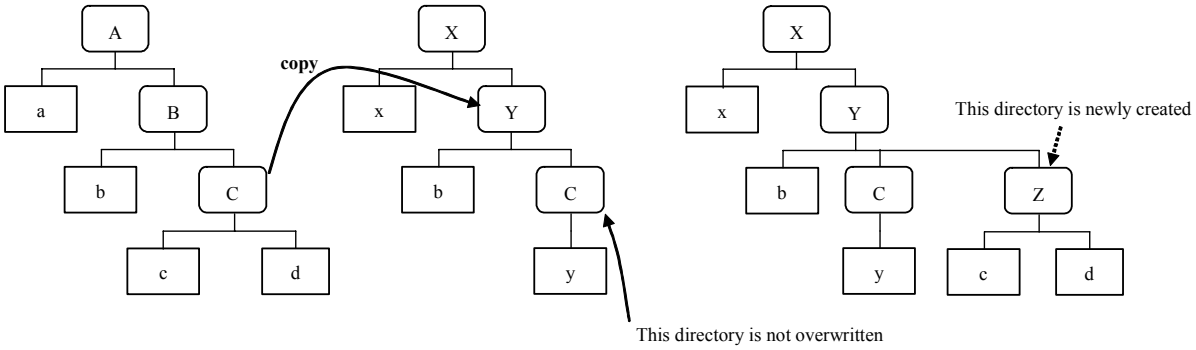
- The following diagram illustrates a sample behaviour of saveDirAs(arib-file://A/B/C, arib-file://X/Y).

When a source directory and a destination directory is specified and the specified directory has a child directory with the same name as the source, this child directory is overwritten.



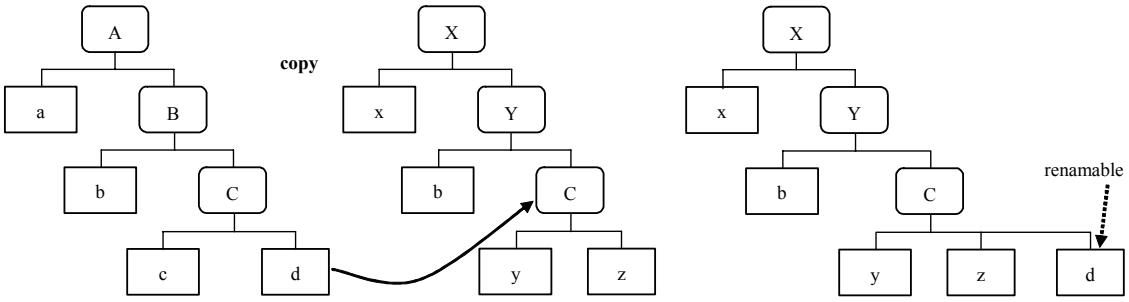
- The following diagram illustrates a sample behaviour of saveDirAs(arib-file://A/B/C, arib-file://X/Y, Z).

  When a source directory and a destination directory are specified and the specified directory does not exist, a new directory is created to contain the copied information. The new directory uses the specified destination's name. When a source directory and a destination directory are specified and the specified directory exists, the existing directory and the contained files are deleted and the copied directory and its files are saved.



- The following diagram illustrates a sample behaviour of saveFileAs(arib-file://A/B/C/d, arib-file://X/Y/C).

  When a source file and a destination directory are specified, the source file is copied to a location immediately under the specified directory.



A1-8.2.4 Sample behaviours for copying information in data carousel to another location

The following summarizes where copied information is saved.

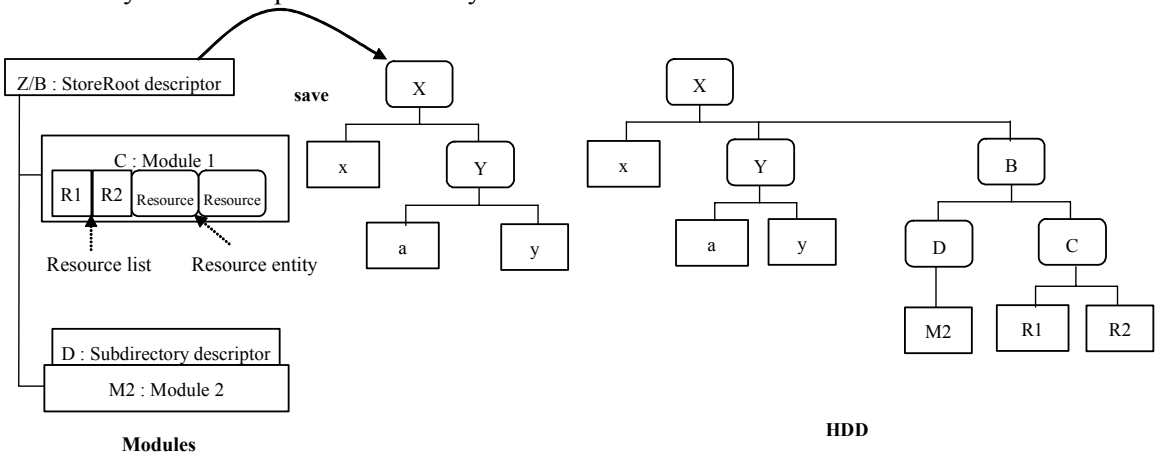
Source	StoreRoot descriptor		Result	Remarks
Data carousel	Specified	Root name/Directory name	Directory identified with the specified directory name without the root name	See the following diagram
		Root name	Modules are saved under the specified directory	
	Not specified	-	Location with a specified name and under the specified directory	

This section outlines how the extended APIs (saveCarouselAs(), saveCarousel(), saveModuleAs(), saveModule(), saveResourceAs(), saveResource())that copy information contained in a data carousel act on a receiver.

1. When an API copying information contained from a data carousel is executed, a latest DII/DDB is obtained. If information specified to be copied (a data carousel, modules, or resources) exists, the API copies the information as specified. If information specified to be copied does not exist, an error is returned.
2. During a copy operation, the ES containing the specified information is monitored. When either of the following occurs, the copy operation is aborted and an error is returned.
   (a) The data_event_id identification of the concerned ES has been updated.
   (b) The version of the DII describing the concerned ES has been updated.
3. When a content that is currently presented (a content that has invoked a function transmitting a copy operation) is discarded during the copy operation (e.g. when data_event_id of a currently presented content is updated), the copy operation aborts. In this case, an error is not capable of being returned.

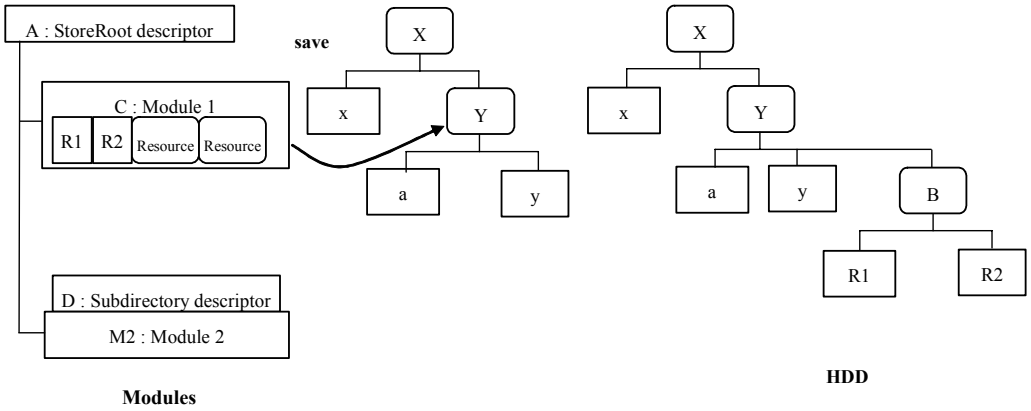
- The following diagram illustrates a sample behaviour of saveCarouselAs(/<component_tag>,arib-file://X).

When a data carousel is specified as a source and the StoreRoot descriptor is not specified, the information contained in the source is saved in a location immediately under the specified directory. When a data carousel is specified as a source and the StoreRoot descriptor is also specified, the information is saved in a location identified without using the specified root name. If the destination is specified with only a root name, the copied information is saved as modules that are located immediately under the specified directory.



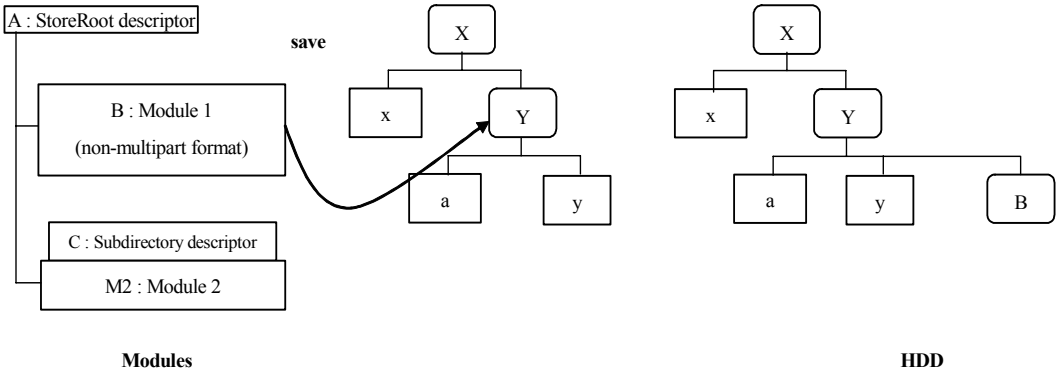
- The following diagram illustrates a sample behaviour of saveModuleAs(arib-dc://...../B, arib-file://X/Y).

  When modules are specified as a source, the information contained in the source is saved in a location immediately under the directory specified as a destination. Module B and the other copied modules are saved under Directory Y.



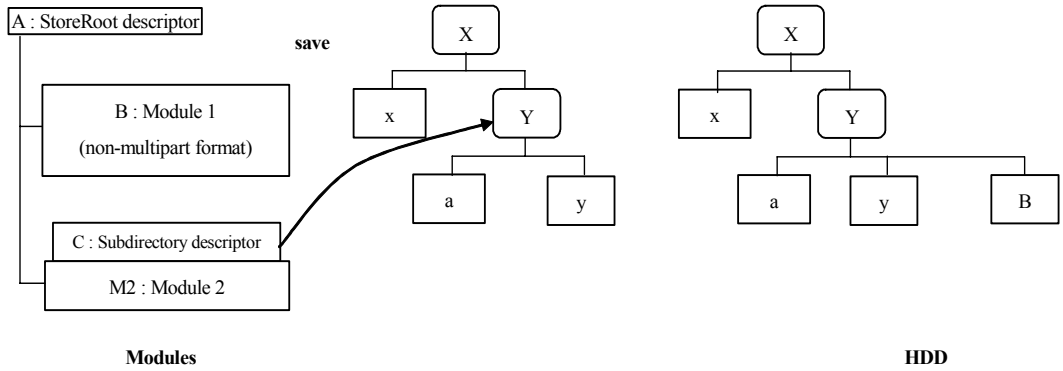
- The following diagram illustrates a sample behaviour of saveModuleAs(arib-dc://...../B, arib-file://X/Y).

  When modules are specified as a source, the information contained in the source is saved in a location immediately under the directory specified as a destination. Module B and the other copied modules are saved under Directory Y.



- The following diagram illustrates a sample behaviour of saveModuleAs(arib-dc://...../C, arib-file://X/Y).

  When modules are specified as a source, the information contained in the source is saved in a location immediately under the directory specified as a destination. Module C and the other copied modules are saved under Directory Y.

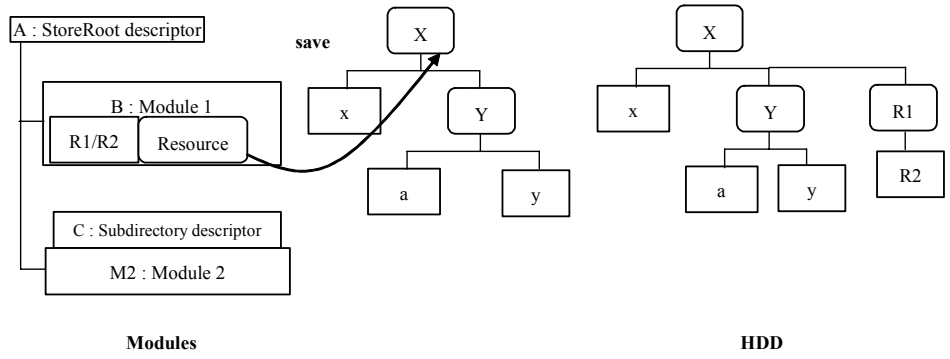


When resources are specified as a source, what are saved depends on the usage of "/".

Source	"/"	What are saved
Resource	Used	Files and a directory
	Not used	Files without a directory

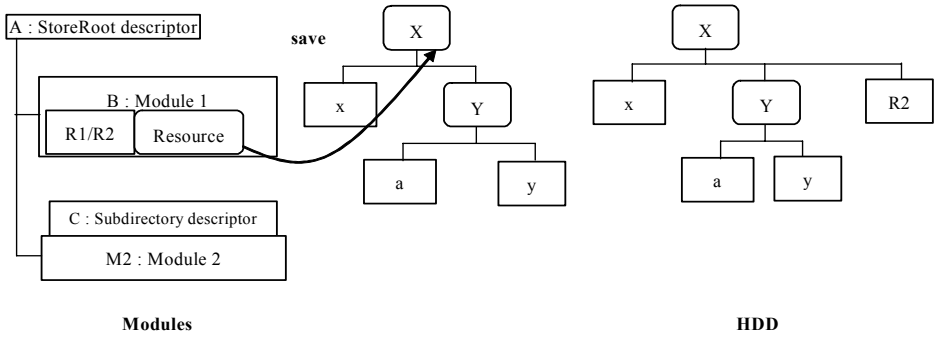
- The following diagram illustrates a sample behaviour of saveResourceAs(arib-dc://...../B/R1, arib-file://X).

When resources are specified as a source, the information contained in the source is saved in a location immediately under the directory specified as a destination. Resource R1 and the other copied resources are saved under Directory X.



- The following diagram illustrates a sample behaviour of saveResourceAs(arib-dc://...../B/R1/R2, arib-file://X).

Resource 2 and the following resources are saved under Directory X as files.

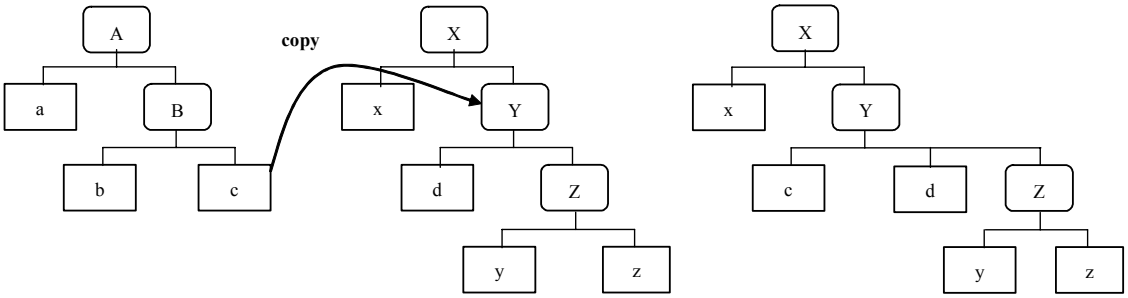


A1-8.2.5 Sample behaviours for copying file on HTTP/FTP server to another location

When files on an HTTP/FTP server are specified as a source, they are treated the same as files in a storage device.

- The following diagram illustrates a sample behaviour of saveHttpServerFileAs(http://A/B/c,arib-file://X/Y).

When files on an HTTP/FTP server are specified as a source, the information contained in the source is saved in a location immediately under the directory specified as a destination. File c is saved under Directory Y.



A1-8.3 Guidelines on Extended Functions for Reserving Records or Storing Information

A receiver must have a user interface that prompts the end user to explicitly accept or deny a writing access executed by a function acting on the receiver to reserve a recording operation or store information. This requirement implies that no playback operation of recorded programs nor copy operation is executed unless it is confirmed by the end user. Detailed specification of the user interface is not covered by this section.

A1-8.4 Guidelines on Bookmarks

A1-8.4.1 Identification of Bookmark Area

When an MM service needs to read/write information from/to a bookmark area,

readBookmarkArray()/writeBookmarkArray() is used to read/write information in variable-length block units. Note that a single variable-length block is treated as a single file. To identify a variable-length block, use URI in the following format.

nvram://bookmark/<block number>

<block number> : 0 ~ N (Detailed usage is defined in an operational guideline.)

When an extended function for broadcasting (readBookmarkArray() or writeBookmarkArray()) is executed by specifying URI in other formats than the format described above, no reading access nor writing access to NVRAM is performed and null (readBookmarkArray() failed) or NaN (writeBookmarkArray() failed) is returned as a return value.

The Sections 8.4.2 and 8.4.3 contains detailed information about operations related to bookmark services.

A1-8.4.2 NVRAM operations in bookmark services

A1-8.4.2.1 Number of bookmark areas

- A maximum number of bookmarks that can be recorded in a receiver is defined in an operational guideline.

A1-8.4.2.2 Access to bookmark area

- A maximum number of data blocks for a bookmark area complies with the specification in Section 8.4.1.

A1-8.4.2.3 Use of bookmark area in NVRAM

- A bookmark area in NVRAM contains no bookmark whose bookmark type has not been registered.

A1-8.4.2.4 Basic data area and extended data area

- A bookmark area in NVRAM consists of basic data areas and extended data areas. A basic data area is applicable to all bookmark types of bookmarks. An extended data area is applicable to limited bookmark types of bookmarks.
- The writeBookmarkArray() function is applicable to both of basic data areas and extended data areas to perform a writing access. The readBookmarkArray() function is also applicable to both of basic data areas and extended data areas to perform a reading access.

A1-8.4.2.5 Structure and operation of basic data area

The structure of a basic data area is specified with structure of writeBookmarkArray(). The format of structure of writeBookmarkArray() is defined in Table 8-2. A basic data element must be up to 128 bytes.

Table 8-2 Structure of Basic Data Area

	Property	ECMAType	Structure	Remarks
Basic Data Area	title	String	S:1V(up to 41Byte)	EUC-JP string (including length)
	dstURI	String	S:1V(up to 61Byte)	String (Note) (including length)
	expire	String	S:10B	YYYYMMDDHH
	registerDate	String	S:10B	YYYYMMDDHH(based on a receiver's configuration)
	bmLock	String	S:1B	a single alphanumeric character (specified through a certain API)
	bmType	String	S:3B	3 numerics
	linkMedia	String	S:1B	1 alphanumeric
	usageFlag	String	S:1B	1 alphanumeric

Note: The available characters are the characters to specify URI, as defined in Section 9.2, Volume 2, ARIB STD-B24.

A basic data area contains arguments of the writing function, writeBookmarkArray, and values specified by a receiver.

- Operation of title (title)
  - The title field contains a string to be displayed in a title column in a bookmark list.
  - The string must be up to 40 bytes (without a length data). When the byte length of a specified string exceeds 40, up to 40 bytes are stored. It is recommended that when the 41st byte is the first byte of a 2-byte character code, the character code is not stored.
  - The title property is non-omittable.
  - The structure is S:1V. Note that the length is not specified.

- Operation of link destination (dstURI)
  - The dstURI field contains up to 60 bytes (without a length data). When the byte length of a specified string exceeds 60, a receiver does not execute the specified writing access and returns NaN.
  - When no link destination URI is specified, an empty string is used.
  - The structure is S:1V. Note that the length is not specified.
- Operation of expire
  - The expire field contains a date and time at which the concerned bookmark information expires.
  - The expire field must contain 10 digits. When a date and time is not a 10-digit representation, the writing access failes.
  - When the expire field contains 10 zeros, the concerned bookmark is assumed to be valid permanently.
  - The expire property is non-omittable.
- Operation of registerDate
  - The registerDate field indicates a time provided by the receiver's internal clock , representing a time at which a writing access function is invoked.
  - When no value is obtained from a receiver's internal clock, the writing access is assumed to have failed.
- Operation of bmLock
  - The bmLock property is a flag that indicates whether or not any deletion operation of the bookmark is explicitly prohibited by an end user.

    0: When immediately after the bookmark has been recorded or after unlockBookmark() has been invoked, 0 must be contained.

    1: When a deletion of the bookmark is prohibited using lockBookmark(), 1 must be contained.
- Operation of bmType
  - It is assumed that a structure of an extended data area of a bookmark area depends on a use of the bookmark. The bmType property is used to identify a bookmark, whose structure of the extended data area may vary.
  - The bmType property is associated with a structure of an extended data in a one-to-one relationship in order to uniquely identify an extended data structure.
  - Bookmark types are divided into bookmark types commonly operated by any broadcaster (common bookmark types) and bookmark types proprietarily operated by each broadcaster (proprietary bookmark type).
  - Any common bookmark type is assumed to be used by any broadcaster and on any receiver. This implies that the system of common bookmark types does not accept future updates nor modifications.
  - Any common bookmark type of bookmark must make a structure of the corresponding extended data area available to any bookmark list. (Any bookmark recorded through a receiver's application also must make a structure of the corresponding extended data area available to any bookmark list.)
  - Table 8-3 list bookmark types.

Table 8-3 Bookmark Types

	Bookmark Types
000	Favourite channel A (only basic data area is used)
001	Favourite channel B (extended data area is also used)
002	Favourite site A (only basic data area is used)
003	Favourite site B (extended data area is also used)
004	Related link service A (only basic data area is used)
005	Related link service B (extended data area is also used)
006	Related link site A (only basic data area is used)
007	Related link site B (extended data area is also used)
008	Memorandum
009	First link site (extended data area is also used)
010 ~ 099	Reserved area I for common bookmark types
100 ~ 198	Reserved area II for common bookmark types
199	Reserved (for registering a call origination)
200 ~ 299	Reserved area I for proprietary bookmark types
300 ~ 899	Reserved area II for proprietary bookmark types
900 ~ 999	For a receiver's application

- The bmType property is non-omittable.
- A proprietary bookmark type may be defined proprietarily by a broadcaster. Any proprietary bookmark is required to be registered.
- Detailed operation for each common bookmark type is described later.

- Operation of linkMedia
  - The linkMedia property identifies a medium of link destination of the bookmark. This identification allows a content determine the media of a link destination without the need to parse nor analyze the specified URI.
  - The string available to linkMedia is any inquiry string available to mediaName of isSupportedMedia() as well as any string shown in Table 8-4.

Table 8-4 Additional Link Media Identifications Applicable to linkMedia

Link media identification	Semantics
0	No link destination exists
A	External Network 1 (Note 1)
B	External Network 2 (Note 2)

Note 1: An external network that contains BML contents applicable to basic services.
Note 2: An external network that contains BML contents applicable to extended services.

- The linkMedia field contains a media type identification of a link destination that offers related information contents.
- Any additional link media identification, a string other than any inquiry string available to mediaName of isSupportedMedia(), is required to be registered.

  - The linkMedia property is non-omittable. When writeBookmarkArray() is used for a writing access, no string other than any inquiry string available to mediaName of isSupportedMedia() nor any string value in Table 8-4 must be specified. When readBookmarkArray() obtains a string other than any inquiry string available to mediaName of isSupportedMedia() and any string value in Table 8-4, the content that has invoked the readBookmarkArray() behaves in the same way as a content has obtained “0”.

- Operation of usageFlag
  - The usageFlag property s a flag indicating whether or not a bookmark registered by a BML content is of no use when only information in a basic data area is available.

    0: The information in a basic data area for the bookmark is not available to a receiver’s application.

    1: The information in a basic data area for the bookmark is available to a receiver’s application.
  - When usageFlag is set to “0”, the title of the bookmark is not contained in a bookmark list to be presented through a receiver’s application unless the receiver can interpret any extended data area for the bookmark.
  - When writeBookmarkArray() is used for a writing access, the available strings are “0” and “1”. When readBookmarkArray() obtains a string other than “0” and “1”, the content that has invoked the readBookmarkArray() behaves in the same way as a content has obtained “0”.

A1-8.4.2.6 Structure and operation of extended data area

- An extended data area of a bookmark contains up to 192 bytes per block, despite of bmType. When the size of a block exceeds 192-byte, a receiver does not execute the specified writing access and returns NaN.
- An extended data area is configured based on extendedStructure and extendedData.
- The structure property of an extended area may be either variable-length or fixed-length.
- Detailed conventions on the structure of an extended data area, except those on the common bookmark types, are defined by a broadcaster that has registered the bookmark type.

A1-8.4.2.7 Operation of common bookmark type of bookmark

This section provides guidelines on producing a content. Whether or not an API verifies a constraint of each argument for each bookmark type described in this section depends on an implementation. When a constraint is verified, a behaviour following a detected violation depends on an implementation.

- Operation of Favourite channel A (bmType:000)
  - The Favourite channel A bookmark type is used to record URI representing a favourite service and to jump to the service by selecting the service from a bookmark list. The bookmark type is available when a link destination exists permanently.
  - No extended data area is used.

  (Operation of basic data area)

  - A service or a component may be specified as a link destination. No smaller entity, including ES, must be specified.
  - The available values to linkMedia are “1-9” representing a broadcasting media.
  - The usageFlag field must contain “1”, indicating that a basic data area is available.

- As a general rule, The expire field contains “0”.
- A content or a receiver may set bmLock to “1” that indicates the bookmark is not allowed to be deleted without explicitly prompting an end user to confirm the deletion.

- Operation of Favourite channel B (bmType:001)
  - This bookmark type is used to record URI representing a favourite service and to jump to the service by selecting the service from a bookmark list. The bookmark type is available when a link destination exists permanently. This type of bookmark is associated with an extended area, unlike Favourite channel A. A description of a link destination or other information may be recorded in an extended area.

  (Operation of basic data area)

  - See the guidelines on Favourite channel A.

  (Operation of extended data area)

Table 8-5 Extended Data Area for Favourite Channel B (bmType:001) Bookmark

Element	Information	Bytes	Structure	Value
Category ID	bookmark_category_ID	16	S:16B	
Bookmark description	bookmark_information	Up to 41	S:1V	2-byte string Including the length
Link destination channel information	channel_information	Up to 121	S:1V	2-byte string Including the length
Recording entity URI	original_network_id	2	U:2B	16bit uimsbf
Recording entity URI	transport_id	2	U:2B	16bit uimsbf
Recording entity URI	service_id	2	U:2B	16bit uimsbf
Recording entity URI	event_id	2	U:2B	16bit uimsbf
Reserved	reserved	6	S:6B	
	Total	192		

  - Operation of bookmark_category_ID
    - This ID is used to categorize bookmarks belonging to a same bookmark type.
    - See Section 8.4.2.8 for more information.
  - Operation of bookmark_information
    - This text information outlines the bookmark.
    - This field contains up to 40 bytes.
    - This field contains only 2-byte characters. No 1-byte character is used.
    - This element is omittable.
  - Operation of channel_information
    - This text information outlines a channel specified as a link destination.
    - This field contains up to 120 bytes.
    - This field contains only 2-byte characters. No 1-byte character is used.
    - This element is non-omittable.
  - Operation of original_network_id, transport_stream_id, service_id, or event_id

  - This field contains URI representing a currently received service. The URI is obtained through API.
  - This element is omittable.

- Operation of Favourite site A (bmType:002)
  - The Favourite site A bookmark type is used to record URI representing a favorite external site and to jump to the service by selecting the service from a bookmark list. The bookmark type is available when a link destination exists permanently.
  - No extended data area is used.

  (Operation of basic data area)

  - The linkMedia field contains only a string representing an external site.
  - The usageFlag field must contain “1”, indicating that a basic data area is available.
  - When bmType is set to “002”, a content or a receiver may set bmLock to “1” that indicates the bookmark is not allowed to be deleted without explicitly prompting an end user to confirm the deletion.
- Operation of Favourite site B (bmType:003)
  - This bookmark type is used to record URI representing a favourite site and to jump to the service by selecting the service from a bookmark list. The bookmark type is available when a link destination exists permanently. This type of bookmark is associated with an extended area, unlike Favourite site A. A description of a link destination or other information may be recorded in an extended area.

  (Operation of basic data area)

  - See the guidelines on Favourite site A.

  (Operation of extended data area)

Table 8-6 Extended Data Area for Favourite Site B (bmType:003) Bookmark

Element	Information	Bytes	Structure	Value
Category ID	bookmark_category_ID	16	S:16B	
Bookmark description	bookmark_information	Up to 41	S:1V	2-byte string Including the length
Link destination channel information	site_information	Up to 121	S:1V	2-byte string Including the length
Recording entity URI	original_network_id	2	U:2B	16bit uimsbf
Recording entity URI	transport_id	2	U:2B	16bit uimsbf
Recording entity URI	service_id	2	U:2B	16bit uimsbf
Recording entity URI	event_id	2	U:2B	16bit uimsbf
Reserved	reserved	6	S:6B	
	Total	192		

  - Operation of bookmark_category_ID
    - This ID is used to categorize bookmarks belonging to a same bookmark type.
    - See Section 8.4.2.8 for more information.
  - Operation of bookmark_information
    - This text information outlines the bookmark.

    - This field contains up to 40 bytes.
    - This field contains only 2-byte characters. No 1-byte character is used.
    - This element is omittable.
  - Operation of site_information
    - This text information outlines a site specified as a link destination.
    - This field contains up to 120 bytes.
    - This field contains only 2-byte characters. No 1-byte character is used.
    - This element is non-omittable.
  - Operation of original_network_id, transport_stream_id, service_id, or event_id
    - This field contains URI representing a currently received service. The URI is obtained through API.
    - This element is omittable.

- Operation of Related link service A
  - This bookmark type is used to record URI representing a service responsible for broadcasting information related an event program and to jump to the service by selecting the service from a bookmark list. The bookmark type is available when a link destination exists temporarily.
  - No extended data area is used.

  (Operation of basic data area)
  - A service or a component may be specified as a link destination. No smaller entity including ES must be specified.
  - The linkMedia field contains only a string representing a broadcasting media.
  - The usageFlag field must contain “1”, indicating that a basic data area is available.
  - The expire field must not contain the “0” value that means that the specified link destination exists permanently.
- Operation of Related link service B
  - This bookmark type is used to record URI representing a service responsible for broadcasting information related an event program and to jump to the service by selecting the service from a bookmark list. The bookmark type is available when a link destination exists temporarily.
  - This type of bookmark is associated with an extended area, unlike Related link service A. A description of a link destination or other information may be recorded in an extended area.

  (Operation of basic data area)
  - A service or a component may be specified as a link destination. No smaller entity must be specified.
  - The linkMedia field contains only a string representing a broadcasting media.
  - The usageFlag field must contain “1”, indicating that a basic data area is available.
  - The expire field must not contain the “0” value that means that the specified link destination exists permanently.

  (Operation of extended data area)

Table 8-7 Extended Data Area for Related Link Service B (bmType:005) Bookmark

Element	Information	Bytes	Structure	Value
Category ID	bookmark_category_ID	16	S:16B	
Bookmark description	bookmark_information	Up to 41	S:1V	2-byte string Including the length
Related link information	relational_information	Up to 121	S:1V	2-byte string Including the length
Recording entity URI	original_network_id	2	U:2B	16bit uimsbf
Recording entity URI	transport_id	2	U:2B	16bit uimsbf
Recording entity URI	service_id	2	U:2B	16bit uimsbf
Recording entity URI	event_id	2	U:2B	16bit uimsbf
Reserved	reserved	6	S:6B	
	Total	192		

- Operation of bookmark_category_ID
  - This ID is used to categorize bookmarks belonging to a same bookmark type.
  - See Section 8.4.2.8 for more information.
- Operation of bookmark_information
  - This text information outlines the bookmark.
  - This field contains up to 40 bytes.
  - This field contains only 2-byte characters. No 1-byte character is used.
  - This element is omittable.
- Operation of relational_information
  - This text information outlines information related to a specified link destination.
  - This field contains up to 120 bytes.
  - This field contains only 2-byte characters. No 1-byte character is used.
  - This element is non-omittable.
- Operation of original_network_id, transport_id, service_id, or event_id
  - This field contains URI representing a currently received service. The URI is obtained through API.
  - This element is omittable.

- Operation of Related link site A
  - This bookmark type is used to record URI representing an external site containing information related an event program and to jump to the site by selecting the site from a bookmark list. The bookmark type is available when a link destination exists temporarily.
  - No extended data area is used.

  (Operation of basic data area)
  - The linkMedia field contains either “A” or “B”, representing an external site.
  - The usageFlag field must contain “1”, indicating that a basic data area is available.
  - The expire field must not contain the “0” value that means that the specified link destination exists permanently.
- Operation of Related link site B

- This bookmark type is used to record URI representing an external site containing information related an event program and to jump to the site by selecting the site from a bookmark list. The bookmark type is available when a link destination exists temporarily.
- This type of bookmark is associated with an extended area, unlike Related link site A. A description of a link destination or other information may be recorded in an extended area.

(Operation of basic data area)

- The linkMedia field contains either “A” or “B”, representing an external site.
- The usageFlag field must contain “1”, indicating that a basic data area is available.
- The expire field must not contain the “0” value that means that the specified link destination exists permanently.

(Operation of extended data area)

Table 8-8 Extended Data Area for Related Link Site B (bmType:007) Bookmark

Element	Information	Bytes	Structure	Value
Category ID	bookmark_category_ID	16	S:16B	
Bookmark description	bookmark_information	Up to 41	S:1V	2-byte string Including the length
Related link information	relational_information	Up to 121	S:1V	2-byte string Including the length
Recording entity URI	original_network_id	2	U:2B	16bit uimsbf
Recording entity URI	transport_id	2	U:2B	16bit uimsbf
Recording entity URI	service_id	2	U:2B	16bit uimsbf
Recording entity URI	event_id	2	U:2B	16bit uimsbf
Reserved	Reserved	6	S:6B	
	Total	192		

- Operation of bookmark category ID
  - This ID is used to categorize bookmarks belonging to a same bookmark type.
  - See Section 8.4.2.8 for more information.
- Operation of bookmark_information
  - This text information outlines the bookmark.
  - This field contains up to 40 bytes.
  - This field contains only 2-byte characters. No 1-byte character is used.
  - This element omittable.
- Operation of relational_information
  - This text information outlines information related to a specified link destination.
  - This field contains up to 120 bytes.
  - This field contains only 2-byte characters. No 1-byte character is used.
  - This element is non-omittable.
- Operation of original_network_id, transport_id, service_id, or event_id
  - This field contains URI representing a currently received service. The URI is obtained through API.
  - This element is omittable.

- Operation of Memorandum

- This bookmark type is designed to record only information related an event program and is not applicable to recording URI.

(Operation of basic data area)

- The linkMedia field must contain “0”.
- The usageFlag field must contain “0”, indicating that using only a basic data area is prohibited.
- The expire field must not contain the “0” value that means that the specified link destination exists permanently.

(Operation of extended data area)

Table 8-9 Extended Data Area for Memorandum (bmType:008) Bookmark

Element	Information	Bytes	Structure	Value
Category ID	bookmark_category_ID	16	S:16B	
Bookmark description	bookmark_information	Up to 41	S:1V	2-byte string Including the length
Memorandum information	memorandum_information	Up to 121	S:1V	2-byte string Including the length
Recording entity URI	original_network_id	2	U:2B	16bit uimsbf
Recording entity URI	transport_id	2	U:2B	16bit uimsbf
Recording entity URI	service_id	2	U:2B	16bit uimsbf
Recording entity URI	event_id	2	U:2B	16bit uimsbf
Reserved	reserved	6	S:6B	
	Total	192		

- Operation of bookmark_information
  - This text information outlines the bookmark.
  - This field contains up to 40 bytes.
  - This field contains only 2-byte characters. No 1-byte character is used.
  - This element is omittable.
- Operation of memorandum_information
  - This text information provides additional information related to visual representations in an event program.
  - This field contains up to 120 bytes.
  - This field contains only 2-byte characters. No 1-byte character is used.
  - This element is non-omittable.
- Operation of original_network_id, transport_id, service_id, or event_id
  - This field contains URI representing a currently received service. The URI is obtained through API.
  - This element is omittable.

- Operation of First link site
  - This bookmark type is designed to transit through a BML content of a data broadcast program to an external site as a first link site.

(Operation of basic data area)

- The dstURI field must identify a component of a data broadcast program. The component assumes that a BML content that reads the first_link_url field of an extended data area in a bookmark area and enable a transition to the URL as soon as the component is presented, is always available as a broadcast content.
- The values applicable to the linkMedia field are 1 through 5, representing "broadcast".
- The usageFlag field must be “1” that means that the basic data area can be used. The expire field must not contain the “0” value that means that the specified link destination exists permanently.

(Operation of extended data area)

Table 8-10 Extended Data Area for First link site (bmType:009) Bookmark

Element	Information	Bytes	Structure	Value
Category ID	bookmark_category_ID	16	S:16B	
Bookmark description	bookmark_information	Up to 41	S:1V	2-byte string Including the length
Related link information	relational_information	Up to 61	S:1V	2-byte string Including the length
First link URL	first_link_url	Up to 61	S:1V	2-byte string Including the length
Recording entity URI	original_network_id	2	U:2B	16bit uimsbf
Recording entity URI	transport_stream_id	2	U:2B	16bit uimsbf
Recording entity URI	service_id	2	U:2B	16bit uimsbf
Recording entity URI	event_id	2	U:2B	16bit uimsbf
Reserved	reserved	5	S:5B	
	Total	192		

- Operation of bookmark category ID
  - This ID is used to categorize bookmarks belonging to a same bookmark type.
  - See Section 8.4.2.8 for more information.
- Operation of bookmark_information
  - This text information outlines the bookmark.
  - This field contains up to 40 bytes.
  - This field contains only 2-byte characters. No 1-byte character is used.
  - This element is omittable.
- Operation of relational_information
  - This text information outlines information related to a specified link destination.
  - This field contains up to 60 bytes.
  - This field contains only 2-byte characters. No 1-byte character is used.
  - This element is non-omittable.
- Operation of first_link_url
  - This field contains up to 60 bytes, not including the length. When the size of this field exceeds 60-byte, a receiver does not execute the specified writing access and returns NaN.
  - This element is non-omittable.
- Operation of original_network_id, transport_id, service_id, or event_id

- This field contains URI representing a currently received service. The URI is obtained through API.
- This element is omittable.

A1-8.4.2.8 Operation of Category ID

To operate bookmarks as efficiently as possible, this section proposes a system of Category IDs that is applicable to an extended area for a common bookmark type of bookmark.

- The upper two bytes of the 16-byte Category ID field are used based on the conventions on major classes and medium classes defined in Annex H “Genre Designation in Content Descriptor”, Volume 2, STD-B10.
- When “0xFF” is specified as a major class and a medium class, the lower 14 digits are enabled to contain a string, as defined in Table 8-10.
- When the upper three digits contain “0xFF0”, the fourth and fifth digits contain an advertiser category, as defined in Table 8-11. The sixth and following digits are not defined.
- When the upper three digits contain “0xFF1”, the fourth and fifth digits contain an store/shop category, as defined in Table 8-12. The sixth and following digits are not defined.
- When the upper three digits contain a string in the range of “0xFFA”-“0xFFE”, the fourth, fifth, sixth, and seventh digits contain a broadcaster identification (broadcaster_id for a BS/CS broadcaster, telestrial_broadcaster_id for a terrestrial broadcaster). The eighth and following digits are defined by a broadcaster.
  For a BS/CS broadcaster, the fourth and fifth digits contain a broadcaster_id string in the hexadecimal notation and the sixth and seventh digits contain “0xFF”. For a terrestrial broadcaster, the fourth, fifth, sixth, and seventh digits contain a telestrial_broadcaster_id string in the hexadecimal notation.

Table 8-11 Classification Applicable to "Others (0xFF in the upper two digits)"

1st-3rd digits	Description
0xFF0	Advertiser Classification
0xFF1	Store/Shop Classification
0xFF2	
0xFF3	
0xFF4	
0xFF5	
0xFF6	
0xFF7	
0xFF8	
0xFF9	
0xFFA	proprietary classification of BS digital broadcast broadcasters
0xFFB	proprietary classification of Broadband CS digital broadcast broadcasters
0xFFC	proprietary classification of terrestrial digital broadcast service providers
0xFFD	
0xFFE	
0xFFF	Others

Table 8-12 Advertiser Classification

1st-3rd digits	4th, 5th digits	
0xFF0	0x00	Energy, Materials, Machinary
0xFF0	0x01	Food
0xFF0	0x02	Beverages
0xFF0	0x03	Pharmaceuticals, Medical devices
0xFF0	0x04	Cosmetics, Toiletry
0xFF0	0x05	Apparel, Accessories
0xFF0	0x06	Precision instruments,
0xFF0	0x07	Home appliance, AV equipment
0xFF0	0x08	Automobiles and related devices
0xFF0	0x09	House wares, Home furnishings
0xFF0	0x0A	Hobby goods, Sports goods
0xFF0	0x0B	Real estate, House equipment
0xFF0	0x0C	Publishing
0xFF0	0x0D	Information, Communication
0xFF0	0x0E	Distributors, Retailers
0xFF0	0x0F	Finance, Insurance
0xFF0	0x10	Transportation, Leisure
0xFF0	0x11	Food services, Other services
0xFF0	0x12	Government/Municipal offices, Foundations
0xFF0	0x13	Education, Health care, Religious
0xFF0	0x14	Classified advertising, Others
0xFF0	0x15	Sustaining advertisement

Table 8-13 Store/Shop Classification

1st-3rd digits	4th, 5th digits	
0xFF1	0x00	Retailers
0xFF1	0x01	Department stores
0xFF1	0x02	Supermarkets
0xFF1	0x03	Discount stores
0xFF1	0x04	Large electrical shop
0xFF1	0x05	Apparel
0xFF1	0x06	Ladies wear
0xFF1	0x07	Menswear
0xFF1	0x08	Children wear
0xFF1	0x09	Cosmetics
0xFF1	0x0A	Bags
0xFF1	0x0B	Shoes
0xFF1	0x0C	Clocks, Watches
0xFF1	0x0D	Eyeglasses
0xFF1	0x0E	Furniture
0xFF1	0x0F	Bedclothes
0xFF1	0x10	Variety goods
0xFF1	0x11	Toys
0xFF1	0x12	Books
0xFF1	0x13	Stationery
0xFF1	0x14	Mobile phones
0xFF1	0x15	Automobiles
0xFF1	0x16	Houses
0xFF1	0x17	
0xFF1	0x18	
0xFF1	0x19	
0xFF1	0x1A	
0xFF1	0x1B	
0xFF1	0x1C	
0xFF1	0x1D	
0xFF1	0x1E	
0xFF1	0x1F	Other retailers (except food retailers)
0xFF1	0x20	Food retailers
0xFF1	0x21	Organic foods
0xFF1	0x22	Meat, Processed meat
0xFF1	0x23	Vegetables
0xFF1	0x24	Seafood
0xFF1	0x25	Bread
0xFF1	0x26	Confectionery
0xFF1	0x27	Beverages (alcohol)
0xFF1	0x28	Tobacco

1st-3rd digits	4th, 5th digits	
0xFF1	0x30	Food services
0xFF1	0x31	Japanese cuisine
0xFF1	0x32	French cuisine
0xFF1	0x33	Italian cuisine
0xFF1	0x34	Chinese cuisine
0xFF1	0x35	Indian cuisine
0xFF1	0x36	Ethnic cuisine
0xFF1	0x37	Western dishes
0xFF1	0x38	Chinese noodles
0xFF1	0x39	Japanese noodles
0xFF1	0x3A	Specialty food service (tonkatsu)
0xFF1	0x3B	Specialty food service (eel)
0xFF1	0x3C	Specialty food service (sushi)
0xFF1	0x3D	Specialty food service (tempura)
0xFF1	0x3E	Pubs, Bars
0xFF1	0x3F	Sweets
0xFF1	0x40	Fast food
0xFF1	0x41	Family restaurant
0xFF1	0x42	
0xFF1	0x43	
0xFF1	0x44	
0xFF1	0x45	
0xFF1	0x46	
0xFF1	0x47	
0xFF1	0x48	
0xFF1	0x49	
0xFF1	0x4A	
0xFF1	0x4B	
0xFF1	0x4C	
0xFF1	0x4D	
0xFF1	0x4E	
0xFF1	0x4F	Other food services
0xFF1	0x50	Broadcasters
0xFF1	0x51	Travel agencies
0xFF1	0x52	Hotels
0xFF1	0x53	Transportation
0xFF1	0x54	Telephone services (carriers)
0xFF1	0x55	Insurance
0xFF1	0x56	Securities
0xFF1	0x57	Banking
0xFF1	0x58	Real estate

0xFF1	0x29	Drugs	0xFF1	0x59	Administration
0xFF1	0x2A		0xFF1	0x5A	Schools
0xFF1	0x2B		0xFF1	0x5B	Hospitals
0xFF1	0x2C		0xFF1	0x5C	Broadcasters, Publishers
0xFF1	0x2D		0xFF1	0x5D	ISPs
0xFF1	0x2E		0xFF1	0x5E	
0xFF1	0x2F	Other food retailers	0xFF1	0x5F	Other broadcasters

A1-8.4.3 Operational Guidelines on Bookmark Service

This section provides guidelines on producing contents related to bookmark services and guidelines on a receiver's application that supports a proprietary bookmark function of the receiver.

A1-8.4.3.1 Overview of bookmark service

- A bookmark service is assumed to be operated across media types.
- A media type used for recording a bookmark may be either the same as that of a link destination or different form that of a link destination.
- A bookmark service is assumed to be triggered by a BML content. However, a bookmark list service or others may be triggered by a receiver's application.
- To avoid any conflict in operating any bookmark service, any broadcaster and manufacturer who provides a bookmark service must obey the operational guidelines in Sections 8.4.3.2 - 8.4.3.8.

A1-8.4.3.2 Relationship between bookmark service by BML content and bookmark service functions by receiver application

- A bookmark service may be triggered by either a BML content or a receiver's application. Any bookmark information recorded through a receiver's application is not accessible to any BML content. This implies that a bookmark recorded through a receiver's application may be ignored by a BML content.
- Both a BML content and a receiver's application can present a bookmark list.
- A bookmark list provided by a receiver can read out both bookmarks recorded by a BML content or bookmarks recorded by a receiver's application to present them.
- A bookmark list provided by a BML content can read out only bookmarks recorded by a BML content unless the bookmark is not accessible depending on the specified usageFlag value.
- Whether a bookmark list provided by a BML content or a bookmark list provided by a receiver's application is used to select a link destination is transparent to the link destination content.
- A receiver's application is not allowed to write nor overwrite a bookmark, except a bmLock value to be stored in a basic data area, to a bookmark area in NVRAM. A receiver's application is allowed to read from a bookmark area in NVRAM.
- For more information about the relationship between bookmark functions in a receiver's application and bookmark services triggered from a BML content, refer to Informative Explanation 5.
- The Sections 8.4.3.3 through 8.4.3.8 provides guidelines applicable to services provided by a BML content, unless otherwise stated. The functions described in Sections 8.4.3.3 through 8.4.3.8 are designed to be implemented by a BML content, unless otherwise stated.

A1-8.4.3.3 Operation of service types for bookmark list data service

- A bookmark list data service is a service specified with a service type (0xAA). The main content is a bookmark list display content.
- The service type (0xAA) is designed to allow a receiver with a bookmark list function as the receiver's application to discriminate services with bookmark lists to be represented by a BML content to prevent these services from being selected ensuring a consistency in bookmarks displayed on a bookmark list user interface. Any operational regulation on transmitting this service type of service and applicable coding schemes must be equivalent to those for a data service (0xC0).
- Whether a receiver with a bookmark list function as the receiver's application receives a 0xAA service or not depends on an implementation.
- It is allowed that a bookmark list data service (0xAA) provides a bookmark list service to enable a receiver to identify services offering a bookmark list. The result is that a receiver is allowed to be equipped with a pre-list service function as the receiver's application.

A1-8.4.3.4 Operation of bookmark record service

- To record a bookmark, the getBookmarkInfo() API must be used to obtain URI representing a free bookmark area before a writing access is executed.
- When no free bookmark area is available, an existing recorded bookmark, one of the following, is allowed to be deleted for a writing access.
  - An overdue bookmark. Provided that a presenting function informs an end user of the bookmark title and notices that the bookmark is going to be deleted because of the overdue.
  - A bookmark whose bmLock is set to ”0” and older than any bookmark with bmLock “0” specified. Provided that a presenting function informs an end user of the bookmark title that is going to be deleted and prompts the end user to explicitly accept the deletion.
- A bookmark record service provided by a BML content must be capable of verifying the media type range supported by a receiver. It must be ensured that when a bookmark whose linkMedia representing a media outside the range is specified, the bookmark is not recorded. In this case, the reason why the specified bookmark is not allowed to be recorded must be presented to the end user.
- When a bookmark record function as a receiver’s application is implemented, an area for containing bookmarks recorded by a BML content and an area for containing bookmarks recorded by the receiver’s application must be separate areas. More detailed specifications on functions of a bookmark record function as a receiver’s application and its operation depend on a receiver.
- When a bookmark title to be contained in a basic data area is longer than 40-byte, any receiver does not contain the 41st and following bytes. It is recommended that when the 41st byte is the first byte of a 2-byte character code, the character code is not stored.
- When a destination link URI to be contained in a basic data area is longer than 60-byte, any receiver makes a writing access by writeBookmarkArray() fail and NaN is returned.

A1-8.4.3.5 Operation of bookmark list service

- The guidelines in this section are applicable to a bookmark list function provided by a BML content as well as a receiver’s application.

- The guidelines in this section are applicable to a bookmark list display content in a TV service and a promotional TV service.
- When a receiver has a bookmark list service application, whether or not the receiver receive the bookmark list service that is specified with service_type=0xAA and provided by a BML content depends on the receiver.
- A bookmark list must have the following functions.
  - A function presenting bookmark titles of recorded bookmarks.
  - A function that allows an end user to select a desired bookmark and switch to a channel based on the link destination URI specified for the selected bookmark.
  - A function for controlling a deletion operation.
- When a link destination is a broadcast service, the link destination is specified as a component or a smaller entity. Note that even if the dstURI field contains an event ID or ID for smaller entity, any event ID or ID for smaller entity must be ignored without any failure. Any bookmark list, including a bookmark list generated by a receiver's application, also must support this level of selection.
- When an end user is going to select a channel through a bookmark list of a receiver’s application or a bookmark list display content, a block number containing the specified bookmark information and the time when the information is written must be saved in Greg Block 0 immediately before the end user does select and switch to the channel. This helps a link destination content to successfully update an extended area of the source bookmark.
  - The following format is used.

    Block Number: 2-byte hexadecimal numeric string
    Time: 12-byte decimal numeric string (YYMMDDHHMMSS)
    No delimiter is used.
  - The time is recorded to prevent a link destination content brought by a bookmark from reading out existing information that is not intended nor desired information. Once an end user has selected a bookmark to switch to a link destination content, the link destination content compares a time value in Greg Block 0 against a current time value obtained in the working environment to verify that Greg contains information that has been saved immediately before the switch. This explains the reason why a source content must obtain and write a time immediately before the switch to Greg Block 0.
- It is recommended that a bookmark list provided by a BML content as well as a receiver’s application has the following function.
  - When a bookmark whose bookmark type is a common bookmark type , information in an extended data area is also presented.
- A bookmark list must present any recorded bookmark expect the following.
  - As a general rule, a bookmark list provided by a receiver’s application does not present any bookmark whose usageFlag is set to “0”. However, it is recommended that a memorandum type bookmark (bmType:008) and the corresponding information saved in an extended data area are presented.
  - A bookmark list provided by a BML content does not present a bookmark whose usageFlag is set to “0” and whose bookmark type is not interpreted by a bookmark list display content. For example, it is assumed that a proprietary bookmark type that is to defined by a broadcaster is not interpreted by a bookmark list display service. A bookmark list must not contain any non-interpretable bookmark type of bookmarks.
  - A bookmark list provided by a BML contest cannot present a bookmark recorded by receiver’s application.

- A bookmark list must present the following elements.
  - Title
  - Expire date
  - Media type of a link destination
  - Prohibition of deletion
  - Description of a bookmark (saved in an extended area)
- Any overdue bookmark must be presented to an end user to notice that the bookmark is overdue. Any overdue bookmark must not accept a selection operation by an end user. It is recommended that a function informs of the end user that the bookmark is going to be deleted because of the overdue and then deletes it.
- When a link destination URI represents a component, a receiver's bookmark list function and a bookmark list content must write the component tag value and the time when the writing access occurs to Greg Block 1 immediately before the component is selected.
  - The following format is used.

    Component Tag Value: 2-byte hexadecimal numeric string
    Time: 12-byte decimal numeric string (YYMMDDHHMMSS)
    No delimiter is used.
  - The time is recorded to prevent a link destination content brought by a bookmark from reading out existing information that is not intended nor desired information. Once an end user has selected a bookmark to switch to a link destination content, the link destination content compares a time value in Greg Block 1 against a current time value obtained in the working environment to verify that Greg contains information that has been saved immediately before the switch. This explains the reason why a source content must obtain and write a time immediately before the switch to Greg Block 1.
- Function for controlling a deletion operation
  A bookmark list display content must provide the following functions for controlling NVRAM area (controlling a deletion operation) for end users.

  (1) Function for manually deleting a bookmark

  It is required that an end user can manually delete a bookmark, as intended.

  (2) Function for automatically deleting a bookmark

  This function automatically an overdue bookmark.

  (3) Function for denying a deletion operation of a bookmark

  It is required that an end user can enable and disable a flag preventing a bookmark from being deleted, as intended.
- It is recommended that when a bookmark list display function is implemented as a receiver's application, a broadcast service content can control a bookmark list display function provided by the broadcast service content to inactivate the function for the purpose of ensuring a consistency in bookmarks displayed on a bookmark list user interface. This implies that the bookmark list in the receiver are given higher priority than any bookmark list provided by any broadcast service content.

A1-8.4.3.6 Operation of pre-list service

- A pre-list is a list of the channels responsible for bookmark list services. An end user can use a pre-list to directly select a desired bookmark list service. When a receiver is not equipped with a

bookmark list display function, a pre-list acts as an auxiliary function to select a desired bookmark list service.

- A pre-list service must list the bookmark list services that are configured to be broadcast through the same media as the pre-list service is broadcast unless a broadcaster prohibit a pre-list service from displaying their bookmark list services. Whether a pre-list service whose media is different from media of a service broadcasting a currently active pre-list service is displayed in a currently active pre-list or not depends on an implementation.
- It is recommended that when a bookmark list display function is implemented as a receiver's application, a pre-list service is hidden.
- More detailed conventions on operating contents related to a pre-list are defined by a broadcaster who operates the pre-list.

A1-8.4.3.7 Operation for erasing bookmarks

The guidelines in this section are applicable to a bookmark list function provided by a BML content as well as a receiver's application.

- As a general rule, any deletion operation of a bookmark is executed only when an end user explicitly accept the deletion.

  A bookmark is allowed to be deleted in the following cases.

  (1) When an end user explicitly requests that a bookmark is deleted.

  (2) When the corresponding information to a bookmark is overdue.
  Provided that a function informs an end user of the bookmark title and notices that the bookmark is going to be deleted because of the overdue.

  (3) When a content to be deleted is presented to an end user and the user explicitly accepts the deletion.
  It is assumed that a bookmark whose bmLock is set to "0" and older than any bookmark with bmLock "0" is presented.

  A manual deletion operation by an end user must be supported by a bookmark list service as well as a receiver's application. Both a bookmark list service and a receiver's application must support a setting operation, in which bmLock is set to "1".

A1-8.4.3.8 Operation for overwriting bookmarks

- Any existing bookmark information must not be directly overwritten.
- When a piece of bookmark information is going to be written in an area that has another information, the existing information must be deleted by an end user's request before the writing access is performed.
- A piece of bookmark information saved in an extended data area may be overwritten by a broadcaster who has recorded the bookmark, provided that each value of the title, dstURI, and linkMedia arguments specified through writeBookmarkArray() is equivalent to the existing counterpart in a basic data area. The writeBookmarkArray() API compares a specified value against the existing value. When any difference is detected, the writing access is not performed. When a string longer than 40-byte is specified as title through writeBookmarkArray(), the writing access is not performed.
- Any information saved in an extended data area may be overwritten, even if the corresponding bmLock is set to "1".

- As a general rule, a receiver's application must not overwrite information saved in an extended data area.

A1-8.5 Guidelines on Other Extended Functions

A1-8.5.1 Operation of operational control functions

This section provides operational guidelines on operational control functions.

- Operation of getBrowserVersion()
    - This function is used to identify a BML browser. To verify whether or not a function is supported by a BML browser, this function is not applicable.
    - A value returned in Array[0], representing MakerID, contains a value given by a registration authority (the same value as MakerID for getIRDIR()).
    - The available values for Array[1], Array[2], and Array[3] depend on a receiver.
    - Any return value of this function must not be presented to an end user.
    - To control a content's behaviour based on information obtained by this function, a prior agreement among the related broadcasters and the related vendors of a concerned receiver is required.
- Operation of epgTune()
    - To specify a media other than that currently used for the service_ref argument, whether or not the desired media is supported by a receiver must be verified before the epgTune() function is executed. It is required that only when the desired media is supported, the function is executed. The isSupportedMedia() function is used for this verification. For a media on which executing isSupportedMedia() is not a standard operation, getBrowserSupport() must be used to verify whether or not isSupportedMedia() is allowed to be executed before isSupportedMedia() is executed.
    - Even if a service to which a content to be executed by epgTune() belongs is the same as that contained in the service_ref argument, the channel must be explicitly selected by a receiver.
    - When the epgTune() function is executed trough a content that has been played from a partial TS, the following behaviour depends on a receiver.
- Operation of epgTuneToComponent()
    - The component_ref argument contains a component containing BML documents that are selectable based on the criteria defined in Section 9.2.2, Volume 2.
    - To specify a media other than that currently used for the component_ref argument, whether or not the desired media is supported by a receiver must be verified before the epgTuneToComponent() function is executed. It is required that only when the desired media is supported, the function is executed. The isSupportedMedia() function is used for this verification.
    - Even if a service to which a content to be executed by epgTuneToComponent() belongs is the same as that contained in the component_ref argument, the channel must be explicitly selected by a receiver. Detailed conventions on the selection are defined in an operational standard regulation.
    - When the epgTuneToComponent() function is executed trough a content that has been played from a partial TS, the following behaviour depends on a receiver.
    - The description "If this object fails to present the component" in Section 7.6.1, Volume 2 means the case where PMT for the service specified with component_ref does not contain information about the specified component in spite of the fact that the service specified with

component_ref is successfully selected and presented. Otherwise, the definition of "failure to present the component" and following behaviours depend on a receiver.

- Operation of launchDocumentRestricted()
  - A BML document presented by executing the launchDocumentRestricted() function is recognized as being outside of a scope of a content group.

A1-8.5.2 Launch of external applications

A1-8.5.2.1 Operation for launching external applications

- Before The extended function for broadcasting especially for launching external application, launchExApp() is excuted, getBrowserSupport() must be used to verify whether or not launchExApp() is allowed to be executed.
  The launchExApp() function is applicable provided that "1" is returned by executing getBrowserSupport() with "ARIB", "APIGroup, and "Ctrl.ExtApp" specified for sPrivider, functionname, and additionalinfo, respectively (each argument is case-sensitive).
- The available arguments to launchExApp() are listed in Table 8-13.
- When the launchExApp() function is implemented on a receiver and has no argument listed in Table 8-13, "0" is returned by executing getBrowserSupport().
- When the launchExApp() function is implemented on a receiver and has one or more arguments listed in Table 8-13, "1" is returned by executing getBrowserSupport(). Then, the launchExApp() function is used to verify whether or not a function specified with MIME-TYPE is supported. When it is not supported, NaN is returned.
- The launchExApp() function returns "1" provided that a receiver supports the concerned external application and a string specified with URI has been successfully passed to the specified external application. This implies that "1" is returned regardless of whether or not an error occurs in the external application that has received the passed string. However, it is possible that no value is retuned if the execution of launchExApp() does not result in an execution of script depending on an implementation on a receiver.
- Behaviours of an external application depend on a receiver. Any option specified with the third argument or the fourth argument is enabled provided that the specified option is allowed to be controlled. When an external application that is not supported is specified for launchExApp(), the following behaviour depends on a receiver.
- When the second argument contains "unfixed", the fifth argument and the sixth argument are used, as defined in Table 8-14. When a receiver cannot identify the specified external application based on the strings in the fifth and sixth arguments, launchExApp() returns NaN.

Table 8-14 Types of Arguments

Argument Number (Information)	Type	Available String	Description	Remarks
First Argument (URI)	URI String	URI	URI to be passed to an external application URI starting with http://, https://, or arib-dc://is available. URI with arib-dc://"~"	Whether a BML or an external application is responsible for URI name resolution depends on an implementation.
Second Argument (MIME-TYPE)	TYPE String	MIME-TYPE(*1)	Launches an external application that supports the specified MIME-TYPE.	Behaviours following any trouble (e.g. version conflict) depends on an external application.

Argument Number (Information)	Type	Available String	Description	Remarks
	Others	"Unfixed"	Service specific to a broadcaster	The Fifth Argument is used to specify an application name
Third Argument (option for launching external application)	Start	"Start"	Regardless of the current status of an external application, it is started up.	
	Update	"Update"	Updates an external application. If not launched, it is started up.	
	Depend on Status	"Depend"	When an external application has been started up, it is updated. Otherwise, it is not launched.	
	No option	Null		
Fourth Argument (option for controlling external application)	Display	"View/" X "/" Y "/" Width "/" Height	Specifies X, Y, Width, and Height for an external application's display operation. X: position against X axis Y: position against Y axis Width: width of a displayed area Height: height of a displayed area	X. Y, Width, and Height are specified in halfwidth alphanumeric (decimal), in this order. The X, Y coordinate represents the upper left corner of a displayed area.
	Sound	"Sound/" Status	Specifies the following as sound related Status for an external application. "mute": mute Z: volume	Z: halfwidth alphanumeric (decimal), 1-10 (relative values, 10 is maximum)
	No option	Null		

*1: A media type string is used, as defined in Volume 2.

Table 8-15 Types of 5th and 6th Arguments

Argument Number (Information)	Type	Available String	Description	Remarks
Fifth Argument (Application Name)	Application Name	<original_network_id>"_"<transport_stream_id>"_"<service_id> "+" <arg1> ["+" <arg2]	An application name is a string consisting of <original_network_id>"_"<transport_streeam_id>"_"<service_id>(*1), "+", and the I character. The J character is used to identify an application in a service that supports two or more services. The "+"J part is omittable. When "WebBrowser" is used as application name, an Internet browser is specified.	The arg1/arg2 string is up to 256 bytes, in halfwidth alphanumeric.
Sixth Argument	Application URI	URI	URI representing an application	
	No URI is specified	Null		

*1: Any string for <original_network_id>, <transport_stream_id>, or <service_id> is described in the hexadecimal notation, as defined in ARIB STD-B10. However, no character (string) identifying the hexadecimal notation, including a prefix ”0x” and a postfix “h” is used.

A1-8.5.2.2 Sample operation for launching external applications

This section illustrates how the launchExApp() function provides a service, assuming that an external application is an Internet browser.

Once “1” has been returned by executing getBrowserSupport() with “ARIB”, “APIGroup, and ”Ctrl.ExtApp” specified, launchExApp() behaves as described below.

(1) When "http://aaa.bbb.ccc", "text/html", "Start", and null are specified, a browser supporting text/html is launched and http://aaa.bbb.ccc is passed as an argument. Once the argument has been successfully passed to the browser, “1” is returned. At this point, the Internet browser behaves in the same way as a browser to which http://aaa.bbb.ccc is entered.

(2) When "arib-dc://ddd", "unfixed", "update", null, "WebBrowser", and null are specified and MIME-TYPE of a module specified with “ddd” is ”image/jpeg , the corresponding still picture module is passed to the launched Internet browser. Once the module has been successfully passed to the browser, “1” is returned. If no Internet browser is launched, the following behaviour depends on a receiver.

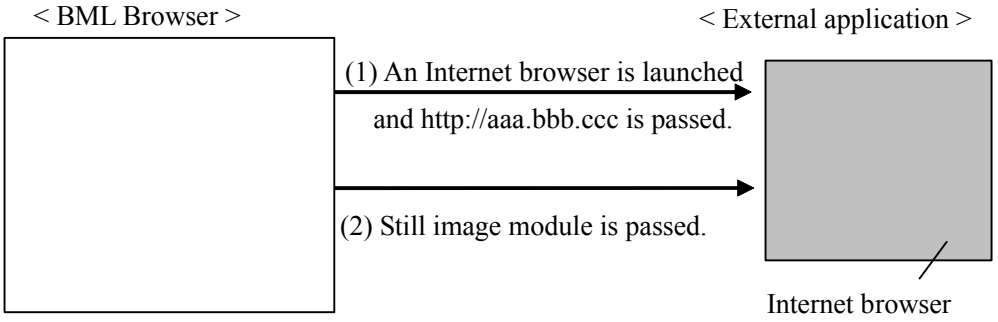


Figure 8-1 Sample of Launched External Application

A BML browser is not responsible for validity nor applicability of information manipulated by an external application, including HTML versions. If a conflict or other problems occur between a presented content and an Internet browser, the following behaviour depends on the external application. Note that how a problem arising from an external application is handled depends on the external application.

A1-8.5.3 Receiver resident application

The values applicable to the argument appName of the extended function for broadcasting getResidentAppVersion are the values applicable to the argument additionalinfo of the extended function for broadcasting getBrowserSupport() when the argument functionname of getBrowserSupport() contain "ResidentApp" .The function getResidentAppVersion returns Array[4], as shown in the following table. When getResidentAppVersion contains a value in appName that is not in the table, getResidentAppVersion returns a null character string in Array[4].

appName	**Values applicable to Array[4]**
ReservedTransmission	One of the following values, in the String type "active": A function for timer-controlled calls resides and the function is active. "inactive": A function for timer-controlled calls resides and the function is inactive. "none": No function for timer-controlled calls resides.

The combinations of applicable values to the arguments appName , showAV, and Ex_info of the extended function for broadcasting startResidentApp is shown in the following table.

appName	**showAV**	**Ex_info**[A.T.3]
HTMLBrowser	No restrictions on the value	a URI string to be sent to an HTML browser
ReservedTransmission	1 (No other value is allowed)	None
MailClient	No restrictions on the value	The fourth argument: Subject of a mail message to be sent (Corresponds to the subject header of the mail message) The fifth argument: Body of a mail message to be sent The sixth argument: Destination address of a mail message to be sent (Corresponds to the to header of the mail message) The seventh argument: Destination address of a copy of a mail message to be sent (Corresponds to the cc header of the mail message) A null character string in each argument represents no specification.
BookmarkList	1 (No other value is allowed)	None

Chapter APPENDIX 1-9 Guidelines on Contents Transmission and Namespace

A1-9.1 List of Resources in Multipart Format Modules

To contain a list of resources in a multipart format module, the list must be placed in the first resource, a resource preceding the other resources in the multipart format module.

The occurrence order of resource information in the resource list must be equivalent to that of the resources within the multipart format module.

A1-9.2 Guidelines on Resources Obtained via Transmission Channel using IP Packet

- When a request to obtain a resource, as specified in Section 9.2.13, Volume 2, by specifying the a element or arguments for launchDocument(), resources are obtained via a transmission channel using IP packets.
- When a communication via IP is required before a PPP connection has not been established, a PPP connection is automatically established to an ISP that is the specified destination specific to a receiver before resources are obtained.
- When connectPPP() is triggered from a content before a PPP connection has not been established, a call is performed according to the connecting parameters specified in connectPPPWithISPParams() to establish the PPP connection.
- When connectPPP() is triggered from a content before a PPP connection has not been established, an ISP that is the specified destination specific to a receiver is called to establish the PPP connection.
- When no packet is send nor received in a specified period of time via a PPP connection or the end user explicitly invokes a disconnecting function of a receiver, the receiver disconnects the PPP connection.
- The disconnectPPP() function is applicable to only a PPP connection that is established by connectPPP()/connectPPPWithISPParams() to disconnect the connection.
- An event, "IPConnectionTerminated occurs only when a PPP has been established by connectPPP()/connectPPPWithISPParams(), indicating that the PPP connection is terminated.

When a receiver establishes a dial-up PPP connection, at the time of connecting and disconnecting described above, it makes a call and terminates the call respectively.

A1-9.3 Namespace Available to BML documents

The following table shows the namespace available to a BML document.

Table 9-1 Types for Namespace

Type	Scheme
Module	arib-dc://<original_network_id>.<transport_stream_id>.<service_id> [;<content_id>][.<event_id>]/<component_tag>/<moduleName>
Resource (Note)	arib-dc://<original_network_id>.<transport_stream_id>.<service_id> [;<content_id>][.<event_id>]/<component_tag>/<moduleName>[/<resourceName>]
MPEG2-TS	arib://<original_network_id>.<transport_stream_id>.<service_id> [;<content_id>][.<event_id>]/<component_tag>[;<channel_id>
MPEG-I frame	arib-ic://<original_network_id>.<transport_stream_id>.<service_id> [;<content_id>][.<event_id>]/<component_tag>/<I-frame_ID>
Service	arib-ic://<original_network_id>.<transport_stream_id>.<service_id>
Event	arib://<original_network_id>.<transport_stream_id>.<service_id>.<event_id>

Type	Scheme
Stored content	arib://<original_network_id>.<transport_stream_id>.<service_id>;<content_id> or arib-dc://<original_network_id>.<transport_stream_id>.<service_id> [;<content_id>][.<event_id>]/<component_tag>/<moduleName>[/<resourceName>]
Program index	arib- node://<original_network_id>.<transport_stream_id>.[<service_id>[.<event_id>]]/ <information_provider_id>/<event_relation_id>/<node_id>
NVRAM	nvram://<filename> nvrams://<filename> nvram://bookmark/<block_number>
Component ES NPT - Generic event	arib-dc://<original_network_id>.<transport_stream_id>.<service_id> [;<content_id>][.<event_id>]/<component_tag>
Series descriptor	arib-series://<series_scope_ref>/<series_id>
Namespace used to obtain resources via IP channel	http://..., https://..., ftp://...
Storage media	arib-file://*(<directoryName>/)<fileName>
External device	peripheral://<peripheral_id>
Server-based content[A.T.4]	crid://<authority>/<data>
Segment	arib-seg:[<authority>/]<segmentId>*(;[<authority>/]<segmentId>)
Segment group	arib-seggrp:[<authority>/]<groupId>

Note: The resource type includes resources in an entity format and resources directly mapped to modules.

A1-9.3.1 Namespace available to attributes of elements or CSS properties specified in BML document

The following table shows the namespace available to attributes of elements and CSS properties.

Table 9-2 Namespace Available to BML elements

Element	Attribute/Property	Namespace													
		Module	Resource	MPEG2-TS	MPEG-I FRAME	Service	Event	Stored Content	local event index	NVRAM	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device
a element	href attribute		O *1					O *1					O *1	O *1	
beitem element	es_ref attribute										O				
	module_ref attribute	O													
body element	background-image property		O *2					O *2					O *2	O *2	
	clut property		O *3	O				O *3					O *3	O *3	

object element	data attribute		O *4	O *4	O *4			O *4					O *4	O *4	
form element	aciton attribute												O		

*1) Limited to BML documents.
*2) Limited to JPEG.
*3) Limited to CLUT.
*4) Limited to monomedia applicable to the object element.

A1-9.3.2 Extended Objects for Broadcasting and Extended Functions for Broadcasting for URI Specification

The following table summarizes the ECMAScript extended objects for broadcasting and the namespace available to them.

Table 9-3 ECMAScript Extended Objects for Broadcasting and Available Namespace

Object Name (Property)	Element for Describing URI	Namespace													
		Module	Resource	MPEG2-TS	MPEG-I FRAME	Service	Event	Stored Content	local event index	NVRAM	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device
CSVTable object (Constructor)	argument table_ref		O					O					O	O	
BinaryTable object (Constructor)	argument table_ref		O					O					O	O	
XMLDoc object (read())	argument XMLDoc_ref		O					O					O	O	
XMLDoc object (write())	argument XMLDoc_ref							O						O	

The following tables summarize the ECMAScript extended functions for broadcasting (Browser pseudo objects) and the namespace available to them.

Table 9-4 Extended Functions for Broadcasting (EPG Functions) and Available Namespace

Function	Element for Describing URI	Namespace													
		Module	Resource	MPEG2-TS	MPEG-I FRAME	Service	Event	Stored Content	local event index	NVRAM	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device

epgGetrEventStartTime()	argument event_ref						O								
epgGetrEventDuration()	argument event_ref						O								
epgTune()	argument service_ref					O									
epgTuneToComponent()	argument component_ref					O *1									
epgTuneToDocument()	argument documentName		O *2												
epgIsReserved()	argument event_ref						O								
epgReserve()	argument event_ref						O								
epgCancelReservation()	argument event_ref						O								
epgRecIsReserved()	argument event_ref						O								
epgRecReserve()	argument event_ref						O								
epgRecCancelReservation()	argument event_ref						O								

*1) No smaller entity than a component of a service is applicable.
*2) Used to specify BML documents.

Table 9-5 Extended Functions for Broadcasting (Event Group Index Functions) and Available Namespace

Function	Element for describing URI	Namespace													
		Module	Resource	MPEG2-TS	MPEG-1 FRAME	Service	Event	Stored Content	local event index	NVRAM	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device
grpIsReserved()	argument node_ref								O						
grpReserve()	argument node_ref								O						
grpCancelReservation()	argument node_ref								O						
grpRecIsReserved()	argument node_ref								O						
grpRecReserve()	argument node_ref								O						
grpRecCancelReservation()	argument node_ref								O						
grpGetNodeEventList()	argument node_ref								O						
grpGetERTNodeName()	argument node_ref								O						
grpGetERTNodeDescription()	argument node_ref								O						

epgXTune()	argument node_ref								O						

Table 9-6 Extended Functions for Broadcasting (Series Reservation Functions) and Available Namespace

Function	Element for Describing URI	Namespace													
		Module	Resource	MPEG2-TS	MPEG-I FRAME	Service	Event	Stored Content	local event index	NVRAM	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device
seriesIsReserved()	argument series_ref											O			
seriesReserve()	argument series_ref											O			
seriesCancelReservation()	argument series_ref											O			
seriesRecIsReserved()	argument series_ref											O			
seriesRecReserve()	argument series_ref											O			
seriesCancelRecReservation()	argument series_ref											O			

Table 9-7 Extended Functions for Broadcasting (Subtitle Presentation Control Functions) and Available Namespace

Function	Element for Describing URI	Namespace													
		Module	Resource	MPEG2-TS	MPEG-I FRAME	Service	Event	Stored Content	local event index	NVRAM	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device
setCCStreamReference()	argument stream_ref			O											
getCCStreamReference()	argument stream_ref			O											

Table 9-8 Extended Functions for Broadcasting (Non-volatile Memory Management Functions) and Available Namespace

Function	Element for describing URI	Namespace													
		Module	Resource	MPEG2-TS	MPEG-I FRAME	Service	Event	Stored Content	local event index	NVRAM	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device
writePersistentString()	argument filename									O *1					
writePersistentNumber()	argument filename									O *1					
writePersistentArray()	argument filename									O *1					
readPersistentString()	argument filename									O *1					
readPersistentNumber()	argument filename									O *1					
readPersistentArray()	argument filename									O *1					
copyPersistent()	argument srcUri,dstUri									O *1					
getPersistentInfoList()	argument type									O *1					
deletePersistent()	argument filename									O *1					
getFreeSpace()	argument type									O *1					

*1: Limited to nvram://.

Table 9-9 Extended Functions for Broadcasting (Functions for Controlling Access-controlled Areas) and Available Namespace

Function	Element for Describing URI	Namespace													
		Module	Resource	MPEG2-TS	MPEG-I FRAME	Service	Event	Stored Content	local event index	NVRAM	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device
setAccessINfoOfPersistentArray()	argument filename									O *1					
checkAccessINfoOfPersistentArray()	argument filename									O *1					
writePersistentArrayWithAccessCheck()	argument filename									O *1					
readePersistentArrayWithAccessCheck()	argument filename									O *1					

*1: Limited to nvram://.

Table 9-10 Extended Functions for Broadcasting (Extended APIs for StoringDirectory Management Functions) and Available Namespace

Function	Element for Describing URI	Namespace													
		Module	Resource	MPEG2-TS	MPEG-I FRAME	Service	Event	Stored Content	local event index	NVRAM	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device
saveDirAs()	argument src_path, dst_path													O	
saveDir()	argument src_path													O	
createDir()	argument path													O	
getPerentDirName()	argument path													O	
getDirName()	argument path													O	
isDirExisting()	argument path													O	

Table 9-11 Extended Functions for Broadcasting (Extended APIs for Storing-File Management Functions) and Available Namespace

Function	Element for Describing URI	Namespace													
		Module	Resource	MPEG2-TS	MPEG-I FRAME	Service	Event	Stored Content	local event index	NVRAM	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device
saveFileAs()	argument src_path													O	
	dst_path													O	
saveFile()	argument src_path													O	
getFileNames()	argument path													O	
isFileExisting()	argument path													O	

Table 9-12 Extended Functions for Broadcasting (Extended APIs for Storing-File Input/Output Functions) and Available Namespace

Function	Element for Describing URI	Namespace													
		Module	Resource	MPEG2-TS	MPEG-I FRAME	Service	Event	Stored Content	local event index	NVRAM	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device
writeArray()	argument filename													O	
readArray()	argument filename													O	

Table 9-13 Extended Functions for Broadcasting (Extended APIs for Storing-Inquiry Functions) and Available Namespace

Function	Element for Describing URI	Namespace													
		Module	Resource	MPEG2-TS	MPEG-I FRAME	Service	Event	Stored Content	local event index	NVRAM	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device
getDirInfo()	argument path													O	
getFileInfo()	argument path													O	
getCarouselInfo()	argument stream_ref							O			O				
getModuleInfo()	argument module_ref	O	O												

Table 9-14 Extended Functions for Broadcasting (Extended APIs for Storing- Data Carousel Storage Function) and Available Namespace

Function	Element for Describing URI	Namespace													
		Module	Resource	MPEG2-TS	MPEG-I FRAME	Service	Event	Stored Content	local event index	NVRAM	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device
saveCarouselAs()	argument src_component							O			O				
readArray()	dest_path													O	

saveCarousel()	argument src_component							O			O				
saveModuleAs()	argument src_module	O						O							
	dest_path													O	
saveModule()	argument src_module	O						O							
saveResouceAs()	argument src_resource		O					O							
	dest_path													O	
saveResouce()	argument src_resource		O					O							

Table 9-15 Extended Functions for Broadcasting (Interaction Channel functions) and Available Namespace

Function	Element for Describing URI	Namespace													
		Module	Resource	MPEG2-TS	MPEG-I FRAME	Service	Event	Stored Content	local event index	NVRAM	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device
saveHttpServerFileAs()	argument src_path												O		
readArray()	dest_path													O	
sendHttpServerFileAs()	argument src_path													O	
	dest_path												O		
saveFtpServerFileAs()	argument src_path												O		
	dest_path													O	
sendFtpServerFileAs()	argument src_path													O	
	dest_path												O		
transmitTextDataOverIP()	argument uri												O		
setDelayedTransmissionData()	argument uri												O		
getDelayedTransmissionStatus()	return value Array[1]												O		
getDelayedTransmissionResult()	return value Array[1]												O		

Table 9-16 Extended Functions for Broadcasting (Operational Control Functions) and Available Namespace

Function	Element for Describing URI	Namespace													
		Module	Resource	MPEG2-TS	MPEG-I FRAME	Service	Event	Stored Content	local event index	NVRAM	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device
lockModuleOnMemory()	argument event_ref	O						O							
unlockModuleOnMemory()	argument module	O						O							
lockModuleOnMemoryEx()	argument module	O						O							
unlockModuleOnMemoryEx()	argument module	O						O							
getLockedModuleInfo()	return value module name	O						O							
setCachePriority()	argument module	O						O							
getTuningLinkageSource()	return value					O									
getLinkSourceServiceStr()	return value					O									
getLinkSourceEventStr()	return value						O								
getActiveDocument()	return value		O					O					O	O	
launchDocument()	argument documentName		O					O					O	O	
launchDocumentRestricted()	argument documentName												O		
detectComponent()	argument component_ref										O				
launchExApp()	argument uriname	O	O					O					O	O	

Table 9-17 Extended Functions for Broadcasting (External Character Functions) and Available Namespace

Function	Element for Describing URI	Namespace													
		Module	Resource	MPEG2-TS	MPEG-I FRAME	Service	Event	Stored Content	local event index	NVRAM	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device
loadDRCS()	argument DRCS_ref		O *1					O *1					O *1	O *1	

*1) Limited to DRCS.

Table 9-18 Extended Functions for Broadcasting (Functions for Controlling External Devices) and Available Namespace

Function	Element for Describing URI	Namespace													
		Module	Resource	MPEG2-TS	MPEG-I FRAME	Service	Event	Stored Content	local event index	NVRAM	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device
passXMLDocToPeripheral()	argument peripheral_ref														O
	XML_ref		O					O					O	O	

Table 9-19 Extended Functions for Broadcasting (Functions for Controlling Bookmark Areas) and Available Namespace

Function	Element for Describing URI	Namespace													
		Module	Resource	MPEG2-TS	MPEG-I FRAME	Service	Event	Stored Content	local event index	NVRAM	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device
writeBookmarkArray()	argument filename									O *1					
	argument destURI		O					O					O	O	
readBookmarkArray()	argument filename									O *1					

	return value Array[1]		O					O					O	O	
deleteBookmark()	argument filename									O *1					
lockBookmark()	argument filename									O *1					
unlockBookmark()	argument filename									O *1					

*1) Limited to nvram://bookmark/

Table 9-20 Extended Functions for Broadcasting (Functions for Printing) and Available Namespace

Function	Element for Describing URI	Namespace													
		Module	Resource	MPEG2-TS	MPEG-I FRAME	Service	Event	Stored Content	local event index	NVRAM	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device
printFile()	argument resource_ref	O													
printTemplate()	argument resource_ref	O													
printUri()	argument uri												O		
saveImageToMemoryCard()	argument src_resource													O	
SaveHttpServerImageToMemoryCard()	argument src_resource													O	
saveStaticScreenToMemoryCard()	argument dst_filename													O	

Table 9-21 Extended Functions for Broadcasting (Storage Schedule Functions) and Available Namespace

Function	Element for Describing URI	Namespace													
		Module	Resource	MPEG2-TS	MPEG-I FRAME	Service	Event	Stored Content	local event index	NVRAM	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device
epgStoreReserve()	argument event_ref								O						
epgStoreCancelReservation()	argument event_ref								O						
epgStoreCheckReservation()	argument event_ref								O						
seriesStoreReserve()	argument series_ref											O			
seriesCancelStoreReservation()	argument series_ref											O			
seriesStoreCheckReservation()	argument series_ref											O			

Table 9-22 Extended Functions for Broadcasting (Storage Functions) and Available Namespace

Function	Element for Describing URI	Namespace													
		Module	Resource	MPEG2-TS	MPEG-I FRAME	Service	Event	Stored Content	local event index	NVRAM	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device
storeStart()	argument es_ref										O				
storeTerminate()	argument es_ref										O				
checkStoreStatus()	argument es_ref										O				

Table 9-23 Extended Functions for Broadcasting (License Functions) and Available Namespace

Function	Element for Describing URI	Namespace											
		Module	Resource	Service	Event	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device	Home server-based content[A.T.5]	Segment	Segment group
getLicense()	argument content_path										O		
	argument src_path							O					
getLicenseLinkInfo()	argument content_path										O		
getEncriptionkeyInfo()	argument conte_path										O		

Table 9-24 Extended Functions for Broadcasting (Home server-based Content[A.T.6] Control Functions) and Available Namespace

Function	Element for Describing URI	Namespace											
		Module	Resource	Service	Event	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device	Home server-based content[A.T.7]	Segment	Segment group
isContentStored()	argument content_path										O		
lockStoredContent()	argument content_path										O		
unlockStoredContent()	argument content_path										O		
isLockedStoredContent()	argument content_path										O		
deleteStoredContent()	argument content_path										O		
exportContent()	argument content_path										O *1		

*1) The argument contains namespace that identifies a resource in home server-based content[A.T.8].

Table 9-25 Extended Functions for Broadcasting (Playback Control Functions) and Available Namespace

Function	Element for Describing URI	Namespace											
		Module	Resource	Service	Event	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device	Home server-based content[ARIB TR-B1]	Segment	Segment group
launchContent()	argument content_path							O			O *1		
	argument ret_content_path							O			O *1		
playSegment()	argument segment_ref											O	
	argument ret_content_path										O *1		
launchDynamicDocument( )	argument src_path							O					

*1) The argument contains namespace that identifies a resource in server-based content.

Table 9-26 Extended Functions for Broadcasting (Metadata Reference Functions) and Available Namespace

Function	Element for Describing URI	Namespace											
		Module	Resource	Service	Event	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device	Home server-based content[ARIB TR-B1]	Segment	Segment group
searchMetadataOnServer()	argument uri							O					
getEntryResourceInformation()	argument content_path										O		

Table 9-27 Extended Functions for Broadcasting (Communication Functions) and Available Namespace

Function	Element for Describing URI	Namespace											
		Module	Resource	Service	Event	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device	Home server-based content[ARIB TR-B14]	Segment	Segment group
downloadContent()	argument src_path							O					
downloadResources()	argument src_path							O					
	argument resource_path							O					
	argument dest_path										O *1		
testNetwork()	argument src_path							O					
getHttpResponseHeader()	argument uri							O					
setServerInfo()	argument uri							O					
getServerInfo()	return value Array[ ][0]							O					

*1) The argument contains namespace that identifies a resource in server-based content.

Table 9-28 Extended Functions for Broadcasting and Available Namespace

Function	Element for Describing URI	Namespace											
		Module	Resource	Service	Event	Component ES/NPT, Generic Event	Series Descriptor	Namespace for Entities on IP Channels	Storage Media	External Device	Home server-based content[ARIB TR-B29]	Segment	Segment group
setTune()	argument service_ref			O									
getDirStructures()	argument path										O *1		

getTransitSource()	return value Array[1] (*2)												

*1) The argument contains namespace that identifies a resource in server-based content.

*2) Values available to the argument are defined in an operational standard regulation.

A1-9.4 Guidelines on Root Certificate

Any encrypted transmission using TLS 1.0 or SSL 3.0 requires a root certificate that authenticates a broadcaster operating a bidirectional Web server. Root certificates are grouped into the two types: generic root certificates and broadcaster-specific root certificates. A generic root certificate is stored in a receiver persistently while a broadcaster-specific certificate is valid only a specific service operated by a broadcaster independently of other services. The term "root certificate" used in this chapter refers to both a generic root certificate and a broadcaster-specific certificate. When each type has to be operated separately, the term "generic root certificate" or the term "broadcaster-specific root certificate" is used explicitly. This section explains guidelines on root certificates.

For more information on transmission of a root certificate, refer to Chapter 3. Details about encrypted transmission and format of a root certificate are defined in an operational standard regulation.

A1-9.4.1 Format of module containing root certificate

The data structure of a root certificate to be contained in a module designed for containing root certificates complies with what is defined in Section 9.1.2 "Resource-to-module mapping", Volume 2. A generic root certificate container number, a root certificate identification number, and a root certificate version number must be mapped to an appropriate resource name.

A resource name must be operated in the following format with no extension.

> [root certificate container number] + "." + [root certificate identification number] + "." + [root certificate version number]

Each of the [root certificate container number], [root certificate identification number] , and [root certificate version number] parts contains a decimal number and '.' is required between two parts.

The valid characters are limited to ’0’ through ’9’, ’-’,’_’, ’A’, ‘B’, and ‘.’ represented with ASCII code. The [root certificate container number] of a broadcaster-specific root certificate shall be equivalent to ’_A’ or ’_B’. The values available to [root certificate container number], [root certificate identification number] , and [root certificate version number] of a generic root certificate are shown in Table 9-29 This implies that the length of a resource is variable. The maximum length must be 24 bytes, which is equal to 2 bytes + '.' + 10 bytes + '.' + 10 bytes.

For example, when a root certificate container number is "0x03", a root certificate identification number is "0x0000003F", and a root certificate version number is "0x0000007D", the resource name is represented as below:

> (Example) 3.63.125

When it comes to a broadcaster-specific root certificate, the resource name is represented as below, as a root certificate identification number is "0xFFFFFFFF" and a root certificate version number is "0xFFFFFFFF（8.2.1.2）:

> (Example) _A.-1.-1. or _B.-1.-1

The maximum value for num_of_resources must be 2. The actual data of a root certificate is contained in entity-body.

Table 9-29

	Allocated bytes	Available values	
		Hexadecimal notation	Decimal notation
Root certificate container number	2	0x0000 － 0x0007	0 － 7
Root certificate identification number	4	0x00000001 － 0x7FFFFFFF	1 － 2147483647
Root certificate version number	4	0x00000001 － 0x7FFFFFFF	1 － 2147483647
	Total 10		

A1-9.4.2 Generic root certificate storage area

A receiver is assumed that it has a generic root certificate storage area to store root certificates. A receiver recognize a root certificate whose root_certificate_type in the root certificate descriptor is 0 as a generic root certificate and stores any generic root certificate obtained from a data carousel in the generic root certificate storage area persistently. A receiver refers to root certificates contained in the generic root certificate storage area to establish an encrypted transmission.

A generic root certificate storage area consists of storage areas for storing more than one root certificate with each sub storage area is indexed with a generic root certificate container number that corresponds with the series of root_certificate_id and root_certificate_version in a root certificate descriptor (See Figure 9-1). When root_certificate_id is 0xFFFFFFFF in a root certificate descriptor, a concerned module designed for containing root certificates has no root certificate that should be contained in a storage area with a corresponding generic root certificate container number.

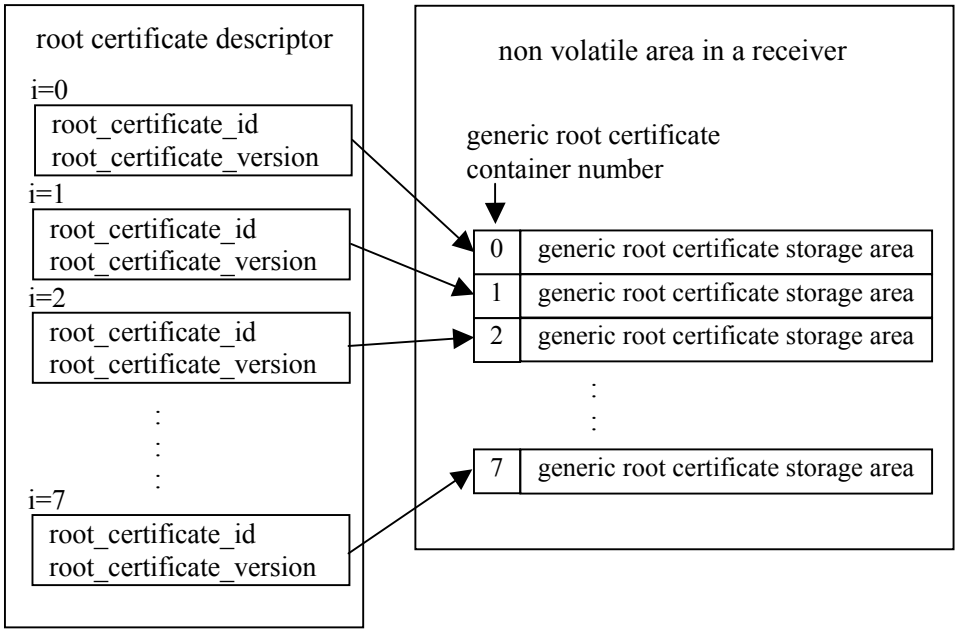


Figure 9-1 Mapping between Root Certificate Descriptor and Generic Root Certificate Storage Area

A1-9.4.3 Generic root certificate contained in receiver

When a receiver receives a module designed for containing root generic certificates, a receiver stores generic root certificate information (a generic root certificate identification number and a generic root certificate version number) contained in the module in a receiver. A generic root certificate container number is assigned to each piece of generic root certificate information (See Figure 9-2). An actual format for storing generic root certificate information depends on an implementation

How an expired root certificate in a generic root certificate storage area is operated depends on an implementation.

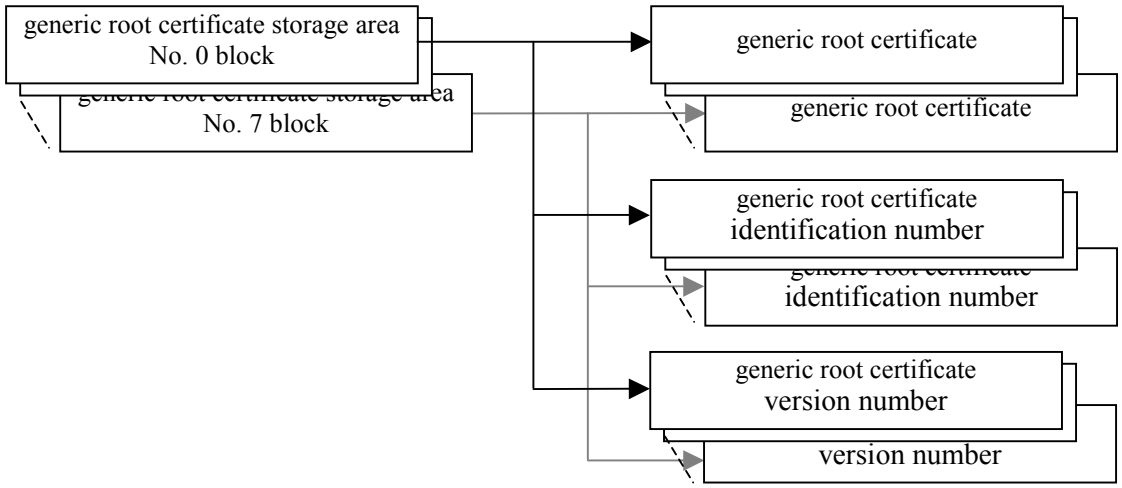


Figure 9-2 Generic Root Certificate Container Number and Related Generic Root Certificate Information

A1-9.4.4 Guidelines on receiver's behaviour for updating generic root certificate

(1) Verifying root certificate descriptor

When a service is selected, a receiver immediately verifies whether a root certificate descriptor exists or not. When a root certificate descriptor is received and root_certificate_type is "0", the receiver recognizes the root certificate as a generic root certificate. Then the receiver verifies the values of the root_certificate_id field in the root certificate descriptor. When each value is root_certificate_id the receiver recognizes that no generic root certificates to be contained exist and does no further processes for the concerned root certificates.

When a receiver detects any value other than "0xFFFFFFF" in a root_certificate_id field the receiver refers to a corresponding root_certificate_version field. When the receiver verifies that the value of the root_certificate_version field is not "0xFFFFFFFF", the receiver does further processes. When the receiver verifies that the value of the root_certificate_version field is "0xFFFFFFFF", the receiver recognizes the concerned generic root certificate as invalid and does no further processes.

(2) Detecting updated generic root certificate

A receiver compares the root_certificate_id value of a root certificate descriptor with a corresponding generic root certificate identification number in a module designed for containing root certificates, and the root_certificate_version value of the root certificate descriptor with a generic root certificate version number in the module. A receiver applies this comparing process to each root_certificate_id whose value is found to be a value other than "0xFFFFFFFF". A module designed for containing root certificates can contain up to two root certificates. This means that up to two root_certificate_id fields, each of which has a value other than "0xFFFFFFFF", can be detected.

When a receiver verifies that what a root certificate descriptor contains is same value as a corresponding piece of generic root certificate information, the receiver recognizes the concerned root certificate has been appropriately processed and does no further processes for the certificate.

When a generic root certificate is not contained in a storage area specified for the certificate or what a root certificate descriptor contains is not same value as a corresponding piece of generic root certificate information, a receiver does further processes.

(3) Verifying validity of module designed for containing root certificates

When a receiver verifies that what a root certificate descriptor contains is not same value as a corresponding piece of generic root certificate information stored in the receiver, the receiver recognizes that new root certificates arrive and receives the concerned module designed for containing root certificates. Then the receiver compares the root_certificate_id value of a root certificate descriptor with a corresponding generic root certificate container number in a module designed for containing root certificates, and the root_certificate_version value of the root certificate descriptor with a corresponding pair of a generic root certificate identification number and a generic root certificate version number in the module. When the receiver detects a difference, the receiver recognizes the concerned module as invalid and does no further processes for the module.

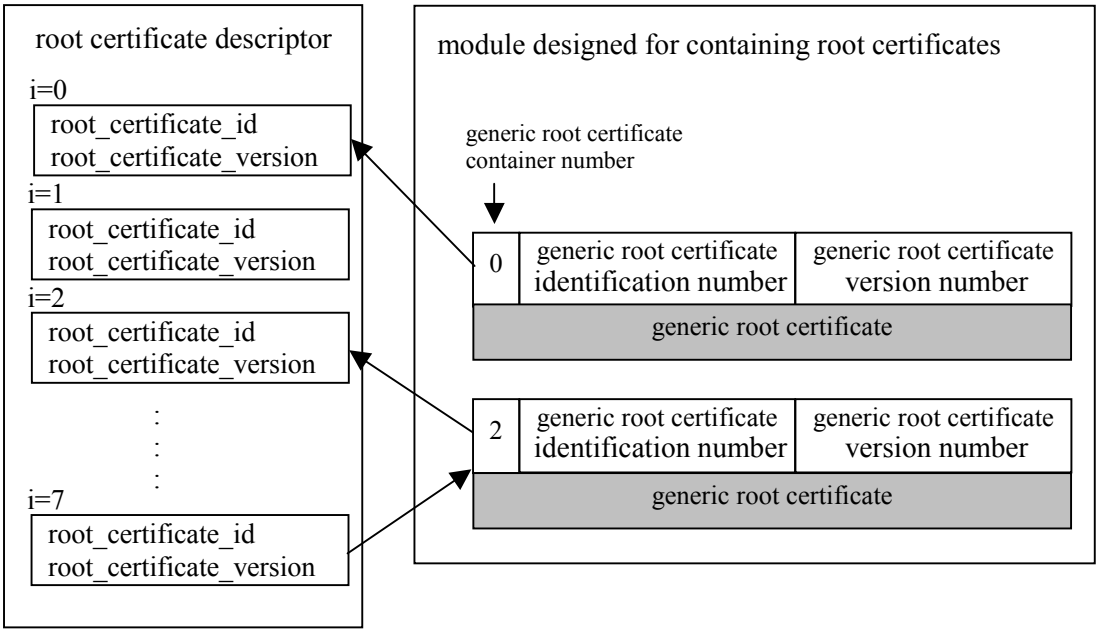


Figure 9-3 Mapping between Module Designed for Containing Root Certificates and Root Certificate Descriptor

(4) Containing generic root certificates

When a receiver verifies the validity of a module designed for containing root certificates, the receiver stores the corresponding generic root certificate in the generic root certificate storage area with a number specified in the concerned root certificate container number. The receiver also stores the generic root certificate identification number and generic root certificate version number specified in the concerned module designed for containing root certificates with assigned to a corresponding generic root certificate container number.

Chapter APPENDIX 1-10 Guidelines on Security of Content

A1-10.1 Guidelines on Security Classes

- When the security class scheme, Class A and Class B is employed in a data broadcasting service, actual definition of the two classes depends on a broadcaster. A broadcaster may specify the classes based on not only the technical reliability of a content but also the expressed and implied meaning in a content, taking into account the actual operation specific to the used media types. That is, it is assumed that a broadcaster applies the same discretion to classifying their contents as that in editing and producing their programs.
- When a Class A content invokes a file/system management function that may result in a damage to a receiver or a loss of benefit of an end user, whether or not the function is executed must be based on a decision made by the receiver or the end user. Even if a Class A content involves such a risk, any use of Class B content must not be restricted nor limited.
  It is assumed that a receiver is equipped with a function that enables an end user to explicitly select a “do it” option to invoke a risky function.
- Sample operations of Class A:
  (a) A content is produced and verified by a broadcaster and distributed by the broadcaster.
  (b) A content that exists on a Web server whose domain is specified by a broadcaster.
  (c) A content whose security is ensured by a certain certificate and which exists on a Web server.
- When a receiver has a function to obtain a content via a telecommunication line over the Internet, no content received through the telecommunication line should be handled as Class A.

A1-10.2 Guidelines on Referred Content

When a receiver is capable of obtaining a content via a telecommunication line or from a data storage service, the receiver can both present a content obtained via a telecommunication line or from a data storage service and obtain a content on the air simultaneously. This means that the following, regarding a stored content or a content obtained via a telecommunication line may result in an authorized use of an on-the-air content:

- A reference activity to a video/audio stream, a still picture, or others by the object element,
- A switching activity of presented documents by the href attribute of the a element, the launchDocument() function, or others.

Therefore, it is recommended that the following be taken into account for any reference/switch activity.

- It is recommended that any Class A content be referenced/switched in accordance with the agreement of the related broadcasters and the content providers.
- It is recommended that any Class B content be referenced/switched under the constraints defined in an operational standard regulation.

Chapter APPENDIX 1-11 Guidelines on Functions Specific to Broadcasters and Others

A1-11.1 Guidelines on Ensured Uniqueness of Identified Extended Function

The following may be defined in an operational standard regulation specific to a broadcaster, a vendor, a consortium, or others.

1. The available values and semantics of the"additionalinfo"argument of getDirInfo(), getFileInfo(), getCarouselInfo(), getModuleInfo(), and getStorageInfo()
2. The available values of the "sProvider" argument, the "functionname" argument, and the "additionalinfo" argument of getBrowserSupport()
3. Extended functions specific to a broadcaster

To avoid that an identical function name is used for different functions among two or more broadcasters, the following guidelines must be obeyed.

A1-11.1.1 Guidelines on ensured uniqueness of identified extended function

1. When a proprietary function defined by a broadcaster, a vendor, a consortium, or others is specified as the "additionalinfo" argument of getDirInfo(), getFileInfo(), getCarouselInfo(), getModuleInfo(), and getStorageInfo(), it must be specified in the following format:

   "<sProviderName>_XXXX" (XXXX is a string)

2. The "sProvider" argument of getBrowserSupport(), <sProviderName> must be specified, as described above.
3. When an extended function specific to a broadcaster is defined, <sProviderName>, it must be specified in the following format:

   X_<sProviderName>_ZZZZ() (ZZZZ is a string)

For example, when <sProviderName> is ABC and a function with a name “newfunction” is defined as an extended function, the following is specified:

X_ABC_newfunction()

To ensure the uniqueness of <sProviderName>, a name to identify a broadcaster, a vendor, a consortium, or others, more detailed conventions on the namespace are defined in an operational standard regulation. However, the “ARIB” string is reserved for <sProviderName> to identify an ARIB function, as defined in ARIB STD-B24.

The available characters to <sProviderName> are listed below.

sProviderName	=	startChar*echar
echar	=	startChar | "-" | "."
startChar	=	lowalpha | upalpha | digit | "_"
lowalpha	=	"a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z"
upalpha	=	"A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z"
digit	=	"0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" |"8" | "9"

The <sProviderName> identification is case-sensitive and at least one character long and up to 16 characters long.

A1-11.1.2 Guidelines on function beyond operational standard regulation specific to media type

To enable a content to use a proprietary function other than the functions that are defined as standard functions in an operational standard regulation specific to a media type, the getBrowserSupport() function must be used to verify whether or not the proprietary function is accepted to ensure that the proprietary function is invoked only when the function is accepted.

- Arguments of getBrowserSupport()

  Table 11-1 summarizes how getBrowserSupport() uses functionname and additionalinfo to verify an extended function when “ARIB” is specified with sProvider.

Table 11-1 Strings Available to Arguments of getBrowserSupport()

functionname	additionalinfo	How getBrowserSupport() works
BMLversion	"major version number.minor version number"	When a browser can play a BML document whose version is specified with additionalinfo, “1” is returned. When addtionalinfo is omitted, “1.0” is assumed.
BXMLversion	"major version number.minor version numbe" - When additionalinfo is omitted, “1.0” is assumed.	When a browser can present an XML document according to the usage of B-XML whose version is specified with additionalinfo, “1” is returned.
MediaDecoder	"scheme name", "media type", See Annex C, Volume 2 for more information about scheme names and media types. For example, for a MPEG-2 I frame transmitted in a still picture carousel, "arib-dc" as a scheme name and "image/X-arib-mpeg2-I" as a media type are specified. When "media type" is "audio/X-arib-mpeg2-aac", "sampling frequency (in Hz) is specified as a third additionalinfo. When this additionalinfo is omitted, "48000" is assumed.	When a browser has a function to decode a monomedia identified with the specified media type and scheme name, “1” is returned. When "media type" is "audio/X-arib-mpeg2-aac", "1" is returned as long as the specified sampling frequency is supported.
CharacterEncoding	One of the following: "UTF-8" "UTF-16" "EUC-JP" "Shift-JIS" "JIS8TEXT"	When a browser can accept a character encoding specified with additionalinfo, “1” is returned.
OSDResolution	One of the following: "1920x1080" "1280x720" "960x540" "720x480"	When a browser can present data in a text graphics plane whose resolution is specified with additionalinfo, “1” is returned.

functionname	additionalinfo	How getBrowserSupport() works
APIGroup	See Annex C, Volume 2 for more information about scheme names and media types. For example, for a MPEG-2 I frame transmitted in a still picture carousel, "arib-dc" as ascheme name and "image/X-arib-mpeg2-I" as a media type are specified	When a browser can execute any function of the group specified with additionalinfo, “1” is returned.
TransmissionProtocol	"application", "HTTP" [, HTTPversion]	When a browser has an interactive communication function that uses HTTP whose version is specified with additionalinfo via a preferred line connection for the browser, “1” is returned. When a HTTP version number is omitted, “1.0” is assumed.
	"application","FTP"	When a browser has an interactive communication function that uses FTP, “1” is returned.
	"application", "TLS" [, TLSversion]	When a browser has a secure interactive communication function that uses TLS whose version is specified with additionalinfo via a preferred line connection for the browser, “1” is returned. When a TLS version number is omitted, “1.0” is assumed
	"datalink", "BASIC"	When a browser has an interactive communication function in a code-independent mode of a BASIC procedure (JIS X5002), “1” is returned.
	"physical", "basic"	When a browser has an interactive communication function that uses V.22bis or later via a preferred line connection for the browser, “1” is returned.
Storage	"cachesize", size	When a browser has a content memory whose size is specified with cachesize, “1” is returned. Note: Specify the size in 1024-byte units (for 2 MB, “2048”).
ResidentBookmark		When a receiver has a proprietary function for displaying a bookmark list, “1” is returned.
BookmarkButton		When a receiver has a button to operate bookmarks, “1” is returned.
KanaInput		When a browser supports a Japanese character input function, "1" is returned.
AudioFile	filesize	When an audio file whose size is equal or less than the size (in bytes) specified with filesize, "1" is returned.
ResidentApp	"HTMLBrowser"	When a receiver has an HTML browser as a receiver resident application, "1" is returned.

functionname	additionalinfo	How getBrowserSupport() works
	“ReservedTransmission”	When a receiver has a receiver resident application for timer-controlled calls, "1" is returned.
	“MailClient”	When a receiver has a receiver resident application for sending and receiving mail messages, "1" is returned.
	"Bookmark"	When a receiver has a function for listing bookmarks as a receiver resident application, "1" is returned.

When MediaDecoder of the functionname of getBrowserSupport() is used, getBrowserSupport() verifies whether or not a browser has a function for decoding/presenting an external file whose media type is specified with a “scheme name”/”media type” set contained in MediaDecoder.

- APIGroup

When functionname is set to APIGroup in getBrowserSupport(), a string specified with additional info represents an extended function group to be specified.

Table 11-2 Extended Functions for Broadcasting, Classes and Extended Functions Group Specification

API	Extended function group (additionalinfo argument)
Class	
BinaryTable	Class.BinaryTable
CSVTable	Class.CSVTable
XMLDoc	Class.XMLDoc
EPG functions (7.6.1)	
epgGetEventStartTime()	EPG.Basic
epgGetEventDuration()	EPG.Basic
epgTune()	EPG.Basic
epgTuneToComponent()	EPG.Basic2
epgTuneToDocument()	EPG.Ext
epgIsReserved()	EPG.Basic
epgReserve()	EPG.Basic
epgCancelReservation()	EPG.Basic
epgRecIsReserved()	EPG.Basic
epgRecReserve()	EPG.Basic
epgCancelRecReservation()	EPG.Basic
Event group index functions (7.6.2)	
grpIsReserved()	EPG.Group
grpReserve()	EPG.Group
grpCancelReservation()	EPG.Group
grpRecIsReserved()	EPG.Group
grpRecReserve()	EPG.Group
grpRecCancelReservation()	EPG.Group

grpGetNodeEventList()	EPG.Group
grpGetERTNodeName()	EPG.Group
grpGetERTNodeDescription()	EPG.Group
epgXTune()	EPG.Group
Series reservation functions (7.6.3)	
seriesIsReserved()	EPG.Series
seriesReserve()	EPG.Series
seriesCancelReservation()	EPG.Series
seriesRecIsReserved()	EPG.Series
seriesRecReserve()	EPG.Series
seriesRecCancelReservation()	EPG.Series
Subtitle presentation control functions (7.6.4)	
setCCStreamReference()	CC.Stream
getCCStreamReference()	CC.Stream
setCCDisplayStatus()	CC.Control
getCCDisplayStatus()	CC.Control
getCCLanguageStatus()	CC.Control
Non-volatile memory functions (7.6.5)	
Non-volatile memory functions - Functions for controlling non-access-controlled areas(7.6.5.1)	
writePersistentString()	Persistent.Ext
writePersistentNumber()	Persistent.Ext
writePersistentArray()	Persistent.Basic
readPersistentString()	Persistent.Ext
readPersistentNumber()	Persistent.Ext
readPersistentArray()	Persistent.Basic
copyPersistent()	Persistent.Ext
getPersistentInfoList()	Persistent.Ext
deletePersistent()	Persistent.Ext
getFreeSpace()	Persistent.Ext
Non-volatile memory functions - Functions for controlling access-controlled areas(7.6.5.2)	
setAccessInfoOfPersistentArray ()	Persistent.MediaSupport
checkAccessInfoOfPersistentArray ()	Persistent.MediaSupport
writePersistentArrayWithAccessCheck()	Persistent.MediaSupport
readPersistentArrayWithAccessCheck()	Persistent.MediaSupport
Extended APIs for Storing (7.6.6)	
Extended APIs for Storing - Directory Management Functions (7.6.6.1)	
saveDirAs()	Storage.Dir.Dest
saveDir()	Storage.Dir
createDir()	Storage.Dir
getParentDirName()	Storage.Dir
getDirNames()	Storage.Dir.Ext
isDirExisting()	Storage.Dir
Extended APIs for Storing - File Management Functions (7.6.6.2)	

saveFileAs()	Storage.File.Dest
saveFile()	Storage.File
getFileNames()	Storage.File.Ext
isFileExisting()	Storage.File
Extended APIs for Storing - File Input/Output Functions (7.6.6.3)	
writeArray()	Storage.IO
readArray()	Storage.IO
Extended APIs for Storing - Inquiry Functions (7.6.6.4)	
getDirInfo()	Storage.Dir
getFileInfo()	Storage.File
getContentSource()	Storage.Basic
getStorageInfo()	Storage.Basic
getCarouselInfo()	Storage.Carousel
getModuleInfo()	Storage.Module
Extended APIs for Storing - Data Carousel Storage Function (7.6.6.5)	
saveCarouselAs()	Storage.Carousel.Ext
saveCarousel()	Storage.Carousel
saveModuleAs()	Storage.Module.Ext
saveModule()	Storage. Module
saveResourceAs()	Storage.Resource.Ext
saveResource()	Storage.Resource
Interaction channel functions (7.6.7)	
Interaction channel functions - Communication Functions assuming simple protocols including BASIC procedures (7.6.7.1)	
connect()	Com.BASIC.Basic
disconnect()	Com. BASIC.Basic
sendBinaryData()	Com.BASIC.Ext
receiveBinaryData()	Com.BASIC.Ext
sendTextData()	Com. BASIC.Basic
receiveTextData()	Com. BASIC.Basic
Interaction channel functions - Delayed call functions assuming simple protocols including BASIC procedures (7.6.7.2)	
registerTransmission()	Com. BASIC.Delay
registerTransmissionStatus()	Com. BASIC.Delay
getTransmissionStatus()	Com. BASIC.Delay
setDelayedTransmissionDataOverBASIC()	Com. BASIC. Delayed
Interaction channel functions - Communication functions using the mass calls reception service (7.6.7.3)	
vote()	Com. BASIC.Vote
Interaction channel functions - Functions for encrypted communication using CAS (7.6.7.4)	
startCASEncryption()	Com. BASIC.CAS
endCASEncryption()	Com. BASIC.CAS
transmitWithCASEncryption()	Com. BASIC.CAS

Interaction channel functions - Functions for communication with public key encryption not using CAS (7.6.7.5)	
setEncryptionKey()	Com. BASIC.Enc
beginEncryption()	Com. BASIC.Enc
endEncryption()	Com.BASIC.Enc
Interaction channel functions - Communication functions assuming TCP/IP (7.6.7.6)	
setISPParams()	Com.IP.Params
getISPParams()	Com.IP.Params
connectPPP()	Com.IP.Connect
connectPPPWithISPParams()	Com.IP.Connect.Ext
disconnectPPP()	Com.IP.Connect
getConnectionType()	Com.IP.GetType
isIPConnected()	Com.IP
saveHttpServerFileAs()	Com.IP.Http.Ext
saveHttpServerFile()	Com.IP.Http
sendHttpServerFileAs()	Com.IP.Http
saveFtpServerFileAs()	Com.IP.Ftp.Ext
saveFtpServerFile()	Com.IP.Ftp
sendFtpServerFileAs()	Com.IP.Ftp
sendTextMail()	Com.IP.Sendmail
trasmitTextDataOverIP	Com.IP.Transmit
setDelayedTransmissionData	Com.IP.Delayed
sendMIMEMail()	Com.IP.Sendmail
setCacheResourceOverIP()	Com.IP.SetCache
Interaction channel functions - Delayed call functions applicable to BASIC procedures and IP connections (7.6.7.7)	
getDelayedTransmissionStatus	Com.Common.Delayed
getDelayedTransmissionResult	Com.Common.Delayed
Interaction channel functions - Function for obtaining line connection status (7.6.7.8)	
getPrefixNumber()	Com.Line.Prefix
Interaction channel functions - Root-certificate-related functions for encrypted transmission (7.6.7.9)	
isRootCertificateExisting()	Com.Certificate
getRootCertificateInfo()	Com.Certificate
Operational control functions (7.6.8)	
reloadActiveDocument()	Ctrl.Basic
getNPT()	Ctrl.NPT
getProgramRelativeTime()	Ctrl.Time
isBeingBroadcast()	Ctrl.Basic
lockExecution()	Ctrl.Exec
unlockExecution()	Ctrl.Exec
lockModuleOnMemory()	Ctrl.Cache
unlockModuleOnMemory()	Ctrl.Cache
setCachePriority()	Ctrl.Cache

getTuningLinkageSource()	Ctrl.Link
getTuningLinkageType()	Ctrl.Link
getLinkSourceServiceStr()	Ctrl.PgmHyperlink
getLinkSourceEventStr()	Ctrl.PgmHyperlink
getIRDID()	Ctrl.Basic
getBrowserVersion()	Ctrl.Version
getProgramID()	Ctrl.Basic
getActiveDocument()	Ctrl.Basic
lockScreen()	Ctrl.Screen
unlockScreen()	Ctrl.Screen
getBrowserSupport()	Ctrl.Basic
launchDocument()	Ctrl.Basic
launchDocumentRestricted()	Ctrl.Com
quitDocument()	Ctrl.Quit
launchExApp()	Ctrl.ExtApp
getFreeContentsMemory()	Ctrl.Cache.Ext
isSupportedMedia()	Ctrl.Media
detectComponent()	Ctrl.Basic2
lockModuleOnMemoryEx()	Ctrl.Cache2
unlockModuleOnMemoryEx()	Ctrl.Cache2
unlockAllModulesOnMemory ()	Ctrl.Cache2
getLockedModuleInfo()	Ctrl.Cache2
setFullDataDisplayArea()	Ctrl.MobileDisplay
getDataDisplayAreaSize()	Ctrl.MobileDisplay
getResidentAppVersion	Ctrl.AppVersion
startResidentApp	Ctrl.s[A.T.13]tartResidentApp
Receiver sound control (7.6.9)	
playRomSound()	RomSound.Basic
Timer functions (7.6.10)	
sleep()	Timer.Basic
setTimeout()	Timer.Ext
setInterval()	Timer.Basic
clearTimer()	Timer.Basic
pauseTimer()	Timer.Basic
resumeTimer()	Timer.Basic
setCurrentDateMode()	Timer.DateMode
External character functions (7.6.11)	
loadDRCS()	Misc.DRCS
unloadDRCS()	Misc.DRCS.unload
Functions for controlling external devices(7.6.12)	
enumPeripherals()	Misc.Peripheral
passXMLDocToPeripheral	Misc.Peripheral.pass

getArrayFromPeripheral()	Misc.Peripheral.Array
Functions for controlling bookmark areas(7.6.13)	
writeBookmarkArray()	Bookmark.Basic
readBookmarkArray()	Bookmark.Basic
deleteBookmark()	Bookmark.Extended
lockBookmark()	Bookmark.Extended
unlockBookmark	Bookmark.Extended
getBookmarkInfo()	Bookmark.Basic
startResidentBookmarkList()	Bookmark.Resident
Other functions (7.6.14)	
random()	Misc.Basic
subDate()	Misc.Basic
addDate()	Misc.Basic
formatNumber()	Misc.Basic
Ureg pseudo object properties (7.6.15)	
Ureg[]	Misc.Ureg
Greg pseudo object properties (7.6.16)	
Greg[]	Misc.Greg
Functions for printing (7.6.17)	
getPrinterStatus()	Print.Basic
printFile()	Print.Basic
printTemplate()	Print.Basic
printUri()	Print.Basic
printStaticScreen()	Print.Basic
saveImageToMemoryCard()	Print.MemoryCard
saveStaticScreenToMemoryCard()	Print.MemoryCard
saveHttpServerImageToMemoryCard()	Print.MemoryCard

A1-11.2 Guidelines on Uniqueness of Namespace for Pre-downloaded Data Services

A root directory in the namespace for stored data services defined in Section 9.2.15.3, Volume 2 is associated with <rootName> of a special two-tire directory specified with the StoreRoot descriptor in DII. To ensure the uniqueness in the namespace, more detailed specifications of <rootName> are defined in an operational standard regulation.

Chapter APPENDIX 1-12 Guidelines on Functions for printing

A1-12.1 Reference model for functions for printing

A reference model for functions for printing is illustrated in Figure 12-1.

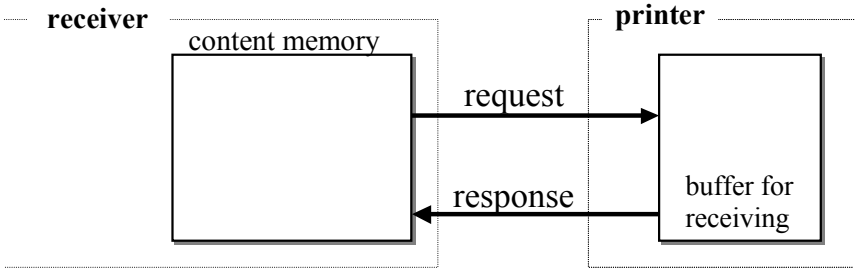


Figure 12-1 Reference Model

An API for printing must be responsible for data transmission from a receiver to a printer. This type of data transmission is a client/server model, which consists of a receiver as a client and a printer as a server. In this model, a single API is mapped to a single logical request/response transaction. And no data structure analysis of files for printing occurs in a receiver. Interpreting a data structure of a file for printing and rasterizing it is responsible for a printer. Any printing process is actually asynchronous to an API execution.

The following subsections outline how an API is operated in a separate situation.

A1-12.1.1 Transmitting file for printing obtained from broadcast stream to printer

1) Locking module

   A file for printing consists of one or more modules. Any module which is part of a file for printing must be locked in a content memory before executing an API for printing.

2) Transmitting a file for printing

   Executing an API for printing triggers transmitting a file for printing from a receiver to a printer.

3) In this situation, specified modules are transmitted to a printer when the concerned API for printing is executed. The result is a single printing job model. No queues or other methods for processing two or more printing jobs are managed through an API.

When an API is executed, the sequence of operations arises as illustrated in Figure 12-2.

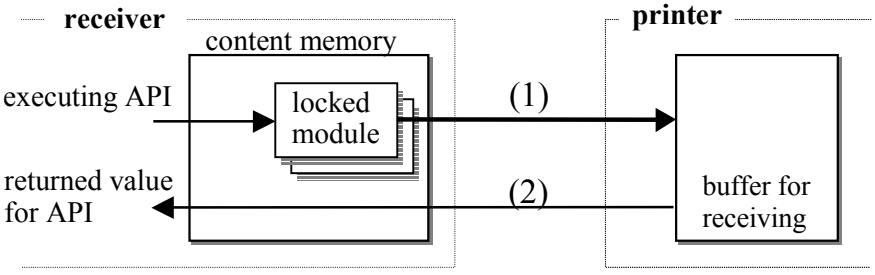


Figure 12-2Sequence of Operations for Printing #1

(1) Executing an API for printing triggers transmitting modules to be printed from a content memory to a buffer for receiving in a printer.

(2) A receiver determines a return value of the API by interpreting a response from a printer.

A1-12.1.2 Transmitting no file for printing from receiver to printer

When no files for printing are transmitted, no modules are required to be locked. When an API is executed in this situation, the sequence of operations arises as illustrated in Figure 12-3.

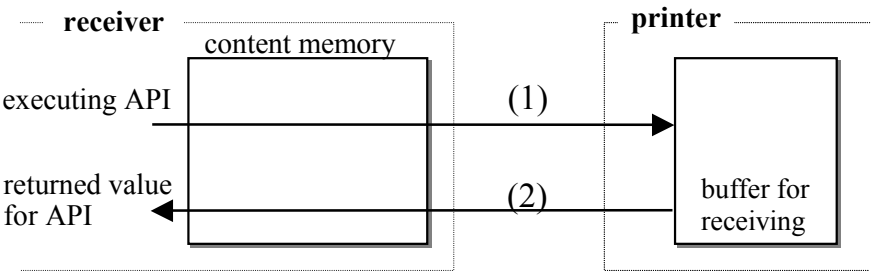


Figure 12-3Sequence of Operations for Printing #2

(1) Executing an API for printing triggers transmitting information for operation control to a printer.

(2) A receiver determines a return value of the API by interpreting a response from a printer.

A1-12.1.3 Capturing and transmitting image

To print an image composed of still picture planes and text and graphics planes, do the following steps.

1 Use Encoder to compose still picture planes and text and graphics planes into an image. Then convert the image into an XHTML format file to transmit it to a printer.

2 A receiver determines a return value of the API by interpreting a response from a printer.

A1-12.2 Operational Sequence Model

An operational sequence for printing consists of the following two phases. In Figure 12-4, a horizontal axis is a flow of data and a vertical axis is a time scale.

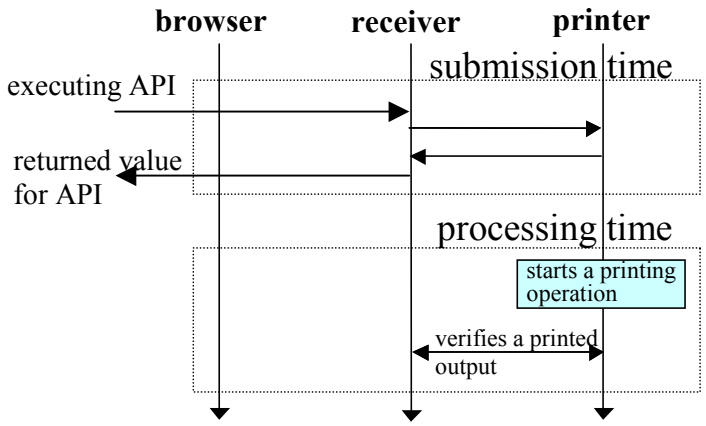


Figure 12-4Two Phases for Printing

1) Submission time

Executing an API triggers transmitting a request for printing from a receiver to a printer. The printer interprets the request and immediately responds to the receiver by bringing back what the printer can determine at the point of time. The browser determines a return value for the API based on the internal status of the receiver and the response from the printer.

For example, when an API for requesting a printing job is executed, a period as "submission time" lasts until a response which shows whether the request is accepted or not is returned. When an API for inquiring about a printer status is executed, a period as "submission time" lasts until a response which shows the status is returned.

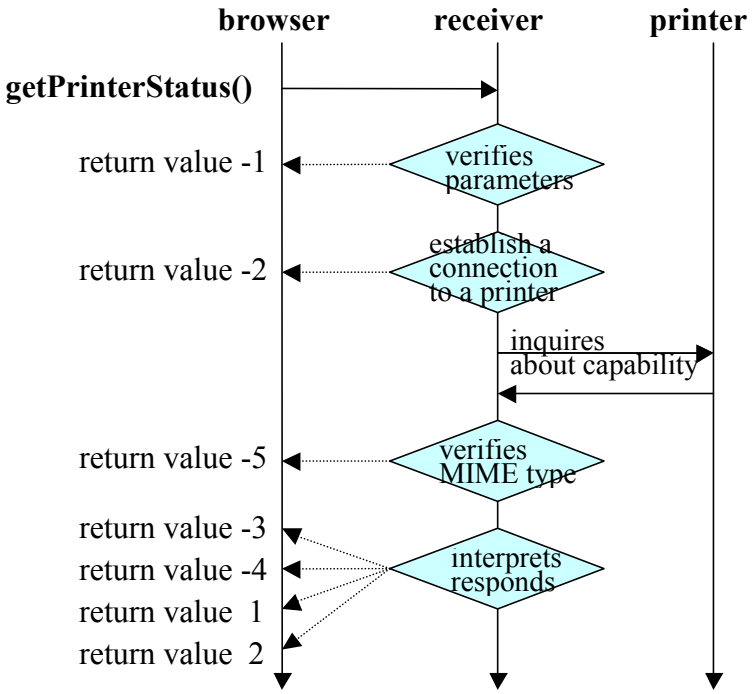
2) Processing time

In the processing time phase, actual printing operation arises. During the phase, no operations are controlled by the concerned API for printing, a receiver's application is responsible for obtaining and reacting to the printed output. Paper jamming, no ink left, failure to obtain URIs and other troubles that can occur once an API for printing has been executed are processed in the "processing time" phase.

A1-12.3 Operation of Functions for printing and Sample Behaviour

A1-12.3.1 Operation of getPrinterStatus() and Sample Behaviour

The following illustrates how the getPrinterStatus() extended function for broadcasting behaves.

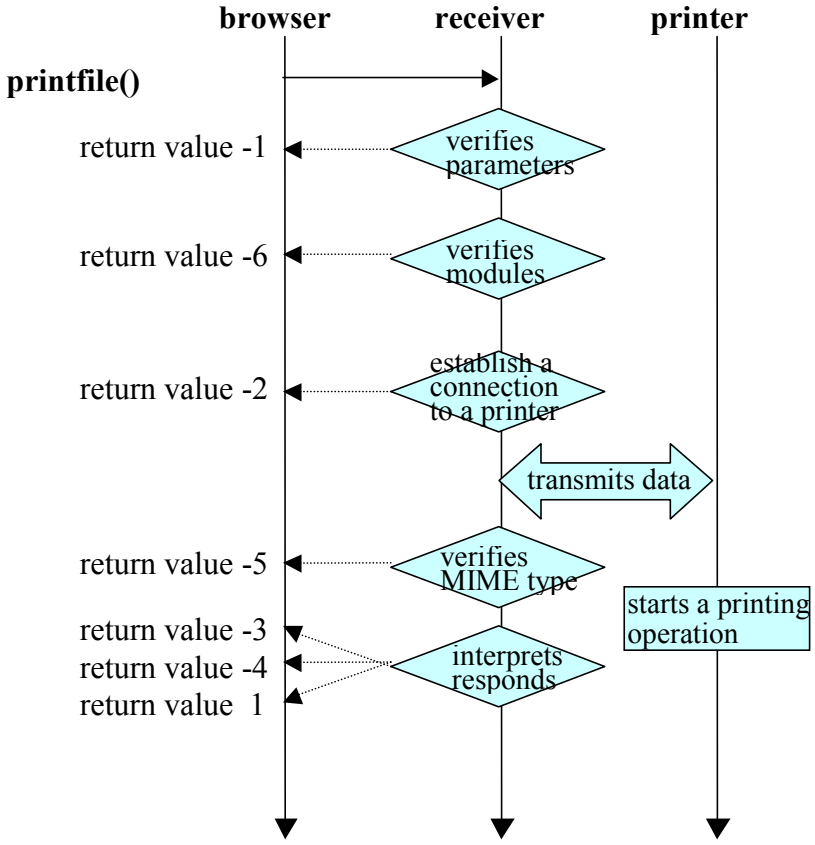


When the function has an argument specifying a MIME type, the function not only inquires about a printer's status, but also verifies whether the printer supports the specified MIME type or not. When the function detects that the specified MIME type is not supported by the printer after a connection to the printer was established, the function must return an error to indicate that the MIME type is not supported regardless of the current printer's status.

A1-12.3.2 Operation of printFile() and Sample Behaviour

Any module to be printed must be locked before this API is executed.

The following illustrates how the printFile() extended function for broadcasting behaves.



A1-12.3.3 Operation of printTemplate() and Sample Behaviour

- Any module to be printed must be locked before this API is executed.
- A replacing operation of this API may directly edit some modules locked in a content memory. This means that before this API is used to replace part or the whole of a stored file with an alternate piece of data or file, the concerned modules must be unlocked and locked again.
- Any property name used for keyword must comply with the naming conventions defined in 4.3 Section, Appendix 1, Volume 2, B24.
- The following is a sample use of printTemplate(). It is assumed that a template file is weather.html and a BML document with scripts is weather.bml. In this sample, keywords are set in a template file, an associative array which has key strings to be replaced and value strings are generated by scripts in the BML document, and the printTemplate() is executed by specifying the associative array as an argument. Executing printTemplate() triggers transmitting an output file whose content has been replaced according to the argument to a printer.

Template File [weather.html]	Script [weather.bml]
... weather in &place; - in the morning : - in the afternoon : 	var kword = new Array(); kword[”place”] = ” Tokyo”; kword[”morning”] = ”fine.jpg”; kword[”afternoon”] = ”rain.jpg”; ... browser.printTemplate(”/50/0300/weather.html”,

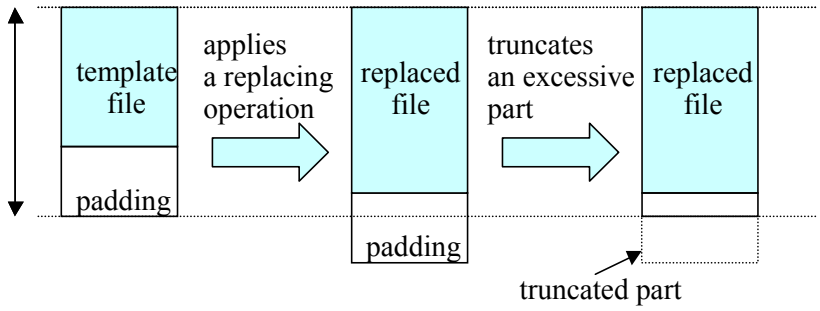
 ...	kword, "/50/0300");

Figure 12-5 Sample Usage of Template File and Script for printTemplate()

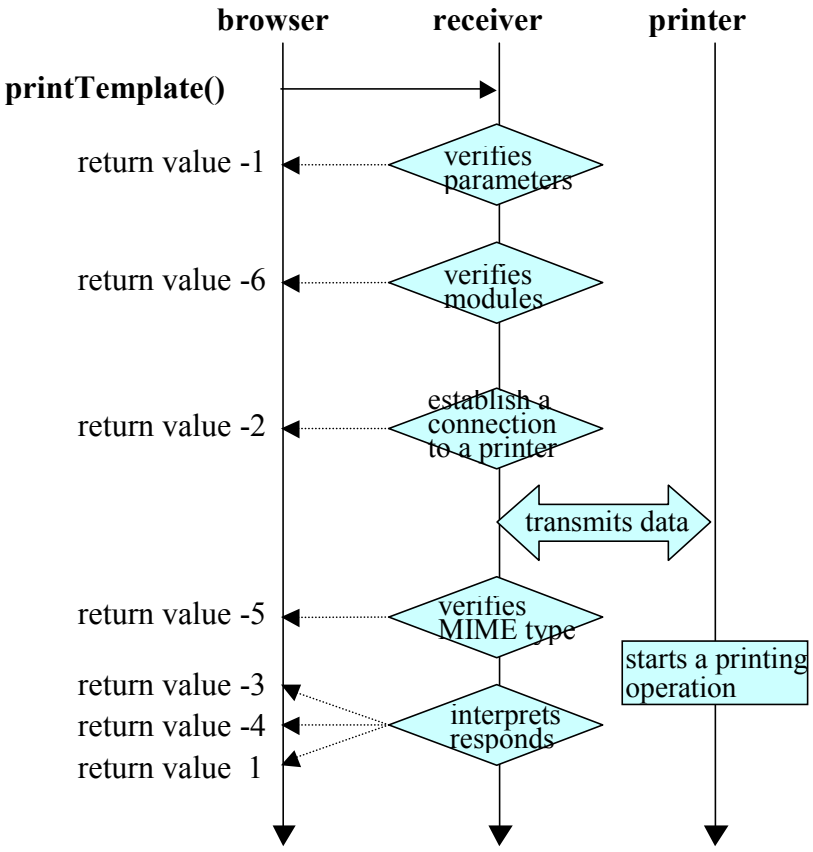
Output file[weather.html]
... weather in Tokyo   - in the morning :   - in the afternoon :   ...

Figure 12-6 Sample File with Replaced Content

It is allowed that an implementation truncates an excessive part of a replaced file whose size exceeds the size of the original template file. This requires that any template file for printing contain enough space for pad characters (space characters) at the end of the file.

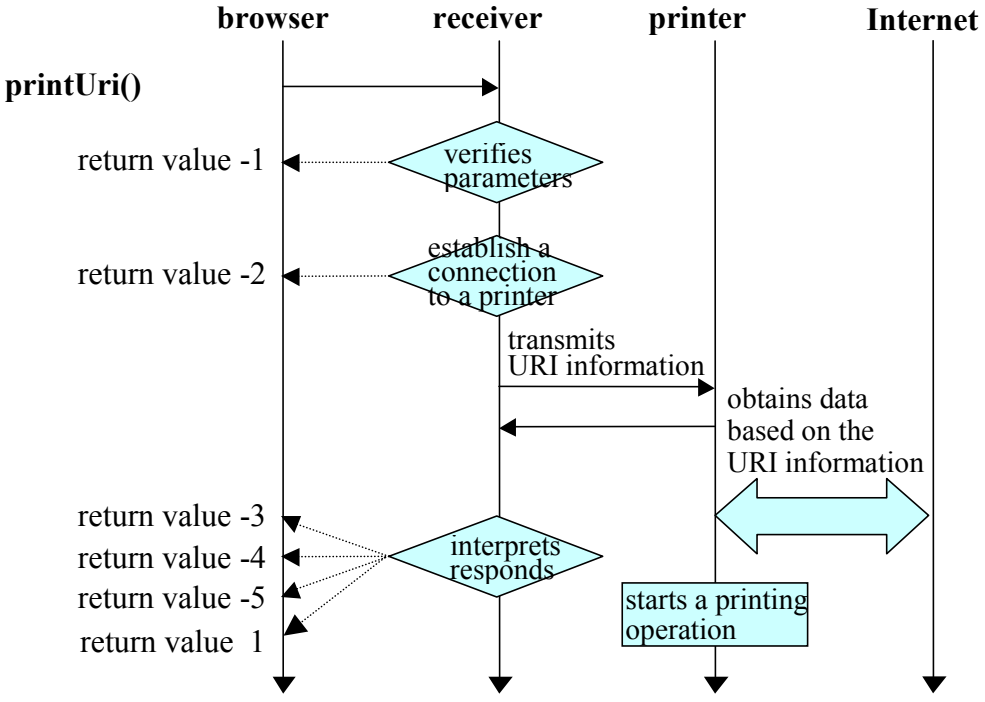


The following illustrates how the printTemplate() extended function for broadcasting behaves.



A1-12.3.4 Operation of printUri() and Sample Behaviour

The following illustrates how the printUri () extended function for broadcasting behaves.



A1-12.3.5 Operation of printStaticScreen() and Sample Behaviour

It is assumed that a printer does a printing job based on a printing layout pattern specified with the pattern argument. However, when an end user explicitly specify another printing layout pattern by using a setting function in a receiver or a printer, the end user's selected pattern can be processed prior to the pattern specified with the pattern argument. No alpha blending between planes is required. No video planes are composed.

A1-12.3.6 Operation of saveImageToMemoryCard(), saveStaticScreenToMemoryCard(), and saveHttpServerImageToMemoryCard()

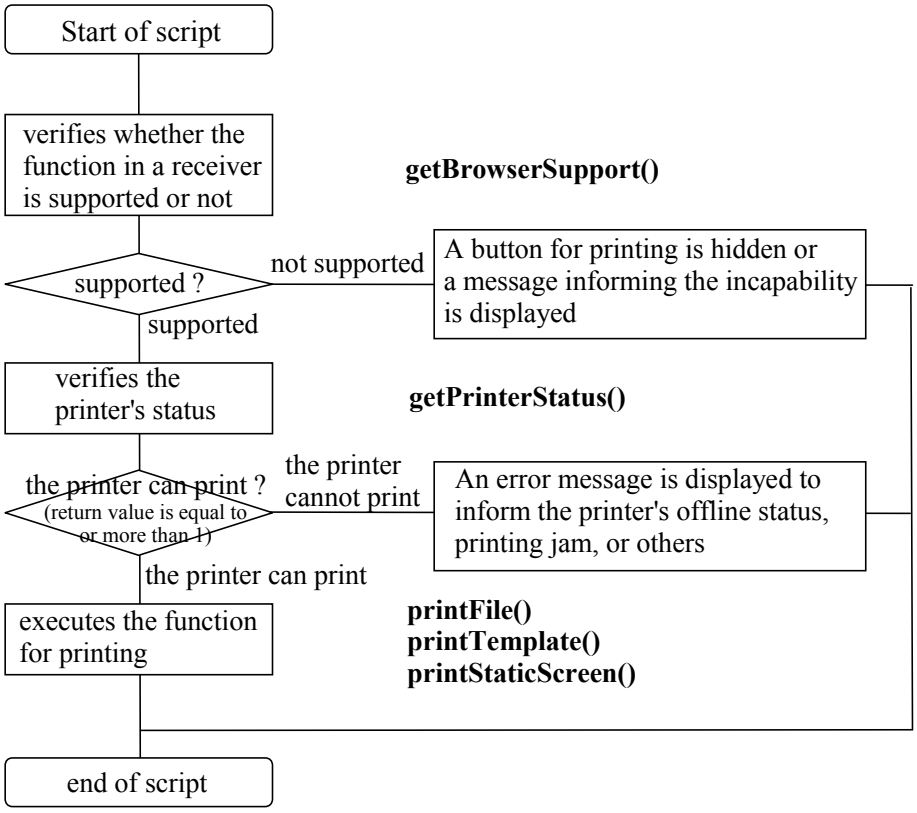
- The saveImageToMemoryCard() function assumes that files to be saved are locked in a content memory before the function is executed.
- The dst_filename argument can contain up to eight alphanumeric characters. Any alphabetical character must be a lower-case character. Note that the first character of the string must be an alphabetical character, not a numeric. Any image file must have an appropriate extension based on a coding specification.
- The saveStaticScreenToMemoryCard () function assumes that no alpha blending between planes is required nor no video planes are composed.

A1-12.4 Sample Behaviours of Functions for Printing

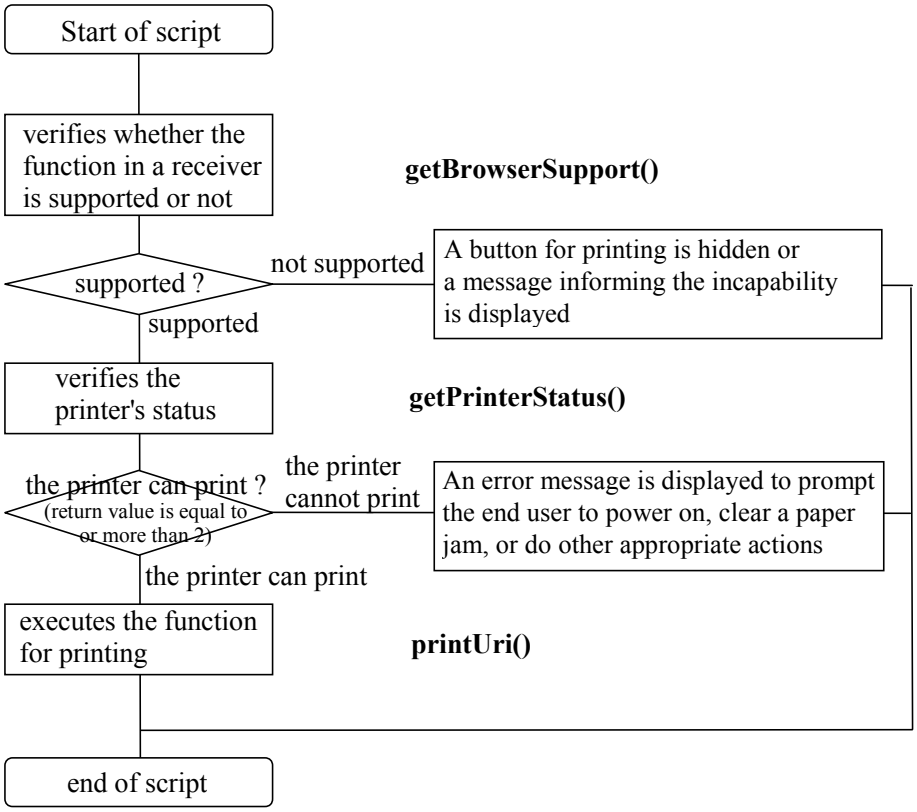
This section provides sample behaviours of functions for printing.

(1) Sample script for printFile, printTemplate, and printStaticScreen

The following is a sample script in a BML document.



(2) Sample script for printUri



Chapter APPENDIX 1-13 Operation relating to Coding of XHTML document for Printing

A1-13.1 Coding of Document for Printing

Any coding of a document to be printed must comply with the following specifications:

XHTML-Print(http://www.w3.org/TR/xhtml-print)

CSS Print Profile (http://www.w3.org/TR/css-print) including the CSS print-Enhanced part

A1-13.2 Character Coding of Printing Document

Any document to be printed must be coded with either the EUC-JP or Shift_JIS character coding scheme to be transmitted. Only one coding scheme of the two must be used for a single document to be printed and external documents to which are referred by the document to be printed. The points and sections representing characters available to a document to be printed are defined in Table 7-4, Part 2, Volume 1, ARIB B24.

Note that the sections from 90 to 94 are not operated, being replaced with the 13 section shown in Table 13-1. To allow script in a BML document to convert what a file contain dynamically, only the character codes that are supported by both a character coding scheme that is applied to a document to be printed and a character coding scheme that is supported the concerned BML document.

A1-13.2.1 EUC-JP

Operational guidelines on EUC-JP are outlined in Table 13-1.

Table 13-1 EUC Code Set

Code Set	Character Set	Operated
Code Set 0 Byte Range 21 - 7F	JIS X 0201-1997 (JIS Roman characters)	O
Code Set 1 Available to First Byte A1 - FE Available to Second Byte A1 - FE	JIS X 0208-1990	O
Code Set 2 Available to First Byte 8E Available to Second Byte A1 - DF	JIS X 0201-1997 (Half-size katakana characters)	-
Code Set 3 Available to First Byte 8F Available to Second Byte A1 - FE Available to Third Byte A1 - FE	JIS X 0212-1990	-
Control Code	Space character (20)	O
	Delete character (7F)	-
	New line (0D0A)	O
	Tab (09)	O

A1-13.2.2 Shift_JIS

Operational guidelines on Shift_JIS are outlined in Table 13-2.

Table 13-2 Shift_JIS Code Set

Code Set	Character Set	Operated
For single-byte Character: Available Bytes 21 - 7F A1 - DF	JIS X 0201-1997 (JIS Roman characters) JIS X 0201-1997 (Half-size katakana)	O Except the A1-DF codes (Half-size katakana), which are not operated.
For double-byte Character: Available to First Byte 81 - 9F E0 - EF Available to Second Byte 40 - 7E 80 - FC	JIS X 0208-1990	O
Control Code	Space character (20)	O
	Delete character (7F)	-
	New Line (0D0A)	O
	Tab (09)	O

A1-13.2.3 Additional Signs

The following signs which may be useful to a document to be printed are added to the 13 Section.

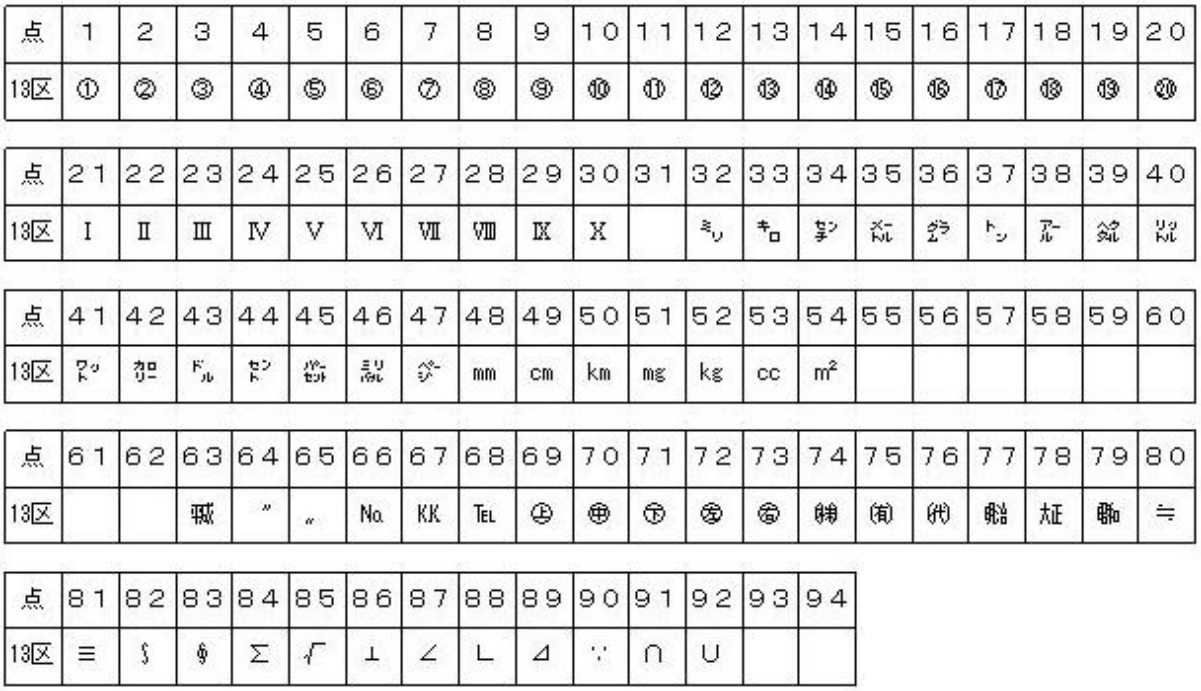


Figure 13-2 Addiritonal Signs for Printing

A1-13.2.4 External characters

No external characters are operated.

A1-13.2.5 Operated Character Reference

No characters are referenced with "&#".

A1-13.3 Operated Media Types and Monomedia

Media types that are assumed to be necessary for a printing service are shown in Table 13-3. The table also shows operational guidelines on a monomedia coding specification defined in the object element or the img element.

Table 13-3 Operated Media Types and Monomedia

Media Type	Operated	Operated (referred with the object element)	Operated (referred with the img element)	Remarks
application/xhtml+xml; charset=”euc-jp”	O	-	-	
application/xhtml+xml; charset=”Shift_JIS”	O	-	-	(Note 1)
text/css	O	-	-	(Note 2)
image/jpeg	O	O	O	
image/png	O	O	O	(Note 3)

Note 1: The charset value is non-omittable.
Note 2: To specify CSS charset, the @charset rule is used. When the @charset rule is omitted, a character coding specification that is for a referring XHTML document to be printed.
Note 3: The ARIB STD-B24 specification extends the XHTML-Print specification to use PNG.

A1-13.4 Bitmap

A1-13.4.1 PNG

Any operation of PNG must comply with the definition in the 5.3 section , Part 2, Volume 1, ARIB STD-B24, except that a PLTE chunk is required for PNG data when the colour type is "3" (pallet index).

A1-13.4.2 JPEG

To use JPEG for a bitmap coding, the methodology defined in ISO/IEC 10918-1 is used. Only the Baseline form can be used.

A1-13.5 XHTML Document for Printing

A1-13.5.1 XML declaration

In the XML declaration, the version of XML must be specified as "1.0". When the EUC-JP character coding scheme is used for a document to be printed, "EUC-JP" must be specified as character coding scheme identifier.

(Example)

<?xml version=”1.0” encoding=”EUC-JP”?>

When the Shift_JIS character coding scheme is used for a document to be printed, " Shift_JIS " must be specified as character coding scheme identifier.

(Example)

<?xml version=”1.0” encoding=”Shift_JIS”?>

A character coding scheme identifier , either of "EUC-JP" or " Shift_JIS ", must be specified.

A1-13.5.2 Document type declaration

When an XHTML document to be printed is not a valid document, this document must be a well-formed document that may have no DTD. To contain a document type declaration, the following document type declaration must be used.

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML-Print 1.0//EN"
"http://www.w3.org/MarkUp/DTD/xhtml-print10.dtd">

A1-13.5.3 Namespace for XHTML document to be printed

The root element for an XHTML document to be printed must be "html". The xmlns attribute defaults to xmlns=”http://www.w3.org/1999/xhtml”. This is assumed to be the xmlns attribute value, when no value is expressed.

(Example)

<html xmlns=”http://www.w3.org/1999/xhtml” xml:lang=”ja”>

A1-13.6 Elements for XHTML Document to Be Printed

This section provides operational guidelines on elements for an XHTML document to be printed.

A1-13.6.1 Entities

The following entities are interpreted as shown below without no explicit definitions. No other entity is required to be operated.

- lt <
- gt >
- amp &
- apos ‘
- quot “

A1-13.6.2 Operational guidelines on elements

The elements that can be used for an XHTML document to be printed are listed below. Any modularization conforms to the “Modularization of XHTML” W3C Recommendation.

- Structure module

  body, head, html, title

- Text module

  abbr, acronym, address, blockquote, br, cite, code, dfn, div, em,
  h1, h2, h3, h4, h5, h6, kbd, p, pre, q, samp, span, strong, var

- Hypertext module

  a

- List module

dl, dt, dd, ol, ul, li

- Presentation module

  b, big, hr, i small, sub, sup, tt

- Basic Forms module

  form, input, label, select, option, textarea

- Basic Table module

  caption, table, td, th, tr

- Image module

  img
  The img element must include width and height as CSS properties.

- Object module

  object, param
  The img element must include width and height as CSS properties.

- Metainformation module

  meta

- Style Sheet module

  style

- Style Attribute module

  style attribute

- Link module

  link

- Base module

  base

A1-13.6.3 Operational guidelines on attributes

- The xml:lang attribute must be fixed to "ja".
- The id attribute is a string of which maximum length is 128 bytes.

A1-13.7 Operated CSS-based Style Sheets

A1-13.7.1 Media type

The @media rule is used to specify "print" as media type.

A1-13.7.2 Box model

This section lists operational guidelines on properties for a box model.

- margin-top, margin-right, margin-bottom, margin-left

  Available value : <margin-width> | inherit

  Initial value : 0

Inherited : yes

The <margin-width> represents one of <length> (it may contain a negative value), <percentage> (it must not contain a negative value), or "auto".

- margin

  Value : <margin-width>{1,4} | inherit

  Initial : 0

  Inherited : no

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

  Value : <padding-width> | inherit

  Initial : 0

  Inherited : no

  The <padding-width> represents either of <padding-width> (Any negative value is invalid, as defined in CSS2) or <percentage> (Any negative value is invalid, as defined in CSS2).

- padding

  Value : <padding-width> {1,4} | inherit

  Initial : 0

  Inherited : no

- border-top-width, border-right-width, border-bottom-width, border-left-width

  Value : <border-width> | inherit

  Initial : medium

  Inherited : no

  The <border-width> represents one of "thin", "medium", "thick" or <length> (any negative value is invalid). Actual thickness specified with "thin", "medium", or "thick" depends on an implementation.

- border-width

  Value : <border-width>{1,4} | inherit

  Initial : medium

  Inherited : no

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

  Value : <colour> | transparent | inherit

  Initial : Depends on the value for the colour property

  Inherited : no

- border-color

  Value : [<colour> | transparent]{1,4} | inherit

  Initial : Refer to each description of the concerned properties

  Inherited : no

- border-top-style, border-right-style, border-bottom-style, border-left-style

  Value : <border-style> | inherit

  Initial : none

Inherited : no

The <border-style> notation represents either of "none" or "solid".

- border-style

  Value : <border-style>{1,4} | inherit

  Initial : none

  Inherited : no

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

  Value : [<border-width> || <border-style> || [<colour> | transparent]] | inherit

  Initial : Refer to each description of the concerned properties

  Inherited : no

- border

  Value : [<border-width> || <border-style> || [<colour> | transparent]] | inherit

  Initial : Refer to each description of the concerned properties

  Inherited : no

A1-13.7.3 Visual formatting model

This section lists operational guidelines on properties for a visual formatting model.

- display

  This property must have a fixed value. For details, see the default style sheet in Annex A.

- position

  Value : static | absolute | relative | inherit

  Initial : static

  Inherited : no

  An element which is a parent of an element containing "absolute" must contain "absolute" or "relative", or must be a body element. It is recommended to contain "relative" in case of using a new coordinate system per every page. To specify "absolute", a page break instruction must be explicitly inserted. Once a page break instruction has been executed, a new coordinate system is set to be used.

- left, top

  Value : <length> | <percentage> | auto | inherit

  Initial : auto

  Inherited : no

- right, bottom

  This property must be set to the following fixed value.

  Fixed value : auto

- float

  Value : left | right | none | inherit

  Initial : none

Inherited : no

- clear

  Value : none | left | right | both | inherit

  Initial : none

  Inherited : no

- width, height

  Value : <length> | <percentage> | auto | inherit

  Initial : auto

  Inherited : no

- line-height

  Value : normal | <number> | <length> | <percentage> | inherit

  Initial : normal

  Inherited : yes

  When this property is sets to "normal", it represents a value which is 1.2 times as great as "font-size". Note that no <number> and <percentage> can accept a negative value which is defined as invalid in the CSS2 specification.

- vertical-align

  Value : baseline | super | sub | top | middle | bottom | inherit

  Initial : baseline

  Inherited : no

A1-13.7.4 Visual effects

This section lists operational guidelines on properties for visual effects.

- overflow

  Value : visible | hidden | inherit

  Initial : visible

  Inherited : no

- clip

  This property must be set to the following fixed value.

  Fixed value : auto

- visibility

  Value : visible | hidden | inherit

  Initial : inherit

  Inherited : yes

  When this property of a parent element is set to "hidden", not only the parent element but also its child elements are not presented to end users. For the body element, this property must be set to a fixed value, "visible".

A1-13.7.5 Generated content, automatic numbering, lists

This section lists operational guidelines on properties for generated content, automatic numbering, and lists.

- content
  - Value : [<string> | counter(pages)]+ | inherit
  - Initial : a null string
  - Inherited : no

  note: Only the single identifier "pages" that represents the current page number is required.
- counter-increment
  - Value : [pages | pages + an integer] | none | inherit
  - Initial : none
  - Inherited : no
- counter-reset
  - Value : [a counter identifier | pages + an integer] | none | inherit
  - Initial : none
  - Inherited : no
- list-style-type
  - Value : disc | decimal | lower-alpha | upper-alpha | none | inherit
  - Initial : disc
  - Inherited : yes
- list-style-image
  - Value : <uri> | none | inherit
  - Initial : none
  - Inherited : yes
- list-style-position
  - Value : inside | outside | inherit
  - Initial : outside
  - Inherited : yes
- list-style
  - Value : [<list-style-type> || <list-style-position> || <list-style-image>] | inherit
  - Initial : Refer to each description of the concerned properties
  - Inherited : yes

A1-13.7.6 Colours and Backgrounds

This section lists operational guidelines on properties for colours and backgrounds.

- colour
  - Value : <colour> | inherit

Initial : black

Inherited : yes

- background-color

  Value : <colour> | transparent | inherit

  Initial : transparent

  Inherited : no

- background-image

  Value : <uri> | none | inherit

  Initial : none

  Inherited : no

- background-position

  This property is not operated.

- background-repeat

  This property must be set to the following fixed value.

  Fixed value : repeat

- background

  Value : [<background-color> || <background-image>] | inherit

  Initial : Refer to each description of the concerned properties

  Inherited : no

A1-13.7.7 Fonts

The four fonts must be implemented; monospaced mincho (mincho), monospaced gothic (gothic), proportional mincho (P mincho), and proportional gothic (P gothic). Each character in the Code Set 0 in EUC-JP and each single-byte character in Shift JIS must be presented in a half width of the normal width for a used font. This section lists operational guidelines on properties for fonts.

- font-family

  Value : [[<family-name> | <generic-family>],]* [<family-name> | <generic-family>] | inherit

  Initial : "Ｐ明朝", serif

  Inherited : yes

  The <family-name> notation represents one of "明朝", "ゴシック", "Ｐ明朝", "Ｐゴシック", "丸ゴシック", "Ｐ丸ゴシック", "太丸ゴシック", or "角ゴシック". The <generic-family> notation represents one of "serif", "sans-serif", "cursive", "fantasy" or "monospace".

  Note: The "Ｐ " character used for the family-name property must be a full-size uppercase character.

- font-size

  Value : <absolute-size> | inherit

  Initial : 10.5pt

Inherited : yes

The <absolute-size> must be an absolute value in points. The available values must range from 8pt to 72pt, with one decimal place considered as a significant digit.

- font-style

  Value : normal | italic | inherit

  Initial : normal

  Inherited : yes

  How "italic" makes a character slope to the right depends on an implementation.

- font-weight

  Value : normal | bold | inherit

  Initial : normal

  Inherited : yes

  How thick "bold" makes lines of a character depends on an implementation.

- font

  Value :[['font-style'|| 'font-weight']? 'font-size'
  ['/'line-height']? 'font-family'] | inherit

  Initial : Refer to each description of the concerned properties

  Inherited : yes

- font-variant

  Value : normal | small-caps | inherit

  Initial : normal

  Inherited : yes

  How large "small-caps" makes a character depends on an implementation.

A1-13.7.8 Text

This section lists operational guidelines on properties for text.

- text-align

  Value : left | right | center | inherit

  Initial : left

  Inherited : yes

- text-decoration

  Value : none | underline | inherit

  Initial : none

  Inherited : no

  Actual position and thickness of an underline specified with "underline" depends on an implementation.

- text-indent

  Value : <length> | inherit

Initial : 0

Inherited : yes

- letter-spacing

  This property must be set to the following fixed value.

  Fixed value : normal

- white-space

  Value : normal | pre | inherit

  Initial : normal

  Inherited : yes

  A tabulation character must represent four half-size spaces. How large a single half-size space is rendered depends on an implementation.

- text-transform

  Value : capitalize | uppercase | lowercase | none | inherit

  Initial : none

  Inherited : yes

A1-13.7.9 Tables

This section lists operational guidelines on properties for a table.

- caption-side

  Value : top | bottom | inherit

  Initial : top

  Inherited : yes

- table-layout

  This property must be set to the following fixed value.

  Fixed value : auto

- border-collapse

  Value : collapse | separate | inherit

  Initial : collapse

  Inherited : yes

- border-spacing

  Value : <length> <length>? | inherit

  Initial : 0

  Inherited : yes

A1-13.7.10 Paged media

This section lists operational guidelines on properties to control a page.

- page

Value : <identifier> | auto

Initial : auto

Inherited : yes

- page-break-after

Value : auto | always | inherit

Initial : auto

Inherited : no

- page-break-before

Value : auto | always | inherit

Initial : auto

Inherited : no

- page-break-inside

Value : avoid | auto | inherit

Initial : auto

Inherited : no

A1-13.7.11 Page formatting

The @page rule in CSS2 is used to format pages. The size property for @page is used to specify a target page size and a target page orientation.

- size

Value : <length>{1,2} | auto | portrait | landscape | inherit

Initial : auto

Inherited : no

A sheet size is specified with <length>. Specifying a width value and a height value in <length> must work as specifying a page orientation. More specifically, specifying a width value that is greater than a height value must be interpreted as specifying "landscape" as page orientation. Specifying a height value that is greater than a width value must be interpreted as specifying "portrait" as page orientation. When a sheet actually fed into a printer is different from what <length> of the size property specifies, how a printer behaves depends on an implementation, which must support the A4 size.

Table 13-4 Page Orientation Specified with size Property

Sheet Size	Page Orientation	Sample Specification with @page
A4	portrait	@page { size: 210mm 297mm; }
A4	landscape	@page { size: 297mm 210mm; }
postcard	portrait	@page { size: 100mm 148mm; }
postcard	landscape	@page { size: 148mm 100mm; }

A1-13.7.12 Guidelines on page layout

The following figure provides guidelines on page layout. A page box (to be rendered on a sheet) defined with the margin properties in the @page rule is a content area. This area work as a parent box

which can contain different types of child boxes, including boxes which are rendered text and/or still pictures. To locate each box, the margin properties for the box must be specified. The margin properties in the @page rule can be used to define a Top margin box at the top of a page box and a Bottom margin box at the bottom of a page box.

Using @top to set a header in a Top margin box must be accepted while using @bottom to set a footer in a Bottom margin box must be accepted.

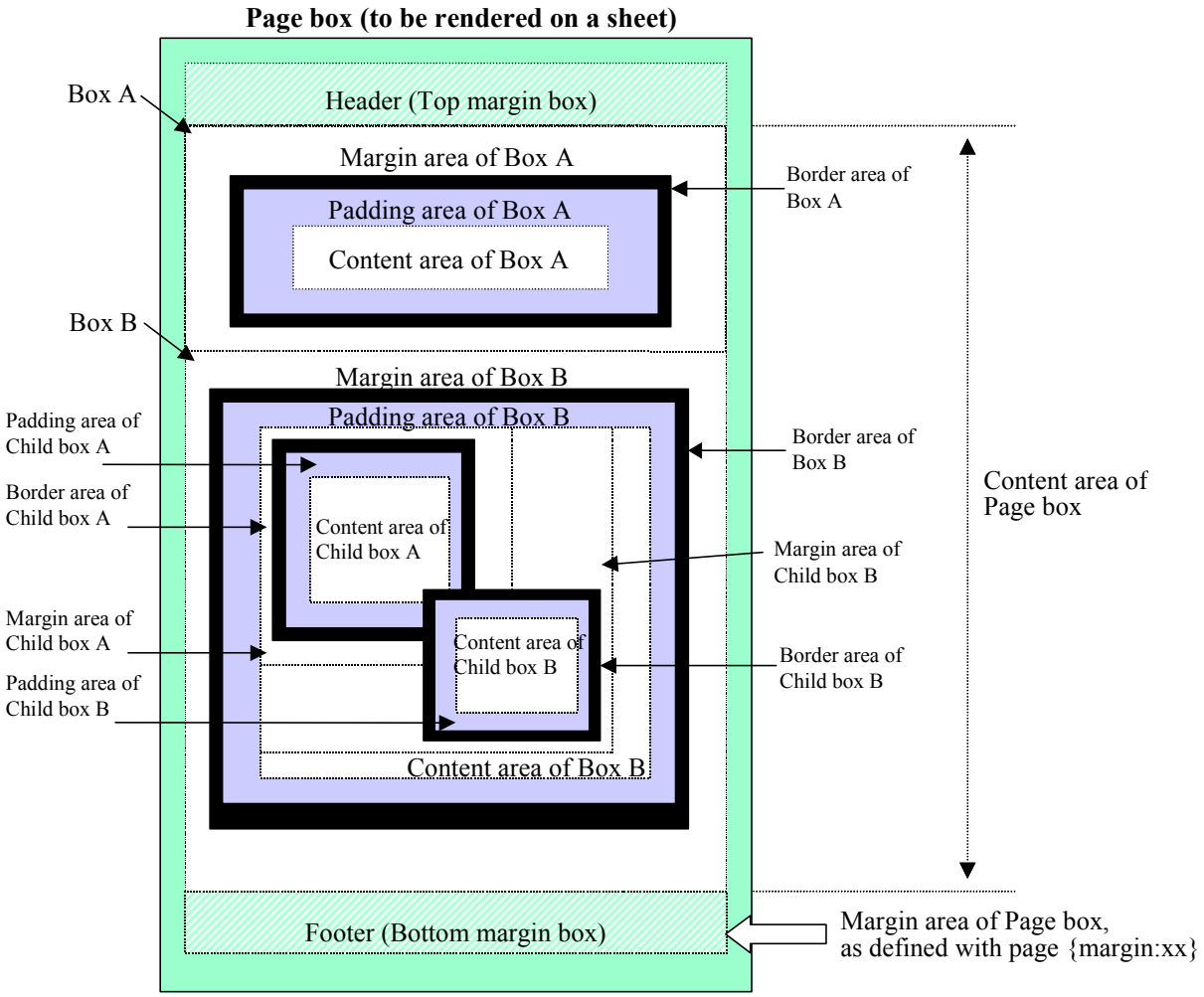


Figure 12-6 Sample File with Replaced Content

A1-13.7.13 Operational guidelines on page controlling

When two or more pages are needed to contain intended information, necessary page breaks are automatically occur. Actual position where a page break occurs depends on an implementation. Inserting "page-break-after / page-break-before" or other explicit instruction to control page breaks are recommended.

A1-13.7.14 Default style sheet

The default style sheet is provided in Annex A Default Style Sheet. This style sheet is installed in a printer, requiring no need to be transmitted.

Some printers may accept additional specifications other than what the default style sheet in Annex A contains.

A1-13.7.15 Operational guidelines on length

Usage of <length> is limited to specify one of the following units:

em 'font-size' for the concerned font

ex the height of the lowercase character 'x' in the concerned font

px How a value in "px" is converted into a value for absolute positioning depends on an implementation (It is not recommended)

in in inches (1 in. = 25.4 mm)

mm in millimeters

cm in centimeters

pt in points (1 pt. = 25.4/72 mm)

A1-13.7.16 Considerations for preparing XHTML document to be printed

- The total size of an XHTML document to be printed (excluding resources to which referred by the document) and concerned CSS files must not exceed 256 KB.
- It is recommended that a single table element has up to one hundred rows and ten columns in a portrait-oriented sheet and thirty rows and thirty columns in a landscape-oriented sheet.

A1-13.8 Namespace for Content to Be Printed

This section defines the namespace used for allowing a part of content to be printed in an XHTML document to be printed to refer to external resources.

A1-13.8.1 Obtaining XHTML document to be printed through HTTP schema

When a URI where content to be printed exists is obtained from a receiver, an XHTML document to be printed as a set of content to be printed is obtained through the HTTP schema. To refer to resources in an XHTML document to be printed, the HTTP schema must be used.

A1-13.8.2 Obtaining XHTML document to be printed through ARIB-DC schema

When resources to be printed are obtained from a receiver, concerned XHTML documents to be printed as a set of content to be printed are obtained through the ARIB-DC schema. In this case, a closed namespace, where only the resources in modules specified with an API for printing can be refereed to, is available. The format for referring to a resource is a hierarchical structure whose top element is component_tag, and includes no schema identifiers. Referring using either of an absolute path or a relative path can work, as shown below.

- Referring using an absolute path
  - / <component_tag> / <moduleName> / <resourceName>

    (Example)
- Referring using a relative path

- <resourceName>

  To refer to a resource in a module to which a referred document includes:

  (Example)

- .. / <moduleName> / <resourceName>

  To refer to a resource in an ES to which a referred document includes but in a module to which the document does not include:

  (Example)

- .. / .. / <component_tag> / <moduleName > / <resourceName >

  To refer to a resource in a module in ES to which a referred document does not include:

  (Example)

A1-13.9 Japanese Hyphenation

For the purpose of rendering an XHTML document to be printed, a printer must implement the following Japanese hyphenation functions.

- When a character that must not be the first character of a line is specified to be the beginning of a second or following line, the character must be appended to the end of the previous line. When two or more successive characters that must not be the first character of a line are specified to be the beginning of a second or following line, how the characters are rendered depends on an implementation.
- When a character that must not be the last character of a line is specified to be the end of a line, the character must be part of the next line. When two or more successive characters that must not be the last character of a line are specified to be the end of a line, how the characters are rendered depends on an implementation.
- Each word in alphabetical characters must separated with a space or accept a hyphenation rule as necessary.

The characters that must not be the first character of a line are listed in Table 13-5 while the characters that must not be the last character of a line are shown in Table 13-6.

Table 13-5 Characters not for First Character of Line

Shift JIS Code	21	29	2C	2E	3A	3B	3F	5D	7D
Character	!	)	,	.	:	;	?	]	}
Shift JIS Code	A1	A3	A4	DE	DF	8141	8142	8143	8144
Character	。	」	、	゛	゜	ヽ	ヾ	々	ー
Shift JIS Code	8146	8147	8148	8149	814A	814B	814D	8152	8153
	：	；	？	！	゛	゜	´	｀	¨
Shift JIS Code	8154	8155	8158	816A	816C	816E	8170	8172	8174
	＾	￣	＿	）	］	］	｝	〉	》
Shift JIS Code	8176	8178	817A						
	」	』	】						

Table 13-6 Characters not for Last Character of Line

Shift JIS Code	28	5B	7B	3A	3B	3F	5D	7D	A2
Character	(	[	{	:	;	?	]	}	「
Shift JIS Code	8169	816B	816D	816F	8171	8173	8175	8177	8179
Character	﴾	﴿	﴿	﴿	﴿	﴿	﴿	﴿	﴿

Annex A Default Style Sheet Applied to XHTML Document to Be Printed

```
@page  { size: 210mm 297mm;
      margin : 3mm 3mm 3mm 3mm;
      @top    { text-align: center; }
      @bottom { text-align: center; }
}
/* Structure Module(xhtml-struct-1.mod) */
html   { }
head   { display: none; }
body   { color     : black;
      display: block;
      font-size: 10.5pt;

      font-family: "Ｐ明朝",serif;
      list-style-type: disc;
      text-align: left;
      line-height: 1.2;
      padding: 6pt;
    }
/* Text Module(xhtml-text-1.mod) */
/* Text: Inline Structural Module(xhtml-inlstruct-1.mod) */
br  { }
span   { }
/* Text: Inline Phrasal Module(xhtml-inlphras-1.mod) */
abbr   { font-variant     : small-caps;
      letter-spacing      : 0.1em;
    }
acronym{ font-variant     : small-caps;
      letter-spacing      : 0.1em;
    }
cite   { font-style       : italic; }

code   { font-family      : "ゴシック", monospace; }
em  { font-style    : italic; }

kbd { font-family   : "ゴシック", monospace;}
q   { }

samp   { font-family      : "ゴシック", monospace;}
strong  { font-weight     : bolder; }
var { font-style    : italic; }

/* Text: Block Structural Module(xhtml-blkstruct-1.mod) */
div, p, address, blockquote, pre, h1, h2, h3, h4, h5, h6 {
    display: block;
}
div { }
p   { margin        : 1.33em 0; }

/* Text: Block Phrasal Module(xhtml-blkphras-1.mod) */
address{ font-style       : italic; }
blockquote{ page-break-inside: avoid;
      margin        : 1.33em 40pt;
    }
pre { page-break-inside    : avoid;

      font-family   : "ゴシック", monospace;
      white-space   : pre;
    }
h1  { page-break-inside    : avoid;
      font-size     : 2.00em;
      margin        : 0.67em 0;
      line-height   : 1.00em;
      font-weight   : bolder;
    }
h2  { page-break-inside    : avoid;
```

```
      font-size     : 1.50em;
      margin        : 0.83em 0;
      line-height   : 1.00em;
      font-weight   : bolder;
    }
h3  { page-break-inside    : avoid;
      font-size     : 1.17em;
      margin        : 1.00em 0;
      line-height   : 1.00em;
      font-weight   : bolder;
    }
h4  { page-break-inside    : avoid;
      font-size     : 1.00em;
      margin        : 1.33em 0;
      line-height   : 1.00em;
      font-weight   : bolder;
    }
h5  { page-break-inside    : avoid;
      font-size     : 0.83em;
      margin        : 1.67em 0;
      line-height   : 1.17em;
      font-weight   : bolder;
    }
h6  { page-break-inside    : avoid;
      font-size     : 0.67em;
      margin        : 2.33em 0;
      line-height   : 1.33em;
      font-weight   : bolder;
    }
/* Hypertext Module(xhtml-hypertext-1.mod) */
a   { text-decoration      : underline; }
/* Lists Module(xhtml-list-1.mod) */
dl, dt, dd, ul, ol { display: block; }
dl  { margin        : 1.33em 0;
           }
dt  { }
dd  { margin-left: 40pt; }
ul  { margin : 1.33em 0 1.33em 40pt;
           }
ol  { margin : 1.33em 0 1.33em 40pt;
      list-style-type: decimal;
           }
li  { display       : list-item; }
/* Image Module(xhtml-image-1.mod) */
img { }
/* Basic Table Module(xhtml-basic-table-1.mod) */
table   { display: table; }
caption{ color      : black;
      text-align: center;
      display: table-caption;
    }
tr  { display: table-row;
      page-break-inside: avoid;
    }
th  { display: table-cell;
      font-weight: bolder;
      text-align: center;
      page-break-inside: avoid;
    }
td  { display: table-cell;
      page-break-inside: avoid;
    }
/* Simplified Forms Module(xhtml-basic-form-1.mod) */
form    { display: block; }
form    { margin    : 1.33em 0; }
label   { }
input   { }
select  { }
```

```
option { }

textarea { font-family: "明朝", monospace;
      white-space: pre;
    }
/* Presentation Module(xhtml-pres-1.mod) */
/* Presentational: Inline Presentation Module(xhtml-inlpres-1.mod) */
b   { font-weight   : bolder; }
big { font-size     : 1.7em; }
i   { font-style    : italic; }
small  { font-size : 0.83em; }
sub { font-size     : 0.83em;
      vertical-align       : sub;
    }
sup { font-size     : 0.83em;
      vertical-align       : super;
    }

tt  { font-family   : "ゴシック", monospace; }
/* Presentation: Block Presentation Module(xhtml-blkpres-1.mod) */
hr  { border        : 1pt black solid;
      border-color  : black silver black silver;
      padding-top   : 1pt;
      display: block;
    }
/* Link Element Module(xhtml-link-1.mod) */
link    { display: none; }
/* Document Metainformation Module(xhtml-meta-1.mod) */
meta    { display: none; }
/* Base Element Module(xhtml-base-1.mod) */
base    { display: none; }
/* Style Sheets Module(xhtml-style-1.mod) */
style   { display: none; }
/* Param Element Module(xhtml-param-1.mod) */
param   { display   : none; }
/* Embedded Object Module(xhtml-object-1.mod) */
object  { display: block; }
```

Informative Explanation

1 Using Box Model

The box model that is defined by CSS determines where its content is to be displayed by using the co-ordinates of the top left corner, the width of the border, and the padding property. In this case, changing the width of a border changes the position at which the content is to be displayed. Therefore, displaying the content at a constant position must not be achieved by changing the border width but by specifying a transparent colour that is not drawn as border-color.

2 Example Behaviour for the Properties which describe Remote control Operations

The nav-* property is used as follows:

```
<body>
   <div id="div1" style = "..." >
      <object id="a" style="... nav-index:1; nav-up:2; nav-down:3" />
      <object id="b" style="... nav-index:2; nav-up:1; nav-down:2" />
      <object id="c" style="... nav-index:3; nav-up:4; nav-down:1" />
      <object id="d" style="... nav-index:4; nav-up:3; nav-down:4" />
   </div>
</body>
```

If the down key is pressed when a focus is on object 1 (id=a), the focus is moved to object 3 (id=c) according to the specification of ”nav-down:3.”

3 Guidelines on Time-specified Call Functions using BASIC-based Protocols

- Time-specified call functions

  To send information related to a time-specified call using a BASIC-based protocol, the extended function for broadcasting setDelayedTransmissionDataOverBASIC() is used. The function sets a time at which the specified information is sent to provide an interactive communication service, without depending on an event. It is recommended that a value representing an interval between a start time to send and a corresponding end time be greater than a value representing an interval between retries. It is also recommended that random numbers be used to set data sending times at intervals of an uneven period.

  No specific data must be used for the confirmation of sending and reception at the receiver side.

- Telephone numbers available to setDelayedTransmissionDataOverBASIC()

  The characters allowed to represent telephone numbers specified in setDelayedTransmissionDataOverBASIC() are described below.

Signal Type	Available Characters
Pulse	0 ~ 9, “(”, “)”, “-”, “T”(preceding the strings that are sent in a tone signal), ”, ”(representing a pause)
Tone	0 ~ 9, “(”, “)”, “-”, “*”, “#”, “,”(representing a pause)

  When a character, not listed in the above table is part of a telephone number, the character must not be used. A string representing a telephone number must be a combination of the listed

characters. A string representing a telephone number contains up to 30 bytes. When a telephone number string is longer than 30-byte, NaN (failure) is returned.

- Area controlled by time-specified call functions using BASIC-based protocols

  The structure of area controlled by time-specified call functions is shown below.

	Element	Information	Size	Format
1	regname	Registration name	256 bytes	EUC-JP
2	uri (Note)	URI representing a location where a time-specified call configuration is sent	256 bytes	Alphanumeric characters
3	start_time	Time when sending a time-specified call configuration starts	12 bytes	YYYYMMDDhhmm
4	end_time	Time when sending a time-specified call configuration ends	12 bytes	YYYYMMDDhhmm
5	strtext	Text data to be sent	4096 bytes	Character encoding specified with charset
6	-	Time when a configuration is sent	12 bytes	YYYYMMDDhhmm
7	-	Code representing the result of the sending operation. A return value of connect(), sendTextData(), receiveTextData(), or disconnect().	4 bytes	Alphanumeric characters
8	-	Other information (Detailed usage depends on an implementation)	472 bytes	
9	-	Flag indicating whether a BASIC-based function or IP-based function. Detailed usage depends on an implementation.	1 byte	
		Total	5121 bytes	

Note: No value is specified when setDelayedTransmissionDataOverBASIC() is used to register.

A total area controlled by time-specified call functions must be 51210 bytes in capacity and contain up to 10 time-specified call configurations. When a time-specified call is to be registered while free area is missing, time-specified call configurations that has been sent, if any, are deleted starting from the oldest configuration.

The area is available to time-specified call functions using TCP/IP protocols.

- Verification of status of time-specified call functions registered or processed by BASIC-based protocols

  To obtain a status of a time-specified call registered by setDelayedTransmissionDataOverBASIC(), the extended function for broadcasting, getDelayedTransmissionStatus() is used. To verify whether or not a time-specified call is successfully registered by setDelayedTransmissionDataOverBASIC(), the extended function for broadcasting, getDelayedTransmissionResult() is used. Each function returns Array[1] (for destination URI) containing an empty string. The getDelayedTransmissionResult() function returns Array[3] containing the return values returned by a function used by the concerned BASIC-based protocol (connect(), sendtextData(), receiveTextData(), or disconnect()).

4 Sample Behaviours of Interaction channel Functions for IP Connection

This section provides sample behaviours of interaction channel functions.

(1) Figure 1 shows a sample flow chart to write script for an IP connection triggered from a content.

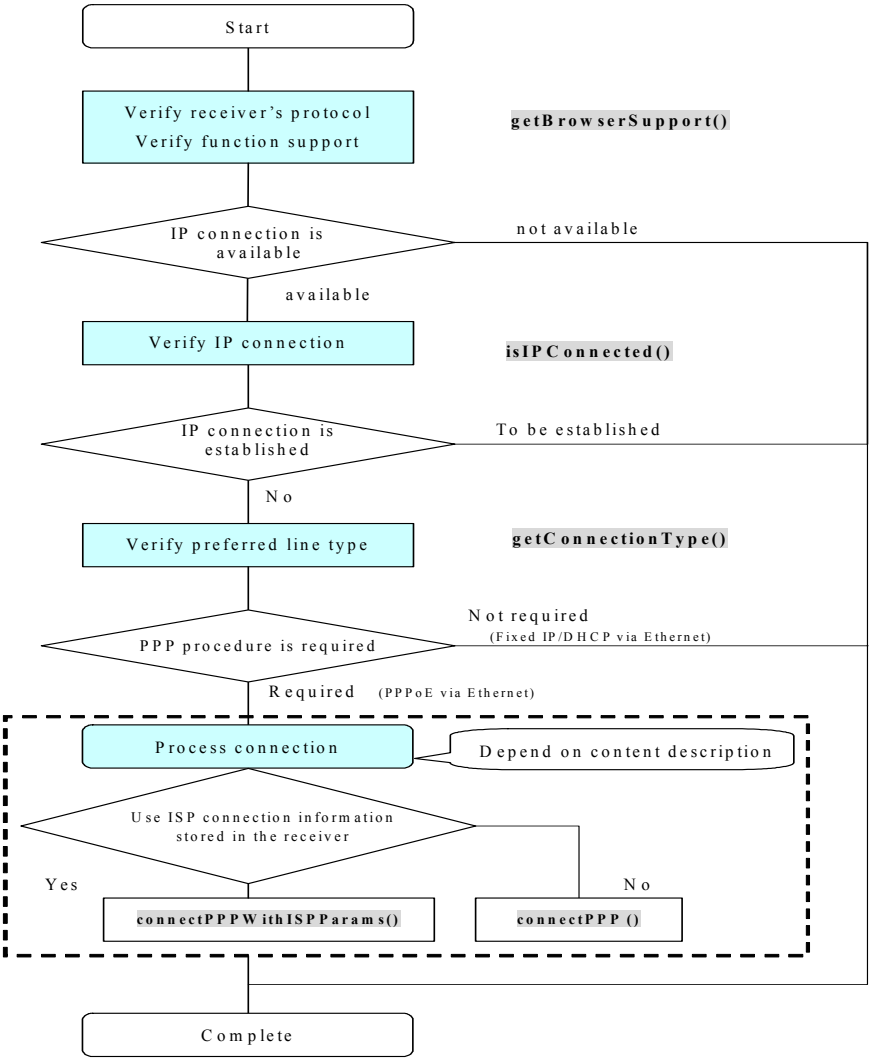


Figure 1 Sample Script for IP Connection

(2) Functions for processing connections

This section provides sample behaviours of extended functions for broadcasting used to process connections.

- Sample behaviour of the isIPConnected() function

Figure 2 shows how the isIPConnected() function works.

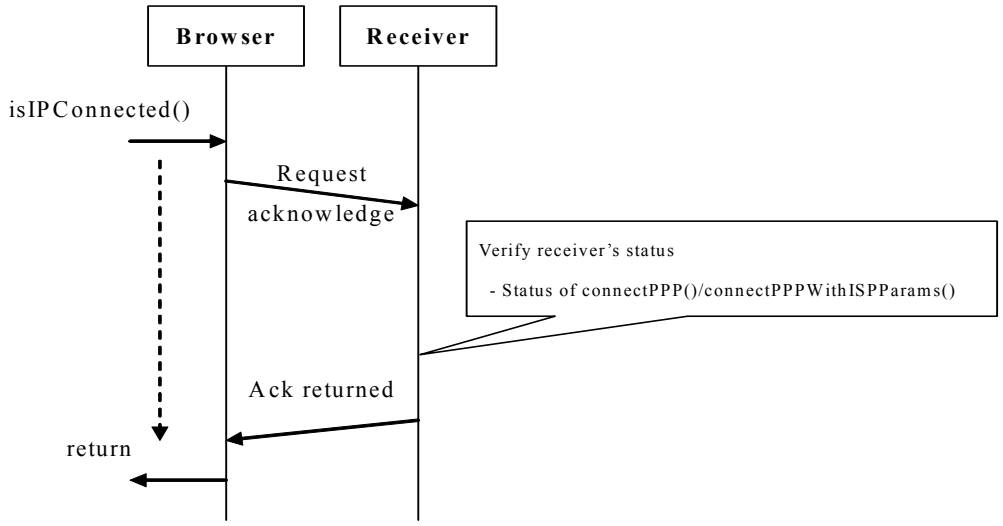


Figure 2 Sample Behaviour of isIPConnected()

- Sample behaviour of the getConnectedType() function

Figure 3 shows how the getConnectedType() function works.

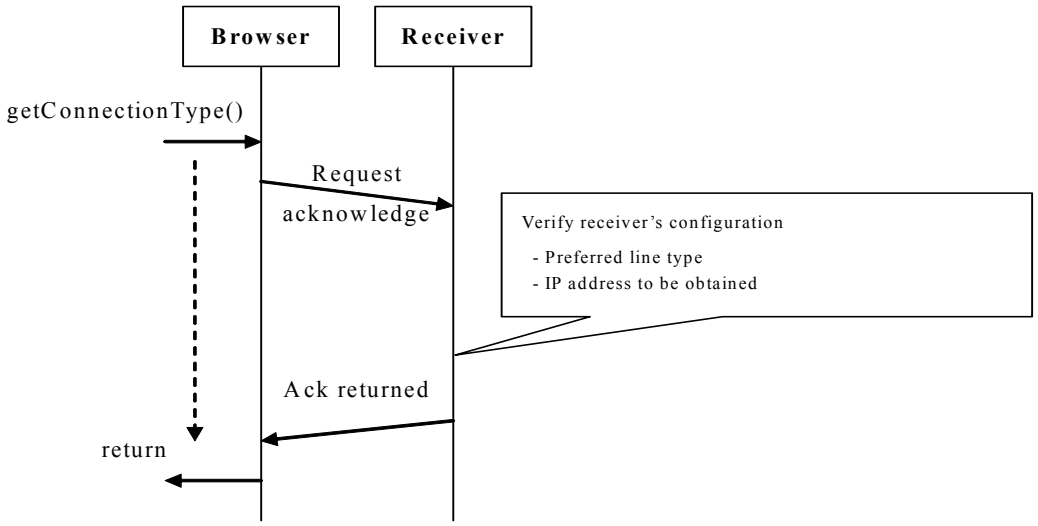


Figure 3 Sample Behaviour of getConnectionType()

- Sample behaviour of the connectPPP() function

  Figure 4 shows how the connectPPP () function works.

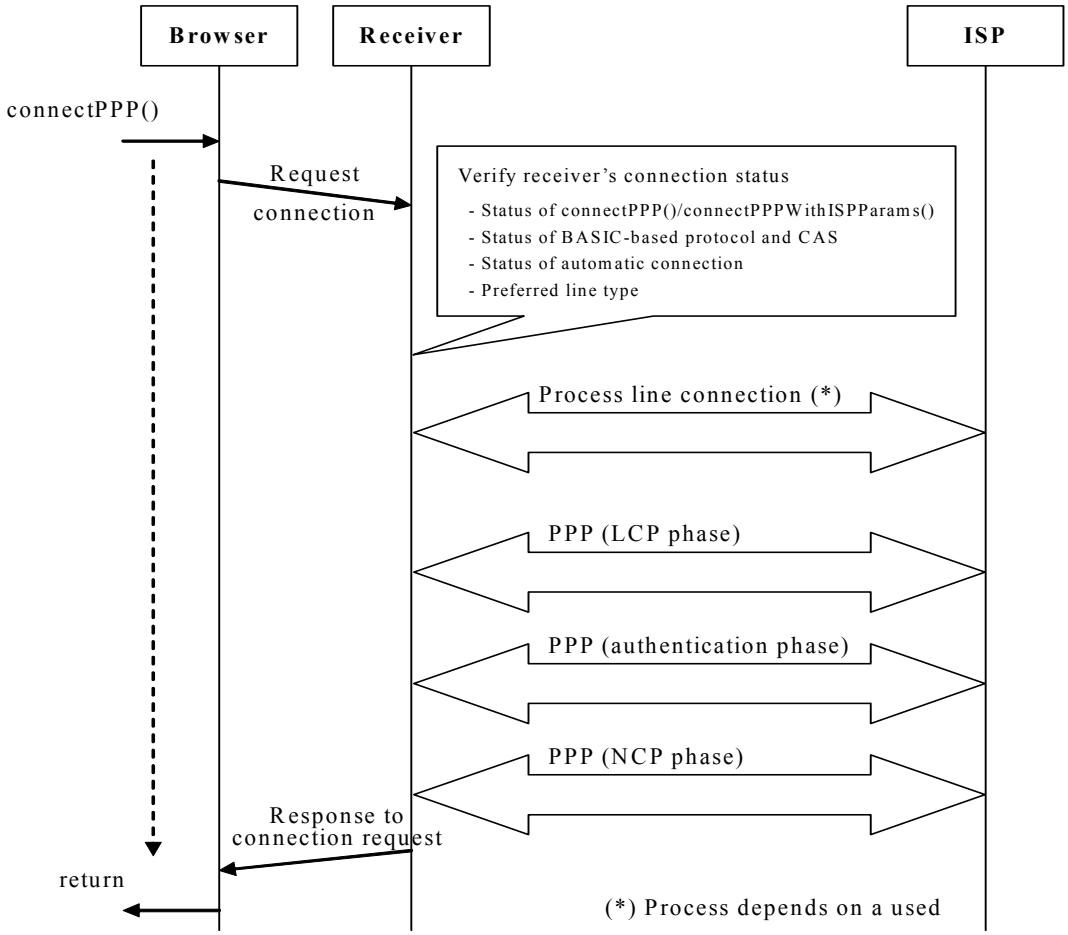


Figure 4 Sample Behaviour of connectPPP ()

- Sample behaviour of the ConnectPPPWithISPParams() function

  Figure 5 shows how the ConnectPPPWithISPParams() function works.

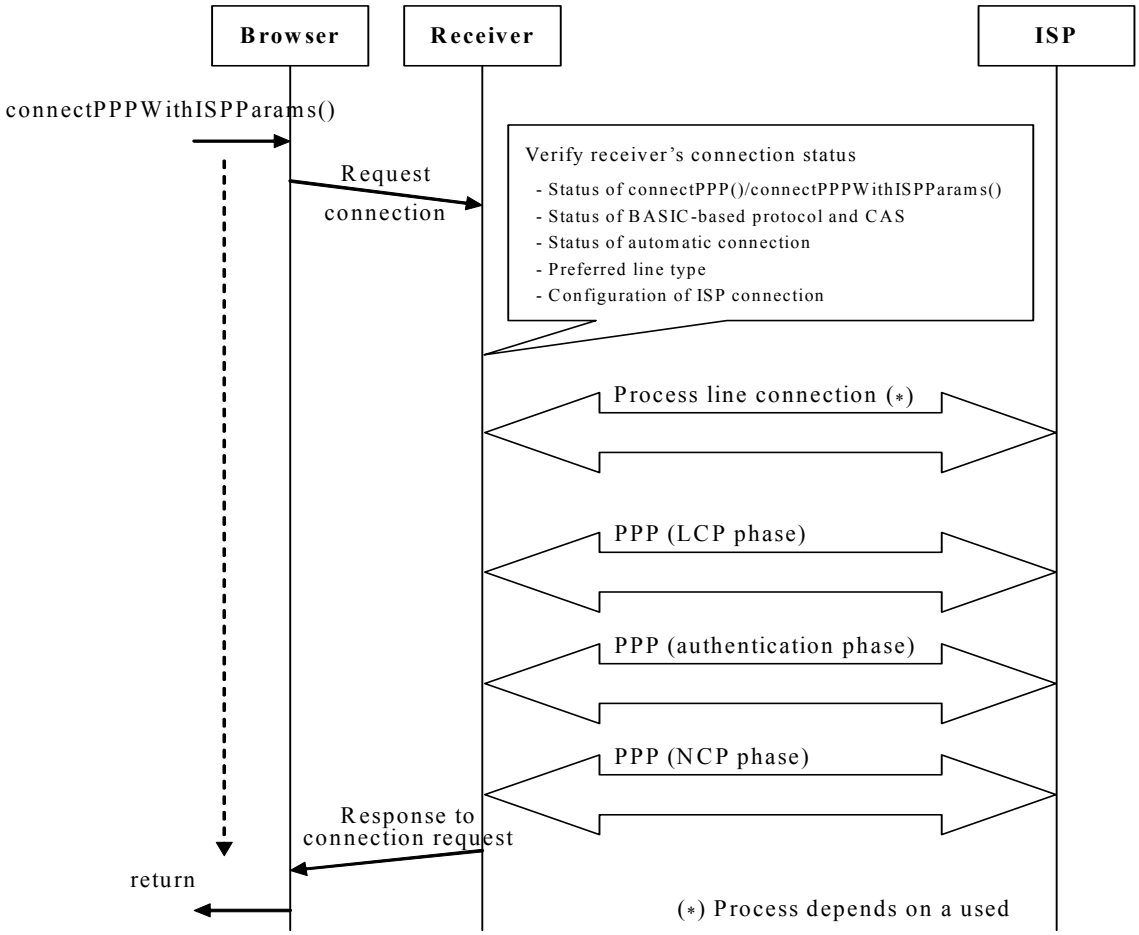


Figure 5 Sample Behaviour of ConnectPPPWithISPParams()

- Sample behaviour of connecting functions specific to channel type

  The followings show how connectPPP()/ConnectPPPWithISPParams() works when the preferred line type is PSTN (PPP), Ethernet(PPPoE), or Ethernet(DHCP/Fixed IP).

  - Sample behavior of connectPPP()/ConnectPPPWithISPParams() with the preferred line type PSTN specified.

    Figure 6 shows how connectPPP()/ConnectPPPWithISPParams() works when PSTN is specified as the preferred line type.

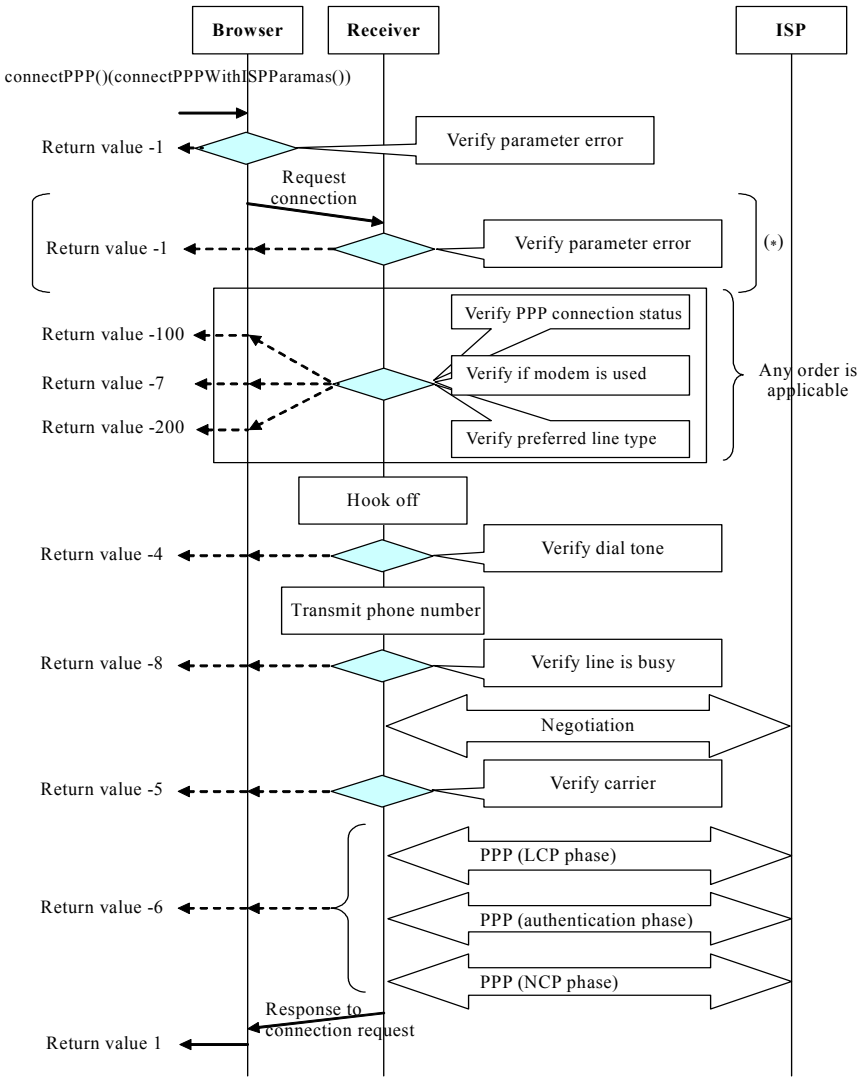


Figure 6 Sample Behaviour of connectPPP()/ConnectPPPWithISPParams() for PSTN

- Sample behaviour of connectPPP()/ConnectPPPWithISPParams() with the preferred line type PSTN and IP address obtained using PPPoE protocol.

  Figure 7 shows how connectPPP()/ConnectPPPWithISPParams() works when PSTN is specified as the preferred line type and IP address is obtained using PPPoE protocol.

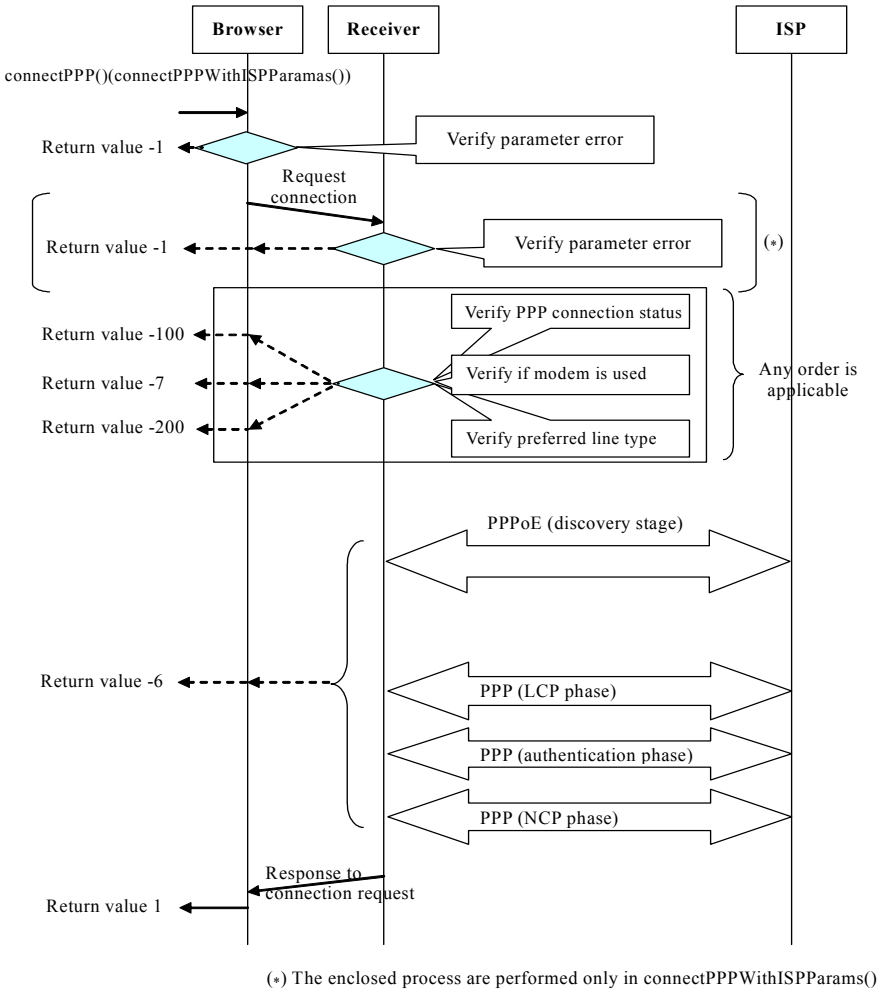


Figure 7 Sample Behaviour of connectPPP()/ConnectPPPWithISPParams() for Ethernet (PPPoE)

- Sample behaviour of connectPPP()/ConnectPPPWithISPParams() with the preferred line type Ethernet and the line type for obtaining IP address DHCP/Fixed IP specified.

  Figure 8 shows how connectPPP()/ConnectPPPWithISPParams() works when Ethernet is specified as the preferred line type and DHCP/Fixed IP is specified as the line type for obtaining IP address.

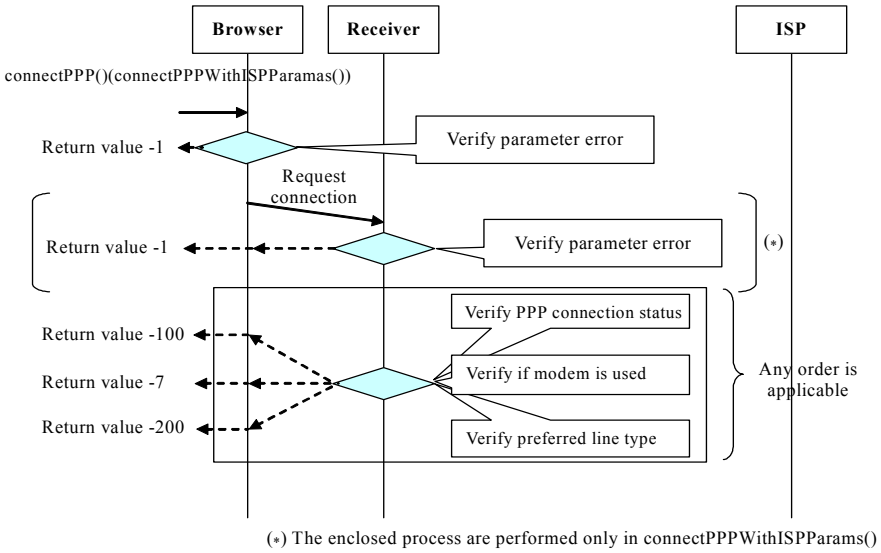


Figure 8 Sample Behaviour of connectPPP()/ConnectPPPWithISPParams() for Ethernet (DHCP/Fixed IP)

(3) Functions for processing time-specified calls

This section provides sample behaviours of time-specified call functions.

- Sample behaviour of the setDelayedTransmissionData() function

Figure 9 shows how the setDelayedTransmissionData() function works.

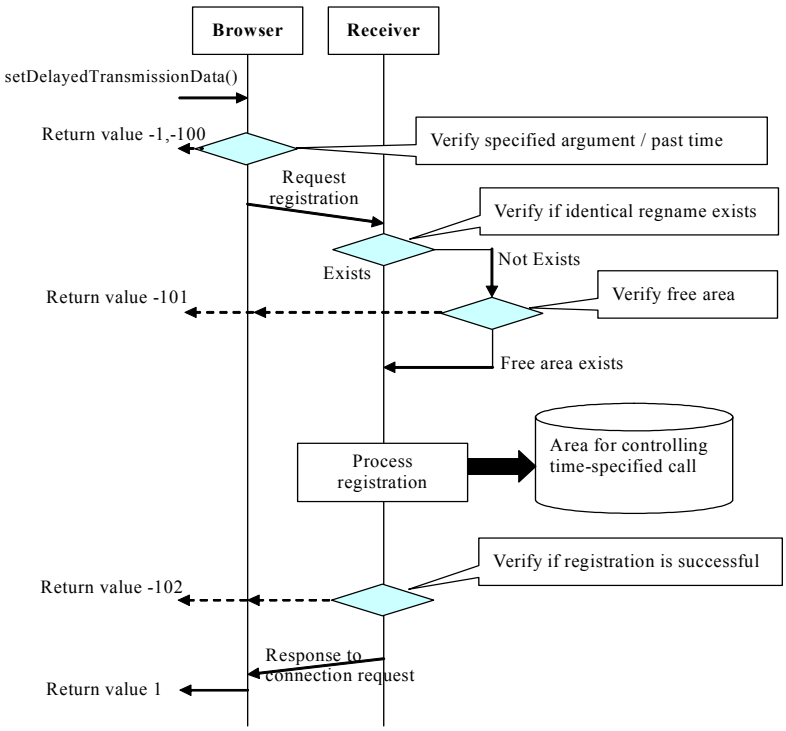


Figure 9 Sample Behaviour of setDelayedTransmissionData()

- Sample behaviour of the getDelayedTransmissionStatus() function

  Figure 10 shows how the getDelayedTransmissionStatus() function works.

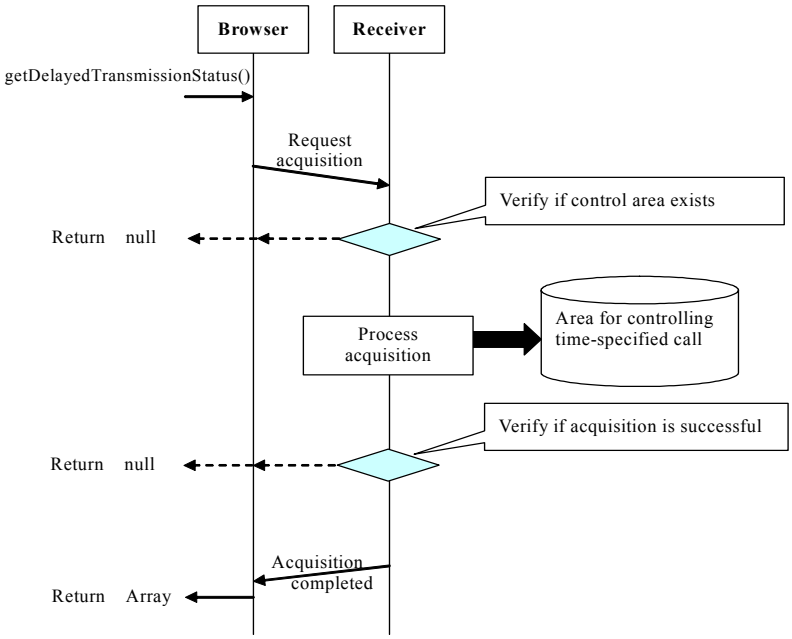


Figure 10 Sample Behaviour of getDelayedTransmissionStatus()

- Sample behaviour of the getDelayedTransmissionResult() function after registration

  Figure 11 shows how the getDelayedTransmissionResult() function works.

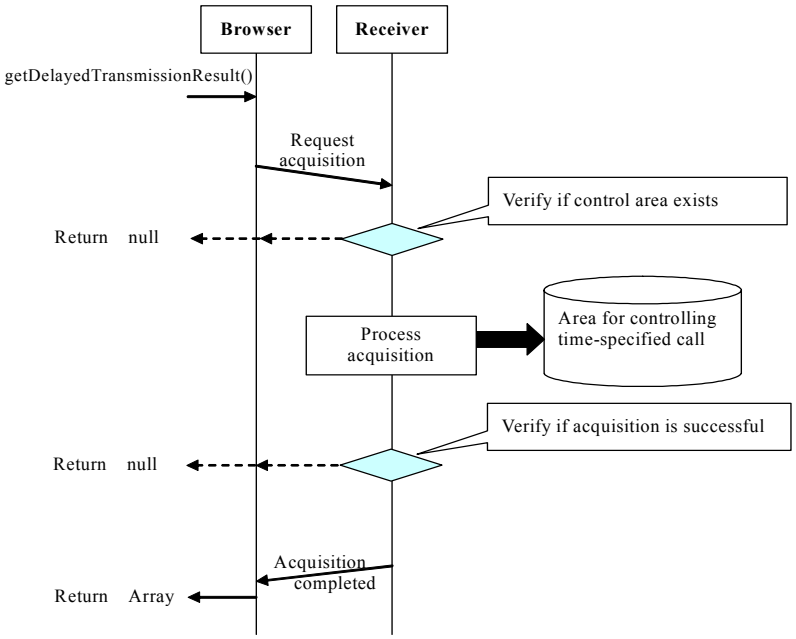
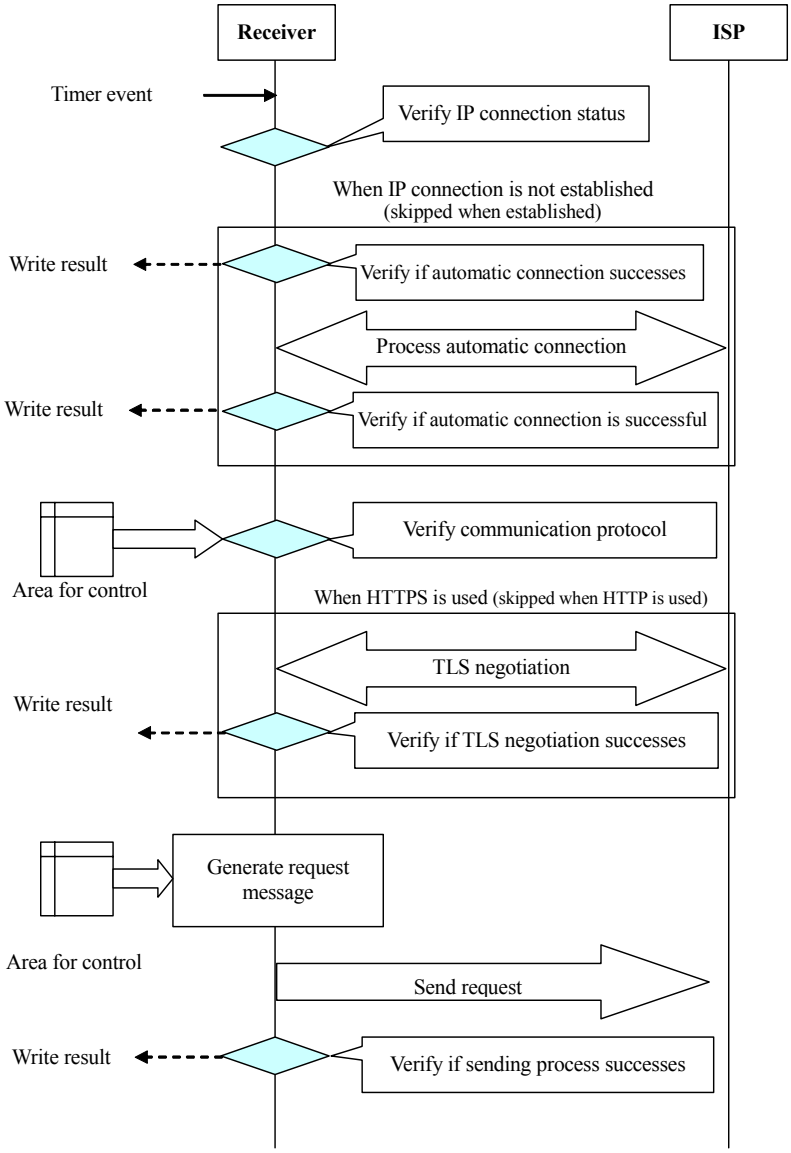


Figure 11 Sample Behaviour of getDelayedTransmissionResult()

- Sample behaviour of sending the datagram registered by the setDelayedTransmissionData() function

  Figure 12 shows how the datagram registered by the setDelayedTransmissionData() function is transmitted.



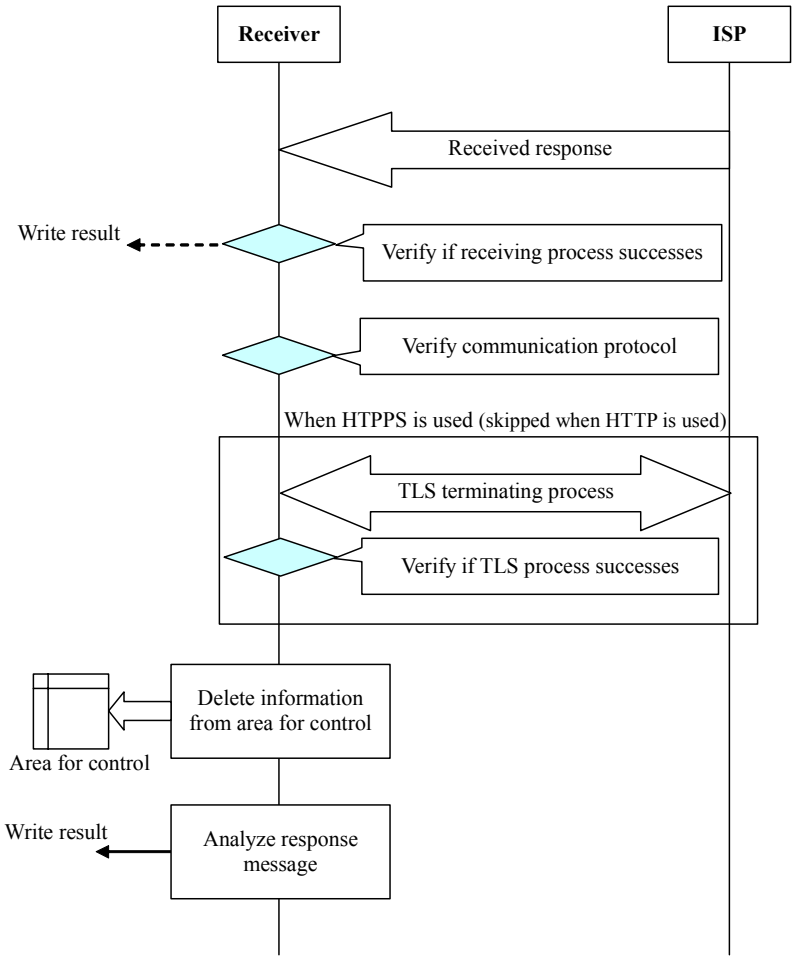
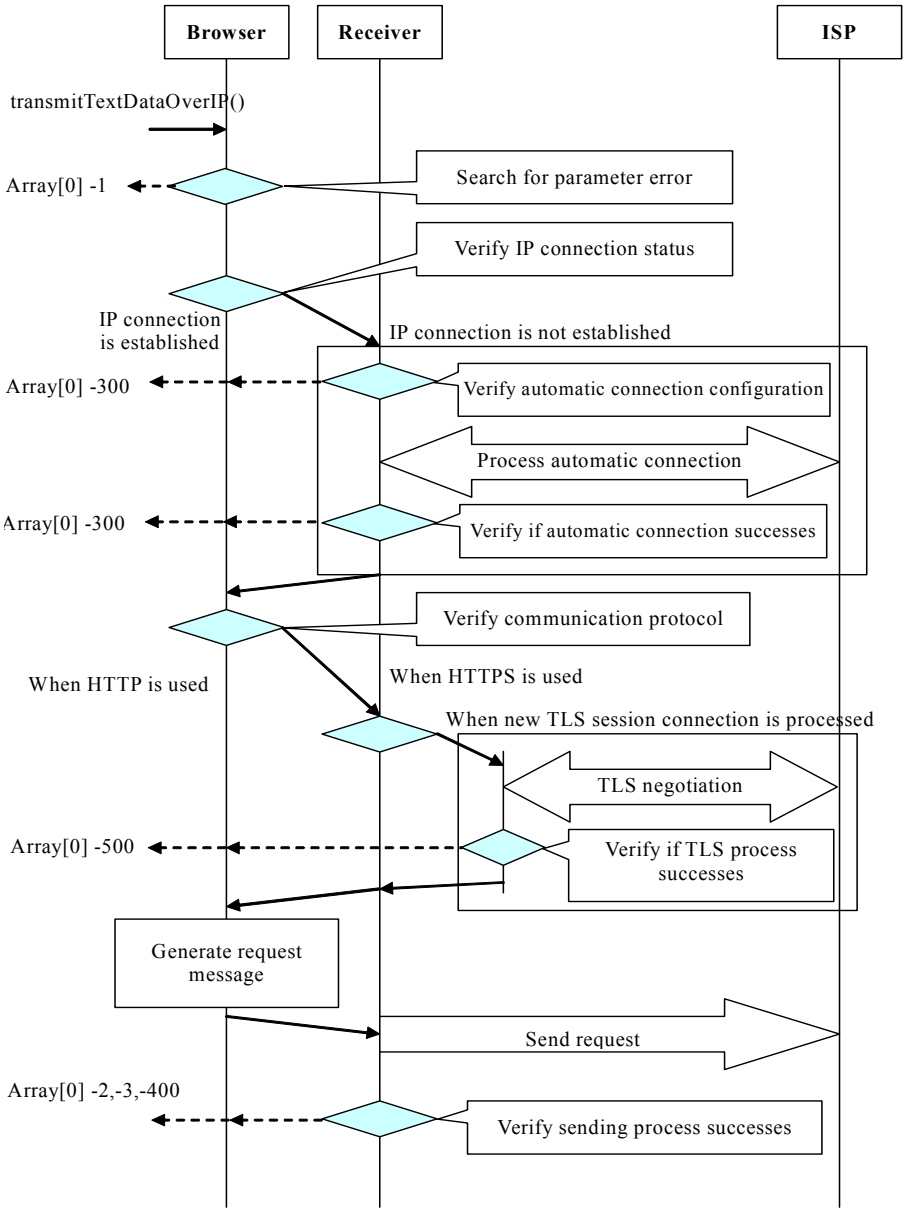


Figure 12 Sample Behaviour for sending datagram registered by time-specified call function

(4) Functions for receiving/sending datagram

- Sample behaviour of the transmitTextDataOverIP() function

Figure 13 shows how the transmitTextDataOverIP() function works.



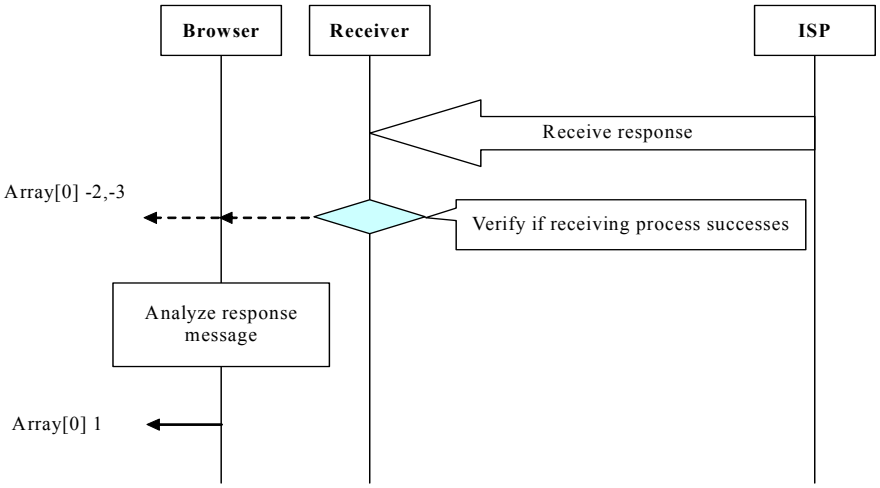


Figure 13 Sample Behaviour of transmitTextDataOverIP()

(5) Function for disconnecting

This section provides a sample behaviour of an extended function for broadcasting used to disconnect.

- Sample behaviour of the disconnectPPP() function

Figure 14 shows how the extended function for broadcasting, disconnectPPP() works.

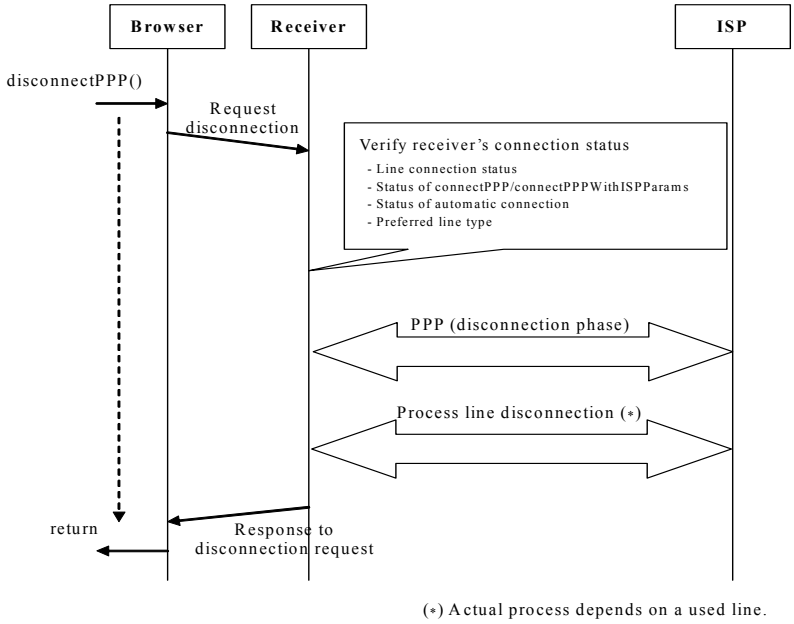


Figure 14 Sample Behaviour of disconnectPPP()

5 Bookmark Services

This section outlines bookmark services. This section also explains the relationship between bookmark functions in a receiver's application and bookmark services triggered from a BML content.

[Overview of bookmark services]

A bookmark service is a service that offers a tool that allows an end user to retrieve information related to a topic in which the end user is interested.

As long as a bookmark service is available, an end user can specify a bookmark using a remote control when the end user comes across an interesting topic in a program. In this case, a BML content for recording bookmarks saves into NVRAM, the bookmark information (URI and others) representing other services where the relevant information is broadcast. Once this is done, the end user can specify the desired bookmark information in a bookmark request window brought by another BML content or a receiver's application and select the related information services to watch. This bookmark function is also used to mark a frequently watched service to select the service with less effort.

The bookmark function is provided through a group of services triggered from a BML content or a receiver's application. This section explains only the services triggered from a BML content.

- Data services required for bookmark function

(1) Bookmark record service

- Bookmarks are recorded by a "bookmark record content". A bookmark record content records a bookmark following these steps:
  - A bookmark icon is presented based on a presenting timing preconfigured in the bookmark record content not to disturb the other presented information.
  - The end user is prompted to use a bookmark by the bookmark icon. When the current displayed information is deemed interesting, a certain button is pressed on a remote control.
  - As specified using the button, the link destinations (URIs) representing the channels responsible for broadcasting information related to the interesting topic are recorded in a bookmark area in NVRAM in a certain format.
- A bookmark record content may be broadcast not only as an independent content containing only the bookmark function but also as part of a content working with an event program or a CM content.
- It is assumed that the link destinations recorded as bookmark information include broadcast contents and contents on external servers.
- The available bookmark types and how they work are defined in an operational standard regulation specific to a broadcaster.
- A key used for data broadcasting is also used as a button to set a bookmark. A button "exclusively used to set a bookmark" may be implemented in a remote control.

(2) Bookmark list service

- A bookmark list service is provided by a "bookmark list display content" broadcast in a special channel (service) or a channel (service) for data added programs. A bookmark list is presented following these steps:
  - An end user selects a bookmark list service using a pre-list or a direct selection function.
  - The bookmark list service is selected to display a bookmark list.
  - The bookmark types available to (interpreted by) the bookmark list service are listed.
  - When the end user selects a desired bookmark from the bookmark list, the bookmark list display content and obtains the URI based on the bookmark information in NVRAM to select a channel responsible for broadcasting the related information.

- Some bookmarks acting as memorandums, notes or others have no link destination and present only related information.

- As a general rule, bookmark list services are broadcast constantly.
- A bookmark list service must list the available bookmarks and allow an end user to select any desired bookmark.
- A bookmark list display contents provide bookmark management functionality (delete bookmark, prohibit deleting, etc) for an end user.
- A broadcaster may customize a format in which a bookmark list is displayed for a better appearance.
- It is assumed that a bookmark list display function is implemented as a receiver's application. In this case, a bookmark list display button may be implemented in a remote control.

(3) Pre-list service

- Definition of pre-list service

  If two or more bookmark list services are broadcast, it is required that a function allows an end user to select a desired bookmark list service among them to successfully switch to the service.

  A pre-list is a list of the channels responsible for bookmark list services and acts as an auxiliary function used to select a desired bookmark list service.

- A pre-list display content displays a pre-list following these steps:
  - When an end user press a pre-list button for a certain channel, a pre-list display content is presented.
  - The pre-list display content displays a list of the channels responsible for bookmark list service (pre-list).
  - When the end user selects a desired bookmark list service from the list, the channel is presented.

(4) Service at the link destination

- A service at the link destination is a service responsible for broadcasting contents related to a bookmarked topic.

[Relationship bookmark functions by receiver's application and bookmark services by BML]

Figure 15 outlines the flow of information used for a bookmark service.

Information about a bookmark is stored in a basic data area, whose structure is common to any bookmark area, and an extended data area, whose structure is customized by a broadcaster. The dashed lines encloses a scope that may be implemented by a receiver. This section explains functions for accessing a bookmark data block.

- Bookmark registered by receiver's application
  - When an end user uses a receiver's application to register a bookmark, a basic data corresponding to the specified information is written to Basic Data Area 1. Basic Data Area 1 is applicable to channels and sites that are explicitly selected by an end user and accept a reading access only by a receiver's application (to secure private information).
- Bookmark registered by bookmark record service (BML content)
  - A content to record a bookmark is required to be broadcast for recording a bookmark. A receiver receives a bookmark record service and executes the transmitted bookmark record

BML contents to write information about a bookmark to Basic Data Area 2 and Extended Data area that is defined by a broadcaster or verify whether or not an extended data area exists.

- Resident bookmark manager
  - A resident bookmark manager provides a bookmark list function triggered by a receiver's application. The bookmark list function is applicable to both Basic Data Area 1 and Basic Data Area 2 to read out information to present it.
  - When a receiver has a bookmark list display function available to a receiver's application, the receiver has a management/control function that allows an end user to manipulate bookmarks. A resident bookmark manager has a function to read information from a basic data area to present it, a function to delete a bookmark saved in a basic data area, and a function to set a flag prohibiting/accepting a deletion operation of a bookmark. Basic Data Area 2 is associated with a group of extended information. When a bookmark saved in Basic Data Area 2 is deleted by an end user, the extended data associated with the bookmark is also deleted. It is recommended that when a writing access occurs and the available area is smaller than needed, a presenting function of a receiver informs an end user of the lack to prompt the user to explicitly delete existing information, as required. In this case, the deletion is executed through a resident bookmark manager.
  - A resident bookmark manager may read information from an extended data area in spite of the fact that Figure 15 does not show this operation.
- Bookmark list service and bookmark area
  - A bookmark list service provides a service through a BML document that has been transmitted by an event program that is responsible for the service. A receiver receives the event program and executes the transmitted BML document to present a list of bookmarks. This allows an end user to select a desired channel. Any bookmark information saved in Basic Data Area 2 and an extended data area defined by a broadcaster can be deleted and updated. Basic Data Area 2 and an extended data area defined by a broadcaster can be configured to set a flag to prohibit/accept a deletion operation of a bookmark. The list of bookmarks can be customized into a desired format by a broadcaster.
  - No bookmark list service is allowed to read information from Basic Data Area 1.
- Prelist service and bookmark area
  - A pre-list service is a service to list the channels responsible for bookmark list services. A pre-list service is transmitted through an event program. A pre-list service allows an end user to locate bookmark list services. It is assumed that two or more broadcasters provide bookmark list services. To allow for the assumption, a pre-list service acts as a portal to various bookmark list services.
  - No pre-list service is allowed to access information in any bookmark area.
- Service at the link destination and bookmark area
  - A service at the link destination is a service responsible for broadcasting contents related to a bookmarked topic. A desired channel is selected and presented by using a bookmark list service.
  - A service at the link destination may update information about a bookmark saved in Basic Data Area 2 and an extended data area.

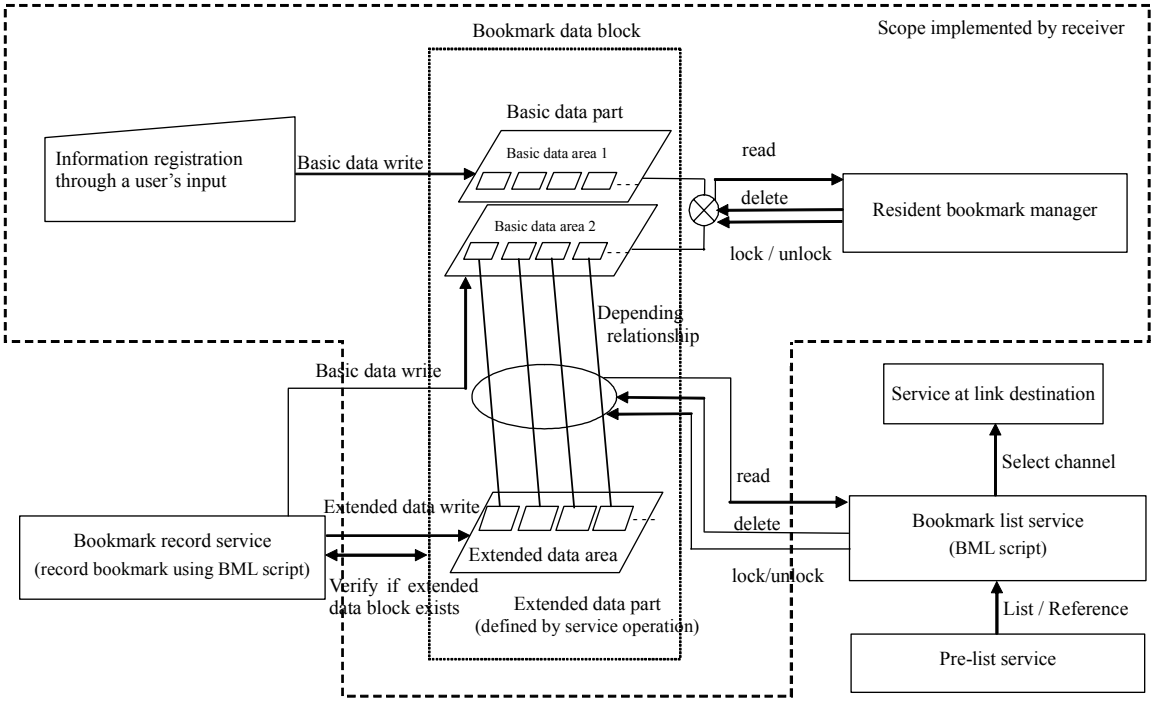


Figure 15 Flow of Data for Bookmark Services

6 Sample Usage of Function for Communicating Information with External Device

This section provides samples of using the getArrayFromPeripheral() function to communicate location information with an external device. Although these samples assume that an external device is a car navigation system, the samples can be applied to any other external device that is equipped with a GPS (Global Positioning System).

To operate the getArrayFromPeripheral() function, the following four arguments of the function must be specified: peripheral_Ref, method, timeout, and data.

The value for peripheral_Ref is obtained by using the enumPeripherals() function. For more information on using this function, refer to the section "5 Sample of controlling external device by using External XML document", Informative Explanation, Volume 2.

How to obtain the value for the timeout argument is defined in an operational standard regulation. Sample values for the method and data arguments are shown below.

(A) To obtain the latitude and the longitude

Input:

method : “position”

data : Omitted

Output:

Array[1] : the Number object representing the latitude

Array[2] : the Number object representing the longitude

(B) To obtain the road code or area code

Input:

method : “areacode”

data : Array[0] : the Number object that:

discriminates toll roads/highways from other roads, when Array[0] = 0

requests the road code representing a road on which the external device is, when Array[0] = 1

requests the municipality code (as defined in the Japanese Industrial Standards) representing an area in which the external device is, when Array[0] = 2

Output:

Array[1] : the Number object containing:

the returned value "1" indicating that the road is identified as a toll road/highway or the returned value "0" indicating that the road is not identified as a toll road/highway, when Array[0] = 0

the road code on which the external device is, when Array[0] = 1

the municipality code (JIS defined) in which the external device is, as defined in JIS, when Array[0] = 2

(C) To verify whether the external device is in a specified rectangle and obtain the returned value

Input:

method : “area”

data :

Array[0] : the Number object representing the latitude of the upper left corner of the specified rectangle

Array[1] : the Number object representing the longitude of the upper left corner of the specified rectangle

Array[2] : the Number object representing the latitude of the lower right corner of the specified rectangle

Array[3] : the Number object representing the longitude of the lower right corner of the specified rectangle

Output:

Array[1] : the Number object containing:

the returned value "0" indicating that the external device is outside of the specified rectangle or the returned value "1" indicating that the external device is in the specified rectangle

- About latitude/longitude representations

Each latitude or longitude is represented as an integer part of a value that is obtained by multiplying a real number in degrees by 10^7.

A positive number is used to represent a North latitude or East longitude. A negative number is used to represent a South latitude or West longitude. For example, "latitude 20 degrees 30 minutes 15 seconds South is represented as "-205041666".

[BLANK]

APPENDIX 2

Operational Guidelines for Implementing Basic Services

Contents

Chapter APPENDIX 2-1 Purpose 467

Chapter APPENDIX 2-2 Scope 468

Chapter APPENDIX 2-3 Legend 469

Chapter APPENDIX 2-4 Guidelines for Coding of BML Documents 470
A2-4.1 Character Codes 470
A2-4.1.1 Character coding scheme used for BML documents 470
A2-4.1.2 External characters 470
A2-4.1.3 Operation of character reference 471
A2-4.1.4 XML Namespace 472
A2-4.2 Operational Range of Media Types and Monomedia 472
A2-4.3 BML Elements 473
A2-4.3.1 Entity 473
A2-4.3.2 Operational guidelines of BML elements 473
A2-4.3.3 Attributes 477
A2-4.3.4 Operation of beitem element 481
A2-4.4 CSS-Based Style Sheet 481
A2-4.4.1 Elements to which each property is applied 483
A2-4.4.2 Operation of selector 484
A2-4.4.3 Media type 485
A2-4.4.4 Box model 485
A2-4.4.5 Visual formatting model 485
A2-4.4.6 Other visual effects 486
A2-4.4.7 Background 487
A2-4.4.8 Fonts 487
A2-4.4.9 Text 488
A2-4.4.10 Pseudo class 488
A2-4.4.11 Extended properties 489
A2-4.4.12 Default style sheet 491
A2-4.4.13 Operation of length property 491
A2-4.5 Operational Guidelines for Procedural Description 491
A2-4.5.1 Operational guidelines of DOM 491
A2-4.5.2 Operation range of built-in objects 503
A2-4.5.3 Operation range of extended objects for broadcasting 506
A2-4.5.4 Operation range of browser pseudo objects 506

A2-4.6 Operational Scope for Navigator pseudo objects 509
A2-4.7 Restrictions on Description of BML Documents 509
A2-4.8 Guidelines for Presentation Control of BML Document 509
A2-4.8.1 Positioning and Restrictions 509
A2-4.8.2 Vertical writing 510
A2-4.8.3 Restrictions on arrangement of elements 511
A2-4.8.4 Restrictions on the elements which are represented on the composition plane of video and still picture 511
A2-4.8.5 Operational range of object element 512

Chapter APPENDIX 2-5 Guidelines for the behaviour of BML Documents 517
A2-5.1 Operational Guidelines of Presentation 517
A2-5.1.1 Operational guidelines on Z-index 517
A2-5.1.2 Presentation plane 517
A2-5.1.3 Operation of monomedia presentation 518
A2-5.1.4 Guideline on clipping 522
A2-5.1.5 Values applicable to CSS2 properties 522
A2-5.1.6 Rules of box model 523
A2-5.1.7 Operation of colour map data 523
A2-5.1.8 Operation of remote control keys 525
A2-5.1.9 Character entry into input element 526
A2-5.2 Guidelines on behavior of management of external characters 526
A2-5.2.1 Management of pattern data 526
A2-5.2.2 Sharing of external characters among documents 527
A2-5.2.3 Presentation of external characters 527
A2-5.3 Behavior of DOM 527
A2-5.3.1 Use of DOM 527
A2-5.3.2 Handling of control codes (blank, line break, tabs) 528
A2-5.3.3 Configuration example of DOM interface 529
A2-5.3.4 Generation of DOM tree 530
A2-5.3.5 Restriction of DOM work memory 530
A2-5.4 Operational Guideline of Script Language 530
A2-5.4.1 Operation of script working environment 530
A2-5.4.2 Data type 531
A2-5.4.3 Effects on basic objects caused by data type restrictions 533
A2-5.4.4 Operational general rule of implementation-dependent behavior 535
A2-5.5 Extended Objects for Broadcasting 536
A2-5.5.1 Operation of CSVTable class 536
A2-5.5.2 Operational guideline on BinaryTable class 536
A2-5.6 Guideline for Behavior of Browser Pseudo Objects 539
A2-5.6.1 Operation of Ureg 539
A2-5.6.2 Guideline for behavior of EPG functions 540
A2-5.6.3 Operation of non-volatile memory function 540
A2-5.6.4 Operation of subtitle presentation control 541
A2-5.6.5 Operation of interaction channel function 541

A2-5.6.6 Line status after the execution of the functions for interaction channel 544
A2-5.6.7 Operation of operational control functions 548
A2-5.6.8 Operation of receiver build-in sound control 551
A2-5.6.9 Operation of timer function 551
A2-5.6.10 Operation of external character function 551
A2-5.6.11 Name of the extended functions which are proper to service operators 551

Chapter APPENDIX 2-6 Contents Transmission and Namespace 552
A2-6.1 Scope Mapping to Transmission System 552
A2-6.2 Contents Transmission 553
A2-6.2.1 Module syntax 553
A2-6.2.2 Resource list in multipart format module 553
A2-6.2.3 Operation on event messages 553
A2-6.3 Namespace 553
A2-6.3.1 Namespace used for acquiring resources via telecommunication channels 553

Annex A Default Style Sheet 554

Annex B DTD for Checking the Operation Range for Basic Services 555
B.1 DTD 555
B.2 Description of DTD declaratives 558

Annex C Counting rule for the restriction on the memory size of ECMAScript 560

[BLANK]

Chapter APPENDIX 2-1 Purpose

The Appendix 2 contains the operational guidelines for implementing basic services by using the XML-based multimedia coding scheme specification responsible for the data broadcasting scheme, part of the digital data broadcasting scheme specified as the standard in Japan.

Chapter APPENDIX 2-2 Scope

The guidelines contained in this appendix are applied to a basic service that uses XML-based multimedia coding for data broadcasting carried out as part of digital data broadcasting.

Chapter APPENDIX 2-3 Legend

The following terms and symbols are used to describe a operational guideline in APPENDIX 2:

O This symbol marks an item required to implement a basic service. It is desirable that a receiver designed for the basic services should represent this symbol normally if the corresponding signal is inserted into the content.

Δ This symbol marks an item required to implement a basic service. However, it is assumed that this symbol is not inserted in the content of basic services and that a receiver designed for the basic services always interprets that the default value for this symbol is designated.

\- This symbol marks an item not required to implement a basic service. It is assumed that this symbol is not inserted in the content of basic services and that a receiver designed for the basic services does not represent it normally if the corresponding signal is inserted into the content.

In addition, when describing the operational restrictions on attributes and properties, the following legend is used:

RW: For implementing basic services, the corresponding attribute or property can be read and written. It is desirable that a receiver for basic services operates normally when writing and reading the corresponding attribute or property in the content.

R: For implementing basic services, the corresponding attribute can be read but cannot be written. It is desirable that a receiver for basic services operates normally when reading the corresponding attribute or property in the content. On the other hand, it is assumed that the writing operation to this item may be ignored.

Chapter APPENDIX 2-4 Guidelines for Coding of BML Documents

As a language, only the language for multimedia representation is required within the range of basic services. Therefore, it is assumed that only BML is operated without using B-XML. The following subsections define the operational guideline on BML elements and attributes for implementation of basic services.

A2-4.1 Character Codes

A2-4.1.1 Character coding scheme used for BML documents

Only the EUC-JP code is operated for basic services. Table 4-1 shows the operational guideline on the EUC-JP code.

Table 4-1 EUC-JP Code Set

Code set	Character set	Operating range
Code set 0 Byte range 21~7E	JIS X 0201-1976 (JIS Roman Alphabets) (Presentation of the bits combination 7/15(DEL) depends on a receiver.)	O
Code set 1 1st byte range A1~FE 2nd byte range A1~FE	JIS X 0208-1990 (Additional symbols defined for the sections 90 to 94 on page 101and 102 (table 10-11 and 10-12) in the ARIB-STD-B5 are specified for the reserved area of the sections 90 to 94. The sections 87 to 88 are used for the area defined by external characters. And the point 13 to 18 of the section 1 and the point 94 of the section 2 are not used.)	O
Code set 2 1st byte 8E 2nd byte range A1~DF	Half-size katakana characters	-
Code set 3 1st byte 8F 2nd byte range A1~FE 3rd byte range A1~FE	JIS X 0212-1990	-
Control codes	Blank character (20)	O
	Delete character (7F)	-
	Line feed (0D0A)	O
	Tab (09)	O
	SS2 (8E)	-
	SS3 (8F)	-

A2-4.1.2 External characters

A DRCS can be used to present external characters. This section provides the operational guideline for coding and using external characters. The operational guideline for managing external character is defined in Section A2-5.2. The operational guideline on the loadDRCS( ) function is described in Section A2-5.6.10The code area available for the extern characters is the sections 87 to 88, which is common to the EUC-JP used for BML document and the 8-bit character code used for the externally referenced text. As it is assumed that the pattern data set in this code area is managed in a common area for the EUC-JP and the 8-bit character codes, the following conditions must be satisfied when the

pattern data for the area used by the external characters is set in the [1] and when the external characters are referenced:

- The external characters referenced by a BML document must be set before the presentation of the document. The loadDRCS( ) function must be executed at the time when global codes of ECMAScript are executed. It is not executed in the event handler of ECMAScript. And, loadDRCS( ) may be executed in global codes not more than twice.
- The external characters referenced by a BML document includes those which are used for the telegram of the interaction channel and those which are used for the string in the privateData area transmitted by an event message as well as the string data of EUC-JP and 8-bit character coded text data.
- Multiple pattern data can be set to the same character code if they are the combination of the different font family and size by the execution of loadDRCS(). If the pattern data are set for the same font family, font size, and the character code with overlap by loadDRCS in the global codes, the process depends on a receiver.

Concerning the structure of DRCS data that are read by loadDRCS( ), the following restrictions are applied to the data structure defined in ARIB STD-B24 Volume1 Annex D.

- The range of the numberOfCode value is from 1 to 188.
- The characterCode value corresponds to from the point 1 of the section 87 to the point 94 of the section 88 in the character code table. The value is a section and point code. For example, 0x2121 is transmitted when transmitting the character in the point 1 of section 1.
- The following value can be specified for the fontID property. It is not set to 0.

Value of fontId	Typeface
1	Round gothic
2	Square Gothic style
3	Bold Round Gothic

- The mode must be 0001. That is, the pattern data is multi-scaled and is not compressed. It is not compressed with geometric coding.
- The depth must be 2. That is, the pattern data has 4-step scaled values.
- The same restrictions as the pixel value for font-size property are applied to width and height.

A2-4.1.3 Operation of character reference

Character reference by “&#” is not operated.

If external characters are set, the EUC-JP codes should be designated directly.

Character reference using “\” of CSS and the SingleEacapeCharacters such as “\a” are not operated.

“% escape” in URI is not operated.

HexEscapeSequence, OctalEscapeSequence, and UnicodeEscapeSequence in ECMAScript are not operated. And, among SingleEscapeCharacters, “\b”, “\t”, and “\f” are not operated but “\””, “\’”, “\r”, “\n”, and “\\” are operated. However,

“\” should be followed by “0x5c”.

“\r” and “\n” should be used as “\r\n”.

---

[1] The purposes of these conditions are to prevent unexpected miss-presentation of characters caused by setting the pattern data with the same external characters again and to avoid confusion at the content presentation due to the fact that receiver cannot guarantee the timing for re-presentation.

A2-4.1.4 XML Namespace

The XML Namespace must not be used.

A2-4.2 Operational Range of Media Types and Monomedia

Table 4-2 shows the operational guideline on the media types required for basic services and the monomedia coding schemes designated by object element.

Table 4-2 Operational guideline on media types and the monomedia coding schemes

Schema	Media type	Operation	Operation (object element)	Remarks
arib-dc:	multipart/mixed	O	-	
	text/xml;charset="euc-jp"	-	-	
	text/xml;charset="UTF-8"	-	-	
	text/xml;charset="UTF-16"	-	-	
	text/xml;charset="Shift_JIS"	-	-	
	text/css	O	-	(Note 1)
	text/xsl;charset="euc-jp"	-	-	
	text/xsl;charset="UTF-8"	-	-	
	text/xsl;charset="UTF-16"	-	-	
	text/xsl;charset="Shift_JIS"	-	-	
	text/X-arib-bml; charset="euc-jp"	O	-	(Note 4)
	text/X-arib-bml; charset="UTF-8"	-	-	
	text/X-arib-bml; charset="UTF-16"	-	-	
	text/X-arib-bml; charset=" Shift_JIS "	-	-	
	text/plain; charset="euc-jp"	-	-	
	text/plain; charset="UTF-8"	-	-	
	text/plain; charset="UTF-16"	-	-	
	text/plain; charset="Shift_JIS"	-	-	
	text/X-arib-jis8text	O	O	
	text/X-arib-ecmascript; charset="euc-jp"	O	-	(Note 2, 4)
	text/X-arib-ecmascript; charset="UTF-8"	-	-	
	text/X-arib-ecmascript; charset="UTF-16"	-	-	
	text/X-arib-ecmascript; charset="Shift_JIS"	-	-	
	image/jpeg	O	O	
	image/png	-	-	
	image/X-arib-png	O	O	
	image/X-arib-mng	O	O	
	image/X-arib-mpeg2-I	-	-	
	audio/X-arib-mpeg2-aac	O	O	
	audio/X-arib-mpeg2-bc	-	-	(Note 3)
	audio/X-arib-aiff	O	O	
	audio/X-arib-additional	-	-	
	video/X-arib-mpeg1	-	-	
	application/X-arib-stream-text; charset="euc-jp"	-	-	
	application/X-arib-stream-text; charset="UTF-8"	-	-	
	application/X-arib-stream-text; charset="UTF-16"	-	-	
	application/X-arib-stream-text; charset="Shift_JIS"	-	-	
	application/X-arib-stream-jis8text	-	-	
	application/X-arib-stream-png	-	-	

Schema	Media type	Operation	Operation (object element)	Remarks
	application/X-arib-stream-jpeg	-	-	
	application/X-arib-stream-mpeg2-I	-	-	
	application/X-arib-bmlclut	O	-	
	application/X-arib-btable	O	-	
	application/X-arib-drcs	O	-	
	application/X-arib-PDI	-	-	
	application/X-arib-resourceList	O	-	
	application/X-arib-contentPlayControl	-	-	
arib:	audio/X-arib-mpeg2-aac	O	O	
	audio/X-arib-mpeg2-bc	-	-	(Note 3)
	audio/X-arib-mpeg4	-	-	
	video/X-arib-mpeg1	O	O	
	video/X-arib-mpeg2	O	O	
	video/X-arib-mpeg4	-	-	
arib-ic:	image/X-arib-mpeg2-I	-	-	
	application/X-arib-stream-mpeg2-I	-	-	
romsound:	audio/X-arib-romsound	O	-	

Note 1: CSS data appears only in BML document and is not transmitted as an independent resource.
Note 2: ECMAScript data appears only in BML document and is not transmitted as independent resources. This is because registerTransmission() is excluded from the operational guideline specified in this ANNEX.
Note 3: It is also assumed that audio/X-arib-mpeg2-bc instead of audio/X-arib-mpeg2-aac is operated, depending on the necessity of each transmission media.
Note 4: The “charset” specification cannot be omitted.

A2-4.3 BML Elements

This section specifies the operational guideline on BML elements and attributes.

A2-4.3.1 Entity

The following entities may be used without any specific definitions and the others are not required to be operated.

- lt <
- gt >
- amp &
- apos '
- quot "

A2-4.3.2 Operational guidelines of BML elements

Operational guidelines of BML elements for basic services are shown in Table 4-3.

Table 4-3 Operational Guideline of BML Elements

Module		BML(Operation)	Element
Core	Structure (required)	O	body
		O	head

Module		BML(Operation)	Element
		-	html
		O	title
	Text (required)	-	abbr
		-	acronym
		-	address
		-	blockquote
		O	br
		-	cite
		-	code
		-	dfn
		O	div
		-	em
		-	h1 ~ h6
		-	kbd
		O	p
		-	pre
		-	q
		-	samp
		O	span
		-	strong
		-	var
	Hypertext (required)	O	a
	List (required)	-	dl
		-	dt
		-	dd
		-	ol
		-	ul
		-	li
Applet (deprecated)		-	applet
		-	param
Text Extension	Presentation	-	b
		-	big
		-	hr
		-	i
		-	small
		-	sub
		-	sup
		-	tt
	Edit	-	del
		-	ins
	Bi-directional	-	bdo
Form	Basic Forms	-	form
		-	input
		-	label
		-	select
		-	option
		-	textarea
	Forms	-	form
		O	input
		-	select
		-	option

Module		BML(Operation)	Element
		-	textarea
		-	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		-	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&*
		-	*form&*
		O	*body&*
		-	*label&*
		O	*input&*
		-	*select&*
		-	*textarea&*
		-	*button&*
Metainformation		O	meta
Scripting		-	noscript
		O	script

Module		BML(Operation)	Element
Stylesheet		O	style
Style Attribute		O	
Link		-	link
Base		-	base
Name Identification (deprecated)		-	*a&*
			applet&
			form&
			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	O	*bml*
		O	*bevent*
		O	*beitem*
		O	*body&*
		O	*div&*
		O	*p&*
		-	*a&*
		-	*bdo&*
		O	*object&*
	Basic BML	-	*bevent*
		-	*beitem*
		-	*body&*
		-	*div&*
		-	*p&*
		-	*span&*

Module		BML(Operation)	Element
		-	*object&*

A2-4.3.2.1 Restriction on the sequence of elements in head element

In this guideline, the following restrictions are set for the elements in the head element:

- In the head element, a meta element, a style element, a script element and a bevent element appear once at maximum according to this sequence following the mandatory title element.

A2-4.3.3 Attributes

Table 4-4 specifies the operational guideline on attributes of the elements (div p a span input object meta title script style body head bml bevent beitem) that are specified for operation in Section A2-4.3.2.

Table 4-4 Operational Guideline regarding Attributes of Elements

element	Attribute	Operation	Operational restriction
Common Attributes			
Core Attributes			
	id	O	Strings whose maximum length is 128 byts
	class	O	
	title	-	
I18N Attributes			
	xml:lang	Δ	Must be fixed to “ja”
Events Attributes			
	onclick	O	
	ondbclick	-	
	onmousedown	-	
	onmouseup	-	
	onmouseover	-	
	onmousemove	-	
	onmouseout	-	
	onkeypress	-	
	onkeydown	O	
	onkeyup	O	
Style Attributes			
	style	O	
Core Modules			
Structure Module			
body	%Common.attrib;		
	%Core.attrib;	O	
	%I18n.attrib;	Δ	
	%Events.attrib;	-	
head	%I18n.attrib;	Δ	
	profile	-	
title	%I18n.attrib;	Δ	

element	Attribute	Operation	Operational restriction
Text Module			
br	%Core.attrib;	O	
div	%Common.attrib;	O	
p	%Common.attrib;	O	
span	%Common.attrib;	O	
Hypertext Module			
a	%Common.attrib;	O	
	accesskey	O	
	charset	Δ	Must be fixed to “EUC-JP”
	href	O	
	hreflang	-	
	rel	-	
	rev	-	
	tabindex	-	
	type	-	
Forms Modules			
Forms Module			
input	%Common.attrib;	O	
	accesskey	O	
	checked	-	
	disabled	O	
	readonly	O	
	maxlength	O	1 through 40. (Note 1)
	alt		
	name	-	
	size	-	
	src	-	
	tabindex	-	
	accept	-	
	type	O	"text" or "password"
	value	O	
Client-side Image Map			
a&	coords	-	
	shape	-	
input&	usemap	-	
object&	usemap	-	
Server-side Image Map			
input&	ismap	-	
Object Module			
object	%Common.attrib;	O	
	archive	-	
	classid	-	
	codebase	-	
	codetype	-	
	data	O	
	declare	-	
	height	-	

element	Attribute	Operation	Operational restriction
	name	-	
	standby	-	
	tabindex	-	
	type	O	
	width	-	
Target Module			
a&	target	-	
Intrinsic Events Module			
a&	onblur	O	
	onfocus	O	
body&	onload	O	
	onunload	O	(Note 2)
input&	onfocus	O	
	onblur	O	
	onselect	-	
	onchange	O	
Metainformation Module			
meta	%I18n.attrib;	Δ	
	http-equiv	-	
	name	O	
	content	O	
	scheme	-	
Scripting Module			
script	charset	Δ	Must be fixed to “EUC-JP”
	type	Δ	Must be fixed to "text/X-arib-ecmascript; charset="euc-jp&quot:”
	src	-	
	defer	-	
	xml:space	-	
Style Sheet Module			
style	%I18n.attrib;	Δ	
	type	Δ	Must be fixed to "text/css"
	media	Δ	Must be fixed to "tv"
	title	-	
	xml:space	-	
BML Module			
bml	%I18n.arrtib;	Δ	
	version	-	
	xmlns	-	
bevent	id	O	
beitem	id	O	
	type	O	Must be one of the following: "EventMessageFired", "ModuleUpdated", "ModuleLocked", "TimerFired", "DataEventChanged", "CCStatusChanged", "MainAudioStreamChanged", "NPTReferred", "MediaStopped", "DataButtonPressed"
	onoccur	O	

element	Attribute	Operation	Operational restriction
	es_ref	O	
	message_group_id	Δ	Must be fixed to "0"
	message_id	O	
	message_version	O	
	module_ref	O	
	language_tag	O	
	register_id	-	
	service_id	-	
	event_id	-	
	peripheral_ref	-	
	time_mode	O	Must be one of the following: "absolute", "origAbsolute", "NPT"
	time_value	O	
	object_id	O	Limited to object id whose type attribute is "audio/X-arib-aiff" or "audio/X-arib-mpeg2-aac", and belongs to the oject element indicating a data send through a data carousel
	subscribe	O	
body&	invisible	O	
div&	accesskey	O	
	onfocus	O	
	onblur	O	
p&	accesskey	O	
	onfocus	O	
	onblur	O	
span&	accesskey	O	
	onfocus	O	
	onblur	O	
a&	effect	-	
bdo&	orientation	-	
object&	streamposition	O	If a monomedia referenced by the object element is MNG(type “image/X-arib-mng”), a frame number must be specified. Otherwise, must be set to “0”
	streamlooping	Δ	Must be fixed to "1"
	streampositionnumerator	-	
	streampositiondenominator	-	
	streamstatus	O	An initial value for a monomedia referenced by the object element must be specified. See Table 4-22.
	streamlevel	-	
	remain	O	Applicability depends on a monomedia referenced by the object element. See Table 4-22.
	accesskey	O	
	onfocus	O	
	onblur	O	

Note 1: If the data for input element is entered exceeding maxlength, the exceeded portion is discarded. When the data exceeds the input frame, the overflowed portion is not presented.

Note 2: The writePersistentArray() function and the unlockModuleOnMemory() function are the two extended functions for broadcastingapplicable to the onunload event handler. It is desirable to limit its process which can be completed in the short period, such as setting for Ureg to enable quick transition of the documents and decision on simple conditions.

A2-4.3.4 Operation of beitem element

To dynamically change attribute values using DOM, first the subscribe attribute is set to “false”. Then, values are changed as required before the subscribe attribute is set to “true”.

A2-4.4 CSS-Based Style Sheet

Table 4-5 shows the operational guideline on CSS 2 properties.

Table 4-5 Operational Guideline of CSS properties

property	operation
Selector	
*	O
E	O
E F	-
E:focus	O
E:active	O
E.myclass	O
#myid	O
Value assignment/ Inheritance	
@import	-
!important	-
Media type	
@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	-
border-right-color	-
border-bottom-color	-

property	operation
border-left-color	-
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	-

property	operation
Other visual effects	
visibility	O
overflow	O
clip	-
Generated content/ Auto numbering/ List	
content	-
quotes	-
counter-reset	-
counter-increment	-
marker-offset	-
list-style-type	-
list-style-image	-
list-style-position	-
list-style	-
Page media	
"@page"	-
size	-
marks	-
page-break-before	-
page-break-after	-
page-break-inside	-
page	-
orphans	-
widows	-
Background	
background	-
background-color	-
background-image	O
background-repeat	O
background-position	-
background-attachment	-
Font	
color	-
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	O

property	operation
word-spacing	-
text-transform	-
white-space	O
Pseudo class/ Pseudo element	
:link	-
:visited	-
:active	O
:hover	-
:focus	O
:lang	-
:first-child	-
:first-line	-
:first-letter	-
:before	-
:after	-
Table	
caption-side	-
border-collapse	-
border-spacing	-
table-layout	-
empty-cells	-
speak-header	-
User interface	
outline-color	-
outline-width	-
outline-style	-
outline	-
cursor	-
Voice style sheet	
volume	-
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	-

property	operation
peak-numeral	-
Extended property	
clut	O
color-index	O
background-color-index	O
border-color-index	-
border-top-color-index	O
border-right-color-index	O
border-bottom-color-index	O
border-left-color-index	O

property	operation
outline-color-index	-
resolution	O
display-aspect-ratio	O
grayscale-color-index	O
nav-index	O
nav-up	O
nav-down	O
nav-left	O
nav-right	O
used-key-list	O

A value defined as a fixed value, should be specified as the most important rule (!important) in the default style sheet. This most important rule always takes priority over the normal rules. This most import rule is not operated in the BML document and the normal rules are operated. Therefore the value specified as a fixed value is always obtained.

A2-4.4.1 Elements to which each property is applied

Table 4-6 shows the correspondence between the properties to be operated and the BML elements to which each property is applied.

Table 4-6 Elements to which CSS properties are applied

	div	p	br	span	a	input	object	body
Box model								
margin	Δ	Δ	-	-	-	Δ	Δ	-
padding-top	Δ	O	-	-	-	O	Δ	-
padding-right	Δ	O	-	-	-	O	Δ	-
padding-bottom	Δ	O	-	-	-	O	Δ	-
padding-left	Δ	O	-	-	-	O	Δ	-
border-width	O	O	-	O	O	O	Δ	-
border-style	O	O	-	O	O	O	Δ	-
Visual formatting model								
position	Δ	Δ	Δ	Δ	Δ	Δ	Δ	-
left	O	O	-	-	-	O	O	-
top	O	O	-	-	-	O	O	-
width	O	O	-	-	-	O	O	-
height	O	O	-	-	-	O	O	-
z-index	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
line-height	-	O	Δ	Δ	Δ	O	-	-
display	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
Other visual effects								
visibility	O	O	-	Δ	Δ	O	O	Δ
overflow	Δ	Δ	-	-	-	Δ	Δ	-
Background								

	div	p	br	span	a	input	object	body
background-image	-	-	-	-	-	-	-	O
background-repeat	-	-	-	-	-	-	-	Δ
Font								
font-family	-	O	-	O	O	O	-	-
font-size	-	O	-	O	O	O	-	-
font-weight	-	O	-	O	O	O	-	-
Text								
text-align	-	O	-	-	-	O	-	-
letter-spacing	-	O	-	Δ	Δ	O	-	-
white-space	-	Δ	-	-	-	Δ	-	-
Extended property								
clut	-	-	-	-	-	-	-	O
color-index	-	O	-	O	O	O	-	-
background-color-index	O	O	-	O	O	O	Δ	O
border-top-color-index	O	O	-	O	O	O	-	-
border-right-color-index	O	O	-	O	O	O	-	-
border-left-color-index	O	O	-	O	O	O	-	-
border-bottom-color-index	O	O	-	O	O	O	-	-
resolution	-	-	-	-	-	-	-	O
display-aspect-ratio	-	-	-	-	-	-	-	O
grayscale-color-index	-	O	-	O	O	O	-	-
nav-index	O	O	-	O	O	O	O	-
nav-up	O	O	-	O	O	O	O	-
nav-down	O	O	-	O	O	O	O	-
nav-left	O	O	-	O	O	O	O	-
nav-right	O	O	-	O	O	O	O	-
used-key-list	-	-	-	-	-	-	-	O

A2-4.4.2 Operation of selector

The selectors are used only in the default style sheet and in the content of the style element. Only the following of the selectors defined in CSS 1 are operated:

- For all elements Ex.: * { color-index: 1 }
- Type selector Ex.: div { color-index: 1 }
- Class selector Ex.: div.class1 { color-index: 1 } or .class1 { color-index: 1 }
- ID selector Ex.: #chap1 { color-index: 1 }
- :active and: focus pseudo class Ex.: focus { color-index: 1 } or Ex : focus { color-index: 1 }

Any usage other than the operation example described above is not allowed. It is not allowed to collectively specify any combination of two or more selectors regardless of using “,” or not. In addition, if a selector is specified, it is applied only to those that are used for the operation in Table 4-6.

A2-4.4.3 Media type

Only the media type of tv must be operated. This media type must be specified according to the @media rule in a style sheet, not by other methods (such as specification under @import rule). If some media type other than tv is specified according to @media rule, any receivers must ignore designation of properties in the @media rule.

A2-4.4.4 Box model

The operational guideline on box model properties is described below.

- margin
  This property is operated with the fixed value as follows:

  Fixed value: 0

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

  Available value: <length> (However, any parts other than 0 are only integers in px units

  Initial value: 0

  Inherited: no

  The div element and the object element should be applied to fixed operation and their values should be 0.

- border-width
  This property is applied to the operation as shown below.

  Available value: <length> (However, any parts other than 0 are only integers in px units

  Initial value: 0

  Inherited: no

  This specification must be common to four sides. However, the object element must be applied to fixed operation and its value must be 0.

- border-style
  This property is applied to the operation as shown below.

  Available value: dotted | dashed | solid | none

  Initial value: none

  Inherited: no

  This specification must be common to four sides. The display format of dotted, dashed, and solid must conform to what the section 8.1 "Geometric", Part 2, Volume 1, specifies. Anything not specified in the Geometric section must depend on a receiver.

  However, the object element must be applied to fixed operation and its value must be none.

A2-4.4.5 Visual formatting model

The following shows the operational guideline on visual formatting models:

- display
  This property must be applied to a different fixed operation depending on an element. However, this must be defined in the default style sheet.

Fixed to block for p, div, body, input and object.

Fixed to inline for br, span and a.

Fixed to none for meta, title, script, style, head, bml, bevent and beitem.

- position
  This property must be applied to a different fixed operation, depending on an element. However, this must be defined in the default style sheet.

  Fixed to absolute for p, div, input and object.

  Fixed to static for br, span and a.

  This is applied only to the above elements.

- left, top, width, height
  This property is applied to the operation as shown below. However, the initial value is defined in the default style sheet.

Available value:	<length> (However, any parts other than 0 are only integers in px units
Initial value:	0
Inherited:	no

- z-index
  This property is applied to fixed operation as shown below.

Fixed value:	Auto

- line-height
  This property is applied to the operation as shown below.

Available value:	normal | <length> (However, any parts other than 0 are only integers in px unit) | inherit
Initial value:	normal
Inherited:	yes

  When normal is specified, a value must be 1.0 time as large as font-size. However, for br, span and a elements, this must be applied to fixed operation and its value must be inherit. The value of inherit for this property must be applied only to these elements.

A2-4.4.6 Other visual effects

The following shows an operational guideline on properties regarding other visual effects:

- visibility
  This property is applied to the operation as shown below.

Available values:	visible | hidden | inherit
Initial value:	inherit
Inherited:	no

  If visibility is set to hidden as a parent element and visible as a child element, only the parent element is hidden. However, the span element and the a element must be applied to fixed operation and their values must be inherit. Besides, the body element must be applied to fixed operation and its value must be visible.

- overflow
  This property is applied to fixed operation as shown below.

Fixed value: hidden

A2-4.4.7 Background

The following shows an operational guideline on background:

- background-image
  This property is applied to the operation as shown below.

  Acquirable value: <uri> | none

  Initial value: none

  Inherited: no

  The image coding scheme designated by <uri> must be jpeg. The start point of drawing is top left of the screen.

- background-repeat
  This property is applied to fixed operation as shown below.

  Fixed value: repeat

A2-4.4.8 Fonts

Table 4-7 shows the fonts specifications available in BML documents. The characters of the code set 0 are displayed with half width of the font size. The external characters specified by DRCS have full width and are displayed with the value of the font size.

Table 4-7 Available Fonts

Font	Font size	Gradation
round gothic	16px, 20px, 24px, 30px, 36px	4 (Note)
bold round gothic	30px	4 (Note)
square gothic	20px, 24px	4 (Note)

Note: It means to specify two colours for the middle gradation colours with the gray-scale-color-index property and also to specify the font of external characters set by loadDRCS to have 4 gradation colours.

The following shows an operational guideline on the properties regarding fonts:

- font-family
  This property is applied to the operation as shown below.

  Available value: “round gothic” | “bold round gothic” | “square gothic” |’round gothic’ | ‘bold round gothic’ | ‘square gothic’ | inherit

  Initial value: “round gothic”

  Inherited: yes

- font-size
  This property is applied to the operation as shown below.

  Available value: <length> (Any parts other than 0 are only integers in px unit) | inherit

  Initial value: 24px

  Inherited: yes

- font-weight
  This property is applied to the operation as shown below.

  Available value: normal | bold | inherit

  Initial value: normal

  Inherited: yes

  The display format for bold must conform to what the section 7.1, Part 2, Volume 1 specifies.

A2-4.4.9 Text

The following shows the operational guideline on properties related to text:

- text-align
  This property is applied to the operation as shown below.

  Available value: left | right | center

  Initial value: left

  Inherited: yes

- letter-spacing
  This property is applied to the operation as shown below.

  Available value: normal | <length> (Any parts other than 0 are only integers in px units) | inherit

  Initial value: normal

  Inherited: yes

  When normal is specified, it must be set to 0. This property represents the blank length between the characters within a specified element. For span and a elements, this property must be applied to fixed operation and its value must be inherit.

- white-space
  This property is applied to fixed operation as shown below.

  Fixed value: normal

A2-4.4.10 Pseudo class

- :active
- :focus
  Table 4-8 shows the CSS properties which can be specified with :active and :focus classes.

Table 4-8 Operational Guideline of CSS Characteristics Specifiable in :focus and :active Classes

	div	p	br	span	a	input	object	body
Box model								
margin	-	-	-	-	-	-	-	-
padding-top	-	-	-	-	-	-	-	-
padding-right	-	-	-	-	-	-	-	-
padding-bottom	-	-	-	-	-	-	-	-
padding-left	-	-	-	-	-	-	-	-

	div	p	br	span	a	input	object	body
border-width	-	-	-	-	-	-	-	-
border-style	-	-	-	-	-	-	-	-
Visual formatting model								
position	-	-	-	-	-	-	-	-
left	O	O	-	-	-	O	O	-
top	O	O	-	-	-	O	O	-
width	O	O	-	-	-	O	O	-
height	O	O	-	-	-	O	O	-
z-index	-	-	-	-	-	-	-	-
line-height	-	-	-	-	-	-	-	-
display	-	-	-	-	-	-	-	-
Other visual effects								
visibility	O	O	-	-	-	O	O	-
overflow	-	-	-	-	-	-	-	-
Background								
background-image	-	-	-	-	-	-	-	-
background-repeat	-	-	-	-	-	-	-	-
Font								
font-family	-	O	-	O	O	O	-	-
font-size	-	O	-	O	O	O	-	-
font-weight	-	O	-	O	O	O	-	-
Text								
text-align	-	-	-	-	-	-	-	-
letter-spacing	-	-	-	-	-	-	-	-
white-space	-	-	-	-	-	-	-	-
Extended characteristics								
clut	-	-	-	-	-	-	-	-
color-index	-	O	-	O	O	O	-	-
background-color-index	O	O	-	O	O	O	-	-
border-top-color-index	O	O	-	O	O	O	-	-
border-right-color-index	O	O	-	O	O	O	-	-
border-left-color-index	O	O	-	O	O	O	-	-
border-bottom-color-index	O	O	-	O	O	O	-	-
resolution	-	-	-	-	-	-	-	-
display-aspect-ratio	-	-	-	-	-	-	-	-
grayscale-color-index	-	O	-	O	O	O	-	-
nav-index	-	-	-	-	-	-	-	-
nav-up	-	-	-	-	-	-	-	-
nav-down	-	-	-	-	-	-	-	-
nav-left	-	-	-	-	-	-	-	-
nav-right	-	-	-	-	-	-	-	-
used-key-list	-	-	-	-	-	-	-	-

A2-4.4.11 Extended properties

The following shows the operational guideline on extended properties:

- clut
  This property is applied to the operation as shown below.

  Available value: <uri>

  Initial value: Undefined

  Inherited: Inherit

  For more information on operation of this property, see 5.1.7.

- color-index
  This property is applied to the operation as shown below.

  Acquirable value: <color-index> (Value range is 0 to 255) | inherit

  Initial value: 0

  Inherited: yes

- background-color-index
  This property is applied to the operation as shown below.

  Available value: <color-index> (Value range is 0 to 255) | inherit

  Initial value: 0

  Inherited: yes

  However, object element must be applied to fixed operation and its value must be 8.

- border-top-color-index, border-right-color-index, border-left-color-index, border-bottom-color-index
  These properties are applied to the operation as shown below.

  Available value: <color-index> (0 to 255) | inherit

  Initial value: 0

  Inherited: no

  The order of drawing the border, is the following sequence: left, right, top and bottom. The corners of four sides are overwritten in that drawing order. If these properties are not specified, it is interpreted as that not the color-index property value of the concerning element but the initial value(0) is specified.

- resolution
  This property is applied to the operation as shown below.

  Available value: 960x540 | 720x480

  Initial value: 960x540

  Inherited: no

- display-aspect-ratio
  This property is applied to the operation as shown below.

  Available value: 16v9 | 4v3

  Initial value: 16v9

  Inherited: no

- greyscale-color-index
  This property is applied to the operation as shown below.

  Available value: <color-index> <color-index>(0 to 255) | inherit

  Initial value: 30 15

Inherited: yes

Four gradation colors are, foreground color, first <color-index> value, second <color-index> value and background color. If a value is set to 8, the background color is displayed.

- nav-index, nav-up, nav-down, nav-left, nav-right
  These properties are applied to the operation as shown below.

  Available value: <integer> (However, usable value ranges from 0 to 32767) | none

  Initial value: none

  Inherited: no

- used-key-list
  This property is applied to the operation as shown below.

  Available value: <key-group>+ | none

  Initial value: basic data-button

  Inherited: no

A2-4.4.12 Default style sheet

The default style sheet is shown in Annex A. This style sheet is assumed to be virtually implemented in a receiver. There is no need to transmit the style sheet.

A2-4.4.13 Operation of length property

A value of the length property must be expressed as an integer pixel value. For example, “100px”. However, if the value is 0, it can be written as “0px” and “0”. The return value of DOM must be “0px”.

A2-4.5 Operational Guidelines for Procedural Description

A2-4.5.1 Operational guidelines of DOM

A2-4.5.1.1 DOM core basic interface group

Table 4-9 lists the operational guideline on DOM core basic interface group. This operational guideline does not cover the mechanism that dynamically changes the tree structure of a document.

Table 4-9 Operational Guideline of DOM Core Basic Interface Group

Interface	Operation
Basic interface group	
DOMException	-
DOMImplementation	O
DocumentFragment	-
Document	O
Node	O
NodeList	-
NamedNodeMap	-
CharacterData	O

Interface	Operation
Attr	-
Element	O
Text	O
Comment	-
Extended interface group	
CDATASection	O
DocumentType	-
Notation	-
Entity	-
EntityReference	-
ProcessingInstruction	-

Table 4-10 shows an operational guideline on the DOM core basic interface attributes:

Table 4-10 Operational Guideline of Attributes and Method (DOM Core Interface Group)

Interface	Attribute/Method	Operation	Restriction
DOMImplementation			
	hasFeature()	O	
Document			
	doctype	-	
	implementation	O	R
	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
	attributes	-	
	ownerDocument	-	
	insertBefore()	-	
	replaceChild()	-	
	removeChild()	-	
	appendChild()	-	

Interface	Attribute/Method	Operation	Restriction
	hasChildNodes()	-	
	cloneNode()	-	
CharacterData			
	data	O	RW (Note)
	length	O	R
	substringData()	-	
	appendData()	-	
	insertData()	-	
	deleteData()	-	
	replaceData()	-	
Element			
	tagName	O	R
	getAttribute()	-	
	setAttribute()	-	
	removeAttribute()	-	
	getAttributeNode()	-	
	setAttributeNode()	-	
	removeAttributeNode()	-	
	getElementsByTagName()	-	
	normalize()	-	
Text			
	splitText()	-	
CDATASection			

Note: The child nodes of script and style are not accessed in the operation. Only the child nodes of p, span, and a can be written in the operation.

A2-4.5.1.2 DOM HTML interface group

Table 4-11 shows an operational guideline on DOM HTML interface:

Table 4-11 Operational Guideline of Interface of DOM HTML Interface Group

Interface	Operation
HTMLCollection	-
HTMLDocument	O
HTMLElement	O
HTMLBlockquoteElement	-
HTMLPreElement	-
HTMLHeadingElement	-
HTMLHRElement	-
HTMLDivElement	O
HTMLParagraphElement	O
HTMLQuoteElement	-
HTMLBRElement	O
HTMLModElement	-
HTMLAnchorElement	O
HTMLBaseElement	-
HTMLLinkElement	-

Interface	Operation
HTMLDListElement	-
HTMLOListElement	-
HTMLUListElement	-
HTMLLIElement	-
HTMLButtonElement	-
HTMLFieldSetElement	-
HTMLFormElement	-
HTMLInputElement	O
HTMLLabelElement	-
HTMLLegendElement	-
HTMLOptGroupElement	-
HTMLOptionElement	-
HTMLSelectElement	-
HTMLTextAreaElement	-
HTMLTableCaptionElement	-

Interface	Operation
HTMLTableColElement	-
HTMLTableElement	-
HTMLTableSectionElement	-
HTMLTableCellElement	-
HTMLTableRowElement	-
HTMLImageElement	-
HTMLAreaElement	-
HTMLMapElement	-
HTMLObjectElement	O
HTMLParamElement	-

Interface	Operation
HTMLFrameSetElement	-
HTMLFrameElement	-
HTMLIFrameElement	-
HTMLMetaElement	O
HTMLTitleElement	O
HTMLScriptElement	O
HTMLStyleElement	O
HTMLBodyElement	O
HTMLHeadElement	O
HTMLHtmlElement	O

Table 4-12 shows an operational guideline on attributes and methods of the DOM HTML interface group:

Table 4-12 Operational Guideline of Attributes and Methods of DOM HTML Interface Group

Interface	Attribute/Method	Operation	Restriction
HTMLDocument			
	title	-	
	referrer	-	
	domain	-	
	URL	-	
	body	-	
	images	-	
	applets	-	
	links	-	
	forms	-	
	anchors	-	
	cookie	-	
	open()	-	
	close()	-	
	write()	-	
	writeln()	-	
	getElementById()	O	
	getElementsByName()	-	
HTMLElement			
	id	O	R
	title	-	
	lang	-	
	dir	-	
	className	O	R
HTMLDivElement			
HTMLParagraphElement			
HTMLBRElement			
HTMLAnchorElement			
	accesskey	O	R
	charset	-	
	coords	-	
	href	O	RW

Interface	Attribute/Method	Operation	Restriction
	hreflang	-	
	name	-	
	rel	-	
	rev	-	
	shape	-	
	tabIndex	-	
	target	-	
	type	-	
	blur()	O	
	focus()	O	
HTMLInputElement			
	defaultValue	O	R
	defaultChecked	-	
	form	-	
	accept	-	
	accesskey	O	R
	alt	-	
	checked	-	
	disabled	O	RW
	maxLength	O	R
	name	-	
	readOnly	O	RW
	size	-	
	src	-	
	tabIndex	-	
	type	O	R
	useMap	-	
	value	O	RW
	blur()	O	
	focus()	O	
	select()	-	
	click()	-	
HTMLObjectElement			
	form	-	
	code	-	
	archive	-	
	codeBase	-	
	codeType	-	
	data	O	RW (Note)
	declare	-	
	height	-	
	name	-	
	standby	-	
	tabIndex	-	
	type	O	R
	useMap	-	
	width	-	

Interface	Attribute/Method	Operation	Restriction
HTMLMetaElement			
	content	O	R
	httpEquiv	-	
	name	O	R
	scheme	-	
HTMLTitleElement			
	text	O	R
HTMLScriptElement			
	text	-	
	htmlFor	-	
	event	-	
	charset	-	
	defer	-	
	src	-	
	type	-	
HTMLStyleElement			
	disabled	-	
	media	-	
	type	-	
HTMLBodyElement			
HTMLHeadElement			
	profile	-	
HTMLHtmlElement			
	version	-	

Note: If the DOM API changes the data attribute of an object concerning the monomedia that is transmitted using data carousel, the data attribute value will be read again even when remaining unchanged. If the module containing a resource specified by the data attribute is locked, the locked data will be applied as it is; otherwise, the presentation must be updated after getting the data from a transmission stream again. Note that dynamically changing type attributes and dynamically changing schemas by changing data attributes for sound are not applicable to the object element.

A2-4.5.1.3 DOM BML interface

Table 4-13 shows an operational guideline on the DOM BML interface:

Table 4-13 Operational Guideline of Interface (DOM BML Interface Group)

Interface	Operation
BMLDocument	O
BMLElement	O
BMLBlockquoteElement	-
BMLPreElement	-
BMLHeadingElement	-
BMLHRElement	-
BMLDivElement	O
BMLSpanElement	O
BMLParagraphElement	O

Interface	Operation
BMLQuoteElement	-
BMLBRElement	O
BMLModElement	-
BMLAnchorElement	O
BMLLinkElement	-
BMLDListElement	-
BMLOListElement	-
BMLUListElement	-
BMLLIElement	-

Interface	Operation
BMLButtonElement	-
BMLFieldSetElement	-
BMLFormElement	-
BMLInputElement	O
BMLLabelElement	-
BMLLegendElement	-
BMLOptGroupElement	-
BMLOptionElement	-
BMLSelectElement	-
BMLTextAreaElement	-
BMLTableCaptionElement	-
BMLTableColElement	-
BMLTableElement	-
BMLTableSectionElement	-
BMLTableCellElement	-
BMLTableRowElement	-
BMLImageElement	-
BMLAreaElement	-
BMLMapElement	-
BMLObjectElement	O
BMLFrameSetElement	-
BMLFrameElement	-
BMLIFrameElement	-
BMLBodyElement	O
BMLBmlElement	O
BMLBeventElement	O
BMLBeitemElement	O
BMLListTableElement	-
BMLItemElement	-

Table 4-14 shows an operational guideline on attributes and methods of the BML interface.

Any defined attribute and method that is not listed in Table 4-14 is assumed that its Operation is "-".

Table 4-14 Operational Guideline of Attributes and Methods (DOM BML Interface Group)

Interface	Attribute/Method	Operation	Remarks
BMLDocument			
	currentFocus	O	R
	currentEvent	O	R (Note 2)
BMLDivElement			
	style	-	
	normalStyle	O	RW (Note1)
	focusStyle	O	RW (Note1)
	activeStyle	O	RW (Note1)
	accessKey	O	R
	focus()	O	
	blur()	O	
BMLSpanElement			
	style	-	
	normalStyle	O	RW (Note1)
	focusStyle	O	RW (Note1)
	activeStyle	O	RW (Note1)
	accessKey	O	R
	focus()	O	
	blur()	O	
BMLParagraphElement			
	style	-	
	normalStyle	O	RW (Note1)
	focusStyle	O	RW (Note1)
	activeStyle	O	RW (Note1)
	accessKey	O	R
	focus()	O	
	blur()	O	
BMLBRElement			
	style	-	
	normalStyle	O	RW (Note1)
	focusStyle	-	
	activeStyle	-	
BMLAnchorElement			
	style	-	
	normalStyle	O	RW (Note1)
	focusStyle	O	RW (Note1)
	activeStyle	O	RW (Note1)
	effect	-	
BMLInputElement			
	style	-	
	normalStyle	O	RW (Note1)
	focusStyle	O	RW (Note1)
	activeStyle	O	RW (Note1)
BMLObjectElement			

Interface	Attribute/Method	Operation	Remarks
	style	-	
	normalStyle	O	RW (Note1)
	focusStyle	O	RW (Note1)
	activeStyle	O	RW (Note1)
	classId	-	
	accessKey	O	R
	remain	O	RW
	streamPosition	O	RW (Note 3)
	streamStatus	O	RW
	streamLooping	-	
	streamSpeedNumerator	-	
	streamSpeedDenominator	-	
	streamLevel	-	
	setSpeed()	-	
	movePosition()	-	
	hasAssociatedIndex()	-	
	assignToLocalEvnet()	-	
	assignToNodePlayMode()	-	
	getMainAudioStream()	O	
	setMainAudioStream()	O	
	focus()	O	
	blur()	O	
BMLBodyElement			
	invisible	O	RW
	style	-	
	normalStyle	O	RW (Note1)
	focusStyle	-	
	activeStyle	-	
BMLBmlElement			
	style	-	
	normalStyle	-	
	focusStyle	-	
	activeStyle	-	
BMLBeventElement			
BMLBeitemElement			
	type	O	R
	esRef	O	RW
	messageGroupId	O	R
	messageId	O	RW
	messageVersion	O	RW
	moduleRef	O	RW
	languageTag	O	RW
	registerId	-	
	serviceId	-	
	eventId	-	
	timeMode	O	R
	timeValue	O	RW
	objectId	O	RW
	subscribe	O	RW

Note 1: These attributes are accessed through the BMLCSS2PropertyInterface. If the attributes are read directly, an object is returned. It is inhibited to write the attributes directly.

Note 2: The values derived from document.currentEvent must not be referenced by other handlers by being substituted to global variables. The result of the substitution is not guaranteed.
Note 3: It can be read and written only if the type attribute is “image/X-arib-mng”.

A2-4.5.1.4 Interface corresponding to BML interrupt event

Table 4-15 shows an operational guideline on the interface corresponding to the BML interrupt events.

Any defined interface that is not listed in Table 4-15 is assumed that its Operation is “-”.

Table 4-15 Operational Guideline of Interfaces Corresponding to BML Interrupt Event

Interface	Attribute/Method	Operation	Remarks
BMLEvent			
	type	O	R
	target	O	R
BMLIntrinsicEvent:			
	keyCode	O	R
BMLBeventEvent:			
	status	O	R
	privateData	O	R
	esRef	O	R
	messageId	O	R
	messageVersion	O	R
	messageGroupId	O	R
	moduleRef	O	R
	languageTag	O	R
	registerId	-	
	serviceId	-	
	eventId	-	
	object	O	R

Note: If the URI string is returned for esRef, moduleRef and so on, the returned value must be in the shortened format (e.g. “/XX”, “/XX/YYYY”).

Table 4-16 lists the correspondence between interrupt event and type attribute of BMLEvent.

Table 4-16 Correspondence between Interrupt Event and type Attribute of BMLEvent

Interrupt event	type value
Remote control key was pressed.	“keydown”
Remote control key was released.	“keyup”
Element was determined by pressing Enter key or access key.	“click”
Focus was set.	“focus”
Focus is out of position.	“blur”
Document was loaded.	“load”
Document unloading was noticed in advance.	“unload”

Interrupt event	type value
When the focus on an input element is out, the change of the value attribute of the concerning input element is detected.	“change”
Event message was received.	“EventMessageFired”
Module update was detected.	“ModuleUpdated”
Module was locked.	“ModuleLocked”
Timer set by beitem caught fire.	“TimerFired”
Process such as getNPT() was enabled.	“NPTReferred”
Monomedia presentation was stopped.	“MediaStopped”
data_event_id update was detected.	“DataEventChanged”
Display status of caption is changed.	“CCStatusChanged”
Main audio stream is changed.	“MainAudioStreamChanged”
Data button was pressed.	“DataButtonPressed”
Execution of global codes was started. Or the functions specified by executing setTimeout() and setInterval() was started.	Undefined (Note)

Note: The target attribute is Null in this case.

A2-4.5.1.5 BMLCSS2Properties interface

Table 4-17 shows an operational guideline on the BMLCSS2Properties interface used in BML.

Table 4-17 Operational Guideline 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	-	
borderRightColor	-	
borderBottomColor	-	
borderLeftColor	-	
borderColor	-	
borderTopStyle	-	
borderRightStyle	-	
borderBottomStyle	-	

Attribute	Operation	Remarks
borderLeftStyle	-	
borderStyle	O	R
borderTop	-	
borderRight	-	
borderBottom	-	
borderLeft	-	
Border	-	
Visual Format model		
Position	-	
Left	O	RW
Top	O	RW
Width	O	RW
Height	O	RW
ZIndex	-	
lineHeight	O	R
verticalAlign	-	
display	-	
bottom	-	
right	-	
cssFloat	-	
clear	-	
direction	-	
unicodeBidi	-	
maxHeight	-	

Attribute	Operation	Remarks
minHeight	-	
maxWidth	-	
minWidth	-	
Other visual effects		
visibility	O	RW
overflow	-	
clip	-	
Generated content/Autonumbering/List		
content	-	
quotes	-	
counterReset	-	
counterIncrement	-	
markerOffset	-	
listStyleType	-	
listStyleImage	-	
listStylePosition	-	
listStyle	-	
Page media		
size	-	
marks	-	
pageBreakBefore	-	
pageBreakAfter	-	
pageBreakInside	-	
page	-	
orphans	-	
widows	-	
Background		
background	-	
backgroundColor	-	
backgroundImage	-	
backgroundRepeat	-	
backgroundPosition	-	
backgroundAttachment	-	
Font		
color	-	
fontFamily	O	RW
fontStyle	-	
fontSize	O	RW
fontVariant	-	
fontWeight	O	RW
font	-	
fontStretch	-	
fontSizeAdjust	-	
Text		
textIndent	-	
textAlign	O	R
textDecoration	-	
textShadow	-	
letterSpacing	O	R
wordSpacing	-	
textTransform	-	
whiteSpace	-	
Table		

Attribute	Operation	Remarks
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 extension		
borderColorIndex	-	
borderTopColorIndex	O	RW
borderRightColorIndex	O	RW
borderLeftColorIndex	O	RW
borderBottomColorIndex	O	RW
backgroundColorIndex	O	RW
colorIndex	O	RW
grayscaleColorIndex	O	RW
outlineColorIndex	-	
clut	O	R
Resolution	O	R
displayAspectRatio	O	R
navIndex	O	R
navUp	O	R
navDown	O	R
navLeft	O	R
navRight	O	R
usedKeyList	O	RW

The above DOM attribute values to be operated conform to the conventions on operation of the CSS2 properties.

A2-4.5.2 Operation range of built-in objects

Table 4-18 shows an operational guideline on the ECMAScript built-in objects:

Table 4-18 Operational Guideline on the ECMAScript Built-in Objects

Built-in object	Method/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 is operated	
	prototype	O	
	Object([value])	O	
	new Object([value])	O	
Object.prototype		All is operated	
	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 is operated	
	constructor	O	
	toString()	O	(Note 3)
Array		All is operated	
	prototype	O	
	Length	O	
	Array(item0, item1, . . .)	O	
	new Array(item0, item1, . . .)	O	
	new Array([len])	O	
Array.prototype		All is operated	
	constructor	O	
	toString()	O	
	join([separator])	O	
	reverse()	O	
	sort([comparefn])	O	
String		All is operated	
	prototype	O	
	Length	O	
	String([value])	O	
	new String([value])	O	

Built-in object	Method/Property	Operation	Remarks
	String.fromCharCode(char0[, char1, . . .])	O	
String.prototype		All is operated	
	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 is operated	
	prototype	O	
	Boolean([value])	O	
	new Boolean([value])	O	
Boolean.prototype		All is operated	
	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])	V	
Number.prototype		All is operated	
	constructor	O	
	toString([radix])	O	
	valueOf()	O	
Math		All is operated	
	E	-	
	LN10	-	
	LN2	-	
	LOG2E	-	
	LOG10E	-	
	PI	-	
	SQRT1_2	-	
	SQRT2	-	
	abs(x)	-	
	acos(x)	-	
	asin(x)	-	
	atan(x)	-	
	atan2(y, x)	-	
	cos(x)	-	
	exp(x)	-	
	floor(x)	-	
	log(x)	-	
	max(x, y)	-	

Built-in object	Method/Property	Operation	Remarks
	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			
	constructor	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)
	setMonth(mon [, date])	O	(Note 6)
	setUTCMonth(mon [, date])	O	(Note 6)

Built-in object	Method/Property	Operation	Remarks
	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: not operated, because it is related to Float.
Note 2: not operated, because it is related to Unicode.
Note 3: Its behaviour is defined in this Appendix.
Note 4: not operated, because it is related to Number.
Note 5: not operated, because it is specified to maintain compatibility with old source codes.
Note 6: operated with restricted specification, because it is related to Number.
Note 7: The radix of parseInt() is 8, 10, and 16 (0 is interpreted as 10).

A2-4.5.3 Operation range of extended objects for broadcasting

Table 4-19 shows an operational guideline on extended objects for broadcasting (CSVTable and BinaryTable) with respect to ECMAScript.

Any defined extended object for broadcasting that is not listed in Table 4-19 is assumed that its Operation is "-".

Table 4-19 Operation of extended objects for broadcasting for implementing basic services

Object	Method/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	

A2-4.5.4 Operation range of browser pseudo objects

Table 4-20 shows an operational guideline on browser pseudo objects. Any defined browser pseudo object that is not listed in Table 4-20 is assumed that its operation is "-".

However, this guideline is not designed to impose a restriction on an operational standard of each media, which defines a standard operation of a function listed in Table 4-20. Instead, this guideline is

aimed to ensure compatibility and interoperability of contents by specifying mandatory functions throughout various types of media.

Table 4-20 Operational Guideline of Browser Pseudo Objects

	Function	Operation	Remarks
Ureg-related function			
	Ureg[]	O	
EPG-related function			
	epgGetEventStartTime()	O	
	epgGetEventDuration()	O	
	epgTune()	O	
	epgIsReserved()	O	
	epgReserve()	O	
	epgCancelReservation()	O	
	epgRecIsReserved()	O	
	epgRecReserve()	O	
	epgRecCancelReservation()	O	
Function related to event group index			
	grpIsReserved()	-	
	grpReserve()	-	
	grpCancelReservation()	-	
	grpRecIsReserved()	-	
	grpRecResrve()	-	
	grpRecCancelReservation()	-	
	grpGetNodeEventList()	-	
	grpGetERTNodeName()	-	
	grpGetERTNodeDescription()	-	
	epgXTune()	-	
Series reservation function			
	seriesIsReserved()	-	
	seriesReserve()	-	
	seriesCancelReservation()	-	
	seriesRecIsReserved()	-	
	seriesRecResrve()	-	
	seriesRecCancelReservation()	-	
Non-volatile memory function			
	readPersistentString()	-	
	readPersistentNumber()	-	
	readPersistentArray()	O	
	writePersistentString()	-	
	writePersistentNumber()	-	
	writePersistentArray()	O	
	copyPersistent()	-	
	getPersistentInfoList()	-	
	deletePersistent()	-	
	getFreeSpace()	-	
Interaction channel function			
Interaction channel function – PPP connection set-up			
	setDialUpParamsForPPP()	-	
	discardDialUpParamsForPPP()	-	
Interaction channel function – Delayed call			
	registerTransmission()	-	
	registerTransmissionStatus()	-	
	getTransmissionStatus()	-	

	Function	Operation	Remarks
Interaction channel function – BASIC procedure			
	connect()	O	
	disconnect()	O	
	sendBinaryData()	-	
	receiveBinaryData()	-	
	sendTextData()	O	
	receiveTextData()	O	
Interaction channel function – TCP/IP			
	connectPPP()	-	
	disconnectPPP()	-	
	setSendmailParams()	-	
	discardSendmailParams()	-	
	mailto()	-	
	setHttpParams()	-	
	discardHttpParams()	-	
	sendHttpRequest()	-	
	sendFtpCommand()	-	
Interaction channel function – Large call reception service			
	vote()	O	
Interaction channel function – Encryption communication using CAS			
	startCASEncryption()	O	
	transmitWithCASEncryption()	O	
	endCASEncryption()	O	
Interaction channel function – Communication with secret key encryption without using CAS			
	setEncryptionKey()	-	
	beginEncryption()	-	
	endEncryption()	-	
Operation control functions			
	reloadActiveDocument()	O	
	getNPT()	O	
	getProgramRelativeTime()	O	
	isBeingBroadcast()	O	
	lockExecution()	-	
	unlockExecution()	-	
	lockModuleOnMemory()	O	
	unlockModuleOnMemory()	O	
	setCachePriority()	O	
	getLinkSourceServiceStr()	-	
	getLinkSourceEventStr()	-	
	getIRDID()	O	
	getProgramID()	O	
	getActiveDocument()	O	
	lockScreen()	O	
	unlockScreen()	O	
	getBrowserSupport()	O	
	launchDocument()	O	
	quitDocument()	-	
Sound control of Receiver			
	playRomSound()	O	
Timer function			
	sleep()	O	
	setTimeout()	-	
	setInterval()	O	

	Function	Operation	Remarks
	clearTimer()	O	
	pauseTimer()	O	
	resumeTimer()	O	
	setCurrentDateMode()	O	
External character function			
	loadDRCS()	O	
	unloadDRCS()	-	
Other functions			
	random()	O	
	subDate()	O	
	addDate()	O	
	formatNumber()	O	
Presentation control function for caption			
	setCCStreamReference()	-	
	getCCStreamReference()	-	
	setCCDisplayStatus()	O	
	getCCDisplayStatus()	O	
	getCCLanguageStatus()	O	

A2-4.6 Operational Scope for Navigator pseudo objects

The following table summarizes operational guidelines on Navigator pseudo objects.

Table 4-21 Operational Guidelines on Navigator Pseudo Objects

Function	Operation	Remarks
AppName	-	
AppVersion	-	

A2-4.7 Restrictions on Description of BML Documents

The restrictions on description of BML documents are described below.

- The style element, the bevent element and the script element appear only once in the head element part. The script element and the style element must be a single CDATA section, enclosing each whole content with “<![CDATA” and “]]>“.
- No arguments are written in the functions used for event handlers.
  That is, the sentence for setting an event handler is as follows:
  Attr = “func( );”
  Note that attr is an attribute name of the interrupt event and func is a function name.
- Automatic semicolon insertion is not mandatory for the operation. (ECMA-262 p.20)

A2-4.8 Guidelines for Presentation Control of BML Document

A2-4.8.1 Positioning and Restrictions

In this section, a guideline concerning designation of position for the elements located in the body element is specified.

The restrictions described below cover only the elements that occupy two-dimensional space by placing them in the body element. This does not mean that including other elements in the body element is inhibited.

In the sentences of restrictions, “the size and position of a box is specified by absolute positioning” is referred to simply as “a position is specified.” The positioning method uses relative coordinates from a parent element (equivalent to containing block of CSS2) as follows:

<Element name style= ’

left:	Distance from the left of box of parent element (number of pixels);
top:	Distance from top of box of parent element (number of pixels);
width:	Width (number of pixels);
height:	Height (number of pixels);
visibility:	“visible”, “hidden” or “inherit”;’ ...>

The position attribute must be fixed to absolute. Therefore, it is not specified in the style part.

In addition, the following terms are used in the sentences below:

boxed element: The followings are named generically ‘boxed elements’:

a) elements having two-dimensional space (input)

b) elements having intrinsic dimension (object)

c) ‘div’ element

d) ‘p’ element

normal flow element: The elements that can be used in operation other than ‘boxed elements’ are named ‘normal flow elements’.

Restriction 1: A position of boxed elements must be always specified.

Restriction 2: A position of normal flow elements must not be specified.

Restriction 3: The child element of body elements must be either div or p element.

Restriction 4: The div element can include only boxed elements. Nesting div element(s) allows nest of position designation.

Restriction 5: The p element is allowed to have one of normal flow elements or a text line as its child element.

Example:

```
<p style=“...box designation...”>
abcde<span style=“...”>fghij</span>
</p>
```

In CSS2, absolute positioning or normal flow can be specified for each element. However, the basic concept of the aforementioned rule is to completely separate the line of position-specified elements from the line of normal flow elements.

A2-4.8.2 Vertical writing

Specifying a writing direction in BML is not indispensable for operation. Vertical writing can be implemented by referring to an external file of 8-bit character codes by the object element and using control codes in it.

A2-4.8.3 Restrictions on arrangement of elements

The elements having translucent pixels must not be placed at the innermost part of the text graphic plane. A translucent color may be specified as a foreground color or scaled color only when the background color is transparent. A transparent color must not be specified as a foreground color or scaled color. As shown in Figure 4-1, if an object having translucent pixels overlaps the top of an object placed on the text graphic plane, how the overlapping part is displayed is unpredictable.

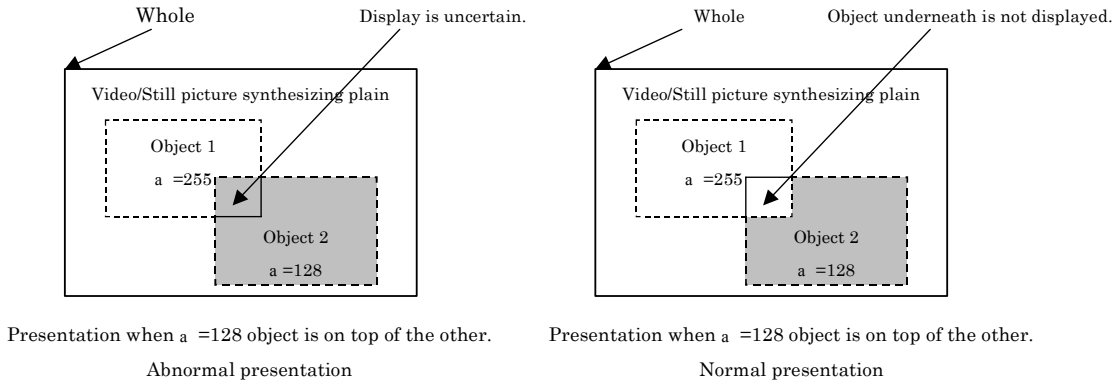


Figure 4-1 Display of translucent pixels

It is desirable that the object element showing a video/still picture is the first element to be presented in the document. If it does not appear first, however, the element must be placed so that a character or graphic object is not on top of it. Figure 4-2 shows a display example.

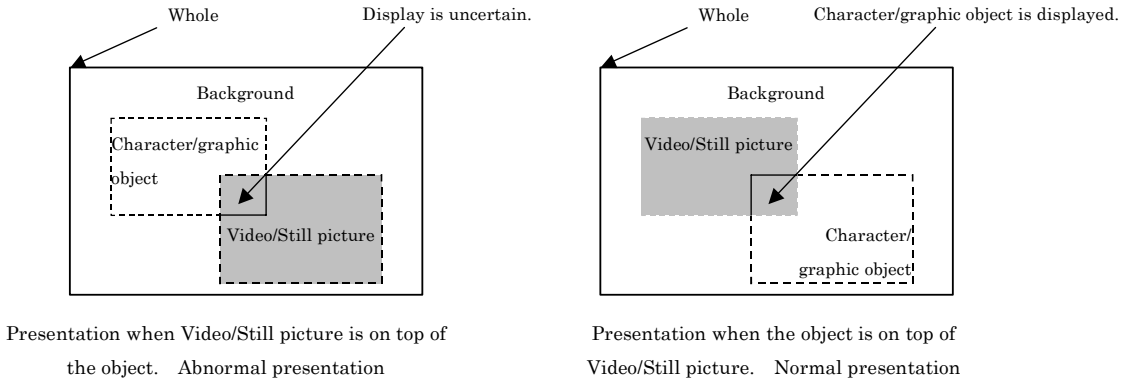


Figure 4-2 Display Example of Object Element that shows Video/Still Picture

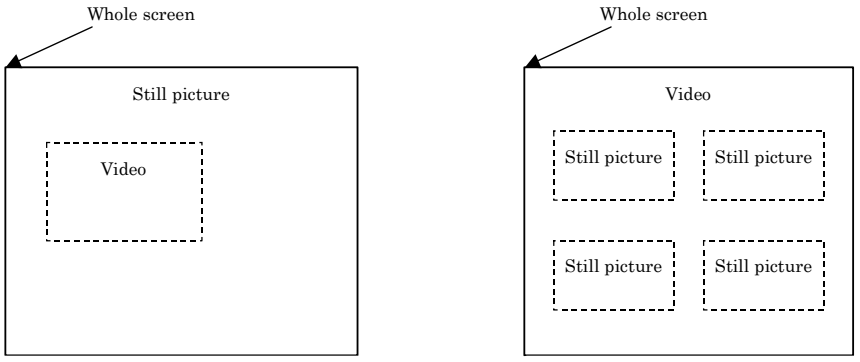
Also, the background color (background-color-index) of the div element holding an object element as a child element that is associated with MNG graphics must always be a non-drawing color. Any objects excluding the div element and video/still picture cannot be displayed in the back of MNG graphics. This is not the case if all pixels overlapping the MNG graphic area are non-drawing colors. Further any elements must not be overlapped in front of MNG graphics.

A2-4.8.4 Restrictions on the elements which are represented on the composition plane of video and still picture

A composition plane of video and still picture that is organized by switching between a video plane and a still picture plane has the restrictions described below. When a still picture and a video are

placed by object element, thus restrictions must be taken into account. Figure 4-3 shows a example presentation of video/still picture combination following these restrictions.

- Only one video is presented. The streamStatus attributes of two or more videoscannot be set to ‘play’ simultaneously.
- If a video is placed on a still picture, the video is displayed in one rectangular area. If a still picture is placed on a video, the still picture is displayed in up to four rectangular areas. If one of the still pictures overlaps with another in this case, the presentation is unpredictable because the overlapped still picture does not make a rectangular area.
- If z-order (including an implicit z-order in which elements have appeared) places a still picture in front of a video and the still picture covers or overlaps the video, the width property and the height property of the object element for the video are modified. The width property is modified to contain a value equivalent to a horizontal resolution value specified with resolution property of the body element. The height property is modified to contain a value equivalent to a vertical resolution value specified with resolution property of the body element.
- If a still picture is placed on a full-screen video, the background-image property is not specified.



Video/Still picture-combined presentation example 1

Video/Still picture-combined presentation example 2

Figure 4-3 Display Example of Combination of Video/Still Picture

A2-4.8.5 Operational range of object element

Note) Presentation of each monomedia that is presented by object elements is specified in Section 5.1.2.

A2-4.8.5.1 Application of remain attribute for each type

Table 4-22 shows a guideline for application of the remain attribute of object element.

TABLE 4-22 Application of the reamain attribute

type attribute	Application of remain attribute	Remarks
video/X-arib-mpeg2	O	
video/X-arib-mpeg1	O	
audio/X-arib-mpeg2-aac	Δ	It is available only when the scheme is “arib: (PES).”
audio/X-arib-aiff	-	
image/jpeg	O	lockModuleOnMemory is used together.

type attribute	Application of remain attribute	Remarks
image/ X-arib-png	-	
image/X-arib-mng	-	
text/X-arib-jis8text	-	

The following conditions are assumed to operate the remain attribute of the object element.

[Condition 1]

The remain attribute of an object element can be made valid only in the case where the object element is used by fulfilling either of the conditions of (1) to (4) listed below

(1) The value of the type attribute is video/X-arib-mpeg1 and belongs to the same content group as the object element

(2) The value of the type attribute is video/X-arib-mpeg2 and belongs to the same content group as the object element

(3) The value of the type attribute is "image/jpeg". In addition, the values of the width property and the height property are equal to the horizontal and vertical resolution specified by the resolution property of the body element, which is an ancestor element, respectively.

(4) The scheme specified by the data attribute is "arib", the data attribute belongs to the same content group as the object element, and the value of the type attribute is "audio/X-arib-mpeg2-aac".

Note that, for the purpose of Condition 1, to make the remain attribute valid is assumed as one of the following.

a) To make a BML document specify the value "remain" for the remain attribute.

b) To set the remain attribute of the BMLObjectElement interface of DOM API to true.

c) To let the remain attribute of the object element in a BML document before a transition be applicable to the object element in the BML document after the transition.

When the remain attribute of the object element is made valid based on any other assumption, how the remain attribute is operated depends on a receiver.

[Condition 2]

The object element whose remain attribute is made valid must be a child element of the div element which is the first child element of the body element.

When the remain attribute of the object element is made valid without meeting this condition, the behaviour depends on a receiver.

[Condition 3]

When the div element has at least one object element whose remain attribute is made valid as a child element, the div element must have not more than one child element meeting the (5) condition described below, not more than one child element meeting the (6) condition described below, and not more than one child element meeting the (7) condition described below. Otherwise, the behaviour depends on a receiver.

(5) The object element meeting the (1) or (2) condition described above.

(6) The object element meeting the (3) condition described above.

(7) The object element meeting the (4) condition described above

[Condition 4]

To make a document transition occur in the case where the remain attribute of the object element satisfying the above condition (3) is made valid, the lockModuleOnMemory() method of a browser

pseudo object must have been applied to the resource specified with the data attribute of the object element. Otherwise, the behaviour depends on a receiver.

[Condition 5]

Even if the object element satisfies the condition (3), its remain attribute must not be made valid when the background-image property of the body element which is an ancestor element of the object element has been specified.

Otherwise, the behaviour depends on a receiver.

[Condition 6]

The remain attribute for a still picture that belongs to two or more document groups must not be made valid.

Otherwise, the behaviour depends on a receiver.

A2-4.8.5.2 Operation of attributes regarding presentation of stream data

Table 4-23 shows an operational guideline on streamposition attribute, streamstatus attribute, and streamlooping attribute of an object element.

Table 4-23 Operation of attributes regarding presentation of stream data

type attribute	streamposition	streamstatus	streamlooping
video/X-arib-mpeg2	Fixed to 0	play (Note 1)	Fixed to 1
video/X-arib-mpeg1	Fixed to 0	play (Note 1)	Fixed to 1
audio/X-arib-mpeg2-aac	Fixed to 0	play, stop (Note 2)	Fixed to 1
image/jpeg	Fixed to 0	play (Note 1)	Fixed to 1
image/ X-arib-png	Fixed to 0	play (Note 1)	Fixed to 1
audio/X-arib-aiff	Fixed to 0	Play, stop (Note 2, 3)	Fixed to 1
image/X-arib-mng	Frame number	Play, stop, pause (Note 2)	Fixed to 1
text/X-arib-jis8text	Fixed to 0	play (Note 1)	Fixed to 1

Note 1: The initial value of streamstatus for an object element referencing media of type attribute which takes ‘play’ only is ‘play’.

Note 2: The initial value of streamstatus for an object element referencing media of type attribute which can take ‘stop’ is ‘stop’. In this case, actually the corresponding sound is not stopped but muted. Note that dynamically changing type attributes and dynamically changing schemas by changing data attributes for sound are not applicable to the object element.

Note 3: When the playback ends, streamstatus is automatically set to ‘stop’.

A2-4.8.5.3 Guidelines on the presentation behaviour of MNG

This section contains guidelines on the interpretation of the attribute and the presentation behaviour when MNG (the value of the type attribute is “image/X-arib-mng”) is specified in the object element.

- Table 4-24 shows how MNG behaves depending on the initial values of the streamstatus attribute and the streamposition attribute.

Table 4-24 Application of the initial values of the streamstatus attribute and the streamposition attribute

Initial value of the streamstatus attribute	Behaviour
play	It plays back from the frame specified by the initial value the streamposition attribute. Playback is repeated in accordance with the repetition number of times calculated with (IterationMax of TERM chunk) × streamlooping. However, it is interpreted with the infinity time repetition at the time of IterationMax = 0x7fffffff of TERM chunk.
stop	A frame is not displayed. Even if the initial value of the streamposition attribute is set to any other than 0, it will be reset in 0.
pause	The frame specified in the initial value of the streamposition attribute is displayed. As for the repetition, it is interpreted with the first round.

- Table 4-25 shows shows how MNG behaves depending on the value of the visibility property.

Table 4-25 MNG presentation depending of the value of the visibility property

Value of the streamstatus attribute	visibility="visible"	visibility="hidden"
play	Normal play back	Not displayed. It depends on receivers whether to proceed play back process (renewal of a frame to be displayed and the update of the repetition number of times) without presentation. And, it is indefinite from which frame the play back starts when the value of the visibility property is set "visible" next time.
stop	Not displayed	Not displayed
pause	The frame specified in the streamposition attribute is displayed.	Not displayed

- Table 4-26 outlines how the streamposition attribute is accessed via DOM API.

Table 4-26 Access to the streamposition attribute

Value of the streamstatus attribute	Read the streamposition attribute	Write the streamposition attribute
play	Not used in the operation	Not used in the operation
stop	Value is fixed to 0	Even if the value except for 0 is set, it is ignored, and the value is fixed to 0.

pause	The number which corresponds to the frame that is set for display.	A frame corresponding to the specified value is displayed. When a bigger value than the number of the frames of MNG is specified, the presentation depends on receivers.

- Table 4-27 shows how MNG is presented when the streamstatus attribute is modified using DOM API.

Table 4-27 Access to the streamstatus attribute

The former value of the streamstatus	Set for play	Set for stop	Set for pause
play	-	- Play back is stopped and the frame is not displayed. - The streamposition attribute is reset in 0. - The repetition number of times is reset in 0 and it is played back from the first round when the streamstatus attribute is set for play next time.	Play back is suspended, and one frame is displayed. It depends on receivers which frame is displayed. The streamposition attribute is set for the value which corresponds to the displayed frame. The repetition number of times is reset in 0 and it is played back from the first round when the streamstatus attribute is set for play next time.
stop	Play back is started from the frame which corresponds to the condition ‘streamposition attribute =0’.	-	Play back is started from the frame which corresponds to the condition ‘streamposition attribute =0’.
pause	- Play back is started from the frame set up in the streamposition attribute.	A frame is not displayed. The streamposition attribute is reset in 0. The repetition number of times is reset in 0 and it is played back from the first round when the streamstatus attribute is set for play next time.	-

- The data attribute can be modified only when the streamstatus attribute is stop.
- The streamstatus attribute is set to pause automatically after the specified times of play back are done.
- It depends on receivers whether a renewal of the presentation frame of MNG is suspended or is continued by the execution of lockScreen() in the period when the renewal of the screen is prohibited.

Chapter APPENDIX 2-5 Guidelines for the behaviour of BML Documents

A2-5.1 Operational Guidelines of Presentation

A2-5.1.1 Operational guidelines on Z-index

The operation rule for object presentation is given below. Objects are presented in the appearance order. In other words, the elements that appeared first in a BML document are presented at the inner part (position far from the viewer) of the screen. The relationship between the DOM object structure and presentation is illustrated below in Fig. 5-1. DOM objects generate a tree structure in the order of element presentation. Objects 1 to 9 are presented in the order of 1, 2, 4, 5, 7, 8, 9, 3 and 6 if having the structure shown below.

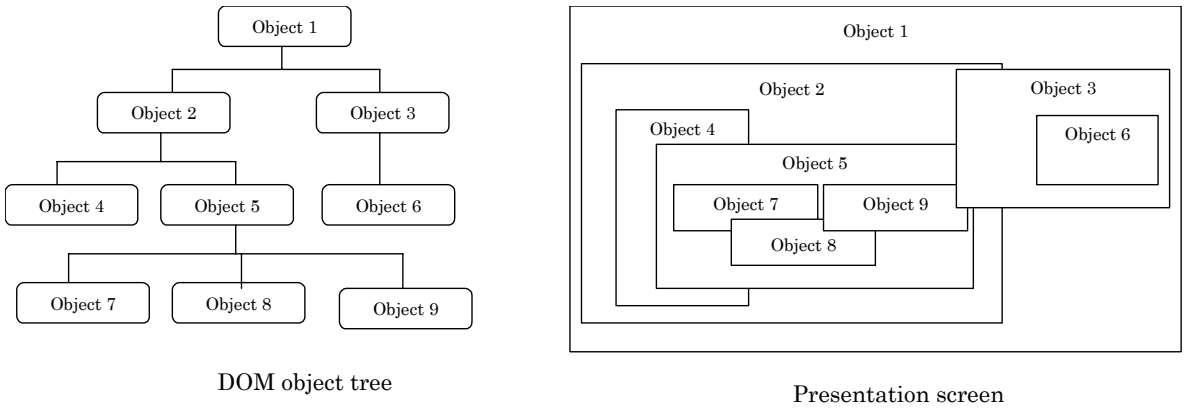


Figure 5-1 The DOM object structure and presentation

A2-5.1.2 Presentation plane

The above objects are drawn according to the presentation plane model shown below in Figure 5-2. Video and still pictures are switched in the rectangular area to organize a video/still picture synthesizing plane. The video/still picture synthesizing plane is blended with the text graphic plane using α value.

The α value output from CLUT indicates α blending ratio of the text graphic plane to the video/still picture plane.

Here, when the α value in 256 steps is 255, it means 100% and when 0, 0%.

The colour expressed by the index with which α is set to 0 means a non-drawing colour. Therefore, nothing is drawn if an index that leads α value to 0 is specified as a pixel of a bit map or a background-color-index or border-color-index value of style properties.

If the invisible attribute of body element is specified, a video is displayed on the full screen but neither text graphic plane nor still picture plane is displayed. If background-color-index is specified to the body element, an index value with α value = 255 must be specified so that the background of video/still picture synthesizing plane appears.

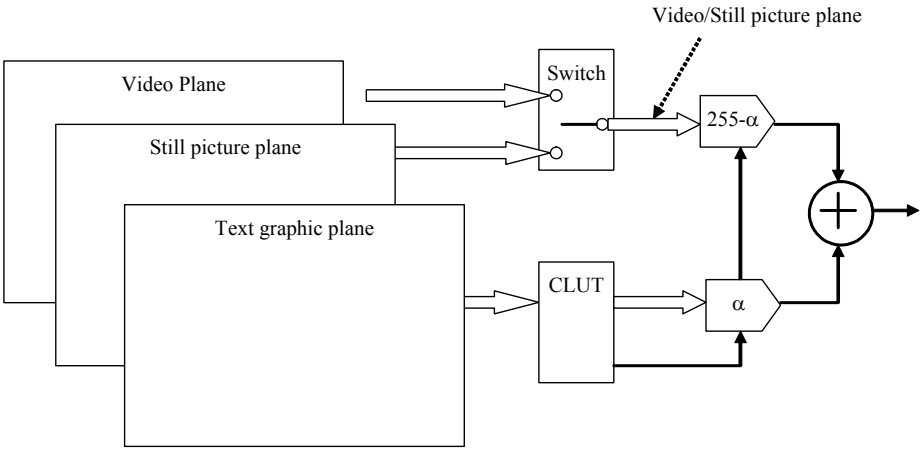


Figure 5-2 Model for drawing planes

A2-5.1.3 Operation of monomedia presentation

The following shows an operation example of monomedia presentation using an object element.

A2-5.1.3.1 Positioning by object element

The following shows an operation example of top, left, width, and height , which are CSS2 properties applied to an object element.

When a video, a still picture or graphic form is presented using an object element, a scaling operation shown below is performed. In the diagram, W indicates the number of horizontal pixels of an image, H indicates the number of vertical pixels of an image. Also, x and y indicate horizontal and vertical coordinates on the presentation screen, respectively. Sx and Sy indicate a horizontal scaling ratio and a vertical scaling ratio, respectively. No value that places a video outside of a corresponding video plane must be specified. Also, no value that places a still picture outside of a corresponding still picture plane must be specified.

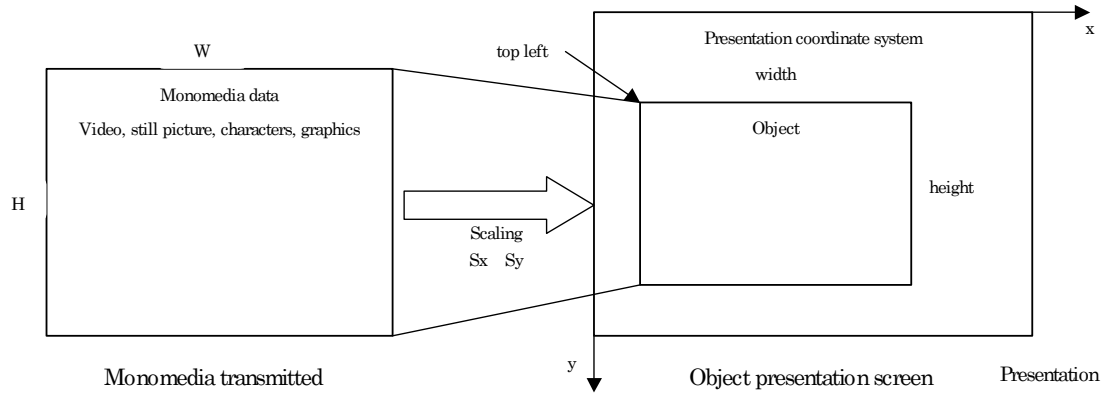


Figure 5-3 Relation between presented coordinates and images

- Combinations of Planes

  The presentation coordinate systems specified by an object element must be as follows: 540x960 coordinate system and 480x720 coordinate system. The combinations of plane resolutions for video, still picture and text graphics are listed below in Table 5-1.

Table 5-1 Combination of plane resolutions

Presentation coordinate system	Text graphics plane resolution	Still Picture plane resolution	Video plane resolution
540x960 coordinate system Horizontal direction(0-959) Vertical direction (0-539)	Horizontal 960 pixels Vertical 540 pixels	Horizontal 1920 pixels (Note) Vertical 1080 pixels (Note)	Horizontal 1920 pixels (Note) Vertical 1080 pixels (Note)
480x720 coordinate system Horizontal direction (0-479) Vertical direction (0-479)	Horizontal 720 pixels Vertical 480 pixels	Horizontal 720 pixels Vertical 480 pixels	Horizontal 720 pixels Vertical 480 pixels

(Note) If (0,0) is specified in the coordinate system, the first horizontal and vertical pixel of a video/still picture is indicated.

- top and left properties

There are restrictions depending on plane characteristics if placing a video, still picture or graphic form on presentation coordinates. The values which are taken in top and left properties according to the type attribute of an object element are listed below.

Table 5-2 The values available for top and left properties

type attribute	top and left properties in 540x960 coordinate system	top and left properties in 480x720 coordinate system
video/X-arib-mpeg2	Arbitrary in both x and y coordinates	Arbitrary even number in both x and y coordinates (Note)
video/X-arib-mpeg1	Arbitrary in both x and y coordinates	Arbitrary even number in both x and y coordinates (Note)
image/jpeg	Arbitrary in both x and y coordinates	Arbitrary even number in both x and y coordinates (Note)
image/ X-arib-png	Arbitrary in both x and y coordinates	Arbitrary in both x and y coordinates
image/X-arib-mng	Arbitrary in both x and y coordinates	Arbitrary in both x and y coordinates

- width and height properties

Scaling operation is restricted depending on a media type or a scaling ratio. The expressions for calculating width and height properties are listed below, provided that the decimals are rounded down. If the values of the width property and the height property which do not satisfy the relationship in the table are specified, the representation on the display depends on a receiver. And, if the scaling ratio of the object element is dynamically changed, the width property and the height property must be changed when the update of the representation is locked by the lockscreen() function.

Table 5-3 Relationship between width/height properties and scaling ratio

type attribute	960x540 coordinate system width property	960x540 coordinate system height property	720x480 coordinate system width property	720x480 coordinate system height property
video/X-arib-mpeg2	(W/2)*Sx	(H/2)*Sy	W*Sx (Note 2)	H*Sy (Note 4)
video/X-arib-mpeg1	(W/2)*Sx	(H/2)*Sy	W*Sx (Note 2)	H*Sy (Note 4)
image/jpeg	W/2 (Note 1)	H/2 (Note 1)	W (Note 3)	H
image/ X-arib-png	W	H	W	H
image/X-arib-mng	W	H	W	H

Note 1: In the case of width=960, height=540, W=960, and H=540, W*Sx and H*Sy must be applied and Sx=Sy=256/128 must be fixed.
Note 2: If the result of calculation is not an even number, W*Sx-1 must be applied.
Note 3: If the result is not an even number, W-1 must be applied.
Note 4: If the result of calculation is not an even number, H*Sy-1 must be applied.

- Scaling ratio

The following values are applicable to Sx and Sy, where Sx=Sy.

Table 5-4 Values applicable to Sx and Sy

Sx	256/128	192/128	160/128	128/128	96/128	80/128	64/128	48/128	32/128
Sy	256/128	192/128	160/128	128/128	96/128	80/128	64/128	48/128	32/128

- Number of horizontal and vertical pixels

The following values are applicable to W and H.

Table 5-5 Values applicable to W and H

	960x540 coordinate system		720x480 coordinate system	
type attribute	W	H	W	H
video/X-arib-mpeg2	1920	1080	720, 360	480, 240
video/X-arib-mpeg1	-	-	360, 180	240, 120
image/jpeg	16-1920	16-1080	16-720	16-480
image/ X-arib-png	2-960	2-540	2-720	2-480
image/X-arib-mng	2-960	2-540	2-720	2-480

A2-5.1.3.2 Operation of video presentation

Video pictures are mapped once when performing the above presentation because those having the number of pixels other than the above W and H must be allowed for presentation. A video with W' having the number of horizontal pixels and H' having the number of vertical pixels is mapped once and converted into that of an intermediate format with W having the number of horizontal pixels and H having the number of vertical pixels. The converted video is scaled as mentioned above. The mapping and scaling processes are shown below.

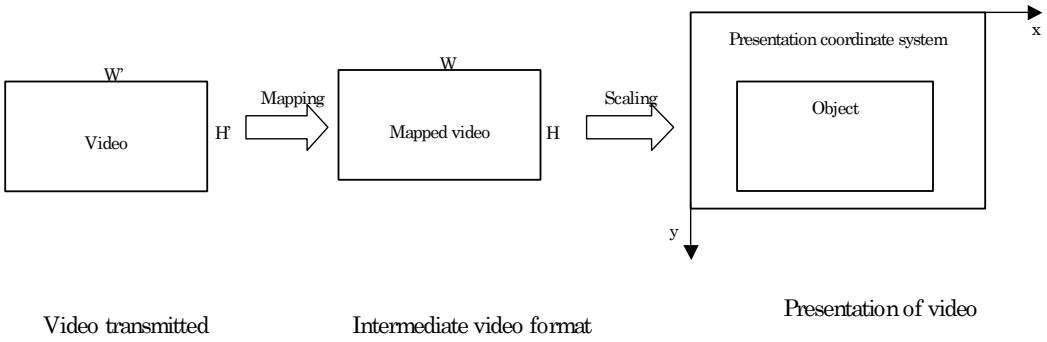


Figure 5-4 Mapping of video

Mapping from W to W' and from H to H' must be provided separately as a vendor rule.

A2-5.1.3.3 Presentation behaviour when the remain attribute is set

The sharing operation in specifying the remain attribute is shown below. This operation is performed by transferring a DOM object from one document to the other.

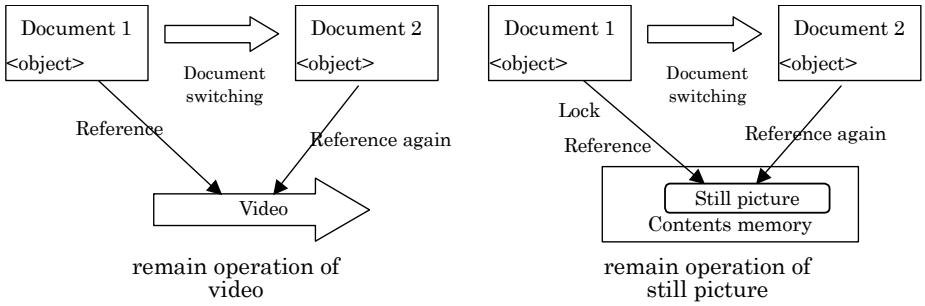


Figure 5-5 Behaviour of remain attribute

The process of the remain attribute when transition from the document 1 to the document 2 occurs is described below.

[1] The presentation of all the elements of the document 1 is finished except for the object elements whose remain attributes are valid and which have the same values of the id attributes in the document 1 and 2.

[2] The presentation of document 2 is started, provided that the document 1 and the document 2 belonging to a same content group share a video or audio monomedia, or the document 1 and the document 2 belonging to a same document group share a still picture. However, when the presentation of the object element D which has the same id attribute value as the object element S whose remain attribute was valid in the document 1, it is handled as the value of the following attributes which an object element S has retained are specified specifically in the start tag of the D element.
This does not depend on what kind of descriptions are included in the start tag of the D element with regard to the concerning attributes.
- type attribute
- data attribute
- remain attribute
- streamstatus attribute
- streamposition attribute

A2-5.1.4 Guideline on clipping

This guideline allows a child element to be clipped by a parent element in order to fix the overflow property to hidden. As shown left in Figure 5-6, the content is clipped if sticking out of the rectangular area of the specified element (object 1). Moreover, as shown right below, if the child element (object 3) is clipped by the parent object (object 2), any areas other than the rectangular area specified by the parent element are hidden.

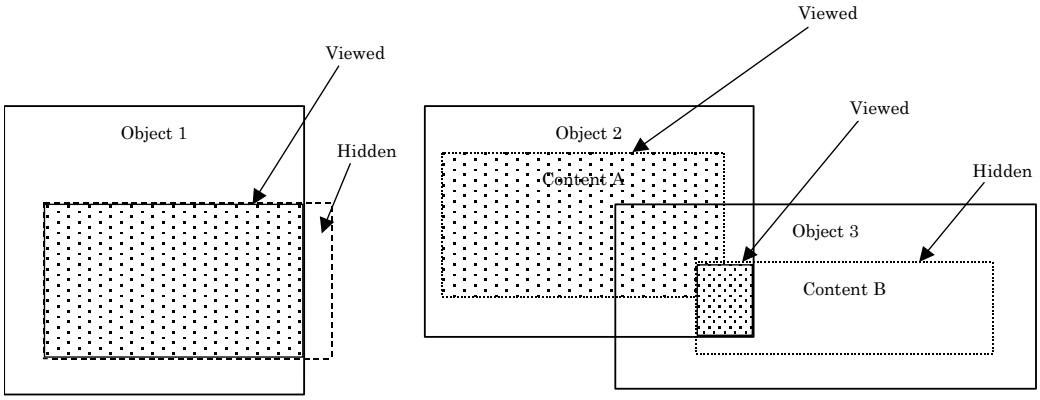


Figure 5-6 Example of Clipping

The above clipping operation is not performed by the parent div element when a video, still picture or MNG image is specified by the object element. In other words, monomedia image specified by object element must not be specified out of area with parent div element. This rule is also applicable to clipping that involves two or more generations. No description must instruct an ancestor element to perform a clipping. When a displayed area of an 8-bit coding string is not contained in the area defined with width and height, line breaks are made, as required, before the string is displayed. If the modified displayed area is still not contained in the area defined with width and height, how the string is displayed depends on a receiver.

A2-5.1.5 Values applicable to CSS2 properties

Table 5-6 shows the values applicable to a CSS2 property hoding the value "<length>".

Table 5-6 Values applicable to CSS2 properties

property	Range of value
top	-960px to 960px
left	-540px to 540px
width	0 to 960px
height	0 to 540px
border-width	0 to 270px
padding-top,bottom	0 to 540px
padding-right,left	0 to 960px
font-size	(See table 4-7)
line-height	16px to 512px
letter-spacing	0 to 944px

A2-5.1.6 Rules of box model

A margin is fixed to 0. Border and padding shown below are used for operation. And any boxed elements described in Section 4.7.1 must satisfy the following condition:

(x0, y0) is the relative position of top left corner of border area for the concerning element to the top left corner of the body element. “w” and “h” are width and height respectively before clipping taking into account the content area, the padding area, and the border area of the concerning element.

$$x0 >= -960,\ y0 >= -540,\ x0 + w <= 1920,\ y0 + h <= 1080,\ 0<=w<=960,\ 0<=h<=540$$

where $w = \text{width} + \text{padding-right} + \text{padding-left} + \text{border-width} * 2$ and
$h = \text{height} + \text{padding-top} + \text{padding-bottom} + \text{border-width} * 2$

If the specified values of CSS2 properties do not satisfy the above conditions, the presentation depends on a receiver.

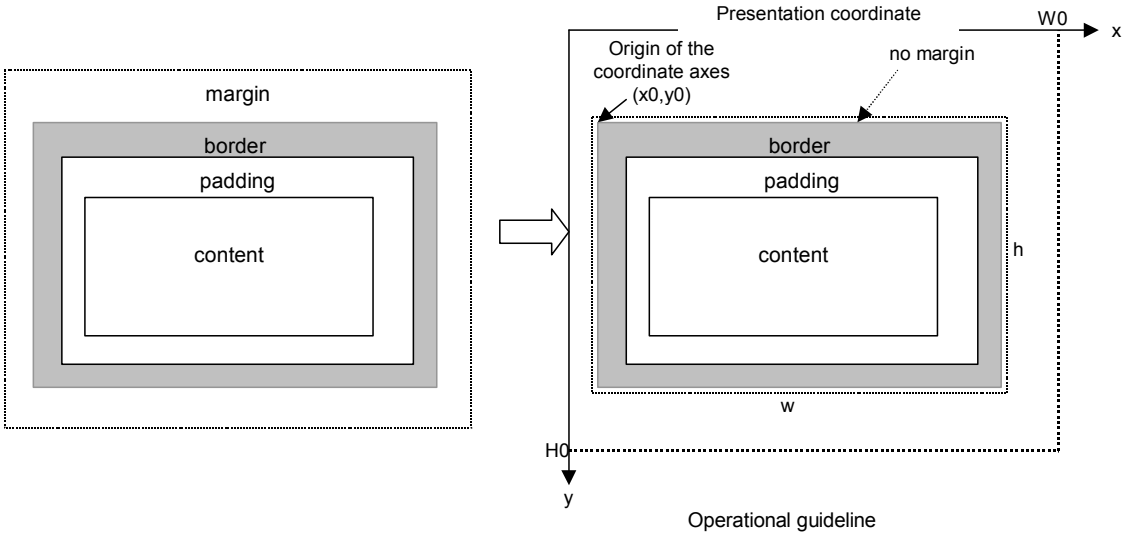


Figure 5-7 Fixed operation of margin

A2-5.1.7 Operation of colour map data

The index colours based on the CLUT with a depth of 8 (up to 256 colours) which employs YCBCR colour space are used for colour specification including a character colour, a background colour and border colour.

The default CLUT which has predefined fixed colours is applied if explicit clut properties are not specified to the body element. If specified, the CLUT defined by colour map data which is identified by URI is applied. If the colour data for all indexes is not contained in the colour map, colour data of the default CLUT is applied to an undefined index.

The default CLUT is listed below. In Table 5-7 the RGB values are used for simple expression. However, that RGB value reconverted into the YCBCR value with equation for conversion defined in ARIB STD-B24 Volume 1, and the YCBCR values are operated.

Table 5-7 Default CLUT Setting

Index	R	G	B	α
0	0	0	0	255
1	255	0	0	255
2	0	255	0	255
3	255	255	0	255
4	0	0	255	255
5	255	0	255	255
6	0	255	255	255
7	255	255	255	255
8	0	0	0	0
9	170	0	0	255
10	0	170	0	255
11	170	170	0	255
12	0	0	170	255
13	170	0	170	255
14	0	170	170	255
15	170	170	170	255
16	0	0	85	255
17	0	85	0	255
18	0	85	85	255
19	0	85	170	255
20	0	85	255	255
21	0	170	85	255
22	0	170	255	255
23	0	255	85	255
24	0	255	170	255
25	85	0	0	255
26	85	0	85	255
27	85	0	170	255
28	85	0	255	255
29	85	85	0	255
30	85	85	85	255
31	85	85	170	255
32	85	85	255	255
33	85	170	0	255
34	85	170	85	255
35	85	170	170	255
36	85	170	255	255
37	85	255	0	255
38	85	255	85	255
39	85	255	170	255
40	85	255	255	255
41	170	0	85	255
42	170	0	255	255

Index	R	G	B	α
43	170	85	0	255
44	170	85	85	255
45	170	85	170	255
46	170	85	255	255
47	170	170	85	255
48	170	170	255	255
49	170	255	0	255
50	170	255	85	255
51	170	255	170	255
52	170	255	255	255
53	255	0	85	255
54	255	0	170	255
55	255	85	0	255
56	255	85	85	255
57	255	85	170	255
58	255	85	255	255
59	255	170	0	255
60	255	170	85	255
61	255	170	170	255
62	255	170	255	255
63	255	255	85	255
64	255	255	170	255
65	0	0	0	128
66	255	0	0	128
67	0	255	0	128
68	255	255	0	128
69	0	0	255	128
70	255	0	255	128
71	0	255	255	128
72	255	255	255	128
73	170	0	0	128
74	0	170	0	128
75	170	170	0	128
76	0	0	170	128
77	170	0	170	128
78	0	170	170	128
79	170	170	170	128
80	0	0	85	128
81	0	85	0	128
82	0	85	85	128
83	0	85	170	128
84	0	85	255	128
85	0	170	85	128

Index	R	G	B	α
86	0	170	255	128
87	0	255	85	128
88	0	255	170	128
89	85	0	0	128
90	85	0	85	128
91	85	0	170	128
92	85	0	255	128
93	85	85	0	128
94	85	85	85	128
95	85	85	170	128
96	85	85	255	128
97	85	170	0	128
98	85	170	85	128
99	85	170	170	128
100	85	170	255	128
101	85	255	0	128
102	85	255	85	128
103	85	255	170	128
104	85	255	255	128
105	170	0	85	128
106	170	0	255	128
107	170	85	0	128
108	170	85	85	128
109	170	85	170	128
110	170	85	255	128
111	170	170	85	128
112	170	170	255	128
113	170	255	0	128
114	170	255	85	128
115	170	255	170	128
116	170	255	255	128
117	255	0	85	128
118	255	0	170	128
119	255	85	0	128
120	255	85	85	128
121	255	85	170	128
122	255	85	255	128
123	255	170	0	128
124	255	170	85	128
125	255	170	170	128
126	255	170	255	128
127	255	255	85	128
128 -255	Undefined	Undefined	Undefined	Undefined

Table 5-8 shows an operation example of colour map data defined in "Annex A Colour Map Data Coding" of the XML-based multimedia coding scheme.

Table 5-8 Operation Example of Colour Map Data

Data	Value	Meaning
clut_type	0	Use YCBCR colour space only.
depth	'01'	Use 8-bit CLUT (up to 256 colours) only.
region_flag	0	Specify no area.
start_end_flag	1	Specify a start/end index and part of elements of CLUT.
start_index	128	Define only 96 colours of index No. 128 to 223.
end_index	223	
alpha	Other than 0	Exclude αvalue=0.

32 colors of index No. 224 to 255 are reserved as colours (receiver-dependent colours) for the receiver to display EPG, etc. and are not used by any BML document.

A2-5.1.8 Operation of remote control keys

Table 5-9 shows the relationship among a remote control key, a key code and a characters specified as an access key.

Table 5-9 Relationship among Remote Control Keys, Key Codes and Access Keys

Remote control key	Key code	Access key character
↑	1	N/A
↓	2	N/A
←	3	N/A
→	4	N/A
0,1,2,3,4,5,6,7,8,9,10,11,12	5-17	N/A
“Enter”	18	N/A
“Back”	19	‘X’
“Data” (Note 1)	20	N/A
Color key (Blue)	21	‘B’
Color key (Red)	22	‘R’
Color key (Green)	23	‘G’
Color key (Yellow)	24	‘Y’
Data button 1	25	‘E’
Data button 2	26	‘F’
Data button 3	27	N/A
Data button 4	28	N/A
Reserved for ARIB data broadcast standard	29-99	reserved
“Bookmark” key (Note 2)	100	N/A
Reserved for future extended features	101-149	reserved
Vendor-dependent	150-	Not defined in this appendix.

Note 1: An event DataButtonPressed occurs and no keydown nor keyup event occurs.
Note 2: mplementing a “bookmark” key is optional.

The above key codes can be taken out by the keyCode attribute of the BMLIntrinsicEvent interface.

A2-5.1.9 Character entry into input element

Table 5-10 shows an operation example of the remote control keys to be at least supported to enter characters into the input element.

Table 5-10 Character Entry into input Element by Remote control Keys

Remote control key	Application to input element
0,1,2,3,4,5,6,7,8,9	Used to enter numbers 0 to 9. However, numeric-tuning must be contained in used-key-list property.
"Back"	Deletes one character (Backspace)

Support of anything other than numeric entry by remote control keys must depend on the receiver.

When in a content where a “BACK” key or a numeric key is specified as an acces key a focus is placed in the input element, the access key is disabled. In this case, removing the focus from the input element enables the access key.

A2-5.2 Guidelines on behavior of management of external characters

A2-5.2.1 Management of pattern data

Both DRCS pattern data invoked by loadDRCS() and transmitted in data unit of 8-bit coding text data to be referenced by the object element are contained in common area of set-top-box. However this area is independent of the area used for subtitling because DRCS pattern data invoked by loadDRCS()

and transmitted in data unit of 8-bit coding text data to be referenced by object element have no influence on pattern data for subtitling. When setting pattern data invoked by loadDRCS() and transmitted in data unit of 8-bit coding text data to be referenced by the object element, updating only pattern data related to character coding contained in DRCS data and having no influence on pattern data have already set.

A2-5.2.2 Sharing of external characters among documents

How external characters are shared between documents is illustrated in Figure 5-8.

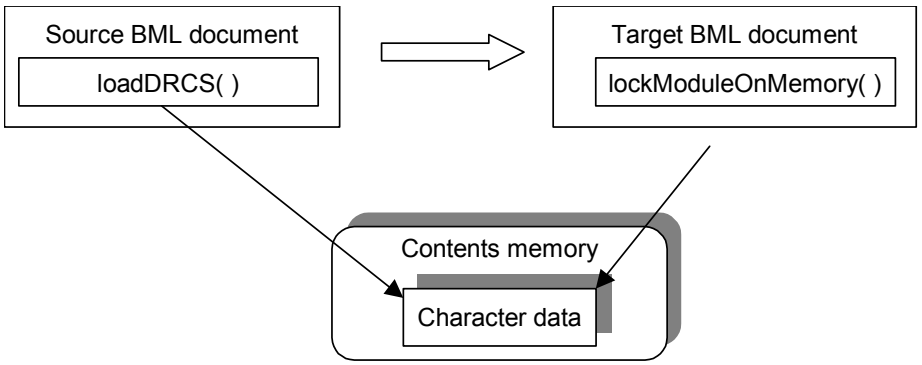


Figure 5-8 Sharing of external characters between documents

[1] External character data is locked on content memory by lockModuleOnMemory( ).

[2] External character data is made available by loadDRCS( ).

[3] One document is switched to the other without starting unloadDRCS( ).

[4] The loadDRCS( ) function is restarted from the jump document to make external character data available.

[5] When unloadDRCS( ) is started or when the scope comes out of the document group, external character data is released.

External character data locked by lockModuleOnMemory( ) is valid within the document group. Therefore, the data can be shared only within the document group.

A2-5.2.3 Presentation of external characters

When the width or height value of an external character do not comply with font-size property which is applied to the p, a, span, and input elements, how the external character is presented depends on a receiver.

A2-5.3 Behavior of DOM

A2-5.3.1 Use of DOM

In using the DOM, it is usual to use the X interface after creating a document structure by the create X() method of Document class if some interface X is utilized. However, generating an explicit structure by the createX() method must be omitted in describing a content conforming to this specification. Therefore, if the DOM is used in the received contents, it is preferable for the receiver to develop the DOM tree in memory before using the DOM.

If an element not defined in the DTD is entered as well, its node is retained as an element with unknown contents but attributes must be allowed to be discarded.

The hasFeature( ) method of the DOMImplementation interface returns true when the feature is "BML" and the version is "1.0", except that there is no distinction between uppercase and lowercase characters.

A2-5.3.2 Handling of control codes (blank, line break, tabs)

If control codes (blank, line break, and tabs) are included in a BML document, DOM is generated with the rule described below.

- Text node is not generated for character strings of control codes.
- If control codes are contained in the character strings of children node of p element, span elements, a elements and input element, the rule in table 5-11 is applied.

Table 5-11 Control codes for character strings of child node of element

Display pattern of control code	Handling when generating DOM	Example
The row of control code just behind of start tag.	All ignored	 abc Character strings data of child node of p element is “abc”
The row of control code just before end tag.	All ignored	defg Character strings data of child node of p element is “defg”.
The row of control code between 2-bytes characters	All ignored	日本語 日本語 Character strings data of child node of p element is “日本語日本語”.
The row of control code between 2-bytes character and 1-byte character.	Blank with 1 byte(0x20)	日本語 Japanese Character strings data of child node of p element is “日本語 Japanese”.
The row of control code between 1-byte characters	Blank with 1 byte(0x20)	abc def Character strings data of child node of p element is “abc def”.

- When span element, a element, and br element are contained in p element, the rules in table 5-12 are applied.

Table 5-12 Control codes in the case of other elements contained in the p element

Display pattern of control code	Handling when generating DOM	Example
The row of control code just after end tag of br element.	All ignored	 abc def All blanks from to “d” are deleted and the character strings of child node of last p element is “def”

Display pattern of control code	Handling when generating DOM	Example
The row of control code just before start tag of span element, an element and br element.	Blank with 1 byte(0x20)	 abc def the character strings of fist child node of p element is “abc ” because 0x20 is contained in the end of “abc”
The row of control code just after end tag of span element, an element.	Blank with 1 byte(0x20)	abcdef ghi def the character strings of last child node of p element is “ def” because 0x20 is contained in the end of “ def”.

- When the characters of one line exceed the width of a block when representing blanks and line breaks of a CDATA section, the exceeded characters of the line will be fold back at the right end.
- Tab is processed as one-byte character code. That is, the width is as same as that of one-byte character codes. Actual representation depends on each implemented receiver.

A2-5.3.3 Configuration example of DOM interface

The following Figure 5-9 shows the inheritance of the DOM interface to be operated. The ellipse in the figure below shows the core DOM interface, and the rectangle shows the interface corresponding to an element.

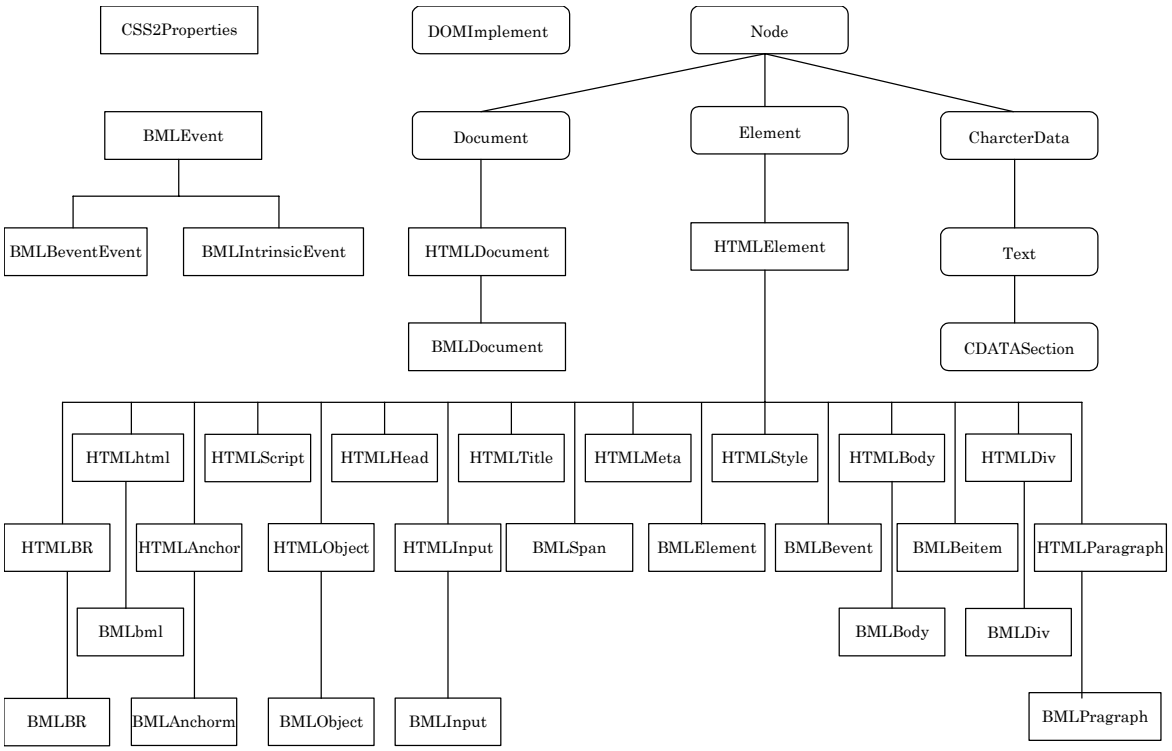


Figure 5-9 Inheritance of DOM Interface

A2-5.3.4 Generation of DOM tree

The following Figure 5-10 shows the order of generating a DOM tree from the BML document. The DOM tree is generated from left to right in the appearance order of BML document. When viewing from object 5, the relationship between node positions is as follows: parentNode is object 2, firstChild is object 7, lastChild is object 9, previousSibling is object 4, and nextSibling is none. If there is no node, null is returned according to the DOM1 rule. When there are only blanks and line feeds between nodes, no node holding such blanks and line-feeds is generated. In addition, the comments must be ignored and no comment nodes must be generated.

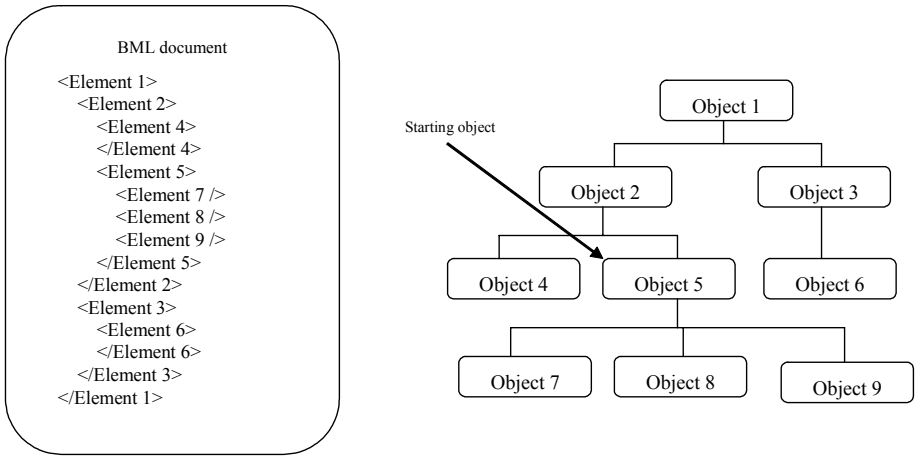


Figure 5-10 DOM Tree Generation

A2-5.3.5 Restriction of DOM work memory

An operation example of DOM objects to be generated is listed in Table 5-13.

Table 5-13 Maximum value of DOM object

Item	Maximum value	Supplement
Number of elements	512 elements	
The total number of nodes	1024 nodes	
Size of character string set to NodeValue	1024 Byte	
Number of child nodes to be connected to one node	255	
Depth of tree structure	16	Count from BMLbmlElement node (this is not contained).

A2-5.4 Operational Guideline of Script Language

A2-5.4.1 Operation of script working environment

An operation example of script work memory is shown in Table 5-14.

TABLE 5-14 Restriction on the script work memory

Item	Maximum value	Supplement
No. of characters of a symbol name	255 Bytes	
Function argument	255	
Local variable	255	

Item	Maximum value	Supplement
Total size of overall character string length	65536Byte	Including the content of character string variable, function name, variable name and properties.
Number of instance of objects	256	Including Object, Number, String, Boolean, Array, Date, BinaryTable and, Instance of function.
Number of properties of one object	256	Restriction against Object, Number, String, Boolean, Array, Date, BinaryTable and, Instance of function. Except the number of property corresponding to the attached character in Instance of Array.
Number of elements of one array	1024	
Nest level with function for invoking	16	Including invocation function by event handler
Number of properties of all objects	4096	The total numbers of property that is the subject of restriction of "total number of property in one Instance of one object. Including the property of activation object and arguments object Except the built-in property of global object(built-in function, built-in object, extended function for broadcasting and extended object broadcasting) and host object for DOM.
Number of global variable	64	
Number of definition of function	64	Definition of function is global. Except event handler.
Buffer for monomedia, data size before decoding	1024kB	
Work memory for ECMAScript	8192 Steps	It must be calculate as specified in Annex C of this Appendix.
Number of CSS selectors	512	
Number of CSS declarations	4096	

The size of an event queue must be up to 64 events. No more events are discarded if over the upper limit.

A2-5.4.2 Data type

It is assumed that any data type has the following restrictions:

- Number must be of single precision (32 bits). Note that any Number object must also be of single precision (32 bits).
- Float must not be supported.
- Math built-in object must not be supported.
- Dynamic type conversion must be restricted.
- The run-time interruption of a script character string must not be supported.
- EUC-JP must be used as a character coding scheme of character string data.
- Use of Unicode Value must be restricted.
- Functions for compatibility with old codes must be restricted.

- The argument and the operation result of addDate() and subDate() must be restricted from 1900/3/31 to 2100/2/28.

A2-5.4.2.1 Character string

For operation of character string data, the following restrictions are assumed:

EUC-JP must be used for description of script itself and character strings of String class.

UnicodeEscapeSequence (\uxxxx) in literal description must not be supported.

EUC codes must be used for magnitude comparison operation of character string data (<, <=, >, >=).

A2-5.4.2.2 Type conversion

The restrictions on dynamic type conversion are assumed below.

- A type of method argument is defined to restrict dynamic type conversion in DOM API and broadcast extended classes (The method of Browser quasi-object and extended object for broadcasting).
- Since ECMAScript has no variable type, there is no structure of examining a type on calling a function. Any data type can be assigned in the case of calling an extended function for broadcasting as argument having value of the other type of definition. Calling a function is prohibited if type can not be converted.
- Four basic types have been defined and regulated as mentioned below according to the operating restrictions.
  [1] Numeric type (Number) must be a 32-bit single precision integer.
  [2] Character string type (String) must be a character string coded by EUC―JP.
  [3] Logical type (Boolean) has no particular restrictions.
  [4] Object type (Object) has no particular restrictions.
- If the argument is specified as NaN where it must be a Number type and there is no particular description, it must be taken as a failure instead of 0.

The rules of conversions are listed in Table 5-15.

Table 5-15 Type conversion rules

Value	Character string	Number	Logic	Object
Character string	-	Numeric value of character string	true	String
Empty character string	-	0	false	String
0	“0”	-	false	Number
NaN	“NaN”	-	false	Number
Number	Numeric character string	-	true	Number
True	“true”	1	-	Boolean
False	“false”	0	-	Boolean
Object	toString()	valueOf()	true	-
Null	“null”	0	false	-
Undefined value	“undefined”	NaN	false	error

A2-5.4.3 Effects on basic objects caused by data type restrictions

Data restriction causes the following language effects:

A2-5.4.3.1 Effects caused by number object of signed 32-bit integer

- Behavior of Date object
  - Calculation of Date object is returned in milli-second order as Number object. However overflow or underflow can be assumed if Number object is under 64bit. subDate() and addDate() are used in calculation of Date object.
  - The precision of Date object must be kept while internally processed (not being affected by the restriction on Number object).
- Operation range of DOM object
  - valueOf() method is not supported.
  - getTime() method is not supported. (A method for individually acquiring hours, minutes and seconds is used instead.)
  - setTime() method is not supported. (A method for individually acquiring hours, minutes and seconds is used instead.)
  - Date.parse() method and Date.UTC() method are not supported. (Numeration of time representation is restricted.)
  - Operations on Date object are not supported. Object the guidelines when operating the Number object of less than 64 bits. (subDate() function or addDate() function is used instead.)
  - Actual [[Value]] property of date object instance depends on an implementation. Therefore the values returned with method in Table 4-18 must depend on implementation.
- Number object
  - MAX_VALUE attribute must be 2147483647. (Maximum value of signed 32-bit integer)
  - MIN_VALUE attribute must be 1. (Minimum value of signed 32-bit integer)
  - If the value exceeds 2˜31, the process depends on the implementation.
  - If the string “Infinity” is not treated particularly and converted by ToNumber, the return value is NaN.
  - The exponential expression is not used in the operation.

A2-5.4.3.2 Behavior in the case of not using Float

- Number object
  - NaN attribute is not expressed with IEEE754 but depends on an implementation.
  - MAX_VALUE attribute and MIN_VALUE attribute are defined with integer number which can be expressed by Number object.
  - NaN is returned in the case of division by 0.
  - NEGATIVE_INFINITY attribute is not supported.
  - POSITIVE_INFINITY attribute is not supported.
  - NaN attribute must be able to be distinguished from an integer which can be expressed by Number object. (Internal representation must depend on implementation)
  - The result of a division by 0 returns NaN.

- No Math objects are supported.
- Global object
  - The infinity attribute is not supported.
  - The parseFloat() function is not supported.
  - The isFinite() function is not supported.

A2-5.4.3.3 Effects by restrictions on run-time interpretation of script character string

- The eval() function of the Global object is not supported.
- No constructor new Function() of the Function object is supported.

A2-5.4.3.4 Behavior in case of using EUC-JP character code

- String object
  - Only EUC-JP is considered as character code of String object.
  - The character is not counted by number of byte but by number of character.

Example:
var mojiretu1 = "123あいう456ういあ";
mojiretu1.length is 12.
mojiretu1.charAt(4) is "い".
mojiretu1.indexOf("う", 0) is 5.
mojiretu1.lastIndexOf("う", 12) is 9.
mojiretu1.split("4") is divided into "123あいう" and "456ういあ".
mojiretu1.substring(4, 9) is "いう456".
var mojiretu2 = String.fromCharCode(0x31, 0x32, 0x41, 0x42, 0xa4a2, 0xa4a4, 0x61, 0x62);
mojiretu2 is "12ABあいab".
mojiretu2.charCodeAt(5) is 0xa4a4.
mojiretu2.toLowerCase() is "12abあいab".
mojiretu2.toUpperCase() is "12ABあいAB".

A2-5.4.3.5 Effects by operation of EUC-JP only

- Global object
  - The escape() function is not supported.
  - The unescape() function is not supported.
- String object
  - Conform to the guidelines when using EUC-JP as a character code.

A2-5.4.3.6 Restrictions of functions left for compatibility with older codes

- Date object
  - The getYear() method is not supported. (The getFullYear() method is used.)
  - The setYear() method is not supported. (The setFullYear() method is used.)
  - The toGMTString() method is not supported. (The toUTCString() method is used.)

A2-5.4.4 Operational general rule of implementation-dependent behavior

This section contains operational general rules of implementation-dependent behaviors.

A2-5.4.4.1 Main syntax

- Sequence of the properties taken out by “for (variable in Expression) statement”(ECMA-262 p.58)
  It must remain implementation-dependent. The “for (variable in Array object)” sequence is also implementation-dependent.

A2-5.4.4.2 Host object

- Host object range
  - DOM object
  - Browser pseudo object
  - Ureg pseudo object
- Result of typeof for host object (ECMA-262 p.43)
  Returns "hostobject."
- Result of new Object(hostobject) for host object (ECMA-262 p.66)
  Returns reference to hostobject in the same manner as when using a normal object as an argument.
- Results of Array.prototype.join(), Array.prototype.reverse() and Array.prototype.sort() for host object (ECMA-262 p.71,72,73, Section 15.4.4.3)
  Inhibits addition of any property to the host object.

  Remarks: There is no need to consider this behavior because it becomes impossible to insert Array.prototype.join, etc. as a new property of the host object.
- [[Class]] of host object
  "hostobject"

A2-5.4.4.3 Built-in object

- [[prototype]] (ECMA-262 p.62) of Global object
  null
- Result of Function.prototype.toString() (ECMA-262 p.69)
  "function FUNCTIONNAME() {}"(FUNCTIONNAME is the name of a specified function.)
- Result of Array.prototype.sort() (ECMA-262 p.72)
  An array element that is not influenced on sorting may not be automatically generated as an undefined. The internal comparison sequence of sort() may depend on implementation.
- Result of Number.prototype.toString(radix) without radix = 10 (ECMA-262 p.81)
  Only radix = 8,10, or 16 is applicable. Otherwise, operation depends on implementation.
  radix = 8: ["0"-"7"]*
  radix = 16: ["0"-"9","a"-"f"]*
- Result of Date.prototype.toString() (ECMA-262 p.95)
  Must be in the format of "DateThoursminutesseconds."
  Date must be YYYY-MM-DD. (Ex. 1999-01-01)
  Hours, minutes and seconds must be hh:mm:ss. (Ex.: 23:01:34)

'T' (character code 0x54) must be used as a delimiter between the date and the hours, minutes and seconds. (Ex.: 1999-01-01T23:01:34)
If the result is a negative value, the low-order four digits are used and the sign (d.c. or a.c.) is ignored.

- Results of Date.prototype.toLocaleString() and Date.prototype.toUTCString() (ECMA-262 p.100, 101)
must be of the same output format as Date.prototype.toString().
- LocalTZA of Date object must be fixed to +9 hours of Japan Standard Time.
- TimeClip() clip range (ECMA-262 15.9.1.14)
Must be within the range of signed 64-bit integers.

A2-5.4.4.4 Implementation of event handler

The code type of the event handler must be an implementation supplied code. Also, neither ImplicitThis nor ImplicitParent must be set. Even if an event handler return value is false, the next processing is performed continuously regardless of its value.

A2-5.5 Extended Objects for Broadcasting

Operation examples of table objects are shown below.

A2-5.5.1 Operation of CSVTable class

No operation is mandatory.

A2-5.5.2 Operational guideline on BinaryTable class

A2-5.5.2.1 Operation of BinaryTable object

An operation example of BinaryTable object is regulated as follows:

Table 5-16 Operational restrictions on BinaryTable object

Item	Value	Supplement
Maximum number of rows	4096 rows	
Maximum record length	1024 Bytes	
Maximum No. of fields in one record	32 fields	
Maximum size of BinaryTable object	255 KB	
Maximum number of comparison conditions	4	

A2-5.5.2.2 Behavior of BinaryTable

As shown in Figure 5-11, constructor BinaryTable allows reference of table data which resides in content memory. This reference is valid until close( ) method is invoked or the document is finished.

- If the same module contains a binary table and a BML document to be referenced, the binary data cannot be released from content memory by the close( ) method.
- If a different module contains a binary table and a BML document to be referenced, the constructor continues to operate until receiving operation is completed unless table data resides in content memory when invoked. If a piece of binary table data consists of one module when close( ) is booted, the data can be released from content memory. If data other than binary table

data is contained in the module and is locked, it cannot be released.
The rows and columns which indicate record length as shown in Figure 5-11 cannot be specified.

- When a BinaryTable object is referenced through a variable, in order to read a new module that contains the BinaryTable object, the close () must be applied to the Binarytable object before the reading process is executed.
  The steps are described below.

  1) Use the beitem element to require a module update to be detected.
     A module update is detected.

  2) Execute the following in this order through handling a ModuleUpdated event.
     - Apply close() to a binary table.
     - Execute unlockModuleOnMemory().
     - Use lockModuleOnMemory() to load a new module in content memory.

  3) A ModuleLocked event is detected.
     Execute the following in this order through handling a ModuleUpdated event.
     - Execute new BinaryTable().

In the second step, close() and lockModuleOnMemory() must be executed in the described order. If they are executed in the inverted order, BinaryTable continues to refer toan existing module, it is not ensured that a new module is loaded, as intended.

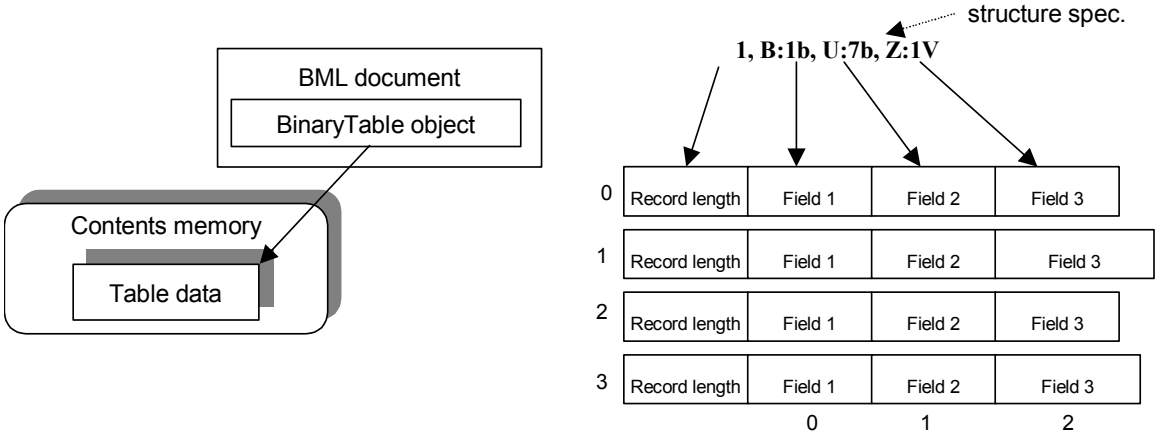


Figure 5-11 Behavior of BinaryTable

A2-5.5.2.3 Inter-document sharing

The following shows an operation example of sharing a binary table between documents in a document group. The binary table and the BML document to be referenced are transmitted in another modules. How to share binary table data is shown below with Figure 5-12.

[1] Using lockModuleOnMemory( ), the binary table data is locked in content memory.

[2] The locked binary table data is retained in the document group.

[3] BinaryTable constructor generates a binary table object.

[4] The document is changed over without invoking the close method.

[5] The binary table data is referenced by the BinaryTable constructor of the target document again.

[6] When the scope comes out of the document group or when close( ) is done, the binary table data is released.

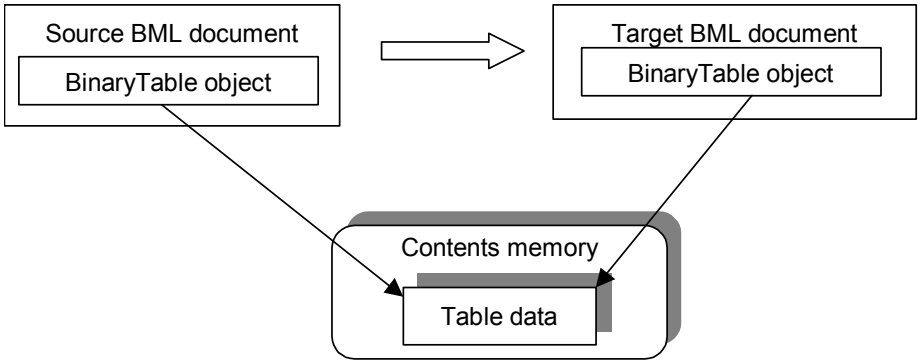


Figure 5-12 Inter-document sharing of binary table

A2-5.5.2.4 Operational restrictions on the structure argument

- The unit of the record length specified with lengthByte is byte. The record length specified with lengthByte does not contain the size of lengthByte itself.
- The value of lengthByte is 0, 1,or 2. And 0 is specified only at the time when all records have the fixed length field. At the same time when all records have the fixed length field, it must be set to 0.
- A hexadecimal character string to be specified to the field length doesn't begin with 0.
- When unit is "V" (a variable length field), the length field does not include its own field. For example, when a character string "ABC" is stored as a variable length character string, '3' is stored in the length field.
- When String-type data are specified to a variable length field, the length data must be either one byte or two bytes.
- The designation of a postal code is "Z : 1V".
- The padding is employed to make the data byte-aligned, and the length must be from 1bit to 7bits. Therefore, the padding with the specified number of bytes ("P:? B") is not employed.
- Even when a BinaryTable is composed of the fixed length field only, the start position of the record is byte-aligned. For example, when the structure is specified with "0, B : 1b", the second byte does not start from the second bit of the first byte, but from the second byte.
- When the field type is "U" (Unsigned Integer) and the length is specified with less than seven bits, it is not necessary to be byte-aligned. For example, when it was specified with U : 1b, U:1b, U:3b, U:4b, U:6b, U2b", the data specified by four bits and two bits extend over the byte alignment.
- When the data length of a String type data is 0-byte, this data is assumed to be an empty string.
- When the data length of a ZipCode type data is 0-byte, this data is assumed not to contain a specified zip code.

A2-5.5.2.5 The property of BinaryTable

- The nrow and ncolumn properties are in Number type.

A2-5.5.2.6 BinaryTable.prototype.search()

- The maximum number of the comparison condition is 4.

- When the number of the retrievals does not reach limitCount, the number of the elements of resultArray is the number of the retrieval result.
- The number of the retrieval result is always set up in length of resultArray. In other words, it is 0 (zero) when no more than 1 case is found, and the number of the results is set up in length of resultArray when any matched items are found.
- When a limitCount-th item is found by the last record, the return value is the position of the last record.
- It judges "It agrees in the retrieval condition." by "The result of = = ToBoolean (operation result) is true." In other words, an operation result is false for Undefined, Null, false, 0 (zero), NaN, the empty character string, and it is true except for them.
- For an operator 6 to 8, the operation result is the data masked by the size of the field.
- An operator 9 to 11 (bit operation with the 1's compliment) operates to have 1's compliment of the data in the table, makes bit-operation with compared, masks with the range of the field size and obtains the operation results.
- It becomes an error when any object except for the Array object is specified in resultArray.
- When two retrieval expressions are specified with OR for a postal code field, 'true' is returned if at least one retrieval condition is satisfied.
- For example, when a table data is "1230001" and the retrieval expression is "postal code = = 1230001 || postal code = = 1230002", 'true' is returned.

A2-5.5.2.7 The behavior after close()

- When the nrow or ncolumn property is referenced, or the toNumber()method is invoked after close(), the following behavior depends on a receiver.

A2-5.6 Guideline for Behavior of Browser Pseudo Objects

The following define the guidelines for behavior of browser pseudo-object. The other methods defined in this chapter are operated as described in the specification.

A2-5.6.1 Operation of Ureg

Ureg values must be valid within the content group. The browser writes a content group identifier into Ureg[0] automatically when acquiring the initial document of content group.

An operation example of Ureg is shown below.

- The browser sets the content group identifier into Ureg[0] automatically upon changeover of the content group. The content identifier is service_id in the hexadecimal character string format of "0xXXXX".
- The browser checks Ureg[0]. Judging the content group identifier to be for own contents, the browser writes marker data as a scope index and marks that the subsequent data is valid.
- When the document is changed over, the browser checks Ureg[0] to see if it is marker data. If so, the browser judges data of Ureg[1] to Ureg[63] to be valid. When a content group identifier is automatically written, the browser judges the data to be invalid.
- A document of non-content group is displayed.
- The browser automatically sets a content group identifier to Ureg[0] again. The browser checks Ureg[0] and judges the data to be invalid.

An access example is shown below.

Ex.: var x = browser.Ureg[1];

A2-5.6.2 Guideline for behavior of EPG functions

An operation example of EPG-related methods is shown below.

- epgGetEventStartTime( )

  The event which can be designated by event_ref must be the one currently displayed.

  This function acquires a start_time value by referencing EIT.

- epgGetEventDuration( )

  The event which can be designated by event_ref must be the one currently displayed.

  This function acquires a duration value by referencing EIT.

- epgTune( )

  Once the event handler that invoked the API, the browser causes an onunload event immediately and places it into the event queue. Then the browser displays a given service after terminating the event handler started by the given onunload.

- epgReserve( )

  The startTime must be designated.

  When a view reserved by this method overlaps with another at the specified time, handling depends on an implementation.

- epgRecReserve()

  The startTime must be designated.

  When a view reserved by this method overlaps with another at the specified time, handling depends on an implementation.

- epgIsReserved( )

  The startTime must be designated.

- epgRecIsReserved( )

  The startTime must be designated.

A2-5.6.3 Operation of non-volatile memory function

A rule of correspondence between the file name specified with the argument filename of the non-volatile memory functions and the area of NVRAM is defined by a service provider.

- writePersistentArray( )

  The period, an optional argument, is not essential for operation. Written data must be allowed to be overwritten without its life controlled.

  Designation of structure is the same as the binaryTable object but has the following restrictions in Table 5-17:

Table 5-17 Restriction on structure specified in writePersistentArray()

Available field type	Corresponding data type	Operational restrictions
B	Boolean	An error results if byte alignment causes a mismatch.
I	Number	The upper bits are cut off if less than 32 bits are specified. An error results if a value of more than 32 bits is set.
U	Number	The upper bits are cut off if less than 32 bits are specified. An error results if a value of more than 31 bits is set. An error results if a negative value is set.
S	String	The latter half of a character string is rounded off if longer than specified character string length; otherwise, a blank (0 x 20) is added. A variable-length character string is specified by S:1V.
P	-	

When actual data type is different from BinaryTable data type, the type conversion rule mentioned in 5.4.2.2 of this Appendix is used. If the rule is not applicable, an error is returned. And, if the array data does not reach the number of fields specified by structure, the result must be an error. Actual data type used when 'P' is specified for the field format depends on an implementation. When an array of two or more dimensions, no succesful operation is guaranteed.

- readPersistentArray( )

  The operation of structure must be the same as writePersistentArray( ). When a structure value that is not what is specified with writePersistentArray( ) is read, no successful operation is guaranteed.

A2-5.6.4 Operation of subtitle presentation control

An operation example of the methods related to presentation of subtitle is shown below. This is used together with CCStatusChanged, which is an interrupt event that notifies selection of a subtitle language and changeover of display status.

- setCCDisplayStatus( )

  If presentation of the current language and another one is specified by executing setCCDisplayStatus() in the receiver which is not capable of presenting multiple languages simultaneously, the current language is quit and the specified language is presented.

A2-5.6.5 Operation of interaction channel function

A2-5.6.5.1 The correspondence of the extended functions for the broadcasting in interaction channel communication using the BASIC protocol and their return values

- connect()

  The detailed descriptions for the return value are as follows.

Table 5-18 Return value of connect()

Return value	meaning	Detailed description
1	success	After dial tone is detected, it is dialled, a host number is sent after the modem negotiation (note 1) , and a link is established. However, when dial-tone detection is made void with the setting of the terminal, it is dialled after more than three seconds are passed.
-1	Error in parameters	The argument error of connect().
-3	Time out is occurred	Link establishment was not completed in the time out value.
-4	No dial tone is detected	No dial tone is detected. However, if dial-tone detection is made void with the setting of the terminal, it does not occur.
-5	No carrier is detected	A dial was done, and a carrier was not detected after the host center had answered. Or response from host center was not detected after dialling.
-6	It is forced to be disconnected	After the host center had answered, the line was disconnected by the center or the receiver.
-7	Modem is in use	Modem is used by other software or it is used by BML browser already.
-8	Line is busy	Busy tone is detected after dialling.
NaN	Failure caused by other reason	The condition of the automatic re-dialling restriction is detected.

Note 1: PIAFS protocol for PHS

When the return key is pressed during the execution of connect(), the connection is stopped, and -6 is returned in the return value. It may be ignored this when any key except for the return key is pushed.

- disconnect()

The detailed descriptions for the return value are as follows.

Table 5-19 Return value of disconnect()

Return value	meaning	Detailed description
1	success	The line could be released normally or was empty already
NaN	failure	The line could not be released normally.

- sendTextData()

The detailed descriptions for the return value are as follows.

Table 5-20 Return value of sendTextData()

Return value	meaning	Detailed description
1	success	Data transmission has been completed.
-1	Error in parameters	Argument error of sendTextData
-2	Line is disconnected during data transfer	The disconnection of the line was detected during the data transfer or the line was disconnected by 3 times retry over after receiving NAK.
-3	Time out is occurred	The data transfer was not completed in the time out value.
NaN	Failure caused by other reason	

- receiveTextData()

The detailed descriptions for the return value are as follows.

Table 5-21 Return value of receiveTextData()

Return value	meaning	Detailed description
Received character string	success	Data reception is completed.
null	failure	Data receiving is not completed in the time out value

- vote()

The detailed descriptions for the return value are as follows.

TABLE 5-22 Return value of vote()

Return value	meaning	Detailed description
1	success	After dial tone is detected, it is dialled, and as a reverse polarity is detected after connection, it is disconnected. However, when dial-tone detection is made void with the setting of the terminal, it is dialled after more than three seconds are passed.
-1	Error in parameters	Argument error of vote
-4	Dial tone is not detected	No dial tone is detected. However, if dial-tone detection is made void with the setting of the terminal, it does not occur.
-6	Line is forced to be disconnected	Before finishing communication normally after dialling, it is disconnected by receivers.
-7	Modem is in use	Modem is used by other software or it is used by BML browser already.
NaN	Failure caused by other reason	

Note 1: If a receiver does not support the detection of a reverse polarity, or if such connection is set up as a reverse polarity cannot be transmitted from the switching machine, it is determined as success (i.e. normal completion) when time-out is detected or the BUSY signal is detected with the disconnection of the line by the switching machine after dialling.

When the return key is pressed during the execution of vote(), the connection is stopped, and -6 is returned in the return value. It may be ignored this when any key except for the return key is pushed.

- startCASEncryption()

The detailed descriptions for the return value are as follows.

Table 5-23 Return value of startCASEncryption()

Return value	meaning	Detailed description
1	success	In order to start IRD data communication, CA broadcaster ID and center ID were notified to IC card and accepted.
-1	Error in parameters	Argument error of startCASEncryption
-50	CAS is not equipped	It is detected that the CAS module is not equipped.
-51	IC card is not inserted	It is detected IC card is not inserted.
NaN	Failure caused by other reason	“CAS error” other than that CAS is not equipped and that IC card is not inserted.

- endCASEncryption()

The detailed descriptions for the return value are as follows.

Table 5-24 Return value of endCASEncryption()

Return value	meaning	Detailed description
1	success	As IRD data communication is finished, the end message is sent to IC card and accepted.
NaN	Failure caused by other reason	"CAS error" other than that CAS is not equipped and that IC card is not inserted.

- transmitWithCASEncryption()

The detailed descriptions for the return value are as follows.

Table 5-25 Return value of transmitWithCASEncryption()

Return value	meaning	Detailed description
1	success	While telegrams are encrypted and decrypted, send and receive to the center is normally finished.
-1	Error in parameters	Argument error of transmitWithCASEncryption()
-2	Line is disconnected during data transfer	The disconnection of the line was detected during the data transfer or the line was disconnected by 3 times retry over after receiving NAK.
-3	Time out is occurred	The data transfer was not completed in the time out value.
-50	CAS is not equipped	It is detected that the CAS module is not equipped.
-51	IC card is not inserted	It is detected IC card is not inserted.
NaN	Failure caused by other reason	"CAS error" other than that CAS is not equipped and that IC card is not inserted.

A2-5.6.6 Line status after the execution of the functions for interaction channel

There are two line statuses, "vacancy" and "busy". And the transition is based on the execution result of the function. Figure 5-13 shows the general concept of the line status transition.

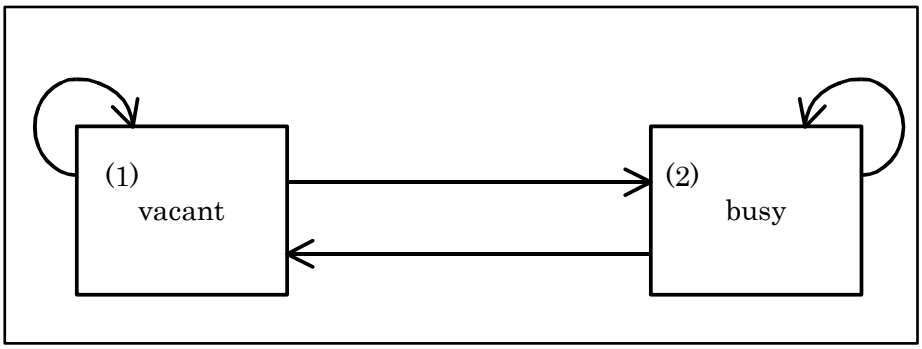


Figure 5-13 Status transition of the line connection

Table 5-26 shows the line status before and after the execution of the functions.

Table 5-26 Line status before and after the execution of the functions

Extended functions for broadcasting	Return value		Line status		Release process of the line	Remarks
			Status number			
			Before execution	After execution		
connect	1	Success	(1)	(2)		
	-1	Parameter error	-	-		
	-3	Time out is occurred	(1)	(1)	required	
	-4	Dial tone is not detected	(1)	(1)		
	-5	No carrier is detected	(1)	(1)	required	
	-6	Line is enforced to be disconnected	(1)	(1)	required	
	-7	Modem is busy	- (CAS)	(2)		Because the line is used by other software, it is not disconnected.
			(2)	(2)		Because the line is used by other software, it is not disconnected. (Note 1)
	-8	Line is busy	(1)	(1)		
	NaN	Failure caused by other reason	(1)	(1)	required	
disconnect	1	Success	(2)	(1)	required	
			(1)	(1)		(Note 2)
	NaN	Failure	(2)	(2)		
sendTextData	1	Success	(2)	(2)		
	-1	Parameter error	-	-		(Note 4)
	-2	Line is disconnected during data transfer	(2)	(1)	required	Line may be disconnected by the center.
	-3	Time out is occurred	(2)	(1)	required	
	NaN	Failure caused by other reason	(2)	(1)	required	
			(1)	(1)		(Note 3)
receiveTextData	Received character string	Success	(2)	(2)		
	null	Failure	(2)	(1)	required	Line may be disconnected by the center.
			(1)	(1)		(Note 3)
vote	1	Success	(1)	(1)	required	Line is used and disconnection process must be executed in the function after the detection of connection as well.
	-1	Parameter error	-	-		(Note 4)

Extended functions for broadcasting	Return value		Line status		Release process of the line	Remarks
			Status number			
			Before execution	After execution		
	-3	Time out is occurred	(1)	(1)	required	
	-4	Dial tone is not detected	(1)	(1)		
	-6	Line is enforced to be disconnected	(1)	(1)	required	
	-7	Modem is busy	- (CAS)	(2)		Because the line is used by other software, it is not disconnected.
			(2)	(2)		Because the line is used by other software, it is not disconnected. (Note 1)
	NaN	Failure caused by other reason	(1)	(1)	required	
startCASEncryption	1	Success	-	-		(Note 5)
	-1	Parameter error	-	-		(Note 5)
	-50	CAS is not equipped	-	-		(Note 5)
	-51	IC card is not inserted	-	-		(Note 5)
	NaN	Failure caused by other reason	-	-		(Note 5)
endCASEncryption	1	Success	(2)	(2)		
	NaN	Failure caused by other reason	(2)	(1)	required	Line may be disconnected by the center.
			(1)	(1)		(Note 3)
transmitWithCASEncryption (Array[0])	1	Success	(2)	(2)		
	-1	Parameter error	-	-		(Note 4)
	-2	Line is disconnected during data transfer	(2)	(1)	required	Line may be disconnected by the center.
	-3	Time out is occurred	(2)	(1)	required	
	-50	CAS is not equipped	(2)	(1)	required	
	-51	IC card is not inserted	(2)	(1)	required	
	NaN	Failure caused by other reason	(2)	(1)	required	Line may be disconnected by the center.
			(1)	(1)		(Note 3)

Note 1: In the case where connection is requested during using a line.
Note 2: In the case where release of the line is requested while it is vacant.
Note 3: In the case where data transmission and reception requested while it is vacant.
Note 4: It does not affect the transition of the connection status of the line.
Note 5: It is executed entirely while the line is vacant, and does not affect the transition of the connection status.

A2-5.6.6.1 Guideline on a telephone number specified in the first argument of connect() and vote()

A valid character for a telephone number is shown in the following.

Telephone line	A valid character
Pulse	0 ~ 9,“(”,“)”,“-”,“T”(The character string of the rest is outputted with tone),“,”(pause designation)
tone	0 ~ 9,“(”,“)”,“-”,“*”,“#”,“,”(pause designation)

When characters except for the valid character listed above are specified, the character string without them is used. The maximum size of the character string specified as a telephone number is 30 bytes, including the characters which can't be used as a telephone number. When it exceeds the size, NaN (failure) is returned.

A2-5.6.6.2 Guideline on the encrypted communication using CAS

The encrypted communication using CAS is executed with the following sequence.

[1] The startCASEncryption() function is executed, and the use of the CAS encryption function is started.

[2] The following series of process is carried out by the execution of the connect() function.

(1) Call to the center is invoked.

(2) A "call connection status notice command" is sent to the IC card.

(3) Data in the IC card are received.

(4) Data received in the process (3) are sent to the center.

(5) Data from the center are received.

(6) A center response command is sent to the IC card.

(7) Data in the IC card are received.

(Success or failure is judged by the status of the data.)

[3] The following series of process is carried out by the execution of the transmitWithCASEncryption() function.

(1) The encryption of the IRD data is done.

(2) Encrypted IRD data are transferred to the center.

(3) Encrypted IRD response data from the center are received.

(4) Encrypted IRD response data is decrypted.

[4] It is possible to execute transmitWithCASEncryption() more than one time in one communication process.

[5] The following series of process is carried out by the execution of the endCASEncryption() function.

(1) A "IRD data transmission completion command" is sent to the IC card.

(2) Data in the IC card are received.

(3) The data received in the process (2) are sent to the center.

(4) Data from the center are received.

(5) A "center response command" is sent to the IC card.

(6) Data in the IC card are received.

(Success or failure is judged by the status of the data.)

[6] Disconnection of call is done by the execution of disconnect().

The plain text cannot be transmitted using sendTextData (), receiveTextData() in the above sequence. And, the maximum data length for transmission and reception in transmitWithCASEncryption() is 4096 bytes, where the length is measured by the number of byte for the plain text, which is not enciphered.

A2-5.6.6.3 Guideline on the transmission and reception of the text data

When the text data is transmitted and received using sendTextData () and receiveTextData(), the following must be obeyed.

- The sendTextData() and receiveTextData() functions are executed in the pair.
- The character code must be in EUC-JP.
- The maximum number of the characters to be transmitted and received by the execution of the function at one time is 4096 bytes.

A2-5.6.6.4 Congestion Avoidance

A delayed call is configured by using random numbers, the timer function, a specified time, any combination of them, or all of them. And, as for the data associated television service, services using interaction channel is possible at any time after the television program completion, setting up the event time of the data broadcasting independently with the event of the television. Therefore, When an event program is expected to cause congestion, making a service time of the data broadcasting service via interaction channels longer is useful to avoid congestion.

A2-5.6.6.5 Others

- The series of communication process starting from connect() and finishing with disconnect(), or that from startCASEncryption() and finishing with disconnect() in the case of the encryption communication using CAS, must be done in one document group. If a selection of other services or document group switching such as the update of the data event occurred on the way to deal with communication, the communication process is stopped, and the call is interrupted. And, when some difficulties occur in the communication channel while the communication process is on the way (the call is in the connected status.), the call is interrupted.
- All connect (), vote() are handled as an target of the automatic redialling restriction, which does not depend on whether they were carried out by the user's operation or it is called automatically by the timer and so on from the contents. The automatic redialling restriction is specified separately by the broadcasters.

A2-5.6.7 Operation of operational control functions

The following shows an operation example of functions for controlling a receiver’s behaviors.

- getProgramRelativeTime()

  Returns a difference between the start time (starttime) of a broadcast program described in PresentEIT and the current time acquired by the receiver. The difference is calculated on a unit of second and shown in a relative time format.

- lockModuleOnMemory( )

  The module locked by this method must be valid within a document group. The basic operation is shown below.

  [1] lockModuleOnMemory( ) is issued to start acquisition of a module. If the specified module is not found, an error is returned.

  [2] Once the specified module is locked, a moduleLocked event occurs and the module is maintained in content memory.

  [3] When unlockModuleOnMemory( ) is invoked or when the scope is out of document group, the module is unlocked and deleted from content memory.

  [4] If an attempt is made to lock the locked module using lockModuleOnMemory( ), a ModuleLocked event is regenerated.

  The operation where an operationally locked module is deleted from the content memory is shown in Figure 5-14.

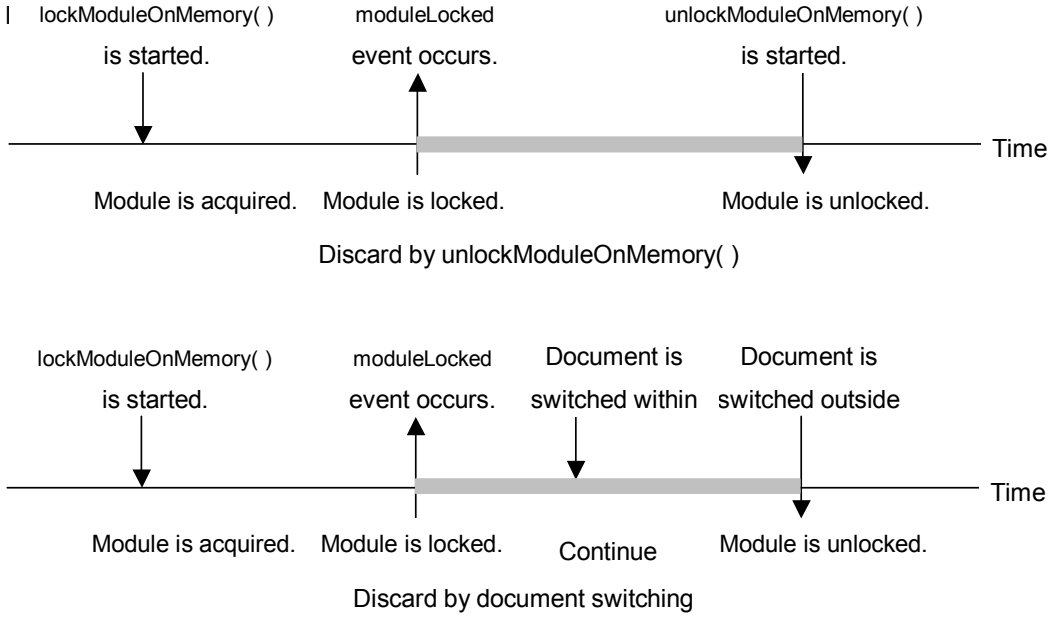


Figure 5-14 Module lock behavior

The following data transmitted from modules can be locked in content memory using lockModuleOnMemory( ):

- Monomedia data other than streams
- Binary table data
- External character data
- CLUT data
- BML document

  Note: An update of the version of the module that is locked in the memory by the execution of lockModuleOnMemory() can be observed. In order to get the new version of data after detecting the update of the version (after ModuleUpdated is occurred), the lock of the old version must be released.

- unlockModuleOnMemory( )

Returns an error when unlockModuleOnMemory( ) is started to a module unlocked in content memory. When unlockModuleOnMemory( ) is executed on a module unlocked in content memory, on which a lock request is being processed, the lock request is cancelled. When unlockModuleOnMemory( ) is executed on a module unlocked in content memory, on which no lock request is being processed, an error is returned.

- setCachePriorty( )

The priority is defined as Table 5-27:

Table 5-27 Cache priority

Priority	Description	Remarks
0	Low priority	Default
63	Medium priority	
127	Cached in top priority	
Minus value	Not cached	

The cache priority is provided merely as a hint. Remember that all values are not cached even when setting top priority.

- getActiveDocument( )

The return value is a URI of shortened form (a relative path starting with “/”).

- lockScreen( )

When lockScreen( ) is started, the presentation screen change which is caused by DOM API is stopped until unlockScreen( ) is started. However, if the event handler which invoked lockScreen( ) is terminated without starting unlockScreen( ), the presentation screen is changed right after termination.

If multiple values are set to the same attribute or if a method is invoked many times, only the last change must be valid.

If lockScreen( ) is invoked multiple times, the invocation is ignored without returning an error. No lock nesting is done.

While lockScreen is being executed, video is presented continuously, however whether terminating updating the presentation frame of MNG or continuing to display the presentation frame of MNG depends on a receiver.

A presentation and manipulation of the object element whose type attribute begins with audio is not subject to a lock status of the screen.

- unlockScreen( )

The invocation is ignored without returning no error if unlockScreen( ) is invoked without invoking lockScreen( ). Even if lockScreen( ) is invoked several times, a module is unlocked by starting unlockScreen( ) once.

- getBrowserSupport( )

When the sProvider argument contains "ARIB" and the functionname argument contains “APIGroup”, any of the following values must not be specified for the additional info argument through a content:

"EPG.Basic", "CC.Control", "Persistent.Basic", "Com.BASIC.Basic", "Com.BASIC.Vote", "Com.BASIC.CAS", "Ctrl.Basic", "Ctrl.NPT", "Ctrl.Cache", "Ctrl.Screen", "RomSound.Basic", "Timer.Basic", "Misc.DRCS", "Misc.Basic", "Misc.Ureg"

This implies that whether or not an extended function for broadcasting that is defined as “O” (required to provide a basic service) is implemented is not verified.

When the functionname argument contains "Storage" and the addtionalinfo argument contains "cachesize", whether or not a content memory value is the sum of a memory size constantly available to a content for locking a module and 1 MB (reserved for obtaining a module) is verified.

- launchDocument( )

  Actual processing a transition effect specified by the second argument “transitionStyle” depends on a receiver.

A2-5.6.8 Operation of receiver build-in sound control

- playRomSound( )

  A sound_id(romsound://<sound_id>) value corresponding to a receiver build-in sound is defined a rule developed by a vendor/service provider.

A2-5.6.9 Operation of timer function

The following shows an operation of timer function:

- setInterval( )

  The first argument is restricted to be only in the “func();” format in the operation, which is the same description of the function call acting as an event handler. Up to 8 timers can be set at the same time. They are in units of 100 msec. The timer precision depends on an implementation.

A2-5.6.10 Operation of external character function

The following shows an operation of external character function:

- loadDRCS( )

  The CharacterCode value for external characters that are retained in a receiver must be 188.

  In the case where DRCS data have been contained in a receiver (e.g. DRCS data and the BML document to be referenced are contained in the same module), loadDRCS() can set the pattern data. However, if DRCS data and the BML document to be referenced have not been contained in receiver (e.g. in the case where they are contained in the different module individually), loadDRCS( ) operation continues until the receiving operation is completed after loadDRCS( ) is invoked.

A2-5.6.11 Name of the extended functions which are proper to service operators

X_sProvider_ZZZZ() is defined by the sProvider value in getBrowserSupport(). For example, if the sProvider value is “ARIB” and the function named as newfunction is added, the extended function of individual provider is X_ARIB_newfunction().

Chapter APPENDIX 2-6 Contents Transmission and Namespace

A2-6.1 Scope Mapping to Transmission System

BML document switching operation is performed when:

- Switching documents using anchor element in the content.
- Switching documents using launchDocument( ) in the script of the content.
- Re-tuning.
- Forcedly switching documents by a signal transmitted.

If the scope identifier corresponding to each scope changes during the above operation, the corresponding scope is removed. The scope mapping to transmission system is operated as follows:

Table 6-1 Transmission and scope mapping

Scope	Scope identifier	Corresponding transmission signal
Within document	Document scope identifier	moduleId, resourceName
Within document group	Document group scope identifier	data_event_id
Within content	Content scope identifier	component_tag
Within content group	Content group scope identifier	service_id

Figure 6-1 shows an example in which content group A includes one entry content and two contents, and content group B has one content being broadcasted. Using the numbers in the figure, a transmission example is described below.

Tuning selects a given content to be started. In the example below, content 2 begins.

[1] In content 2, the operating scope when switching documents within data_event_id is between the documents and within the document group.

[2] In content 2, the operating scope when the document is switched by a different data_event_id is between the document groups and within the contents.

[3] The operating scope is between the contents and within the content group when data_event_id is switched, content 2 is forcedly terminated by signal return_to_entry_flag, and the entry content is started.

[4] If the anchor element or launchDocument( ) in the document causes a jump to content 2, the entry content is terminated and content 2 is started. This operating scope is between contents and within the content group.

[5] If returning by the user causes a jump to another content group, content 2 is terminated and content 1 of the group is started. This operating scope is out of the content group.

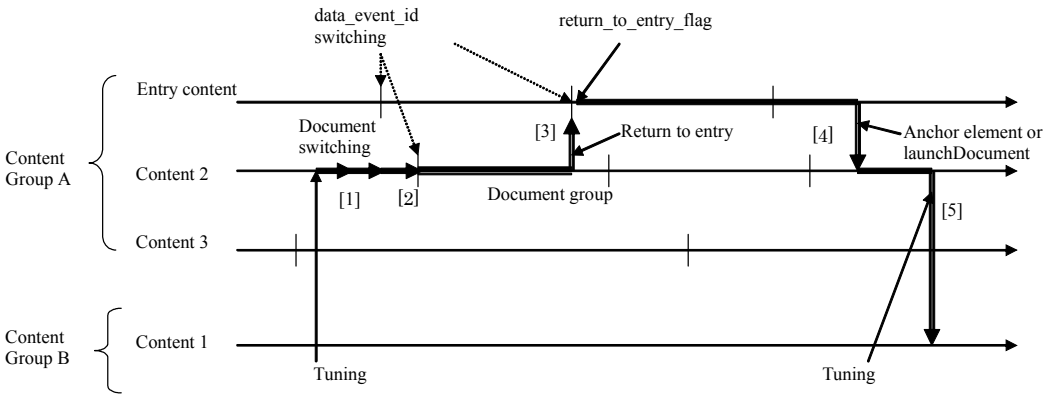


Figure 6-1 Example of contents transmission

A2-6.2 Contents Transmission

A2-6.2.1 Module syntax

- The media type of individual entity-body contained in the entity body using multipart will never be a multipart type.
- The entity headers to be at least interpreted by the basic receiver are as listed in Table 6-2.

Table 6-2 Entity Headers to be at Least Interpreted by Basic Receiver

Entity header	Meaning
Content-Type	Indicates a media type of resource contained in an entity body. Define Content-Type for using multipart as “multipart/mixed” and specify a boundary parameter.
Content-Length	Indicates the byte length of an entity body.
Content-Location	Indicates the identifier of an entity body.

- Content-Base: header and Content-Encoding field must not be essential for operation.
- Content-Location field must be specified relatively from a module and only <resourceName> must be described. <resourceName> must use the namespace of the resource defined in Chapter 9 of the XML-based multimedia coding scheme.

A2-6.2.2 Resource list in multipart format module

A2-A resource list must be always inserted in the first part of a multipart format module.

A2-6.2.3 Operation on event messages

The character code used for private_data_byte of a general event message descriptor must be "EUC-JP."

A2-6.3 Namespace

A2-6.3.1 Namespace used for acquiring resources via telecommunication channels

This is not a mandatory function for basic services.

Annex A Default Style Sheet

```
@media tv{
/* margin */
div, p, input, object {margin: 0 !important}
/* padding */
div, object {padding-top : 0 !important; padding-right : 0 !important; padding-
bottom : 0 !important; padding-left : 0 !important; }
/* border */
div, p, span, a, input {border-width : 0 }
object {border-width : 0 !important; border-style : none !important}
/* display */
meta, title, script, style, head, bml, bevent, beitem, { display :
none !important }
body, div, p, object, input { display : block !important }
br, span, a { display : inline !important }
/* position */
p, div, object, input { position : absolute !important }
br, span, a { position : static !important }
/* top left width height*/
p,div, input, object {top : 0; left : 0; width : 0; height : 0 }
/* z-index */
div, p, br, span, a, input, object, body { z-index : auto !important }
/* line-height */
br, span, a { line-height : inherit !important }
/* visibility */
body { visibility : visible !important }
span, a { visibility : inherit !important }
/* overflow */
p, div, input, object { overflow : hidden !important }
/* background-repeat */
body { background-repeat : repeat !important }
/* text */
p, input { font-family : "round gothic"; font-size : 24px }
span, a { font-family : inherit; font-size : inherit }
p, input { text-align : left }
/* letter-spacing */
span, a { letter-spacing : inherit !important }
/* white-space */
p, input { white-space : normal !important }
/* background-color-index */
body {background-color-index : 0}
div, p, span, a, input { background-color-index : 8 }
object { background-color-index : 0  !important}
/* grayscale-color-index */
p, input { grayscale-color-index : 30 15 }
}
```

Annex B DTD for Checking the Operation Range for Basic Services

B.1 DTD

The following DTD is provided to verify whether or not a BML document is in the operational range defined in this Appendix.

```
<!-- ======== Broadcast Markup Language (BML) 1.0 DTD [OPERATABLE] ======== -->
<!ENTITY % ContentType                  "CDATA" >
<!ENTITY % Charset                      "CDATA" >
<!ENTITY % Character                    "CDATA" >
<!ENTITY % LanguageCode                 "NMTOKEN" >
<!ENTITY % Number                       "CDATA" >
<!ENTITY % URI                          "CDATA" >
<!ENTITY % Script                       "CDATA" >
<!ENTITY % StyleSheet                   "CDATA" >
<!ENTITY % Text                         "CDATA" >

<!ENTITY % Events.attrib
"onclick   %Script;                     #IMPLIED
onkeydown  %Script;                     #IMPLIED
onkeyup    %Script;                     #IMPLIED"
>
<!ATTLIST a
onfocus    %Script;                     #IMPLIED
onblur     %Script;                     #IMPLIED
>
<!ATTLIST input
onfocus    %Script;                     #IMPLIED
onblur     %Script;                     #IMPLIED
onchange   %Script;                     #IMPLIED
>
<!ATTLIST body
onload     %Script;                     #IMPLIED
onunload   %Script;                     #IMPLIED
>
<!ATTLIST div
onfocus    %Script;                     #IMPLIED
onblur     %Script;                     #IMPLIED
accesskey  %Character;                  #IMPLIED
>
<!ATTLIST p
onfocus    %Script;                     #IMPLIED
onblur     %Script;                     #IMPLIED
accesskey  %Character;                  #IMPLIED
>
<!ATTLIST object
onfocus    %Script;                     #IMPLIED
onblur     %Script;                     #IMPLIED
accesskey  %Character;                  #IMPLIED
>
<!ATTLIST span
onfocus    %Script;                     #IMPLIED
```

```
onblur     %Script;                    #IMPLIED
accesskey  %Character;                 #IMPLIED
>

<!ENTITY % Core.attrib
"id ID                                 #IMPLIED
class CDATA                            #IMPLIED
style %StyleSheet;                     #IMPLIED"
>
<!ENTITY % Common.attrib
"%Core.attrib;
%Events.attrib;"
>

<!ENTITY % Inlstruct.class "br | span" >
<!ENTITY % Inline.class
"%Inlstruct.class;
| a"
>
<!ENTITY % Inline-noa.class
"%Inlstruct.class;"
>
<!ENTITY % Blkstruct.class "p | div" >
<!ENTITY % Block.class
"%Blkstruct.class;"
>
<!ENTITY % Boxed.mix
"%Block.class;
| object
| input"
>

<!ENTITY % Br.content "EMPTY" >
<!ELEMENT br %Br.content; >
<!ATTLIST br
%Core.attrib;
>
<!ENTITY % Span.content "( #PCDATA | %Inline.class; )*" >
<!ELEMENT span %Span.content; >
<!ATTLIST span
%Common.attrib;
>

<!ENTITY % Div.content "( %Boxed.mix; )*" >
<!ELEMENT div %Div.content; >
<!ATTLIST div
%Common.attrib;
>
<!ENTITY % P.content "( #PCDATA | %Inline.class; )*" >
<!ELEMENT p %P.content; >
<!ATTLIST p
%Common.attrib;
>
```

```
<!ENTITY % Script.content "( #PCDATA )" >
<!ELEMENT script %Script.content; >


<!ENTITY % Style.content "( #PCDATA )" >
<!ELEMENT style %Style.content; >


<!ENTITY % A.content "( #PCDATA | %Inline-noa.class; )*" >
<!ELEMENT a %A.content; >
<!ATTLIST a
%Common.attrib;
href       %URI;                        #IMPLIED
accesskey  %Character;                  #IMPLIED
>

<!ENTITY % Object.content "EMPTY" >
<!ELEMENT object %Object.content; >
<!ATTLIST object
%Common.attrib;
data           %URI;                    #IMPLIED
type           %ContentType;            #IMPLIED
remain         (remain)                 #IMPLIED
streamposition %Number;    "0"
streamstatus   ( stop | play | pause )  #IMPLIED
>
<!ENTITY % InputType.class
"( text | password )"
>
<!ENTITY % Input.content "EMPTY" >
<!ELEMENT input %Input.content; >
<!ATTLIST input
%Common.attrib;
type       %InputType.class;            'text'
value      CDATA                        #IMPLIED
disabled   (disabled)                   #IMPLIED
readonly   (readonly)                   #IMPLIED
maxlength  %Number;                     "40"
accesskey  %Character;                  #IMPLIED
>

<!ENTITY % Title.content "( #PCDATA )" >
<!ELEMENT title %Title.content; >
<!ATTLIST title

>
<!ENTITY % Meta.content "EMPTY" >
<!ELEMENT meta %Meta.content; >
<!ATTLIST meta
name       NMTOKEN                      #IMPLIED
content    CDATA                        #REQUIRED
>
```

```
<!ENTITY % Head.content "( title, meta?, style?, script?, bevent? )" >
<!ELEMENT head %Head.content; >

<!ENTITY % Body.content
"( div | p )+"
>
<!ELEMENT body %Body.content; >
<!ATTLIST body
%Core.attrib;
invisible  (invisible)                  #IMPLIED
>
<!ENTITY % Bml.content "( head, body )" >
<!ELEMENT bml %Bml.content; >

>

<!ENTITY % bevent.content "( beitem )+" >
<!ELEMENT bevent %bevent.content; >
<!ATTLIST bevent id ID #IMPLIED>
<!ENTITY % BMLEventType
"(EventMessageFired|ModuleUpdated|ModuleLocked|TimerFired
|DataEventChanged|CCStatusChanged|MainAudioStreamChanged|NPTReferred
|MediaStopped|DataButtonPressed)"
>
<!ENTITY % BMLTimeMode
"(absolute|origAbsolute|NPT)"
>
<!ENTITY % beitem.content "EMPTY" >
<!ELEMENT beitem %beitem.content; >
<!ATTLIST beitem
id ID                                   #REQUIRED
type              %BMLEventType;        #REQUIRED
onoccur           %Script;              #REQUIRED
es_ref            %URI;                 #IMPLIED
message_id        %Number;              #IMPLIED
message_version     %Number;            #IMPLIED

module_ref        %URI;                 #IMPLIED
language_tag      %Number;              #IMPLIED
time_mode         %BMLTimeMode;         #IMPLIED
time_value        CDATA                 #IMPLIED
object_id         CDATA                 #IMPLIED
subscribe         (subscribe)           #IMPLIED
>

<!-- End of BML DTD -->
```

B.2 Description of DTD declaratives

The DTD declaratives must be as shown below.

```
<?xml version="1.0" encoding="EUC-JP" ?>
<!DOCTYPE bml PUBLIC
                    "-//ARIB//DTD BML 1.0//JA"
"http://www.arib.or.jp/B24/DTD/bml_1_0.dtd">
<?bml bml-version="1.0" ?>
```

Annex C Counting rule for the restriction on the memory size of ECMAScript

- The same description in ECMA-262 is basically used, as it is described with the same style of yacc grammar file.
- {} means that it is not counted when the formation rule is found. {count++;} means to count it.
- The value of count is set to 0 before parsing starts, and the value of count is examined at the end. This value of count is the number of the program steps of ECMAScript concerned.
- A formation rule must be counted in principle. However, the formation rule which only refers other formation rules isn't counted.

Example 1:

```
AdditiveExpression :
   MultiplicativeExpression
   {}
   AdditiveExpression + MultiplicativeExpression
   { count++; }
   AdditiveExpression - MultiplicativeExpression
   { count++; }
```

The formation rule which only refers the other formation rules is counted.

Example 2:

```
FormalParameterList :
   Identifier
   { count ++; }
   FormalParameterList , Identifier
   { count ++; }
```

Function is counted, but Program is not counted.

Example 3:

```
EmptyStatement :
   ;
   { count++; }
```

The empty formation rule is counted.

The counting rules are shown below. The section number in the comments means the correspondence to ECMA-262.

```
/* 11.1 Primary Expressions */

PrimaryExpression :
     this
     { count++; }
     Identifier
     { count++; }
     Literal
     { count++; }
     ( Expression )
     { count++; }

/* 11.2 Left-Hand-Side Expressions */
```

```
MemberExpression :
      PrimaryExpression
      {}
      MemberExpression [ Expression ]
      { count++; }
      MemberExpression . Identifier
      { count++; }
      new MemberExpression Arguments
      { count++; }

NewExpression :
      MemberExpression
      {}
      new NewExpression
      { count++; }

CallExpression :
      MemberExpression Arguments
      { count++; }
      CallExpression Arguments
      { count++; }
      CallExpression [ Expression ]
      { count++; }
      CallExpression . Identifier
      { count++; }

Arguments :
      ( )
      { count++ }
      ( ArgumentList  )
      {}

ArgumentList :
      AssignmentExpression
      { count++; }
      ArgumentList , AssignmentExpression
      { count++; }

LeftHandSideExpression :
      NewExpression
      {}
      CallExpression
      {}

/* 11.3 Postfix expressions */

PostfixExpression :
      LeftHandSideExpression
      {}
      LeftHandSideExpression ++ /* no LineTerminator here */
      { count++; }
      LeftHandSideExpression -- /* no LineTerminator here */
```

```
      { count++; }

/* 11.4 Unary operators */

UnaryExpression :
      PostfixExpression
      {}
      delete UnaryExpression
      { count++; }
      void UnaryExpression
      { count++; }
      typeof UnaryExpression
      { count++; }
      ++ UnaryExpression
      { count++; }
      -- UnaryExpression
      { count++; }
      + UnaryExpression
      { count++; }
      - UnaryExpression
      { count++; }
      ~ UnaryExpression
      { count++; }
      ! UnaryExpression
      { count++; }

/* 11.5 Multiplicative operators */

MultiplicativeExpression :
      UnaryExpression
      {}
      MultiplicativeExpression * UnaryExpression
      { count++; }
      MultiplicativeExpression / UnaryExpression
      { count++; }
      MultiplicativeExpression % UnaryExpression
      { count++; }

/* 11.6 Additve operators */

AdditiveExpression :
      MultiplicativeExpression
      {}
      AdditiveExpression + MultiplicativeExpression
      { count++; }
      AdditiveExpression - MultiplicativeExpression
      { count++; }

/* 11.7 Bitwize shift operators */

ShiftExpression :
      AdditiveExpression
      {}
```

```
      ShiftExpression << AdditiveExpression
      { count++; }
      ShiftExpression >> AdditiveExpression
      { count++; }
      ShiftExpression >>> AdditiveExpression
      { count++; }

/* 11.8 Relational operators */

RelationalExpression :
      ShiftExpression
      {}
      RelationalExpression < ShiftExpression
      { count++; }
      RelationalExpression > ShiftExpression
      { count++; }
      RelationalExpression <= ShiftExpression
      { count++; }
      RelationalExpression >= ShiftExpression
      { count++; }

/* 11.9 Equality operators */

EqualityExpression :
      RelationalExpression
      {}
      EqualityExpression == RelationalExpression
      { count++; }
      EqualityExpression != RelationalExpression
      { count++; }

/* 11.10 Binary bitwise operators */

BitwiseANDExpression :
      EqualityExpression
      {}
      BitwiseANDExpression & EqualityExpression
      { count++; }

BitwiseXORExpression :
      BitwiseANDExpression
      BitwiseXORExpression ^ BitwiseANDExpression
      { count++; }

BitwiseORExpression :
      BitwiseXORExpression
      {}
      BitwiseORExpression | BitwiseXORExpression
      { count++; }

/* 11.11 Binary logical operators */

LogicalANDExpression :
```

```
        BitwiseORExpression
        {}
        LogicalANDExpression && BitwiseORExpression
        { count++; }

    LogicalORExpression :
        LogicalANDExpression
        {}
        LogicalORExpression || LogicalANDExpression
        { count++; }

    /* 11.12 Conditional operator ( ?: ) */

    ConditionalExpression :
        LogicalORExpression
        {}
        LogicalORExpression ? AssignmentExpression : AssignmentExpression
        { count ++; }

    /* 11.13 Assignment operators */

    AssignmentExpression :
        ConditionalExpression
        {}
        LeftHandSideExpression AssignmentOperator AssignmentExpression
        { count ++; }


    AssignmentOperator :
        =
        *=
        /=
        %=
        +=
        -=
        <<=
        >>=
        >>>=
        &=
        ^=
        |=
        {}

    /* 11.14 Comma operator ( , ) */

    Expression :
        AssignmentExpression
        { count++; }
        Expression , AssignmentExpression
        { count ++; }

    /* 12 Statements */
```

```
Statement :
     Block
     {}
     VariableStatement
     {}
     EmptyStatement
     {}
     ExpressionStatement
     {}
     IfStatement
     {}
     IterationStatement
     {}
     ContinueStatement
     {}
     BreakStatement
     {}
     ReturnStatement
     {}
     WithStatement
     {}

/* 12.1 Block */

Block :
     { StatementList? }
     { count++;  }

StatementList :
     Statement
     { count++; }
     StatementList Statement
     { count++; }

/* 12.2 Variable statement */

VariableStatement :
     var VariableDeclarationList ;
     { count++; }

VariableDeclarationList :
     VariableDeclaration
     { count++; }
     VariableDeclarationList , VariableDeclaration
     { count++; }

VariableDeclaration :
     Identifier Initializer?
     { count++; }

Initializer :
     = AssignmentExpression
     { count++; }
```

```
/* 12.3 Empty statement */

EmptyStatement :
     ;
     { count++; }

/* 12.4 Expression statement */

ExpressionStatement :
     Expression ;
     { count++; }

/* 12.5 The if statement */

IfStatement :
     if ( Expression ) Statement else Statement
     { count++; }
     if ( Expression ) Statement
     { count++; }

/* 12.6 Iteration statement */

IterationStatement :
     while ( Expression ) Statement
     { count++; }
     for ( Expression? ; Expression? ; Expression? ) Statement
     { count++; }
     for ( var VariableDeclarationList ; Expression? ; Expression? ) Statement
     { count++; }
     for ( LeftHandSideExpression in Expression ) Statement
     { count++; }
     for ( var Identifier Initializer? in Expression ) Statement
     { count++; }

/* 12.7 The continue statement */

ContinueStatement :
     continue ;
     { count++; }

/* 12.8 The break statement */

BreakStatement :
     break ;
     { count++; }

/* 12.9 The return statement */

ReturnStatement :
     return  Expression? ; /* no Line Terminator here */
     { count++; }
```

```
/* 12.10 The with statement */

WithStatement :
     with ( Expression ) Statement
     { count++; }

/* 13 Function Definition */

FunctionDeclaration :
     function Identifier ( FormalParameterList? ) Block
     { count++; }

FormalParameterList :
     Identifier
     { count ++; }
     FormalParameterList , Identifier
     { count ++; }

/* 14 Program */

Program :
     SourceElements
     {}

SourceElements
     SourceElement
     {}
     SourceElements SourceElement
     {}

SourceElement :
     Statement
     {}
     FunctionDeclaration
     {}
```

[BLANK]

APPENDIX 3

Operational Guidelines for Implementing Extended Services for Fixed Receiving System

Contents

Chapter APPENDIX 3-1 Purpose 569

Chapter APPENDIX 3-2 Scope 570

Chapter APPENDIX 3-3 Legend 571

Chapter APPENDIX 3-4 Guidelines for Coding of BML Documents 572
A3-4.1 Character Codes 572
A3-4.1.1 Character coding scheme used for BML documents 572
A3-4.1.2 External characters 574
A3-4.1.3 Operation of character reference 575
A3-4.1.4 Operation of character encoding available to contents 575
A3-4.2 Media Types and Operational Range of Monomedia 576
A3-4.2.1 Bitmap 578
A3-4.2.2 Video/Audio 578
A3-4.2.3 8-bit character code 579
A3-4.3 Operation of XML Namespace 580
A3-4.4 Operation of PI 580
A3-4.4.1 Switching presentation model 580
A3-4.5 Operation of BML elements 580
A3-4.5.1 Operational convention on BML elements for XHTML modules 580
A3-4.5.2 Entity 592
A3-4.5.3 Attribute 592
A3-4.6 Operation of CSS-Based Style Sheet 599
A3-4.6.1 Elements applicable to property 601
A3-4.6.2 Operation of selectors 603
A3-4.6.3 Media type 604
A3-4.6.4 !important rule 604
A3-4.6.5 Box model 604
A3-4.6.6 Visual formatting model 606
A3-4.6.7 Other visual effects 608
A3-4.6.8 Generated content, Autonumbering, List 608
A3-4.6.9 Colours and Backgrounds 609
A3-4.6.10 Fonts 610
A3-4.4.11 Text 611

A3-4.6.12 Tables 611
A3-4.6.13 Pseudo classes 612
A3-4.6.14 Extended properties 615
A3-4.6.15 Default style sheet 616
A3-4.6.16 Operation of length 616
A3-4.6.17 Actual values derived from computed values 616
A3-4.6.18 <color-index> 616
A3-4.7 Operational Guidelines on Procedural Description 616
A2-4.7.1 Operational guidelines on DOM 616
A3-4.7.2 Operational scope for built-in objects 644
A3-4.7.3 Operational scope for extended objects for broadcasting 644
A3-4.7.4 Operational scope for browser pseudo objects 644
A3-4.7.5 Operation of Navigator pseudo objects 651
A3-4.8 Restrictions on Description of BML Documents 651
A3-4.8.1 Positioning and Restrictions 651
A3-4.8.2 Vertical writing 652
A3-4.8.3 Restrictions on arrangement of elements 652
A3-4.8.4 Restrictions on elements represented on composition planes of video and still pictures 652
A3-4.8.5 Operation of object element and img element 652

Chapter APPENDIX 3-5 Guidelines on behaviours of BML browsers 655
A3-5.1 Operational guidelines on presentation 655
A3-5.1.1 Presentation model 655
A3-5.1.2 Coordinate system for text and graphics plane 655
A3-5.1.3 Operational guideline on Z-index 655
A2-5.1.4 Presentation plane 656
A3-5.1.5 Operation of monomedia presentation 656
A3-5.1.6 Guideline for clipping 659
A3-5.1.7 Available values to CSS2 properties 659
A3-5.1.8 Rules of box model 659
A3-5.1.9 Operation of colour map data 660
A3-5.1.10 Operation of remote control keys 661
A3-5.1.11 Character entry into input element and textarea element 661
A3-5.1.12 Operation of Focus 662
A3-5.2 Guidelines on behaviours of external characters 662
A3-5.3 Guidelines on behaviours of DOM 662
A3-5.3.1 Use of DOM 662
A3-5.3.2 Handling of control codes (blank, line break, tab) 662
A3-5.3.3 Generation of DOM tree 662
A3-5.3.4 Restriction on DOM work memory 663
A3-5.3.5 Restriction on style description 663
A3-5.4 Operational Guideline on ECMAScript 663
A3-5.4.1 Operation of script working environment 663
A3-5.4.2 Data type 664

A3-5.4.3 Effects on basic objects caused by data type restrictions 664
A3-5.4.4 Operation rule of implementation-dependent portion 664
A3-5.5 Operational guidelines on extended objects for broadcasting 664
A3-5.5.1 Operational guideline on BinaryTable class 665
A3-5.6 Guideline on Behaviour of Browser Pseudo Objects 665
A3-5.6.1 Operation of Ureg 665
A3-5.6.2 Guideline on behaviours of EPG functions 665
A3-5.6.3 Operation of non-volatile memory functions 665
A3-5.6.4 Operation of subtitle presentation control 665
A3-5.6.5 Operation of interaction channel function 666
A3-5.6.6 Guideline on extended functions for broadcasting used to control ISP connecting information 678
A3-5.6.7 Channel status after execution of interaction channel function 680
A3-5.6.8 Operation of operational control functions 682
A3-5.6.9 Operation of receiver build-in sound control 682
A3-5.6.10 Operation of timer function 682
A3-5.6.11 Operation of external character function 682
A3-5.6.12 Operation of Directory Management Functions, File Management Functions, and Data Carousel Storage Functions 683
A3-5.6.13 Operation of module lock among components 683
A3-5.7 Operation of link element 683
A3-5.7.1 Operation of link element for external style sheet specification 683
A3-5.8 Other Restrictions 683
A3-5.8.1 Size of monomedia data 686
A3-5.9 Operation of Cookies 686
A3-5.9.1 Version of Cookies 686
A3-5.9.2 Function enabling Cookies 687
A3-5.9.3 Restrictions on implementation of Cookies 687
A3-5.9.4 Reference of Cookie from DOM 687
A3-5.9.5 Reference of Cookie in broadcast/communication/stored content 687

Chapter APPENDIX 3-6 Guidelines on Contents Transmission and Namespace 688
A3-6.1 Scope Mapping to Transmission System 688
A3-6.1.1 Scope mapping for contents received in real time 688
A3-6.1.2 Scope mapping for contents transmitted via interactive communication networks associated with broadcasting services 689
A3-6.1.3 Scope mapping for stored contents 689
A3-6.2 Guideline on reference across media types 690
A3-6.2.1 Restriction on document transmission among media types 690
A3-6.2.2 Restriction on reference of monomedia and others across media types 690
A3-6.3 Namespace 692
A3-6.3.1 Namespace for contents received in real time 692
A3-6.3.2 Namespace for contents transmitted via interactive communication networks associated with broadcasting services 692
A3-6.3.3 Namespace for stored contents (sub-profile 2) 694

A3-6.4 Guidelines on transmission of contents 694
A3-6.4.1 Syntax of module 694
A3-6.4.2 Resource list in multi-part formatted module 694
A3-6.4.3 Resource list in multi-part formatted module assuming stored data services (sub-profile 2) 695
A3-6.4.4 Operation of event messages 695
A3-6.5 Operational guidelines on communication protocols 695
A3-6.5.1 Operational guidelines on HTTP 695
A3-6.5.2 Operational Guidelines on mapping between API and protocols 697
A3-6.5.3 Operational guidelines on secure communication 698

Annex A Default Style Sheet 699
A.1 Default Style Sheet Common to Index-based Model and Component-based Model 699
A.2 Default Style Sheet Common Specific to Index-based Model 699
A.3 Default Style Sheet Common Specific to Component-based Model 700

Annex B Operated DTD 701
B.1 BML Driver DTD 701
B.2 BML Document Model Module 709
B.3 Description of DTD Declaration 713

Chapter APPENDIX 3-1 Purpose

The Appendix 3 contains the operational guidelines for implementing an integrated service that provides broadcasting services without stored services and services via the Internet for a fixed receiving system, and an integrated service that combines broadcasting services accompanying stored data services with services via the Internet for a fixed receiving system, by using the XML-based multimedia coding scheme specification responsible for the data broadcasting scheme, part of the digital data broadcasting scheme specified as the standard in Japan.

Chapter APPENDIX 3-2 Scope

The guidelines contained in this appendix are applied to an integrated service that provides broadcasting services without stored services and services via the Internet for a fixed receiving system, and an integrated service that combines broadcasting services accompanying stored data services with services via the Internet for a fixed receiving system, by using the XML-based multimedia coding scheme specification responsible for the data broadcasting scheme, part of the digital data broadcasting scheme specified as the standard in Japan.

Chapter APPENDIX 3-3 Legend

The following terms and symbols are used to describe a operational guideline in APPENDIX 3:

Sub-profile 1: This term indicates that a concerned operational guideline in this appendix is a profile used to implement an integrated service that provides broadcasting services without stored services and services via the Internet for a fixed receiving system.

Sub-profile 2: This term indicates that a concerned operational guideline in this appendix is a profile used to implement an integrated service that combines broadcasting services accompanying stored data services with services via the Internet for a fixed receiving system.

O: This symbol marks an item required to implement a service based on a specified sub-profile. It is desirable that an implementation designed for a concerned service should present this item normally when the corresponding signal is contained in a content. However, an item marked by "*" or "Note" may not be available due to restrictions on a media type, a specific implementation, or others.

-: This symbol marks an item not required to implement a service based on a specified sub-profile. Whether or not the item is presented depends on an implementation designed for a concerned service, when the corresponding signal contained in a content to be presented. It is desirable that no signal corresponding to a "-" marked item is contained in a content. However, this does not prevent an operational standard for a specific media type from using a "-" marked item.

Chapter APPENDIX 3-4 Guidelines for Coding of BML Documents

A3-4.1 Character Codes

Character codes are used, as defined in Section 5.1, Volume 2.

A3-4.1.1 Character coding scheme used for BML documents

Any BML document to be transmitted contains character codes in EUC-JP or JIS.

A3-4.1.1.1 EUC-JP

The EUC-JP character codes are used based on Section 4.1.1, Appendix 2.

A3-4.1.1.2 Shift-JIS

Table 4-1 shows the operational guideline on the Shift-JIS character codes.

Table 4-1 Shift-JIS Code Set

Code set	Character set	Operating range
1-byte character Byte range 21~7F, A1~DF	JIS X 0201-1976 (JIS Roman Alphabets) Half-width (Hankaku) Katakana	O However, the byte range A1~DF (Half-width Katakana) is not operated.
2-byte character 1st byte range 81~9F, E0~EF 2nd byte range 40~7E, 80~FC	JIS X 0208-1997 Note that a character with the same kuten code in the Kanji character set is allocated in the corresponding location in the sections 90 ku to 94 ku (free area).	O
Control codes	Blank character (20)	O
	Delete character (7F)	-
	Line feed (0D0A)	O
	Tab (09)	O

A3-4.1.1.3 Conversions of character codes

This section describes operational guidelines for converting a character coding for BML documents into the other.

- Converting EUC-JP into Shift-JIS

  An EUC-JP character code is converted into a Shift-JIS code as described below.

  (1) When an EUC-JP character code belongs to the code set 0:

  The EUC-JP character code is converted into a 1-byte character in Shift-JIS.

  Provided that an EUC-JP character code to be converted is E1 and the resulting Shift-JIS character code is S1, this conversion is expressed as below.

  $$S1 = E1$$

(2) When an EUC-JP character code belongs to the code set 1:

The EUC-JP character code is converted into a 2-byte character in Shift-JIS.

Provided that the first byte of an EUC-JP character code to be converted is E1, the second byte of the EUC-JP code is E2, the first byte of the resulting Shift-JIS character code is S1, the second byte of the Shift-JIS code is S2, and div is a integer division (discarding all digits to the right of the decimal point) this conversion is expressed as below.

a) S1 is obtained as described below.

a-1) When 0xA1 <= E1 <= 0xDE, S1 = (E1 - 1) div 2 + 0x31.

a-2) When 0xDF <= E1 <= 0xFE, S1 = (E1 - 1) div 2 + 0x71.

b) S2 is obtained as described below.

b-1) Provided that E2 is an odd number:

b-1.1) When 0xA1 <= E2 <= 0xDF, S2 = E2 - 0x61.

b-1.2) When 0xE0 <= E2 <= 0xFE, S2 = E2 - 0x60.

b-2) Provided that E2 is an even number:

S2 = E2 - 0x2

(3) When an EUC-JP character code is a control code (a blank, a line break, or a tab):

The EUC-JP character code is converted into a corresponding character code in Shift-JIS.

Provided that an EUC-JP control code to be converted is Ec and the resulting Shift-JIS control code is Sc, this conversion is expressed as below.

Sc = Ec

- Converting Shift-JIS into EUC-JP

(1) When a Shift-JIS character code is a 1-byte character:

The Shift-JIS character code is converted into an EUC-JP character code that belongs to the code set 0.

Provided that a Shift-JIS character code to be converted is S1 and the resulting EUC-JP character code is E1, this conversion is expressed as below.

E1 = S1

(2) When a Shift-JIS character code is a 2-byte character:

The Shift-JIS character code is converted into an EUC-JP character code that belongs to the code set 1.

Provided that the first byte of a Shift-JIS character code to be converted is S1, the second byte of the Shift-JIS code is S2, the first byte of the resulting EUC-JP character code is E1, and the second byte of the EUC-JP code is E2, this conversion is expressed as below.

a) E1 is obtained as described below.

a-1)Provided that 0x81 <= S1 <= 0x9F,

a-1.1) When 0x40 <= S2 <= 0x9E, E1 = (S1 - 0x30) × 2 - 0x1.

a-1.2) When 0x9F <= S2 <= 0xFC, E1 = (S1 - 0x30) × 2.

a-2)Provided that 0xE0 <= S1 <= 0xEF,

a-2.1) When 0x40 <= S2 <= 0x9E, E1 = (S1 - 0x70) × 2 - 0x1.

a-2.2) When 0x9F <= S2 <= 0xFC, E1 = (S1 - 0x70) × 2.

b) E2 is obtained as described below.

b-1)When 0x40 <= S2 <= 0x7E, E2 = S2 + 0x61.

b-2)When 0x80 <= S2 <= 0x9E, E2 = S2 + 0x60.

b-3)When 0x9F <= S2 <= 0xFC, E2 = S2 + 0x2.

(3) When a Shift-JIS character code is a control code (a blank, a line break, or a tab):

The Shift-JIS character code is converted into a corresponding character code in EUC-JP.

Provided that a Shift-JIS control code to be converted is Sc and the resulting EUC-JP control code is Ec, this conversion is expressed as below.

Ec = Sc

A3-4.1.2 External characters

A DRCS can be used to present external characters. This section provides the operational guideline for coding and using external characters. The operational guideline for managing external character is defined in Section A2-5.2, APPENDIX 2. The operational guideline on the loadDRCS( ) function is described in Section A2-5.6.10, APPENDIX 2.

The code area available to the external characters is the sections 87 through 88. A pattern data invoked by loadDRCS() is referenced by a BML document. To configure a pattern data in an area for managing external characters in a content and to make external characters referenced, the following conditions [1] must be satisfied:

- The external characters referenced by a BML document must have been set before the BML document is presented. The loadDRCS( ) function must be executed at the time when global codes of ECMAScript are executed. It is not executed in event handlers of ECMAScript. The loadDRCS function may be executed in global codes not more than twice.
- The external characters referenced by a BML document must be in a string presented by the referencing BML document, a message transmitted via an interaction channel, and a string representing privateData transmitted in an event message.
- Two or more sets of pattern data may be set for a single character code by executing loadDRCS(), provided that each set of pattern data is a different combination of a font family and a font size. When a character code is associated with two or more sets of pattern data and each set is the same combination of a font family and a font size, how the redundant sets are treated depends on an implementation.

A structure of DRCS data that is read by loadDRCS( ) is an extension to a data structure defined in ARIB STD-B24 Volume1 Annex D, to which the following restrictions are also applied:

- The range of the numberOfCode value is from 1 to 188.
- The available values to characterCode correspond to the characters from the point 1 of the section 87 to the point 94 of the section 88 in the character code table. A characterCode value is a section and point code. For example, to transmit the character in the point 1 of section 1, 0x2121 is transmitted.
- The following values can be specified for the fontID property. The fontID property must not set to 0.

---

1 The purposes of these conditions are to prevent a redundant pattern data associated with the same external character code from causing any unexpected miss-presentation of the character and to avoid any disordered presentation of contents due to a receiver that cannot ensure that a redundant pattern data is presented at an intended timing.

Value of fontID	Typeface
1	Round gothic
2	Square Gothic
3	Bold Round Gothic

- The mode must be 0001. That is, the pattern data is multi-scaled and is not compressed. It is not compressed with geometric coding.
- The depth must be 2. That is, the pattern data has 4-step scaled values.
- The same restrictions as the pixel value for font-size property are applied to width and height.

A3-4.1.3 Operation of character reference

A3-4.1.3.1 EUC-JP

- Character reference by "&#" is not operated.
- When external characters are set, corresponding EUP-JP codes must be designated directly.
- Character reference using "\" of CSS and the SingleEacapeCharacters such as "\a" are not operated.
- "% escape" in URI is not operated.
- HexEscapeSequence, OctalEscapeSequence, and UnicodeEscapeSequence in ECMAScript are not operated.
  And, among SingleEscapeCharacters, "¥b", "¥t", and "¥f" are not operated but "¥"", "¥'", "¥r", "¥n", and "¥¥" are operated. However, the "¥" character must be followed by "0x5c". The "¥r" and "¥n" characters must be used as the combination, "¥r¥n".

A3-4.1.3.2 Shift-JIS

- Character reference by "&#" is not operated.
- When external characters are set, corresponding Shift-JIS codes must be designated directly.
- Character reference using "\" of CSS and the SingleEacapeCharacters such as "\a" are not operated.
- HexEscapeSequence, OctalEscapeSequence, and UnicodeEscapeSequence in ECMAScript are not operated.
  And, among SingleEscapeCharacters, "¥b", "¥t", and "¥f" are not operated but "¥"", "¥'", "¥r", "¥n", and "¥¥" are operated. However, the "¥" character must be followed by "0x5c". The "¥r" and "¥n" characters must be used as the combination, "¥r¥n".

A3-4.1.4 Operation of character encoding available to contents

- Which character encoding specification is internally used by an ECMAScript processor depends on an implementation. Which character encoding specification is applied to functions for comparing lengths of strings and sorting strings also depends on an implementation.
- Any character encoding specification used in a Greg or Ureg pseudo object is converted into a character encoding used by a BML document that is presented at the time when the pseudo object is read out.
- The writePersisitentArray() function and readPersistentArray()functions writes and reads information written in EUC-JP to and from NVRAM, respectively.

- Any binary table contains information written in EUC-JP.
- The privateData attribute of EventMessageFired contains strings in EUC-JP.
- The writeBookmarkArray() function and readBookmarkArray() function writes and reads information written in EUC-JP to and from a bookmark area, respectively.
- Any communication function assuming BASIC procedures sends or receives information written in EUC-UP.
- To transmit information via IP:
  - The sendTextMail() function sends or receives information written in ISO 2022-JP.
  - Any information sent to or from a server via the form element is written the same character encoding specification as used in a document that contains the form element.
  - The transmitTextDataOverIP() function sends or receives messages written in a character encoding specification specified in the function.

A3-4.2 Media Types and Operational Range of Monomedia

Table 4-2 shows the operational guideline on the media types required for basic services and the monomedia coding schemes designated by object element.

Table 4-2 Operational guideline on media types and monomedia coding schemes

Schema	Media type	Operation	Operation (referenced by object element)	Operation (referenced by img element)	Remarks
arib-dc:	multipart/mixed	O	-	-	
	text/xml;charset="euc-jp"	-	-	-	
	text/xml;charset="UTF-8"	-	-	-	
	text/xml;charset="UTF-16"	-	-	-	
	text/xml;charset="Shift-JIS"	-	-	-	
	text/css	O	-	-	(Note 1)
	text/xsl;charset="euc-jp"	-	-	-	
	text/xsl;charset=”Shift-JIS”	-	-	-	
	text/xsl;charset="UTF-8"	-	-	-	
	text/xsl;charset="UTF-16"	-	-	-	
	text/html;charset="euc-jp"	O	-	-	
	text/html;charset="UTF-8"	-	-	-	
	text/html;charset="UTF-16"	-	-	-	
	text/html;charset=”Shift_JIS”	O	-	-	
	text/X-arib-bml; charset="euc-jp"	O	-	-	(Note 2)
	text/X-arib-bml; charset="UTF-8"	-	-	-	
	text/X-arib-bml; charset="UTF-16"	-	-	-	
	text/X-arib-bml; charset="Shift-JIS"	O	-	-	(Note 2)
	text/plain; charset="euc-jp"	-	-	-	
	text/plain; charset="UTF-8"	-	-	-	
	text/plain; charset="UTF-16"	-	-	-	
	text/plain; charset="Shift-JIS"	-	-	-	
	text/X-arib-jis8text	-	-	-	
	text/X-arib-ecmascript; charset="euc-jp"	O	-	-	(Note 2)
	text/X-arib-ecmascript; charset= "UTF-8"	-	-	-	

Schema	Media type	Operation	Operation (referenced by object element)	Operation (referenced by img element)	Remarks
	text/X-arib-ecmascript; charset= "UTF-16"	-	-	-	
	text/X-arib-ecmascript; charset="Shift-JIS"	O	-	-	(Note 2)
	image/jpeg	O	O	O	
	image/png	O	O	O	(Note 3)
	image/X-arib-png	O	O	O	(Note 3)
	image/X-arib-mng	O	O	-	
	image/X-arib-mpeg2-I	-	-	-	
	image/X-arib-mpeg4-I-simple	-	-	-	
	image/X-arib-mpeg4-I-core	-	-	-	
	audio/X-arib-mpeg2-aac	O	O	-	
	audio/X-arib-mpeg2-bc	-	-	-	(Note 4)
	audio/X-arib-aiff	O	O	-	
	audio/X-arib-additional	-	-	-	
	video/X-arib-mpeg1	-	-	-	
	application/xhtml+xml; charset="euc-jp"	-	-	-	
	application/xhtml+xml; charset="Shift-JIS"	-	-	-	
	application/xhtml+xml; charset="UTF-8"	-	-	-	
	application/xhtml+xml; charset="UTF-16"	-	-	-	
	application/X-arib-stream-text; charset="euc-jp"	-	-	-	
	application/X-arib-stream-text; charset="UTF-8"	-	-	-	
	application/X-arib-stream-text; charset="UTF-16"	-	-	-	
	application/X-arib-stream-text; charset="Shift-JIS"	-	-	-	
	application/X-arib-stream-jis8text	-	-	-	
	application/X-arib-stream-png	-	-	-	
	application/X-arib-stream-jpeg	-	-	-	
	application/X-arib-stream-mpeg2-I	-	-	-	
	application/X-arib-stream-mpeg4-I-simple	-	-	-	
	application/X-arib-stream-mpeg4-I-core	-	-	-	
	application/X-arib-mpeg2-tts	O	O	-	
	application/X-arib-bmlclut	O	-	-	
	application/X-arib-btable	O	-	-	
	application/X-arib-drcs	O	-	-	
	application/X-arib-PDI	-	-	-	
	application/X-arib-resourceList	O	-	-	
	application/X-arib-storedResourceList	O	-	-	(Note 5)
	application/X-arib-contentPlayControl	-	-	-	
arib:	audio/X-arib-mpeg2-aac	O	O	-	
	audio/X-arib-mpeg2-bc	-	-	-	(Note 4)

Schema	Media type	Operation	Operation (referenced by object element)	Operation (referenced by img element)	Remarks
	audio/X-arib-mpeg4	-	-	-	
	video/X-arib-mpeg1	O	O	-	
	video/X-arib-mpeg2	O	O	-	
	video/X-arib-mpeg4-simple	-	-	-	
	video/X-arib-mpeg4-core	-	-	-	
arib-ic:	image/X-arib-mpeg2-I	-	-	-	
	application/X-arib-stream-mpeg2-I	-	-	-	
	application/X-arib-stream-mpeg4-I-simple	-	-	-	
	application/X-arib-stream-mpeg4-I-core	-	-	-	
romsound:	audio/X-arib-romsound	O	-	-	

Note 1: The same character encoding specification as a concerned BML document is used for a CSS and the charset attribute for the CSS is omitted.
Note 2: The charset attribute must not be omitted.
Note 3: To present a image according to a PLTE chunk specified in PNG, the image/png media type is used.
Note 4: It is also assumed that audio/X-arib-mpeg2-bc instead of audio/X-arib-mpeg2-aac is operated, depending on the necessity of each transmission media.
Note 5: This media type is applicable only to Sub-profile 2.

A3-4.2.1 Bitmap

A3-4.2.1.1 PNG

The conventions on coding and transmitting PNG defined in Section 8.2.3, Volume 2 must be obeyed. Note that in order to use a component-based model (refer to Section A3-5.1.1), a PLTE chunk is required and a tRNS chunk is employed.

The length of Chunk Data in a tRNS chunk must be not less than nine bytes. The ninth byte must contain 00 and the other bytes contain a value other than 00.

A3-4.2.1.2 MNG

The conventions on coding and transmitting MNG defined in Section 8.2.4, Volume 2 must be obeyed. Note that in order to use a component-based model (refer to Section A3-5.1.1), a PLTE chunk is required and a tRNS chunk is employed.

The length of Chunk Data in a tRNS chunk must be not less than nine bytes. The ninth byte must contain 00 and the other bytes contain a value other than 00.

A3-4.2.2 Video/Audio

A3-4.2.2.1 MPEG-2 Video stream

The convention on transmitting MPEG-2 Video data defined in Section 8.2.4, Volume 2 must be obeyed. When video data is output at HD resolution, SD, SIF, or QSIF is presented as required.

The available sizes to present scaled video in a video plane is listed in Table 5-2, Section A3-5.1.5.1.

A3-4.2.2.2 Time-stamped TS file

The convention on transmitting a time-stamped TS file defined in Section 8.1.2.2, Volume 2 must be obeyed. Note that only transmitting at a fixed bit rate is allowed and the fixed bit rate must be contained in TSbitrate of the TimestampedTS descriptor.

To transmit a time-stamped TS file in a module in the multipart format, TimestampedTSdescriptor() must be contained in additionalFileInfo() of storedFileInfo() indicating a time-stamped TS resource by using X-arib-storedResourceList() in the multipart module. TimestampedTSdescriptor() is operated based on Table 4-2.

Table 4-3 Operation of TimestampedTSdescriptor()

Data	Operation
TSbitrate	Required
PCR_PID	Required
default_video_flag	0 (Required when a video stream exists) 1 (When no video stream exist)
default_audio_flag	0 (Required when an audio stream exists) 1 (When no audio stream exist)
timestampedTS_PBctrl_info_flag	Optional (An implementation is not required to interpret a time-stamped TS playback control information.)
videoPID	Required when a video stream exists
audioPID	Required when an audio stream exists

A time-stamped TS file contains a single program, video streams, or vide streams as listed in Table 4-4.

Table 4-4 Operation of time-stamped TS file

Property	Operation
the number of programs	Limited to one
the number of video streams	One or zero
the number of audio streams	Zero through two

in order to play a time-stamped TS file, the object element is operated as described in Table 4-5. When the data attribute value of the object element that specifies a video stream or an audio stream is –1, a stream specified with videoPID or audioPID of TimestampedTSdescriptor() is assumed.

Table 4-5 Operation of object for time-stamped TS playback

Property	Operation
value of the value attribute of the object specifying a time-stamped TS	Limited to “-1”
value of the data attribute of the object specifying a video stream	Limited to “-1”
value of the data attribute of the object specifying an audio stream	“-1” or /<component_tag>[;<channel_id>]

A3-4.2.3 8-bit character code

No 8-bit character code must be operated.

A3-4.3 Operation of XML Namespace

The XML Namespace must operate based on the conventions defined in Chapter 10, Volume 2.

A3-4.4 Operation of PI

A3-4.4.1 Switching presentation model

Presentation models are switched by using PI. To operate a component-based model, PI is used to control how a focused frame is presented. The definition format for bml-model processing instruction is shown below.

- Definition

BmlModelPI	::=	'<?bml-model' (S PseudoAtt)* S? '?>'
PseudoAtt	::=	Name S? '=' S? PseudoAttValue
PseudoAttValue	::=	("" ([^"<&] | CharRef)* "" | '"' ([^'<&] | CharRef)* "') - (Char* '?>' Char*)

- PseudoAtt, PseudoAttValue

  presentation-model(index-base|component-base) "component-base"

  focus-visibility (UAcontrol | invisible) "UAcontrol"

  The focus-visibility described above is applicable only to a component-based model. When an index-based model is presented, no focus is presented regardless of whether or not a browser supports the focus presentation.

- Semantics

Table 4-6 PI to switch presentation models

PseudoAtt	PseudoAttValue	Semantics
presentation-model	index-base	Presentation is based on an index-based model.
	component-base	Presentation is based on a component-based model.
focus-visibilty	UAcontrol	Presentation of a focus is controlled by a browser.
	invisible	A browser does not present a focus.

Only prolog of a BML document must contain a bml- model processing instruction.

Example1: To use an index-based model for presentation,

<?bml-model presentation-model="index-base"?>

Example2: To use a component-based model for presentation,

<?bml-model presentation-model="component-base"?>

A3-4.5 Operation of BML elements

A3-4.5.1 Operational convention on BML elements for XHTML modules

Any BML element must be operated to ensure consistency with XHTML 1.0. Any BML element is operated by selecting a module based on the W3C Recommendation, Modularization of XHTML except that the behaviours of the the onclick, ondblclick, onmousedown, onmouseup, onmouseover,

onmousemove, onmouseout events, each of those is defined with Events as described Section 5.1 Attribute Collection, the Modularization of XHTML recommendation, depend on an implementation.

Table 4-7 Operation of XHTML modules

Module		Operation	Available elements to module	Applicable attributes to element
Core	Structure (required)	O	body	%Common;
			head	%I18N;, profile
			html	%I18N;, version, xmlns
			title	%I18N;
	Text (required)	O	abbr	%Common;
			acronym	%Common;
			address	%Common;
			blockquote	%Common;, cite
			br	%Core;
			cite	%Common;
			code	%Common;
			dfn	%Common;
			div	%Common;
			em	%Common;
			h1 ~ h6	%Common;
			kbd	%Common;
			p	%Common;
			pre	%Common;, xml:space="preserve"
			q	%Common;, cite
			samp	%Common;
			span	%Common;
			strong	%Common;
			var	%Common;
	Hypertext (required)	O	a	%Common;, accesskey, charset, href, hreflang, rel, rev, tabindex, [1] type
	List (required)	O	dl	%Common;
			dt	%Common;

[1] Interpretation of tabindex depends on a receiver.

Module		Operation	Available elements to module	Applicable attributes to element
			dd	%Common;
			ol	%Common;
			ul	%Common;
			li	%Common;
Applet (deprecated)		-	applet	%Core;, alt*, archive, code, codebase, height*, object, width*
			param	id, name*, type, value, valuetype
Text Extension	Presentation [2]	O	b	%Common;
			big	%Common;
			hr	%Common;
			i	%Common;
			small	%Common;
			sub	%Common;
			sup	%Common;
			tt	%Common;
	Edit	-	del	%Common;, cite, datetime
			ins	%Common;, cite, datetime
	Bi-directional	-	bdo	%Core;, dir*
Form	Basic Forms	O	form	%Common;, action*, method, enctype

2 How it is presented depends on a receiver. Presenting it is not required.

Module		Operation	Available elements to module	Applicable attributes to element
			input	%Common; accesskey, checked, maxlength, name, size, src, tabindex, type, value
			label	%Common;, accesskey, for
			select	%Common;, multiple, name, size, tabindex
			option	%Common;, selected, value
			textarea	%Common;, accesskey, cols*, name, rows*, tabindex
	Forms	-	form	%Common;, accept, accept-charset, action*, method, enctype
			input	%Common;, accept, accesskey, alt,
				checked, disabled, maxlength,
				name, readonly, size,
				src, tabindex,
				type, value

Module		Operation	Available elements to module	Applicable attributes to element
Table	Basic Tables	O	select	%Common;, disabled, multiple, name, size, tabindex
			option	%Common;, disabled, label, selected, value
			textarea	%Common;, accesskey, cols*, disabled, name, readonly, rows*, tabindex
			button	%Common;, accesskey, disabled, name, tabindex, type, value
			fieldset	%Common;
			label	%Common;, accesskey, for
			legend	%Common;, accesskey
			optgroup	%Common;, disabled, label*
			caption	%Common;
			table	%Common;, summary, width
			td	%Common;, abbr, align, axis, colspan, headers, rowspan, scope, valign

Module		Operation	Available elements to module	Applicable attributes to element
			th	%Common;, abbr, align, axis, colspan, headers, rowspan, scope, valign
			tr	%Common;, align, valign
	Tables	-	caption	%Common;
			table	%Common;, border, cellpadding, cellspacing, datapagesize, frame, rules, summary, width
			td	%Common;, abbr, align, axis, char, charoff, colspan, headers, rowspan, scope, valign
			th	%Common;, abbr, align, axis, char, charoff, colspan, rowspan, scope, valign
			tr	%Common;, align, char, charoff, valign

Module		Operation	Available elements to module	Applicable attributes to element
			col	%Common;, align, char, cahroff, span, valign, width
			colgroup	%Common;, align, char, charoff, span, valign, width
			tbody	%Common;, align, char, charoff, valign
			thead	%Common;, align, char, charoff, valign
			tfoot	%Common;, align, char, charoff, valign
Image		O	img	%Common;, alt*, height, longdesc, src*, width
CS Image Map		-	*a&*	coords, shape
			area	%Common;, accesskey, alt*, coords, href, nohref, shape, tabindex
			img&	usemap
			Input&	usemap

Module		Operation	Available elements to module	Applicable attributes to element
			map	%I18N;, %Events;, class, id*, title
			object&	usemap
SS Image Map		-	*img&*	ismap
			input&	ismap
Object		O	object	%Common;, archive, classid, codebase, codetype, data, declare, height, name, standby, tabindex, type, width
			param	id, name*, type, value, valuetype
Frames		-	frameset	%Core;, cols, rows
			frame	%Core;, frameborder, longdesc, marginheight, marginwidth, noresize, scrolling, src
			noframes	%Common;
Target		-	*a&*	target
			area&	target
			base&	target
			link&	target
			form&	target

Module		Operation	Available elements to module	Applicable attributes to element
Iframe		-	iframe	%Core;, frameborder, height, longdesc, marginheight, marginwidth, scrolling, src, width
Intrinsic Events		O [3]	*a&*	onblur, onfocus
			area&	onblur, onfocus
			frameset&	onload, onunload
			form&	onreset, onsubmit
			body&	onload, onunload
			label&	onblur, onfocus
			input&	onblur, onchange, onfocus, onselect
			select&	onblur, onchange, onfocus
			textarea&	onblur, onchange, onfocus, onselect
			button&	onblur, onfocus
Metainformation		O	meta	%I18N;, content*, http-equiv, name, scheme
Scripting		O	noscript	%Common;

3 The behaviors of an event controlled by a mouse or other input devices whose movement directions are not predefined depend on an implemenation.

Module		Operation	Available elements to module	Applicable attributes to element
			script	charset, defer, src, type*, xml:space="preserve"
Stylesheet		O	style	%I18N;, media, title, type*, xml:space="preserve"
Style Attribute		O		%Style (activate)
Link		O	link	%Common;, charset, href, hreflang, media, rel, rev, type
Base		O	base	href*
Name Identification (deprecated)		-	*a&*	name
			applet&	name
			form&	name
			frame&	name
			iframe&	name
			img&	name
			map&	name
Legacy (deprecated)		-	basefont	color, face, id, size
			center	%Common;
			dir	%Common;, compact
			font	%Core;, %I18N;, color, face, size
			isindex	%Core;, %I18N;, prompt
			menu	%Common;, compact
			s	%Common;
			strike	%Common;
			u	%Common;

Module		Operation	Available elements to module	Applicable attributes to element
			body&	alink, background, bgcolor, link, text, vlink
			br&	clear
			caption&	align
			div&	align
			dl&	compact, type
			h1-h6&	align
			hr&	align, noshade, size, width
			img&	align, border, hspace, vspace
			input&	align
			legend&	align
			li&	type, value
			ol&	compact, start, type
			p&	align
			pre&	width
			script&	language
			table&	align, bgcolor
			tr&	bgcolor
			th&	bgcolor, height, nowrap, width
			td&	bgcolor, height, nowrap, width
			ul&	compact, type
BML	BML	-	*bml:bml*	%I18N;, version, xmlns
			bml:bevent	id

Module		Operation	Available elements to module	Applicable attributes to element
			bml:beitem	id*, type*, onoccur*, es_ref, message_group_id, message_id, message_version, module_ref, language_tag, register_id, service_id, event_id, time_mode, time_value, object_id, subscribe
			body&	bml:invisible
			div&	bml:onfocus, bml:onblur, bml:accesskey
			p&	bml:onfocus, bml:onblur, bml:accesskey
			span&	bml:onfocus, bml:onblur, bml:accesskey
			a&	bml:effect
			bdo&	bml:orientation
			object&	bml:remain, bml:streamstatus, bml:streamposition, bml:streamlooping, bml:streamspeednumerator, bml:streamspeeddenominator, bml:streamlevel, bml:onfocus, bml:onblur, bml:accesskey
BML	Basic BML	O	*bml:bevent*	id

Module		Operation	Available elements to module	Applicable attributes to element
			bml:beitem	id*, type*, onoccur*, es_ref, message_group_id, message_id, message_version, module_ref, language_tag, time_mode, time_value, object_id, subscribe,
			body&	bml:invisible,
			div&	bml:onfocus, bml:onblur, bml:accesskey
			p&	bml:onfocus, bml:onblur, bml:accesskey
			span&	bml:onfocus, bml:onblur, bml:accesskey
			object&	bml:remain, bml:streamstatus, bml:streamposition, bml:streamlooping, bml:onfocus, bml:onblur, bml:accesskey

A3-4.5.2 Entity

Any entity must be operated based on the conventions defined in Section A2-4.3.1, APPENDIX 2, Volume 2.

A3-4.5.3 Attribute

For a module to which operational restrictions are applied, the attributes applicable to the elements of the module must be operated based on Table 4-8. Modules not listed in Table 4-8 are defined in the corresponding standard.

Table 4-8 Operational Restrictions on Attributes

Module Classification	Element Entity	Attribute	Operational Restrictions
Basic Entities	coreattrs	id	String (up to 128 bytes)
		class	
		style	

Module Classification	Element Entity	Attribute	Operational Restrictions
		title	Whether or not it is rendered depends on an implementation.
	i18n	xml:lang	Must be set to “ja”
		lang	Must be set to “ja”
		dir	Must be set to “ltr”
		orientation	Must be set to “horiz”
	events	onclick	
		ondblclick	When it is applied, the behaviour depends on an implementation.
		onmousedown	When it is applied, the behaviour depends on an implementation.
		onmouseup	When it is applied, the behaviour depends on an implementation.
		onmouseover	When it is applied, the behaviour depends on an implementation.
		onmousemove	When it is applied, the behaviour depends on an implementation.
		onmouseout	When it is applied, the behaviour depends on an implementation.
		onkeypress	When it is applied, the behaviour depends on an implementation.
		onkeydown	
		onkeyup	
Forms			
Basic Forms	form	%coreattrs;	
		%i18n;	
		%events;	
		action	
		method	
		enctype	The method attribute must be “post”, and the input element must be “application/x-www-form-urlencoded” or “multipart/form-data”.
	input	%coreattrs;	
		%i18n;	
		%events;	
		type	Must be set to one the following: “text”, “password”, “checkbox”, “radio”, “submit”, “reset”, “hidden”
		name	
		value	
		checked	
		size	This element is available only when the type attribute is “text” or “password”. Must be an integer from 1 to 40.
		maxlength	Must be set to an integer from 1 to 40
		src	
		tabindex	
		accesskey	

Module Classification	Element Entity	Attribute	Operational Restrictions
	label	%coreattrs;	
		%i18n;	
		%events;	
		for	
		accesskey	When it is applied, the behaviour depends on an implementation.
	select	%coreattrs;	
		%i18n;	
		%events;	
		name	
		size	Must be set to an integer from 1 to 16
		multiple	
		tabindex	
	option	%coreattrs;	
		%i18n;	
		%events;	
		selected	
		value	
	textarea	%coreattrs;	
		%i18n;	
		%events;	
		name	
		rows	Must be set to an integer from 1 to 16
		cols	Must be set to an integer from 1 to 32
		tabindex	
		accesskey	
Table			
Basic Tables	caption	%coreattrs;	
		%i18n;	
		%events;	
	table	%coreattrs;	
		%i18n;	
		%events;	
		summary	
		width	The applicable range to pixel specification is 0 through 960. The applicable range to percentage specification is 0 % through 100%.
	tr	%coreattrs;	
		%i18n;	
		%events;	
		align	Must be set to on one of he following: “left”, “center”, “right”
		valign	Must be set to on one of he following: “top”, “middle”, “bottom”
	th	%coreattrs;	
		%i18n;	

Module Classification	Element Entity	Attribute	Operational Restrictions
		%events;	
		abbr	
		axis	
		headers	
		scope	
		rowspan	Must be an integer from 1 to 32
		colspan	Must be set to an integer from 1 to 32
		align	Must be set to one of the following: “left”, “center”, “right”
		valign	Must be set to one of the following: “top”, “middle”, “bottom”
	td	%coreattrs;	
		%i18n;	
		%events;	
		abbr	
		axis	
		headers	
		scope	
		rowspan	Must be set to an integer from 1 to 32
		colspan	Must be set to an integer from 1 to 32
		align	Must be set to one of the following: “left”, “center”, “right”
		valign	Must be set to one of the following: “top”, “middle”, “bottom”
Image			
	img	%coreattrs;	
		%i18n;	
		%events;	
		src	
		alt	
		longdesc	
		height	The applicable range to pixel specification is 0 through 540. No percentage specification is operated.
		width	The applicable range to pixel specification is 0 through 960. No percentage specification is operated.
Object			
	object	%coreattrs;	
		%i18n;	
		%events;	
		declare	
		classid	
		codebase	
		data	
		type	
		codetype	

Module Classification	Element Entity	Attribute	Operational Restrictions
		archive	
		standby	
		height	The applicable range to pixel specification is 0 through 540. No percentage specification is operated.
		width	The applicable range to pixel specification is 0 through 960. No percentage specification is operated.
		tabindex	
	param	id	
		name	Must be one of the following: “program_number”,“stream_ref”
		value	
		valuetype	Must be set to one of the following: “data”,“object”
		type	
Intrinsic Events			
	a&	onblur	
		onfocus	
	area&	onblur	Invalid because no CS Image Map module is used
		onfocus	Invalid because no CS Image Map module is used
	frameset&	onload	Invalid because no Frames module is used
		onunload	Invalid because no Frames module is used
	form&	onreset	
		onsubmit	
	body&	onload	
		onunload	
	label&	onblur	When it is applied, the behaviour depends on an implementation.
		onfocus	When it is applied, the behaviour depends on an implementation.
	input&	onblur	
		onchange	
		onfocus	
		onselect	When it is applied, the behaviour depends on an implementation.
	select&	onblur	
		onchange	
		onfocus	
	textarea&	onblur	
		onchange	
		onfocus	
		onselect	When it is applied, the behaviour depends on an implementation.
	button&	onblur	Invalid because no Forms module is used.
		onfocus	Invalid because no Forms module is used.
Scripting			
	noscript	%coreattr;	

Module Classification	Element Entity	Attribute	Operational Restrictions
		%i18n;	
		%events;	
	script	charset	Must be set to “EUC-JP” or “Shift-JIS”
		type	Must be set to 'text/X-arib-ecmascript; charset=”euc-jp”' or 'text/X-arib-ecmascript; charset=”Shift_JIS”
		src	
		defer	
		event	
		xml:space	
Stylesheet			
	style	%i18n;	
		type	Must be set to "text/css"
		media	
		title	
		xml:space	
Link			
	link	%coreattrs;	
		%i18n;	
		%events;	
		charset	
		href	
		hreflang	
		type	
		rel	Must be set to “Stylesheet”
		rev	
		media	
Base			
	base	href	
Basic BML			
	bevent	id	
	beitem	id	
		type	Must be set to one of the following: “EventMessageFired”, “ModuleUpdated”, “ModuleLocked”, “TimerFired”, “DataEventChanged”, “CCStatusChanged”, “MainAudioStreamChanged”, “NPTReferred”, “MediaStopped”, “DataButtonPressed”
		onoccur	
		es_ref	
		message_group_id	Must be set to “0”
		message_id	
		message_version	
		module_ref	
		language_tag	

Module Classification	Element Entity	Attribute	Operational Restrictions
		time_mode	Must be set to one of the following: “absolute”, “origAbsolute”, “NPT”
		time_value	
		object_id	
		subscribe	
	body&	invisible	
	div&	onfocus	
		onblur	
		accesskey	
	p&	onfocus	
		onblur	
		accesskey	
	span&	onfocus	
		onblur	
		accesskey	
	object&	remain	
		streamstatus	
		streamposition	
		streamlooping	
		onfocus	
		onblur	
		accesskey	

A3-4.6 Operation of CSS-Based Style Sheet

Operational guidelines on properties of CSS-based style sheets are shown in Table 4-9.

Table 4-9 Operational Guidelines on Properties of CSS-Based Style Sheet

Property	Operation
Box model	
margin-top	O
margin-right	O
margin-bottom	O
margin-left	O
margin	O
padding-top	O
padding-right	O
padding-bottom	O
padding-left	O
padding	O
border-top-width	O
border-right-width	O
border-bottom-width	O
border-left-width	O
border-width	O
border-top-color	O
border-right-color	O
border-bottom-color	O
border-left-color	O
border-color	O
border-top-color-index	O
border-right-color-index	O
border-bottom-color-index	O
border-left-color-index	O
border-color-index	O
border-top-style	O
border-right-style	O
border-bottom-style	O
border-left-style	O
border-style	O
border-top	O
border-right	O
border-botom	O
border-left	O
border	O
Visual formatting model	
display	-
position	O

Property	Operation
top	O
right	-
bottom	-
left	O
float	O
clear	O
z-index	-
direction	-
unicode-bidi	-
width	O
height	O
min-width	-
max-width	-
min-height	-
max-height	-
line-height	O
vertical-align	O
Other visual effects	
overflow	O
clip	-
visibility	O
Generated content/Autonumbbering/List	
content	-
quotes	-
counter-reset	-
counter-increment	-
marker-offset	-
list-style-type	O
list-style-image	-
list-style-position	-
list-style	O
Page media	
size	-
marks	-
page-break-before	-
page-break-after	-
page-break-inside	-
page	-
orphans	-
widows	-

Property	Operation
Colours and Background	
clut	O
color	O
color-index	O
background-color	O
background-color-index	O
background-image	O
background-repeat	-
background-attachment	-
background-position	-
background	O
Fonts	
font-family	O
font-style	-
font-variant	-
font-weight	O
font-stretch	-
font-size	O
font-size-adjust	-
font	O
Text	
text-indent	-
text-align	O
text-decoration	-
text-shadow	-
letter-spacing	O
word-spacing	-
text-transform	-
white-space	-
Tables	
caption-side	O
table-layout	-
border-collapse	-
border-spacing	O
empty-cells	-
speak-header	-
User Interface	
cursor	-
outline-color	-
outline-color-index	-
outline-width	-
outline-style	-
outline	-
Aural style sheets	
volume	-
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	-
speak-numeral	-
Extended properties	
resolution	O
display-aspect-ratio	O
grayscale-color-index	O
nav-index	O
nav-up	O
nav-down	O
nav-right	O
nav-left	O
used-key-list	O

A3-4.6.1 Elements applicable to property

Which BML elements are applicable to each available property is shown in Table 4-10.

Table 4-10 BML elements applicable to CSS property

Legend
O Refer to Chapter A3-3.
I This symbol marks a BML element that must be operated only in a content to which an index-based model is specified.
C This symbol marks a BML element that
Δ This symbol marks a BML element that
- Refer to Chapter A3-3.

Property	body	h1-h6, address, blockquote, pre	div, p	em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, b, big, i, small, sub, sup, tt	q	br	span, a	ol, ul	dl, dt, dd	li	hr	form	input	label	select	option	textarea	caption	table	tr	th, td	img	object
Box model																							
margin-top	Δ	O	O	Δ	Δ	Δ	Δ	O	O	O	O	O	O	Δ	O	Δ	O	O	O	-	-	O	O
margin-right	Δ	O	O	Δ	Δ	Δ	Δ	O	O	O	O	O	O	Δ	O	Δ	O	O	O	-	-	O	O
margin-bottom	Δ	O	O	Δ	Δ	Δ	Δ	O	O	O	O	O	O	Δ	O	Δ	O	O	O	-	-	O	O
margin-left	Δ	O	O	Δ	Δ	Δ	Δ	O	O	O	O	O	O	Δ	O	Δ	O	O	O	-	-	O	O
Margin	Δ	O	O	Δ	Δ	Δ	Δ	O	O	O	O	O	O	Δ	O	Δ	O	O	O	-	-	O	O
padding-top	Δ	O	O	Δ	Δ	Δ	Δ	O	O	O	Δ	O	Δ	Δ	Δ	Δ	Δ	O	O	Δ	O	Δ	Δ
padding-right	Δ	O	O	Δ	Δ	Δ	Δ	O	O	O	Δ	O	Δ	Δ	Δ	Δ	Δ	O	O	Δ	O	Δ	Δ
padding-bottom	Δ	O	O	Δ	Δ	Δ	Δ	O	O	O	Δ	O	Δ	Δ	Δ	Δ	Δ	O	O	Δ	O	Δ	Δ
padding-left	Δ	O	O	Δ	Δ	Δ	Δ	O	O	O	Δ	O	Δ	Δ	Δ	Δ	Δ	O	O	Δ	O	Δ	Δ
Padding	Δ	O	O	Δ	Δ	Δ	Δ	O	O	O	Δ	O	Δ	Δ	Δ	Δ	Δ	O	O	Δ	O	Δ	Δ
border-top-width	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O
border-right-width	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O
border-bottom-width	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O
border-left-width	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O
border-width	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O
border-top-color	Δ	C	C	C	C	Δ	C	C	C	C	C	C	C	C	C	Δ	C	C	C	-	C	C	C
border-right-color	Δ	C	C	C	C	Δ	C	C	C	C	C	C	C	C	C	Δ	C	C	C	-	C	C	C
border-bottom-color	Δ	C	C	C	C	Δ	C	C	C	C	C	C	C	C	C	Δ	C	C	C	-	C	C	C
border-left-color	Δ	C	C	C	C	Δ	C	C	C	C	C	C	C	C	C	Δ	C	C	C	-	C	C	C
border-color	Δ	C	C	C	C	Δ	C	C	C	C	C	C	C	C	C	Δ	C	C	C	-	C	C	C
border-top-color-index	Δ	I	I	I	I	Δ	I	I	I	I	I	I	I	I	I	Δ	I	I	I	-	I	I	I
border-right-color-index	Δ	I	I	I	I	Δ	I	I	I	I	I	I	I	I	I	Δ	I	I	I	-	I	I	I
border-bottom-color-index	Δ	I	I	I	I	Δ	I	I	I	I	I	I	I	I	I	Δ	I	I	I	-	I	I	I
border-left-color-index	Δ	I	I	I	I	Δ	I	I	I	I	I	I	I	I	I	Δ	I	I	I	-	I	I	I

Property	body	h1-h6, address, blockquote, pre	div, p	em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, b, big, i, small, sub, sup, tt	q	br	span, a	ol, ul	dl, dt, dd	li	hr	form	input	label	select	option	textarea	caption	table	tr	th, td	img	object
border-color-index	Δ	I	I	I	I	Δ	I	I	I	I	I	I	I	I	I	Δ	I	I	I	-	I	I	I
border-top-style	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O
border-right-style	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O
border-bottom-style	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O
border-left-style	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O
border-style	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O
border-top	Δ	C	C	C	C	Δ	C	C	C	C	C	C	C	C	C	Δ	C	C	C	-	C	C	C
border-right	Δ	C	C	C	C	Δ	C	C	C	C	C	C	C	C	C	Δ	C	C	C	-	C	C	C
border-bottom	Δ	C	C	C	C	Δ	C	C	C	C	C	C	C	C	C	Δ	C	C	C	-	C	C	C
border-left	Δ	C	C	C	C	Δ	C	C	C	C	C	C	C	C	C	Δ	C	C	C	-	C	C	C
border	Δ	C	C	C	C	Δ	C	C	C	C	C	C	C	C	C	Δ	C	C	C	-	C	C	C
Visual formatting model																							
display	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
position	Δ	O	O	Δ	Δ	Δ	Δ	O	O	Δ	O	O	O	Δ	Δ	Δ	Δ	Δ	O	Δ	Δ	Δ	O
top	-	O	O	-	-	-	-	O	O	-	O	O	O	-	-	-	-	-	O	-	-	-	O
right	-	Δ	Δ	-	-	-	-	Δ	Δ	-	Δ	Δ	Δ	-	-	-	-	-	Δ	-	-	-	Δ
bottom	-	Δ	Δ	-	-	-	-	Δ	Δ	-	Δ	Δ	Δ	-	-	-	-	-	Δ	-	-	-	Δ
left	-	O	O	-	-	-	-	O	O	-	O	O	O	-	-	-	-	-	O	-	-	-	O
float	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	O	O
clear	-	O	O	-	-	-	-	O	O	-	O	O	-	-	-	-	-	-	O	-	-	-	-
z-index	-	Δ	Δ	-	-	-	-	Δ	Δ	-	Δ	Δ	Δ	-	-	-	-	-	Δ	-	-	-	Δ
width	Δ	O	O	-	-	-	-	O	O	Δ	O	O	O	-	Δ	Δ	Δ	Δ	O	-	Δ	O	O
height	Δ	O	O	-	-	-	-	O	O	Δ	O	O	O	-	Δ	Δ	Δ	Δ	O	Δ	Δ	O	O
line-height	O	O	O	Δ	Δ	Δ	Δ	O	O	Δ	-	O	O	Δ	Δ	Δ	Δ	Δ	O	Δ	Δ	-	-
vertical-align	-	-	-	Δ	Δ	Δ	Δ	-	-	Δ	-	-	Δ	Δ	Δ	Δ	Δ	Δ	O	O	O	O	O
Other visual effects																							
overflow	Δ	O	O	-	-	-	-	O	O	O	-	O	-	-	-	-	Δ	O	O	O	O	O	O
clip	Δ	Δ	Δ	-	-	-	-	Δ	Δ	Δ	-	Δ	-	-	-	-	Δ	Δ	Δ	Δ	Δ	Δ	Δ
visibility	Δ	O	O	Δ	Δ	Δ	Δ	O	O	O	O	O	O	O	O	Δ	O	O	O	O	O	O	O
Generated content/Autonumbering/List																							
content	-	-	-	-	Δ	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
quotes	-	-	-	-	Δ	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
list-style-type	-	-	-	-	-	-	-	O	-	Δ	-	-	-	-	-	-	-	-	-	-	-	-	-
list-style-image	-	-	-	-	-	-	-	Δ	-	Δ	-	-	-	-	-	-	-	-	-	-	-	-	-
list-style-position	-	-	-	-	-	-	-	Δ	-	Δ	-	-	-	-	-	-	-	-	-	-	-	-	-
list-style	-	-	-	-	-	-	-	O	-	Δ	-	-	-	-	-	-	-	-	-	-	-	-	-
Colours and Background																							
clut	I	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
color	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
color-index	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I

Property	body	h1-h6, address, blockquote, pre	div, p	em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, b, big, i, small, sub, sup, tt	q	br	span, a	ol, ul	dl, dt, dd	li	hr	form	input	label	select	option	textarea	caption	table	tr	th, td	img	object
background-color	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
background-color-index	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
background-image	O	Δ	Δ	Δ	Δ	-	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
background-repeat	Δ	Δ	Δ	Δ	Δ	-	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
background	C	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Fonts																							
font-family	O	O	O	O	O	-	O	O	O	O	-	O	O	O	O	O	O	O	O	O	O	-	-
font-style	Δ	Δ	Δ	Δ	Δ	-	Δ	Δ	Δ	Δ	-	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	-	-
font-weight	O	O	O	O	O	-	O	O	O	O	-	O	O	O	O	O	O	O	O	O	O	-	-
font-size	O	O	O	O	O	-	O	O	O	O	-	O	O	O	O	O	O	O	O	O	O	-	-
font	O	O	O	O	O	-	O	O	O	O	-	O	O	O	O	O	O	O	O	O	O	-	-
Text																							
text-align	O	O	O	-	-	-	-	O	O	O	-	O	O	-	-	-	-	O	O	O	O	-	-
text-decoration	Δ	Δ	Δ	Δ	Δ	-	Δ	Δ	Δ	Δ	-	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	-	-
letter-spacing	O	O	O	Δ	Δ	-	Δ	O	O	O	-	O	O	Δ	Δ	Δ	Δ	O	O	O	O	-	-
white-space	Δ	Δ	Δ	-	-	-	-	Δ	Δ	Δ	-	Δ	Δ	-	-	-	-	Δ	Δ	Δ	Δ	-	-
Tables																							
caption-side	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	O	-	-	-	-	-
table-layout	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Δ	-	-	-	-
border-collapse	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Δ	-	-	-	-
border-spacing	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	O	-	-	-	-
Extended properties																							
resolution	O	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
display-aspect-ratio	O	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
grayscale-color-index	I	I	I	I	I	-	I	I	I	I	-	I	I	I	I	I	I	I	I	I	I	I	I
nav-index	-	-	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
nav-up	-	-	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
nav-right	-	-	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
nav-down	-	-	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
nav-left	-	-	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
used-key-list	O	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

A3-4.6.2 Operation of selectors

This section lists the available selectors to an external style sheet or the style element among the selectors defined in CSS2. The * symbol represents a universal selector. The term, element, represents a type selector. The term, .class, represents a class selector. The term, #id, represents an ID selector. The term, :pseudoClass, represents an :active pseudo-class or a :focus pseudo-class. Any order in describing #id, :pseudoClass, and .class in a single selector is accepted. To operate a selector, the

available combinations of a CSS property and a BML element are limited to the combinations marked by O in Table 4-10. How a signal that is not operable is processed depends on an implementation.

*

element

.class

#id

:pseudoClass

*.class

*:pseudoClass

*#id

element.class

element:pseudoClass

.class:pseudoClass

#id:pseudoClass

*.class:pseudoClass

element.class:pseudoClass

*#id:pseudoClass

Note that this section does not restrict any usage nor operation of a default style sheet.

A3-4.6.3 Media type

Among the @ rules, only @media rules must be operated. The other @ rules, @import rules and the @page rules must not operated. The available media type to @media rules is limited to tv. When a media type other than tv is specified according to a @media rule, any receiver must ignore the designated properties in the @media rule.

A3-4.6.4 !important rule

No !important rule must be operated.

A3-4.6.5 Box model

This section provides operational guidelines on box model properties.

- margin-top, margin-right, margin-bottom, margin-left

  Available value : <margin-width>

  Initial value : 0

  Inherited : no

  The <margin-width> value type may take one of the following values: <length> (not a negative value), <percentage> (not a negative value), and auto. However, the auto value is not operated in a BML document based on an index-based model.

- margin

  Available value : <margin-width>{1,4}

Initial value : not defined for shorthand properties

Inherited : no

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

  Available value : <padding-width>

  Initial value : 0

  Inherited : no

  The <padding-width> value type may take one of the following values: <length> (not a negative value that is not supported by CSS2) or <percentage> (not a negative value that is not supported by CSS2). However, the auto value is not operated in a BML document based on an index-based model.

- padding

  Available value : <padding-width>{1,4}

  Initial value : not defined for shorthand properties

  Inherited : no

- border-top-width, border-right-width, border-bottom-width, border-left-width

  Available value : <border-width>

  Initial value : medium

  Inherited : no

  The <border-width> value type may take one of the following values: thin, medium, thick, or <length> (not a negative value that is illegal). The thin value, the medium value, the thick value must be interpreted as 1px, 2px, 5px, respectively.

- border-width

  Available value : <border-width>{1,4}

  Initial value : refer to individual properties

  Inherited : no

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

  Available value : <color> | transparent | inherit

  Initial value : the value of the colour property

  Inherited : no

  This property is applicable only to a component-based model and not operated in an index-based model. An order in rendering borders depends on an implementation.

- border-color

  Available value : <color>{1,4} | transparent | inherit

  Initial value : refer to individual properties

  Inherited : no

  This property is applicable only to a component-based model and not operated in an index-based model.

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

  Available value : <color-index> | inherit

Initial value : 0

Inherited : no

This property is applicable only to an index-based model and not operated in a component-based model. An order in rendering borders depends on an implementation. When this property is omitted, the initial value, 0, is assumed instead of the color-index value of the concerned element.

- border-color-index

Available value : <color-index>{1,4} | inherit

Initial value : refer to individual properties

Inherited : no

This property is applicable only to an index-based model and not operated in a component-based model.

- border-top-style, border-right-style, border-bottom-style, border-left-style

Available value : none | dotted | dashed | solid

Initial value : none

Inherited : no

How the specified dotted, dashed, or solid line style is presented depends on an implementation.

- border-style

Available value : [none | dotted | dashed | solid]{1,4}

Initial value : refer to individual properties

Inherited : no

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

Available value : <'border-width'> || [none | dotted | dashed | solid] || [<color> | transparent] | inherit

Initial value : refer to individual properties

Inherited : no

This property is applicable only to a component-based model and not operated in an index-based model.

- border

Available value : <'border-width'> || [none | dotted | dashed | solid] || [<color> | transparent] | inherit

Initial value : refer to individual properties

Inherited : no

This property is applicable only to a component-based model and not operated in an index-based model.

A3-4.6.6 Visual formatting model

This section provides operational guidelines on visual formatting models.

- display

This property must contain a fixed value specific to an element. Refer to Annex A.

- position

  Available value : static | absolute

  Initial value : static

  Inherited : no

  A parent element whose child element takes the absolute value must take the absolute value or must be the body element. However, depending on an element, a fixed value may be applied. See Table 4-10 for elements that may take fixed values. Refer to Annex A for values.

- left, top

  Available value : <length> | <percentage> | auto

  Initial value : auto

  Inherited : no

- right, bottom

  This property must take the following fixed value.

  Fixed value : auto

- float

  Available value : left | right | none

  Initial value : none

  Inherited : no

- clear

  Available value : none | left | right | both

  Initial value : none

  Inherited : no

- z-index

  This property must take the following fixed value.

  Fixed value : auto

- width, height

  Available value : <length> | <percentage> | auto

  Initial value : auto

  Inherited : no

- line-height

  Available value : normal | <number> | <length> | <percentage>

  Initial value : normal

  Inherited : yes

  When this property takes normal, a value that is 1.2 times as great as font-size is assumed. However, depending on an element, the fixed value, inherit, may be applied. Any negative value is illegal for <number> and <percentage>, as defined in CSS2.

- vertical-align

  Available value : baseline | top | middle | bottom

Initial value : baseline

Inherited : no

The baseline value is not applicable to the table related elements except the caption element. The values, sub, super, and inherit, are applicable only to a default style sheet. How the concerned element is rendered when sub or super is specified depends on an implementation.

A3-4.6.7 Other visual effects

This section provides operational guidelines on other visual effects.

- overflow

  Available value : visible | hidden

  Initial value : visible

  Inherited : no

- clip

  This property must take the following fixed value.

  Fixed value : auto

- visibility

  Available value : visible | hidden | inherit

  Initial value : inherit

  Inherited :no

  When the visibility property of a parent element is set to hidden and the visibility property of the child element is set to visible, only the parent element is not presented. It is allowed that this property contains the fixed value, inherit, depending on an element. For more information, refer to Table 4-10. Note that the visibility property of the body element must contain the fixed value, visible.

A3-4.6.8 Generated content, Autonumbering, List

This section provides operational guidelines on generated content, autonumbering, and lists.

- content

  This property must contain a fixed value specific to an element. Refer to Annex A.

- quotes

  This property must contain a fixed value specific to an element. Refer to Annex A.

- list-style-type

  Available value : disc | circle | square | decimal | lower-roman | upper-roman | lower-alpha | lower-latin | upper-alpha | upper-latin | inherit

  Initial value : disc

  Inherited : yes

- list-style-image

  This property must take the following fixed value.

  Fixed value : none

- list-style-position

  This property must take the following fixed value.

  Fixed value : outside

- list-style

  Available value : <'list-style-type'> | inherit

  Initial value : not defined for shorthand properties

  Inherited : yes

A3-4.6.9 Colours and Backgrounds

This section provides operational guidelines on colours and backgrounds.

- clut

  This property must be operated as described below.

  Available value : <uri>

  Initial value : variable

  Inherited : yes

  For detailed operational conventions on this property, refer to Section 5.1.9. This property is applicable only to an index-based model and not operated in a component-based model.

- colour

  Available value : <color> | inherit

  Initial value : black

  Inherited : yes

  This property is applicable only to a component-based model and not operated in an index-based model.

- color-index

  This property must be operated as described below.

  Available value : <color-index> | inherit

  Initial value : 0

  Inherited : yes

  This property is applicable only to an index-based model and not operated in a component-based model.

- background-color

  Available value : <color> | transparent | inherit

  Initial value : transparent

  Inherited : no

  This property is applicable only to a component-based model and not operated in an index-based model.

- background-color-index

  Available value : <color-index> | inherit

Initial value : 0

Inherited : yes

This property is applicable only to an index-based model and not operated in a component-based model.

- background-image

  Available value : <uri> | none

  Initial value : none

  Inherited : no

  The graphic image coding specification specified with the <uri> value type must be limited to JPEG. This property is applicable only to the body element. The upper left corner of the presented area must be rendered first.

- background-repeat

  This property must take the following fixed value.

  Fixed value : repeat

- background

  Available value : <'background-color'> || <'background-image'>

  Initial value : not defined for shorthand properties

  Inherited : no

  This property is applicable only to a component-based model and not operated in an index-based model.

A3-4.6.10 Fonts

The available specifications for a font in a BML document are listed in Table 4-10. Any character of the code set 0 in EUC-JP and the 1-byte characters in Shift-JIS is rendered with a half width of the font size. Any external charcter specified by using DRCS is a full-width character and rendered with the corresponding font size.

Table 4-10 Available Fonts

Font family	Font size	Gradation (Note)
round gothic	16px, 20px, 24px, 30px, 36px	4
bold round gothic	30px	4
square gothic	20px, 24px	4

Note: The gradation value for a component-based model depends on an implementation.

Operational guidelines on properties related to fonts are described below.

- font-family

  This property must be operated as described below.

  Available value : "round gothic" | "bold round gothic" | "square gothic" |'round gothic' | 'bold round gothic' | 'square gothic' | inherit

  Initial value : "round gothic"

  Inherited : yes

- font-style

This property must contain a fixed value, which is defined in a default style sheet. Which font family is applied to the characters to which this property is applied depends on an implementation.

- font-weight

  Available value : normal | bold | inherit

  Initial value : normal

  Inherited : yes

  How a bold character is rendered complies with the definition of text with a control code (8-bit character code) specified with the text/X-arib-jis8text media type.

A3-4.4.11 Text

This section provides operational guidelines on properties related to text.

- text-align

  Available value : left | right | center

  Initial value : left

  Inherited : yes

- text-decoration

  This property must take a fixed value. Refer to Annex A.

- letter-spacing

  Available value : normal | <length> | inherit

  Initial value : normal

  Inherited : yes

  When normal is specified, 0 is assumed as a computed value. The <length> value type must not take a negative value. This property specifies the length of a space between characters in the specified elements. This property may take the fixed value, inherit depending on an element.

- white-space

  This property must take a fixed value. Refer to Annex A.

A3-4.6.12 Tables

This section provides operational guidelines on properties related to tables.

- caption-side

  Available value : top | bottom

  Initial value : top

  Inherited : yes

- table-layout

  This property must take the following fixed value.

  Fixed value : auto

- border-collapse

  This property must take the following fixed value.

Fixed value : separate

- border-spacing

  Available value : <length> <length>?

  Initial value : 0

  Inherited : yes

A3-4.6.13 Pseudo classes

The available CSS properties to the :active and :focus classes are listed in Table 4-11.

Table 4-11 Operational Guideline on Available CSS Properties to :focus and :active Classes

Legend O: Refer to Chapter 3.
If any restriction exists, it is expressed in the corresponding operational guideline.
-: Refer to Chapter 3.

Property	body	h1-h6, address, blockquote, pre	div	p	em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, b, big, i, small sub, sup, tt	q	br	span, a	ol, ul	dl, ,dt, dd	li	hr	form	input	label	select	option	textarea	caption	table	tr	th, td	img	object
Box model																								
margin-top	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
margin-right	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
margin-bottom	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
margin-left	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
margin	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
padding-top	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
padding-right	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
padding-bottom	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
padding-left	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
padding	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-top-width	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-right-width	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-bottom-width	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-left-width	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-width	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-top-color	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
border-right-color	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
border-bottom-color	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
border-left-color	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
border-color	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-top-color-	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O

Property	body	h1-h6, address, blockquote, pre	div	p	em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, b, big, i, small sub, sup, tt	q	br	span, a	ol, ul	dl, ,dt, dd	li	hr	form	input	label	select	option	textarea	caption	table	tr	th, td	img	object
index																								
border-right-color-index	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
border-bottom-color-index	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
border-left-color-index	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
border-color-index	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-top-style	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-right-style	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-bottom-style	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-left-style	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-style	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-top	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-right	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-bottom	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-left	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Visual formatting model																								
display	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
position	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
top	-	-	O	O	-	-	-	-	-	-	-	-	-	O	-	-	-	-	-	-	-	-	-	O
right	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bottom	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
left	-	-	O	O	-	-	-	-	-	-	-	-	-	O	-	-	-	-	-	-	-	-	-	O
float	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
clear	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
z-index	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
width	-	-	O	O	-	-	-	-	-	-	-	-	-	O	-	-	-	-	-	-	-	-	-	O
height	-	-	O	O	-	-	-	-	-	-	-	-	-	O	-	-	-	-	-	-	-	-	-	O
line-height	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
vertical-align	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Other visual effects																								
overflow	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
clip	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
visibility	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Generated content/Autonumbering/List																								
content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
quotes	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Property	body	h1-h6, address, blockquote, pre	div	p	em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, b, big, i, small sub, sup, tt	q	br	span, a	ol, ul	dl, dt, dd	li	hr	form	input	label	select	option	textarea	caption	table	tr	th, td	img	object
list-style-type	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
list-style-image	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
list-style-position	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
list-style	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Colors and Backgrounds																								
clut	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
color	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
color-index	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
background-color	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
background-color-index	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
background-image	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
background-repeat	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
background	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Fonts																								
font-family	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	-
font-style	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
font-weight	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	-
font-size	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	-
font	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Text																								
text-align	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
text-decoration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
letter-spacing	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
white-space	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Tables																								
caption-side	-	-	-		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
table-layout	-	-	-		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-collapse	-	-	-		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-spacing	-	-	-		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
empty-cells	-	-	-		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Extended properties																								
resolution	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
display-aspect-ratio	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
grayscale-color-index	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	-
nav-index	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
nav-up	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
nav-right	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Property	body	h1-h6, address, blockquote, pre	div	p	em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, b, big, i, small sub, sup, tt	q	br	span, a	ol, ul	dl, ,dt, dd	li	hr	form	input	label	select	option	textarea	caption	table	tr	th, td	img	object
nav-down	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
nav-left	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
used-key-list	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

A3-4.6.14 Extended properties

This section provides operational guidelines on extended properties.

- resolution

  Available value : 960x540 | 720x480

  Initial value : 960x540

  Inherited : no

- display-aspect-ratio

  Available value : 16v9 | 4v3

  Initial value : 16v9

  Inherited : no

- grayscale-color-index

  Available value : <color-index><color-index> | inherit

  Initial value : the value of the color-index property

  Inherited : yes

  The four gradation colours are, the foreground colour, the first <color-index> value, the second <color-index> value and the background colour. When 8 is specified, the background colour is displayed.

  This property is applicable only to an index-based model and not operated in a component-based model.

  Note that when DRCS is used for a content based on a component-based model, which colours are applied to the medium grey scale specifications in DRCS depends on an implementation.

- nav-index, nav-up, nav-down, nav-left, nav-right

  Available value : <integer> | none

  Initial value : none

  Inherited : no

  The available values to the <integer> value type are 0 through 32767.

- used-key-list

Available value : <key-group>+ | none

Initial value : basic data-button

Inherited : no

A3-4.6.15 Default style sheet

The available default style sheets are shown in Annex A. Each default style sheets is assumed to be virtually implemented in a receiver. There is no need to transmit any of the default style sheets. However, depending on an individual receiver, additional specifications may be contained in the default style sheet.

A3-4.6.16 Operation of length

The <length> value type must be operated in order to specify a length in em, ex, or px units. Which unit is used as a basic unit when em or ex is specified depends on a receiver. When 0 is specified, 0px is assumed as a computed value.

A3-4.6.17 Actual values derived from computed values

An actual value derived from a computed value that is a decimal fraction depends on a receiver.

A3-4.6.18 <color-index>

The <color-index> value type must contain an integer, 0 through 223.

A3-4.7 Operational Guidelines on Procedural Description

A2-4.7.1 Operational guidelines on DOM

This section provides tables that indicate the available interface, attribute, and method is applicable and use the following symbols:

The O mark means that the interface, attribute, or method must be operated.

The – mark means that the interface, attribute, or method must not be operated.

The R character means that any access to the interface, attribute, or method is limited to a reading access.

The RW string means that any access to the interface, attribute, or method may be a reading access or a writing access.

A3-4.7.1.1 Operation of DOM core interfaces

This section provides the table that indicates which DOM core interfaces are allowed to be operated. This section also provides the table that describes which and how attributes and methods of a DOM core interface are allowed to be operated.

Interface Classification	Interface	Operation
Core Interfaces	DOMException	-
	DOMImplementation	O

Interface Classification	Interface	Operation
	DocumentFragment	-
	Document	O
	Node	O
	NodeList	-
	NamedNodeMap	-
	CharacterData	O
	Attr	-
	Element	O
	Text	O
	Comment	-
Extended Interfaces	CDATASection	O
	DocumentType	-
	Notation	-
	Entity	-
	EntityReference	-
	ProcessingInstruction	-

Interface	Attribute/Method	Operation	Restriction
DOMImplementation	hasFeature()	O	
Document	doctype	-	
	implementation	O	R
	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()	-	

Interface	Attribute/Method	Operation	Restriction
	replaceChild()	-	
	removeChild()	-	
	appendChild()	-	
	hasChildNodes()	-	
	cloneNode()	-	
CharacterData	data	O (Note 1)	RW (Note 2)
	length	O (Note 1)	R
	substringData()	-	
	appendData()	-	
	insertData()	-	
	deleteData()	-	
	replaceData()	-	
Element	tagName	O	R
	getAttribute()	-	
	setAttribute()	-	
	removeAttribute()	-	
	getAttributeNode()	-	
	setAttributeNode()	-	
	removeAttributeNode()	-	
	getElementsByTagName()	-	
	normalize()	-	
Text	splitText()	-	
CDATASection			

Note 1: No attribute in the above table is applicable to any CDATA section node that is a child node of the style element node or the script element node.

Note 2: The restriction level R is applicable to any CDATA section node that is a child node of the option element node or any text node that is a child node of the option element node.

A3-4.7.1 2 Operation of DOM HTML interfaces

Operation of DOM HTML interfaces applicable to events

This section provides the table that indicates which DOM HTML interfaces are applicable to events. This section also contains the table that describes what and how values of the type attribute and the target attribute are obtained by using the BMLEvent interface.

Interface	Attribute/Method	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
	languageTag	O	R
	registerId	-	

Interface	Attribute/Method	Operation	Restriction
	serviceId	-	
	eventId	-	
	object	O	R

Note 1: The format of a value obtained through this attribute must be /<component_tag>, as defined in Section 9.2, Volume 2.
Note 2: The format of a value obtained through this attribute must be /<component_tag>/<module_id>, as defined in Section 9.2, Volume 2.

Condition for reading access			Value of type attribute	Value of target attribute
Event handler onkeydown is executed			"keydown"	Element specifying the event handler
Event handler onkeyup is executed			"keyup"	Element specifying the event handler
Event handler onclickis executed			"click"	Element specifying the event handler
Event handler onfocusis executed			"focus"	Element specifying the event handler
Event handler onbluris executed			"blur"	Element specifying the event handler
Event handler onloadis executed			"load"	Element specifying the event handler
Event handler onunloadis executed			"unload"	Element specifying the event handler
Event handler onchangeis executed			"change"	Element specifying the event handler
Event handler onsubmitis executed			"submit"	Element specifying the event handler
Event handler onresetis executed			"reset"	Element specifying the event handler
Event handler Onoccur is executed		EventMessageFired	"EventMessageFired"	
		ModuleUpdated	"ModuleUpdated"	
		ModuleLocked	"ModuleLocked"	
		TimerFired	"TimerFired"	
		NPTReferred	"NPTReferred"	Element specifying the handler
		MediaStopped (Note)	"MediaStopped"	
		DataEventChanged	"DataEventChanged"	
		CCStatusChanged	"CCStatusChanged"	
		MainAudioStreamChanged	"MainAudioStreamChanged"	
		DataButtonPressed	"DataButtonPressed"	
Global code is executed			undefined	null
Specified procedure description is executed upon invoking setTimeout(), extended function for broadcasting			undefined	null
Specified procedure description is executed upon invoking setInterval(), extended function for broadcasting			undefined	null

Note: A MediaStopped event arises from monomedia: audio/X-arib-mpeg2-aac, audio/X-arib-aiff, and application/X-arib-mpeg2-tts.

Operation of interfaces applicable to document nodes

The following table indicates which and how DOM HTML interface and DOM BML interface is applicable to document nodes.

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

Note When a value obtained through each attribute in the above table is assigned to a global variable, how the interface behaves depends on an implementation.

Operation of DOM HTML interfaces applicable to element nodes

The following table describes which and how DOM HTML interfaces are applicable to element nodes.

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
html	HTMLElement	id	-	
		className	-	
		title	-	
		lang	-	
		dir	-	
	HTMLHtmlElement	version	-	
	BMLBmlElement	style	-	
		normalStyle	-	
		focusStyle	-	
		activeStyle	-	
bml:bevent	HTMLElement	id	O	R
		className	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		title	-	
		lang	-	
		dir	-	
	BMLBeventElement			
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
		languageTag	O	RW
		registerId	-	
		serviceId	-	
		eventId	-	
		timeMode	O	R
		timeValue	O	RW
		objectId	O	RW
		subscribe	O	RW
body	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLBodyElement			
	BMLBodyElement	invisible	O	RW
		style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
head	HTMLElement	id	-	
		className	-	
		title	-	
		lang	-	
		dir	-	
	HTMLHeadElement	profile	-	
title	HTMLElement	id	-	
		className	-	
		title	-	
		lang	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		dir	-	
	HTMLTitleElement	text	O	R
h1 h2 h3 h4 h5 h6	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLHeadingElement			
	BMLHeadingElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
address	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
blockquote	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLBlockquoteElement	cite	-	
	BMLBlockquoteElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
div	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLDivElement			
	BMLDivElement	style	-	
		normalStyle	O	R
		focusStyle	O	R
		activeStyle	O	R
		accessKey	O	R
		focus()	O	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		blur()	O	
p	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLParagraphElement			
	BMLParagraphElement	style	-	
		normalStyle	O	R
		focusStyle	O	R
		activeStyle	O	R
		accessKey	O	R
		focus()	O	
		blur()	O	
pre	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLPreElement			
	BMLPreElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
abbr	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
acronym	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
br	HTMLElement	id	O	R
		className	O	R

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		title	-	
		lang	-	
		dir	-	
	HTMLBRElement			
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
cite	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
code	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
dfn	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
em	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
kbd	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
q	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLQuoteElement	cite	-	
	BMLQuoteElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
samp	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
span	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLSpanElement	style	-	
		normalStyle	O	R
		focusStyle	O	R
		activeStyle	O	R
		accessKey	O	R
		focus()	O	
		blur()	O	
strong	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
var	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
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()	O	
		focus()	O	
	BMLAnchorElement	style	-	
		normalStyle	O	R
		focusStyle	O	R
		activeStyle	O	R
		effect	-	
dl	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLDListElement			

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
	BMLDListElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
dt	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
dd	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
ol	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLOListElement			
	BMLOListElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
ul	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLUListElement			
	BMLUListElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
li	HTMLElement	id	O	R
		className	O	R

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		title	-	
		lang	-	
		dir	-	
	HTMLLIElement			
	BMLLIElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
b	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
big	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
hr	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLHRElement			
	BMLHREElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
i	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		activeStyle	-	
small	HTMLElement	Id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
sub	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
sup	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
tt	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
form	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLFormElement	elements	-	
		length	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		name	-	
		acceptCharset	-	
		action	O	RW
		enctype	O	RW
		method	O	R
		submit()	O	
		reset()	O	
	BMLFormElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
		accept	O	R
input	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLInputElement	defaultValue	O	R
		defaultChecked	O	R
		form	O	R
		accept	-	
		accessKey	O	R
		alt	O	RW
		checked	O	RW
		disabled	-	
		maxLength	O	R
		name	-	
		readOnly	-	
		size	O	R
		src	O	RW(*1)
		tabIndex	-	
		type	O	R
		useMap	-	
		value	O	RW
		blur()	O	
		focus()	O	
		select()	-	
		click()	-	
	BMLInputElement	style	-	
		normalStyle	O	R
		focusStyle	O	R
		activeStyle	O	R
label	HTMLElement	id	O	R
		className	O	R

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		title	-	
		lang	-	
		dir	-	
	HTMLLabelElement	form	O	R
		accessKey	-	
		htmlFor	O	R
	BMLLabelElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
select	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLSelectElement	type	-	
		selectedIndex	O	R
		value	-	
		length	-	
		form	O	R
		options	-	
		disabled	-	
		multiple	O	R
		name	O	R
		size	O	R
		tabIndex	-	
		add()	-	
		remove()	-	
		blur()	O	
		focus()	O	
	BMLSelectElement	style	-	
		normalStyle	O	R
		focusStyle	O	R
		activeStyle	O	R
option	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLOptionElement	form	O	R
		defaultSelected	O	R
		text	O	R
		index	O	R
		disabled	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		label	-	
		selected	O	RW
		value	O	R
	BMLOptionElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
textarea	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLTextAreaElement	defaultValue	O	R
		form	O	R
		accessKey	O	R
		cols	O	R
		disabled	-	
		name	O	R
		readOnly	-	
		rows	O	R
		tabIndex	-	
		type	-	
		value	O	RW
		blur()	O	
		focus()	O	
		select()	-	
	BMLTextAreaElement	style	-	
		normalStyle	O	R
		focusStyle	O	R
		activeStyle	O	R
caption	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLTableCaptionElement			
	BMLTableCaptionElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
table	HTMLElement	id	O	R
		className	O	R

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		title	-	
		lang	-	
		dir	-	
	HTMLTableElement	caption	-	
		tHead	-	
		tFoot	-	
		rows	-	
		tBodies	-	
		border	-	
		cellPadding	-	
		cellSpacing	-	
		frame	-	
		rules	-	
		summary	-	
		Width	-	
		createTHead()	-	
		deleteTHead()	-	
		createTFoot()	-	
		deleteTFoot()	-	
		createCaption()	-	
		deleteCaption()	-	
		insertRow()	-	
		deleteRow()	-	
	BMLTableElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
tr	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLTableRowElement	rowIndex	-	
		sectionRowIndex	-	
		cells	-	
		align	-	
		Ch	-	
		chOff	-	
		vAlign	-	
		insertCell()	-	
		deleteCell()	-	
	BMLTableRowElement	style	-	
		normalStyle	O	R
		focusStyle	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		activeStyle	-	
th td	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLTableCellElement	cellIndex	-	
		abbr	-	
		align	-	
		axis	-	
		ch	-	
		chOff	-	
		colSpan	-	
		headers	-	
		rowSpan	-	
		scope	-	
		vAlign	-	
	BMLTableCellElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
img	HTMLElement	id	O	R
		className	O	R
		Title	-	
		Lang	-	
		dir	-	
	HTMLImageElement	lowSrc	-	
		name	-	
		alt	-	
		height	-	
		isMap	-	
		longDesc	-	
		src	-	
		useMap	-	
		width	-	
	BMLImageElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
object	HTMLElement	Id	O	R
		className	O	R
		Title	-	
		Lang	-	
		Dir	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
	HTMLObjectElement	form	-	
		code	-	
		archive	-	
		codeBase	-	
		codeType	-	
		data	O	RW
		declare	-	
		height	-	
		name	-	
		standby	-	
		tabIndex	-	
		type	O	R
		useMap	-	
		width	-	
	BMLObjectElement	style	-	
		normalStyle	O	R
		focusStyle	O	R
		activeStyle	O	R
		classId	-	
		accessKey	O	R
		remain	O	RW
		streamPosition	O(*2)	RW
		streamStatus	O	RW
		streamLooping	-	
		streamSpeedNumerator	-	
		streamSpeedDenominator	-	
		streamLevel	-	
		setSpeed()	-	
		movePosition()	-	
		hasAssociatedIndex()	-	
		assignToLocalEvnet()	-	
		assignToNodePlayMode()	-	
		getMainAudioStream()	O	
		setMainAudioStream()	O	
		focus()	O	
		Blur()	O	
param	HTMLElement	Id	-	
		className	-	
		title	-	
		lang	-	
		dir	-	
	HTMLParamElement	name	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		type	-	
		value	-	
		valueType	-	
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	-	
style	HTMLElement	id	-	
		className	-	
		title	-	
		lang	-	
		dir	-	
	HTMLStyleElement	disabled	-	
		media	-	
		type	-	
link	HTMLElement	id	-	
		className	-	
		title	-	
		lang	-	
		dir	-	
	HTMLLinkElement	disabled	-	
		charset	-	
		href	-	
		hreflang	-	
		media	-	
		rel	-	
		rev	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		type	-	
	BMLLinkElement	style	-	
		normalStyle	-	
		focusStyle	-	
		activeStyle	-	
base	HTMLElement	id	-	
		className	-	
		title	-	
		lang	-	
		dir	-	
	HTMLBaseElement	href	-	

*1) This attribute is allowed to be rewritten provided that the type of a referenced monomedia retains before and after the rewriting operation.

*2) This attribute is allowed to be applicable only to a node whose type attribute is set to “image/X-arib-mng”.

Operation of BMLCSS2Properties interface

The BMLCSS2Properties interface is allowed to be operated based on a presentation model (an index-based model or a component-based model), as described later. The following tables summarize operational guidelines on the BMLCSS2Properties interface.

Symbol	Semantics
RW	Operated as readable and rewritable
cRW	In a component-based model, operated as readable and rewritable. In an index-based model, not operated.
iRW	In an index-based model a component-based model, operated as readable and rewritable. In a component-based model, not operated.
R	Operated only as readable.
cR	In a component-based model, operated only as readable. In an index-based model, not operated.
iR	In an index-based model a component-based model, operated only as readable. In a component-based model, not operated.

Upper Row: via the normalStyle attribute Lower Row: via the focusStyle or activeStyle attribute	body	h1, h2, h3, h4, h5, h6, address, blockquote, pre	div, p	em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, b, big, i, small, sub, sup, tt	q	br	span, a	ol, ul, dl	dt, dd	li	hr	form	input	label	select	option	textarea	caption	table	tr	th, td	img	object
marginTop	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
marginRight	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Upper Row: via the normalStyle attribute Lower Row: via the focusStyle or activeStyle attribute	body	h1, h2, h3, h4, h5, h6, address, blockquote, pre	div, p	em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, b, big, i, small, sub, sup, tt	q	br	span, a	ol, ul, dl	dt, dd	li	hr	form	input	label	select	option	textarea	caption	table	tr	th, td	img	object
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
marginBottom	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
marginLeft	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
margin	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
paddingTop	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
paddingRight	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
paddingBottom	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
paddingLeft	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
padding	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderTopWidth	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	R	R	R
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderRightWidth	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	R	R	R
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderBottomWidth	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	R	R	R
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderLeftWidth	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	R	R	R
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderWidth	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderTopColor	cR	cRW	cRW	cRW	cRW	cR	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cR	cRW	cRW	cRW	-	cRW	cRW	cRW
	-	-	cRW	-	-	-	cRW	-	-	-	-	-	cRW	-	cRW	-	cRW	-	-	-	-	-	cRW
borderRightColor	cR	cRW	cRW	cRW	cRW	cR	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cR	cRW	cRW	cRW	-	cRW	cRW	cRW
	-	-	cRW	-	-	-	cRW	-	-	-	-	-	cRW	-	cRW	-	cRW	-	-	-	-	-	cRW

Upper Row: via the normalStyle attribute Lower Row: via the focusStyle or activeStyle attribute	body	h1, h2, h3, h4, h5, h6, address, blockquote, pre	div, p	em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, b, big, i, small, sub, sup, tt	q	br	span, a	ol, ul, dl	dt, dd	li	hr	form	input	label	select	option	textarea	caption	table	tr	th, td	img	object
borderBottomColor	cR	cRW	cRW	cRW	cRW	cR	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cR	cRW	cRW	cRW	-	cRW	cRW	cRW
	-	-	cRW	-	-	-	cRW	-	-	-	-	-	cRW	-	cRW	-	cRW	-	-	-	-	-	cRW
borderLeftColor	cR	cRW	cRW	cRW	cRW	cR	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cR	cRW	cRW	cRW	-	cRW	cRW	cRW
	-	-	cRW	-	-	-	cRW	-	-	-	-	-	cRW	-	cRW	-	cRW	-	-	-	-	-	cRW
borderColor	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderTopColorIndex	iR	iRW	iRW	iRW	iRW	iR	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iR	iRW	iRW	iRW	-	iRW	iRW	iRW
	-	-	iRW	-	-	-	iRW	-	-	-	-	-	iRW	-	iRW	-	iRW	-	-	-	-	-	iRW
borderRightColorIndex	iR	iRW	iRW	iRW	iRW	iR	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iR	iRW	iRW	iRW	-	iRW	iRW	iRW
	-	-	iRW	-	-	-	iRW	-	-	-	-	-	iRW	-	iRW	-	iRW	-	-	-	-	-	iRW
borderBottomColorIndex	iR	iRW	iRW	iRW	iRW	iR	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iR	iRW	iRW	iRW	-	iRW	iRW	iRW
	-	-	iRW	-	-	-	iRW	-	-	-	-	-	iRW	-	iRW	-	iRW	-	-	-	-	-	iRW
borderLeftColorIndex	iR	iRW	iRW	iRW	iRW	iR	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iR	iRW	iRW	iRW	-	iRW	iRW	iRW
	-	-	iRW	-	-	-	iRW	-	-	-	-	-	iRW	-	iRW	-	iRW	-	-	-	-	-	iRW
borderColorIndex	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderTopStyle	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	R	R	R
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderRightStyle	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	R	R	R
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderBottomStyle	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	R	R	R
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderLeftStyle	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	R	R	R
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderStyle	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Upper Row: via the normalStyle attribute Lower Row: via the focusStyle or activeStyle attribute	body	h1, h2, h3, h4, h5, h6, address, blockquote, pre	div, p	em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, b, big, i, small, sub, sup, tt	q	br	span, a	ol, ul, dl	dt, dd	li	hr	form	input	label	select	option	textarea	caption	table	tr	th, td	img	object
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderTop	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderRight	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderBottom	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderLeft	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
display	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
position	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
top	-	RW	RW	-	-	-	-	RW	RW	-	RW	RW	RW	-	-	-	-	-	RW	-	-	-	RW
	-	-	RW	-	-	-	-	-	-	-	-	-	RW	-	-	-	-	-	-	-	-	-	RW
right	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bottom	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
left	-	RW	RW	-	-	-	-	RW	RW	-	RW	RW	RW	-	-	-	-	-	RW	-	-	-	RW
	-	-	RW	-	-	-	-	-	-	-	-	-	RW	-	-	-	-	-	-	-	-	-	RW
float	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
clear	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
zIndex	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
width	R	RW	RW	-	-	-	-	RW	RW	R	RW	RW	RW	-	R	R	R	R	RW	-	R	RW	RW

Upper Row: via the normalStyle attribute Lower Row: via the focusStyle or activeStyle attribute	body	h1, h2, h3, h4, h5, h6, address, blockquote, pre	div, p	em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, b, big, i, small, sub, sup, tt	q	br	span, a	ol, ul, dl	dt, dd	li	hr	form	input	label	select	option	textarea	caption	table	tr	th, td	img	object
	-	-	RW	-	-	-	-	-	-	-	-	-	RW	-	-	-	-	-	-	-	-	-	RW
height	R	RW	RW	-	-	-	-	RW	RW	R	RW	RW	RW	-	R	R	R	R	RW	R	R	RW	RW
	-	-	RW	-	-	-	-	-	-	-	-	-	RW	-	-	-	-	-	-	-	-	-	RW
lineHeight	R	R	R	R	R	R	R	R	R	R	-	R	R	R	R	R	R	R	R	R	R	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
verticalAlign	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
overflow	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
clip	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
visibility	R	RW	RW	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	R	RW	RW	RW	RW	RW	RW	RW
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
quotes	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
listStyleType	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
listStyleImage	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
listStylePosition	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
listStyle	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
clut	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
color	cRW	cRW	cRW	cRW	cRW	-	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW
	-	-	cRW	-	-	-	cRW	-	-	-	-	-	cRW	-	cRW	-	cRW	-	-	-	-	-	cRW

Upper Row: via the normalStyle attribute Lower Row: via the focusStyle or activeStyle attribute	body	h1, h2, h3, h4, h5, h6, address, blockquote, pre	div, p	em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, b, big, i, small, sub, sup, tt	q	br	span, a	ol, ul, dl	dt, dd	li	hr	form	input	label	select	option	textarea	caption	table	tr	th, td	img	object
colorIndex	iRW	iRW	iRW	iRW	iRW	-	iRW	iRW	iRW	iRW	-	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW
	-	-	iRW	-	-	-	iRW	-	-	-	-	-	iRW	-	iRW	-	iRW	-	-	-	-	-	iRW
backgroundColor	cRW	cRW	cRW	cRW	cRW	-	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW
	-	-	cRW	-	-	-	cRW	-	-	-	-	-	cRW	-	cRW	-	cRW	-	-	-	-	-	cRW
backgroundColorIndex	iRW	iRW	iRW	iRW	iRW	-	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW
	-	-	iRW	-	-	-	iRW	-	-	-	-	-	iRW	-	iRW	-	iRW	-	-	-	-	-	iRW
backgroundImage	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
backgroundRepeat	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
background	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
fontFamily	RW	RW	RW	RW	RW	-	RW	RW	RW	RW	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	-	-
	-	-	RW	-	-	-	RW	-	-	-	-	-	RW	-	RW	-	RW	-	-	-	-	-	-
fontStyle	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
fontWeight	RW	RW	RW	RW	RW	-	RW	RW	RW	RW	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	-	-
	-	-	RW	-	-	-	RW	-	-	-	-	-	RW	-	RW	-	RW	-	-	-	-	-	-
fontSize	RW	RW	RW	RW	RW	-	RW	RW	RW	RW	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	-	-
	-	-	RW	-	-	-	RW	-	-	-	-	-	RW	-	RW	-	RW	-	-	-	-	-	-
font	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
textAlign	R	R	R	-	-	-	-	R	R	R	-	R	R	-	-	-	-	R	R	R	R	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
textDecoration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Upper Row: via the normalStyle attribute Lower Row: via the focusStyle or activeStyle attribute	body	h1, h2, h3, h4, h5, h6, address, blockquote, pre	div, p	em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, b, big, i, small, sub, sup, tt	q	br	span, a	ol, ul, dl	dt, dd	li	hr	form	input	label	select	option	textarea	caption	table	tr	th, td	img	object
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
letterSpacing	R	R	R	R	R	-	R	R	R	R	-	R	R	R	R	R	R	R	R	R	R	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
whiteSpace	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
captionSide	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
tableLayout	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderCollapse	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderSpacing	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
resolution		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
displayAspectRatio		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
grayscaleColorIndex	iRW	iRW	iRW	iRW	iRW	-	iRW	iRW	iRW	iRW	-	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW
	-	-	iRW	-	-	-	iRW	-	-	-	-	-	iRW	-	iRW	-	iRW	-	-	-	-	-	-
navIndex	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
navUp	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
navRight	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
navDown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
navLeft	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
usedKeyList	RW	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

A3-4.7.2 Operational scope for built-in objects

The operational scope for built-in objects complies with the convention defined in Section A2-4.5.2, APPENDIX 2.

A3-4.7.3 Operational scope for extended objects for broadcasting

The operational scope for extended objects for broadcasting complies with the convention defined in Section A2-4.5.3, APPENDIX 2.

A3-4.7.4 Operational scope for browser pseudo objects

A3-4.7.4.1 Script Description

Any document must contain not more than one Global Object.

The applicable operations of browser pseudo objects regarding to script descriptions are limited to describing procedures in a CDATA section node that is the sole child element of the script element and reading an external script file via the src attribute.

The script element may be contained only in a header object. Two ore more script elements may exist. However, any document may not more than one script element that directrly describes procedures.

Refer to Chapter 3 for what the O and - symbols mean.

A3-4.7.4.2 Operational scope for browser pseudo objects

The following table provides operational guidelines on browser pseudo objects.

	Function	Sub profile 1	Sub profile 2	Remarks
Ureg function				
	Ureg[]	O	O	
Greg function				
	Greg[]	O	O	
EPGfunctions				
	epgGetEventStartTime()	O	O	
	epgGetEventDuration()	O	O	
	epgTune()	O	O	
	epgTuneToComponent()	O	O	
	epgTuneToDocument()	-	-	
	epgIsReserved()	O	O	
	epgReserve()	O	O	
	epgCancelReservation()	O	O	
	epgRecIsReserved()	O	O	
	epgRecReserve()	O	O	
	epgRecCancelReservation()	O	O	
Program group index functions				
	grpIsReserved()	-	-	
	grpReserve()	-	-	
	grpCancelReservation()	-	-	

	Function	Sub profile 1	Sub profile 2	Remarks
	grpRecIsReserved()	-	-	
	grpRecReserve()	-	-	
	grpRecCancelReservation()	-	-	
	grpGetNodeEventList()	-	-	
	grpGetERTNodeName()	-	-	
	grpGetERTNodeDescription()	-	-	
	epgXTune()	-	-	
Series reservation functions				
	seriesIsReserved()	O	O	(Note)Verify if this function is supported in an implementation.
	seriesReserve()	O	O	(Note)Verify if this function is supported in an implementation.
	seriesCancelReservation()	O	O	(Note)Verify if this function is supported in an implementation.
	seriesRecIsReserved()	O	O	(Note)Verify if this function is supported in an implementation.
	seriesRecReserve()	O	O	(Note)Verify if this function is supported in an implementation.
	seriesRecCancelReservation()	O	O	(Note)Verify if this function is supported in an implementation.
Non-volatile memory functions				
	readPersistentString()	-	-	
	readPersistentNumber()	-	-	
	readPersistentArray()	O	O	(Note) Applicable only to contents received from real-time broadcasting service or stored from broadcasting services
	writePersistentString()	-	-	
	writePersistentNumber()	-	-	
	writePersistentArray()	O	O	(Note) Applicable only to contents received from real-time broadcasting service or stored from broadcasting services
	copyPersistent()	-	-	
	getPersistentInfoList()	-	-	
	deletePersistent()	-	-	
	getFreeSpace()	-	-	

	Function	Sub profile 1	Sub profile 2	Remarks
Functions for controlling access-controlled areas				
	isSupportedPersistentType()	-	-	
	setAccessInfoOfPersistentArray()	O	O	(Note)Verify if this function is supported in an implementation.
	checkAccessInfoOfPersistentArray()	O	O	(Note)Verify if this function is supported in an implementation.
	writePersistentArrayWithAccessCheck()	O	O	(Note)Verify if this function is supported in an implementation.
	readPersistentArrayWithAccessCheck()	O	O	(Note)Verify if this function is supported in an implementation.
Interaction channel functions				
Interaction channel functions-Delayed call functions				
	registerTransmission()	-	-	
	registerTransmissionStatus()	-	-	
	getTransmissionStatus()	-	-	
	setDelayedTransmissionDataOverBASIC()	O	O	(Note)Verify if this function is supported in an implementation.
Interaction channel functions-Communication Functions assuming BASIC procedure				
	connect()	O	O	
	disconnect()	O	O	
	sendBinaryData()	-	-	
	receiveBinaryData()		-	
	sendTextData()	O	O	
	receiveTextData()	O	O	
Interaction channel functions-Communication functions assuming TCP/IP				
	setISPParams()	O	O	(Note)Verify if this function is supported in an implementation.
	getISPParams()	O	O	(Note)Verify if this function is supported in an implementation.
	connectPPP()	O	O	(Note)Verify if this function is supported in an implementation.
	connectPPPWithISPParams()	O	O	(Note)Verify if this function is supported in an implementation.

	Function	Sub profile 1	Sub profile 2	Remarks
	disconnectPPP()	O	O	(Note)Verify if this function is supported in an implementation.
	getConnectionType()	O	O	(Note)Verify if this function is supported in an implementation.
	isIPConnected()	O	O	
	saveHttpServerFileAs()	-	O	(Note)Verify if this function is supported in an implementation.
	saveHttpServerFile()	-	O	
	sendHttpServerFileAs()	-	-	
	saveFtpServerFileAs()	-	-	
	saveFtpServerFile()	-	-	
	sendFtpServerFileAs()	-	-	
	sendTextMail()	-	-	
	transmitTextDataOverIP()	O	O	(Note)Verify if this function is supported in an implementation.
	setDelayedTransmissionData()	O	O	(Note)Verify if this function is supported in an implementation.
Interaction channel functions-Delayed call functions applicable to BASIC procedure and IP connections				
	getDelayedTransmissionStatus( )	O	O	(Note)Verify if this function is supported in an implementation.
	getDelayedTransmissionResult( )	O	O	(Note)Verify if this function is supported in an implementation.
Interaction channel functions-Function for obtaining line connection status				
	getPrefixNumber()	O	O	(Note)Verify if this function is supported in an implementation.
Interaction channel functions-Communication functions using the mass call reception service				
	vote()	O	O	
Interaction channel functions-Functions for encrypted communication using CAS				
	startCASEncryption()	O	O	
	transmitWithCASEncryption()	O	O	
	endCASEncryption()	O	O	
Interaction channel functions-Functions for communication with public key encryption not using CAS				
	setEncryptionKey()	-	-	

	Function	Sub profile 1	Sub profile 2	Remarks
	beginEncryption()	-	-	
	endEncryption()	-	-	
Operational control functions				
	reloadActiveDocument()	O	O	
	getNPT()	O	O	
	getProgramRelativeTime()	O	O	
	isBeingBroadcast()	O	O	
	lockExecution()	-	-	
	unlockExecution()	-	-	
	lockModuleOnMemory()	O	O	
	unlockModulesOnMemory()	O	O	
	setCachePriority()	O	O	
	getTuningLinkageSource()	-	-	
	getTuningLinkageType()	-	-	
	getLinkSourceServiceStr()	-	-	
	getLinkSourceEventStr()	-	-	
	getIRDID()	O	O	
	getBrowserVersion()	O	O	
	getProgramID()	O	O	
	getActiveDocument()	O	O	
	lockScreen()	O	O	
	unlockScreen()	O	O	
	getBrowserSupport()	O	O	
	launchDocument()	O	O	
	launchDocumentRestricted()	O	O	(Note)Verify if this function is supported in an implementation.
	quitDocument()	O	O	(Note)Verify if this function is supported in an implementation.
	launchExApp()	O	O	(Note)Verify if this function is supported in an implementation.
	getFreeContentsMemory()	O	O	(Note)Verify if this function is supported in an implementation.
	isSupportedMedia()	O	O	
	detectComponent()	O	O	(Note)Verify if this function is supported in an implementation.
	lockModuleOnMemoryEx()	O	O	(Note)Verify if this function is supported in an implementation.

	Function	Sub profile 1	Sub profile 2	Remarks
	unlockModuleOnMemoryEx()	O	O	(Note)Verify if this function is supported in an implementation.
	unlockAllModulesOnMemorys()	O	O	(Note)Verify if this function is supported in an implementation.
	getLockedModuleInfo()	O	O	(Note)Verify if this function is supported in an implementation.
Receiver sound control				
	playRomSound()	O	O	
Timer functions				
	sleep()	O	O	
	setTimeout()	-	-	
	setInterval()	O	O	
	clearTimer()	O	O	
	pauseTimer()	O	O	
	resumeTimer()	O	O	
	setCurrentDateMode()	O	O	
External character functions				
	loadDRCS()	O	O	
	unloadDRCS()	-	-	
Functions for controlling external devices				
	enumPeripherals()	-	-	
	passXMLDocToPeripheral()	-	-	
	getArrayFromPeripheral()	-	-	
Other functions				
	random()	O	O	
	subDate()	O	O	
	addDate()	O	O	
	formatNumber()	O	O	
Subtitle presentation control functions				
	setCCStreamReference()	-	-	
	getCCStreamReference()	-	-	
	setCCDisplayStatus()	O	O	
	getCCDisplayStatus()	O	O	
	getCCLanguageStatus()	O	O	
Directory management functions				
	saveDir()	-	O	
	saveDirAs()	-	O	(Note)Verify if this function is supported in an implementation.
	createDir()	-	-	

	Function	Sub profile 1	Sub profile 2	Remarks
	getParentDirName()	-	O	(Note)Verify if this function is supported in an implementation.
	getDirNames()	-	O	(Note)Verify if this function is supported in an implementation.
	isDirExisting()	-	O	
File management functions				
	saveFile()	-	O	
	saveFileAs()	-	O	(Note)Verify if this function is supported in an implementation.
	getFileNames()	-	O	(Note)Verify if this function is supported in an implementation.
	isFileExisting()	-	O	
File input/output functions				
	writeArray()	-	O	(Note)Verify if this function is supported in an implementation.
	readArray()	-	O	(Note)Verify if this function is supported in an implementation.
Inquiry functions				
	getDirInfo()	-	O	(Note)Verify if this function is supported in an implementation.
	getFileInfo()	-	O	(Note)Verify if this function is supported in an implementation.
	getCarouselInfo()	-	O	(Note)Verify if this function is supported in an implementation.
	getModuleInfo()	-	O	(Note)Verify if this function is supported in an implementation.
	getContentSource()	-	O	(Note)Verify if this function is supported in an implementation.
	getStorageInfo()	-	O	
Data carousel storage functions				
	saveCarouselAs()	-	-	
	saveCarousel()	-	-	
	saveModuleAs()	-	O	(Note)Verify if this function is supported in an implementation.

	Function	Sub profile 1	Sub profile 2	Remarks
	saveModule()	-	O	
	saveResourceAs()	-	O	(Note)Verify if this function is supported in an implementation.
	saveResource()	-	O	
Functions for controlling bookmark area				
	writeBookmarkArray()	O	O	
	readBookmarkArray()	O	O	
	deleteBookmark()	O	O	
	lockBookmark()	O	O	
	unlockBookmark()	O	O	
	getBookmarkInfo()	O	O	

A3-4.7.5 Operation of Navigator pseudo objects

Navigator pseudo objects are operated. Refer to Chapter 3 for what the O and – symbols mean.

A3-4.7.5.1 Operational Scope for Navigator pseudo objects

The following table summarizes operational guidelines on Navigator pseudo objects.

	Function	Sub-profile 1	Sub-profile 2	Remarks
	AppName	O	O	
	AppVersion	O	O	

A3-4.7.5.2 Operational guidelines on Navigator pseudo objects

As a Navigator pseudo object, navigator.appName and navigator.appVersion are allowed to be operated. When a BML-compatible browser receives navigator.appName or navigator.appVersion, the following value is returned.

- The navigator.appName pseudo object returns a string containing the ”BMLBrowser” string.
- What navigator.appVersion pseudo object returns depends on an implementation.

A3-4.8 Restrictions on Description of BML Documents

The restrictions on description of BML documents complies with the convention defined in Section A2-4.6, APPENDIX 2.

A3-4.8.1 Positioning and Restrictions

Any positioning and any restriction on positioning are governed by the W3C Recommendation, CSS 2.0. The additional restriction is provided in Section A3-5.1.

A3-4.8.2 Vertical writing

No vertical writing is operated.

A3-4.8.3 Restrictions on arrangement of elements

Any arrangement of elements and any restriction on arrangement of elements are governed by CSS 2.0, the convention defined in Section A3-4.8.3, APPENDIX 2, and the following additional restrictions.

Provided that an element is configured to overlap with an MNG object whose streamstatus attribute is set to play, no operation that changes how the element is presented is allowed. Provided that an MNG object has a streamstatus attribute whose value is play, no operation that changes the background color of the MNG object is allowed.

A3-4.8.4 Restrictions on elements represented on composition planes of video and still pictures

The restrictions on elements represented on a composition plane of video and still picture comply with the convention defined in Section A3-4.7.4, APPENDIX 2.

A3-4.8.5 Operation of object element and img element

How an individual monomedia that is specified with the object element or the img element is presented is governed by the convention in Section A3-5.1.5. How the input element whose type attribute is set to image is presented is also governed by the convention in Section A3-5.1.5.

The available monomedia type to the src attribute for both the img element and the input element is limited to image/jpeg, image/png, and image/X-arib-png. No operation that modifies a type of a monomedia, that is specified with the src attribute of the img element and the src attribute of the input element, is allowed.

A3-4.8.5.1 Application of remain attribute for each type

The following Table 4-13 summarizes guidelines for applying the remain attribute of the object element.

Table 4-13 Application of the remain attribute

type attribute	Applicability of remain attribute	Remarks
video/X-arib-mpeg2	O	Applicable only when a broadcasting service is referenced
video/X-arib-mpeg1	O	Applicable only when a broadcasting service is referenced
audio/X-arib-mpeg2-aac	Δ	Applicable only when the scheme specification is arib:(PES)
audio/X-arib-aiff	-	
image/jpeg	O	
image/png	-	
image/X-arib-png	-	
image/X-arib-mng	-	
application/X-arib-mpeg2-tts	-	

The following conditions are assumed to operate the remain attribute of the object element.

[Condition 1]

The remain attribute of the object element can be made valid only in the case where the object element is used by fulfilling either of the conditions of (1) to (4) listed below.

(1) The value of the type attribute is video/X-arib-mpeg1 and belongs to the same content group as the object element.

(2) The value of the type attribute is video/X-arib-mpeg2 and belongs to the same content group as the object element.

(3) The value of the type attribute is "image/jpeg". In addition, the values of the width property and the height property are equal to the horizontal and vertical resolution specified by the resolution property of the body element, which is an ancestor element, respectively.

(4) The scheme specified by the data attribute is "arib", the data attribute belongs to the same contents group as the object element, and the value of the type attribute is "audio/X-arib-mpeg2-aac".

Note that, for the purpose of Condition 1, to make the remain attribute valid is assumed as one of the following.

a) To make a BML document specify the value "remain" for the remain attribute.

b) To set the remain attribute of the BMLObjectElement interface of DOM API to true.

c) To let the remain attribute of the object element in a BML document before a transition be applicable to the object element in the BML document after the transition.

When the remain attribute of the object element is made valid based on any other assumption, how the remain attribute is operated depends on an implementation.

[Condition 2]

The object element whose remain attribute is made valid must be a child element of the div element which is the first child element of the body element.

When the remain attribute of the object element is made valid without meeting this condition, the behaviour depends on an implementation.

[Condition 3]

When the div element has at least one object element whose remain attribute is made valid as a child element, the div element must have not more than one child element meeting the (5) condition described below, not more than one child element meeting the (6) condition described below, and not more than one child element meeting the (7) condition described below. Otherwise, the behaviour depends on an implementation.

(5) The object element meeting the (1) or (2) condition described above.

(6) The object element meeting the (3) condition described above.

(7) The object element meeting the (4) condition described above.

[Condition 4]

To make a document transition occur in the case where the remain attribute of the object element satisfying the above condition (3) is made valid, the lockModuleOnMemory() method of a browser pseudo object must have been applied to the resource specified with the data attribute of the concerned object element. Otherwise, the behaviour depends on an implementation.

[Condition 5]

Even if the object element satisfies the condition (3), its remain attribute must not be made valid when the background-image property of the body element which is an ancestor element of the concerned object element has been specified.

Otherwise, the behaviour depends on an implementation.

[Condition 6]

The remain attribute for a still picture that belongs to two or more document groups must not be made valid.

Otherwise, the behaviour depends on an implementation.

A3-4.8.5.2 Operation of attributes responsible for stream presentation

The following Table 4-14 summarizes operational guidelines on the streamposition attribute, the streamstatus attribute, and the streamlooping attribute of an object element.

Table 4-14 Operation of attributes responsible for stream presentation

type attribute	streamposition	streamstatus	streamlooping
video/X-arib-mpeg2	Fixed to 0	play (Note 1)	Fixed to
video/X-arib-mpeg1	Fixed to 0	play (Note 1)	Fixed to
audio/X-arib-mpeg2-aac	Fixed to 0	play, stop (Note 2)	Fixed to
image/jpeg	Fixed to 0	play (Note 1)	Fixed to
image/ X-arib-png	Fixed to 0	play (Note 1)	Fixed to
audio/X-arib-aiff	Fixed to 0	play, stop (Note 2, 3)	Fixed to
image/X-arib-mng	Indicates frame number	play, stop, pause (Note 2)	Fixed to
application/X-arib-mpeg2-tts	Fixed to 0	play, stop (Notes 4, 5)	Fixed to

Note 1: The initial value of streamstatus for an object element referencing media of type attribute which takes ‘play’ only is ‘play’.

Note 2: The initial value of streamstatus for an object element referencing media of type attribute which can take ‘stop’ is ‘stop’. In this case, actually the corresponding sound is not stopped but muted. Note that dynamically changing type attributes and dynamically changing schemas by changing data attributes for sound are not applicable to the object element.

Note 3: When the playback ends, streamstatus is automatically set to “stop”.

Note 4: When the playback ends, streamstatus is automatically set to “pause”.

Note 5: If streamstatus for an object element is pause when the concerned BML document is read and streamstatus for the referenced video object is play, a still picture in a frame near the beginning of the stream is presented. Once streamstatus for the object element has been reconfigured to play, the playback starts at the position immediately following the still picture or begins with the very beginning of the stream.

A3-4.8.5.3 Guideline on MNG presentations

Any presentation of MNG must comply with the convention defined in Section A2-4.8.5.3, APPENDIX 2.

Chapter APPENDIX 3-5 Guidelines on behaviours of BML browsers

A3-5.1 Operational guidelines on presentation

A3-5.1.1 Presentation model

This section defines the two presentation models, an index-based model and a component-based model. Each model must be operated based on the three-plane-based configuration (a video plane, a still picture plane, a text and graphics plane), as defined in Chapter 7, Part 1, Volume 1.

- Index-based model

  This presentation model directly specifies the CLUT data of a text and graphics plane. In an index-based model, the whole area of a screen is used to present a content and no scrolling occurs.

- Component-based model

  This presentation model does not directly specify the CLUT data of a text and graphics plane. In a component-based model, only part of a screen is used to present a content and a displayed area is scrolled to make the whole content available to an end user.

Background colours, characters, and borders of each element other than the body element behave as if they are rendered in a text and graphics plane, and take colours that specified with colour names or RGB values. A browser automatically allocates the specified colours to CLUT of the text and graphics plane. When PNG, MNG, or JPEG is to be presented, it behaves as if it is rendered in a still picture plane. However, no translucent colour is applicable.

A3-5.1.2 Coordinate system for text and graphics plane

The coordinate system for a text and graphics plane must be either the 960x540 coordinate system or the 720x480 coordinate system. Operational guidelines on presenting each model is summarized in the following table.

Presentation		Index-based	Component-based
1	Screen Resolution	HD(960x540) or SD(720x480) is available	HD(960x540) or SD(720x480) is available
2	Effective Presentation Area	Entire screen is used for presentation.	Size of displayed area and how surrounding area is processed depend on an implementation.

A3-5.1.3 Operational guideline on Z-index

Operational guidelines on Z-index comply with Section A2-5.1.1, APPENDIX 2.

Various objects are composed in each presentation model as described below.

Operation of Objects Composition		Index-based	Component-based
1	Composition of video objects and other objects	When a video object acts as the background and JPEG objects are presented in front of the video object in Z order, the video object must be presented in the full screen and not more than four JPEG objects (including a object referencing a JPEG file) may be presented. (Note)	When a video object acts as the background and JPEG objects are presented in front of the video object in Z order, the video object must be presented in the full screen and not more than four JPEG objects (including a object referencing a JPEG file) may be presented. (Note)

Note: Besides the JPEG object, the input element whose type attribute is image and the img element may reference a JPEG file.

No object must be presented in front of a video object. That is, no overlaying in Z order is allowed. No part of a video object must be hidden by any other object.

A2-5.1.4 Presentation plane

Any presentation plane is operated as defined in Section A2-5.1.2, APPENDIX 2.

A2-5.1.4.1 top property and left property

The following table contains the difference from the convention defined in Table 5-2 in APPENDIX 2.

Table 5-1 Available values to top and left properties

type attribute	top and left properties in 960x540 coordinate system	top and left properties in 720x480 coordinate system	
		Index-based	Component-based
video/X-arib-mpeg2	arbitrary number for x coordinate value/y coordinate value	arbitrary even number for x coordinate value/y coordinate value	arbitrary even number for x coordinate value/y coordinate value
video/X-arib-mpeg1	arbitrary number for x coordinate value/y coordinate value	arbitrary even number for x coordinate value/y coordinate value	arbitrary even number for x coordinate value/y coordinate value
image/jpeg	arbitrary number for x coordinate value/y coordinate value	arbitrary even number for x coordinate value/y coordinate value	arbitrary even number for x coordinate value/y coordinate value
image/ X-arib-png	arbitrary number for x coordinate value/y coordinate value	arbitrary even number for x coordinate value/y coordinate value	arbitrary even number for x coordinate value/y coordinate value
image/X-arib-mng	arbitrary number for x coordinate value/y coordinate value	arbitrary even number for x coordinate value/y coordinate value	arbitrary even number for x coordinate value/y coordinate value

A3-5.1.5 Operation of monomedia presentation

This section contains the difference from the convention defined in Section A2-5.1.3, APPENDIX 2.

In each presentation model, a monomedia is allowed to be presented as described below.

Operation of Objects Composition		Index-based	Component-based
1	Presentation of PNG	- When PLTEchunk is not specified, CLUT data specified with the clut property of the body element is used. A text and graphics plane is used for the presentation. - When PLTEchunk is specified, PLTEchunk associated with PNG data is used. In either case, the presentation size is	PLTEchunk associated with PNG data is used. The presentation size is defined based on the coordinate system of the concerned text and graphics plane.

		defined based on the coordinate system of the concerned text and graphics plane. (Note)	
2	Presentation of MNG	CLUT data specified with the clut property of the body element is used. The presentation behaves as if it is rendered in a text and graphics plane. The presentation size is defined based on the coordinate system of the concerned text and graphics plane. PLTEchunk associated with MNG data is not used.	PLTEchunk associated with MNG data is used. The presentation size is defined based on the coordinate system of the concerned text and graphics plane.
3	Presentation of JPEG	The presentation size is defined based on the coordinate system of the concerned text and graphics plane.	The presentation size is defined based on the coordinate system of the concerned text and graphics plane.

Note: Whether or not PNG data including PLTEchunk is used in an index-based model is defined in an operational standard.

Sample behaviours of a text and graphics plane in which a monomedia is presented are provided below.

Sample Behaviours of Presentation		Index-based	Component-based
1	Sample Presentation of JPEG	The image data is scaled two times in a still picture plane while a solid rectangle in the colour (R,G,B,□)=(0,0,0,0) with the same size as the image data is rendered at the same position as the image data in a text and graphics plane.	The image data is scaled two times in a still picture plane while a solid rectangle in the colour (R,G,B,□)=(0,0,0,0) with the same size as the image data is rendered at the same position as the image data in a text and graphics plane.
2	Sample Presentation of PNG	The image data is rendered in a text and graphics plane.	The image data excluding the non-drawing colours is presented in a still picture plane while the corresponding graphic excluding the pixels associated with non-drawing colours is also rendered in the colour (R,G,B,α)=(0,0,0,0) at the same position of the image data in a text and graphics plane.
3	Sample Presentation of Video	A solid rectangle in the colour (R,G,B,α)=(0,0,0,0) with the same size as the video data is rendered at the same position as the video data in a text and graphics plane.	A solid rectangle in the colour (R,G,B,α)=(0,0,0,0) with the same size as the video data is rendered at the same position as the video data in a text and graphics plane.

Note that

(1) The conventions on the width property and the height property contained in Section A2-5.1.3, APPENDIX 2 are applicable to the width attribute and the height attribute, respectively.

(2) The above description is also applied to a monomedia that is obtained through the input element whose type attribute is “image”, and a monomedia obtained through the img element.

A3-5.1.5.1 width and height attributes and width and height properties

The following table contains the difference form the convention provided in Table 5-3 in APPENDIX 2.

Table 5-2 Operation of sizes of scaled video on video planes

Lines (vertical size)	Horizontal Size		Operation per Signal Type				Description
	16:9 signal	4:3 signal (excluding HD)	HD 1080 lines	SD 480 lines	SIF 240 lines	QSIF 120 lines	
1080	1920	1440	O	- (Note)	-	-	HD standard
960	1706	1280	-	O	-	-	SD maximum
810	1440	1080	O	O	-	-	
674	1200	900	O	O	-	-	
540	960	720	O	O	-	-	
480	852	640	-	O	O	-	SD standard SIF maximum
404	720	540	O	O	O	-	
270	480	360	O	O	O	-	
240	426	320	-	O	O	O	SIF standard QSIF maximum
120	212	160	-	O	O	O	QSIF standrad

O : Operated, - : Not operated

When a 4:3 signal is valid and a 16:9 size is specified, a receiver must retain the intended aspect ratio by automatically inserting side panels or performing other adjustments.

Note: In order not to deteriorate the image quality, the maximum vertical scaling ratio must be two. For more information about HD, refer to APPENDIX 2.

Table 5-3 Relation between width/height Properties and Scaling Ratios

	960x540 coordinate system		750x480 coordinate system			
			Index-based		Component-based	
type attribute	width property	height property	width property	height property	width property	height property
video/X-arib-mpeg2	1/2 of the horizontal size in Table 5-2	1/2 of the vertical size in Table 5-2	W*Sx (Note 1)	H*Sy (Note 3)	W*Sx (Note 1)	H*Sy (Note 3)
video/X-arib-mpeg1	1/2 of the horizontal size in Table 5-2	1/2 of the vertical size in Table 5-2	W*Sx (Note 1)	H*Sy (Note 3)	W*Sx (Note 1)	H*Sy (Note 3)

	960x540 coordinate system		750x480 coordinate system			
			Index-based		Component-based	
image/jpeg	auto or W	auto or H	auto or W (Note 2)	auto or H	auto or W (Note 2)	auto or H
image/ X-arib-png	auto or W	auto or H	auto or W	auto or H	auto or W (Note 2)	auto or H
image/X-arib-mng	auto or W	auto or H	auto or W	auto or H	auto or W (Note 2)	auto or H

Note 1: When a computed value is not an even number, W*Sx-1 is valid.
Note 2: When a computed value is not an even number, W-1 is valid.
Note 3: A computed value is consistent with the definition in Table 5-2. When a computed value is not an integer, all digits the right of the decimal point are discarded.
Note 4: The width property and the height property of the input element whose type attribute is image, the img element, and the object element must not contain % nor em. When % or em is specified, the behaviour depends on an implementation.

The available values to the width property of image/jpg is 16 through 960 in the 960x540 coordinate system and 16 through 720 in the 720x480 coordinate system. The available values to the height property of image/jpg is 16 through 540 in the 960x540 coordinate system and 16 through 480 in the 720x480 coordinate system.

The above restrictions on the width property and the height property are the same as the width attribute and the height attribute expect that the auto value is not allowed to be specified.

A3-5.1.6 Guideline for clipping

Clipping in an index-based model is governed by the guidelines provided in Section A2-5.1.4, APPENDIX 2. Clipping in a component-based model is defined in an operational standard.

A3-5.1.7 Available values to CSS2 properties

The available values to each presentation model are listed below.

Operation of CSS		Index-based	Component-based
1	xx-color-indexs specification	Allowed	Not allowed
2	xx-color specification	Not allowed	Allowed (See Section A3-5.1.9)
3	System colours defined in Section 18.2, CSS2	Not allowed	Not allowed

A3-5.1.8 Rules of box model

Any box model is governed by CSS 2.0. However, when an index-based model takes 0 for margin, the convention defined in Section A2-5.1.6, APPENDIX 2 is applied.

Operation of Objects Composition		Index-based	Component-based
1	Specification of position	Only the block element positioned immediately under the body element is allowed. The position and the size of each block element must be specified.	Not required

A3-5.1.9 Operation of colour map data

This section provides operational restrictions in order to make a text and graphics plane available on a hardware device with a 256-colour CLUT in each presentation model.

- Operation of CLUT in an index-based model
    - The default colours of CLUT comply with the default setting defined in APPENDIX 2.
    - Index colours 224 through 255 depend on an implementation.
    - Index colours 17 through 223 may be configured as required.
    - A colour map data is allowed to be operated in an index-based model as defined in Table 5-4.

Table 5-4 Operation of Colour Map Data

Data	Value	Description
clut_type	0	Among the colour spaces, only YCbCr colour space is allowed.
depth	'01'	Limited to 8-bit CLUT (up to 256 colours)
region_flag	0	No region is specified.
start_end_flag	1	Start/end index and some elements of CLUT are specified.
start_index	17 or 128	Defines 207 colours from index number 17 to 223
end_index	223	
alpha	Except 0	αvalue=0 is not allowed

- Operation of CLUT in a component-based model
    - The 17 colours in Table 5-5 are reserved. It is allowed to specify ColorName in a content.
    - Each colour may be allocated index as required provided that only one translucent colour may be allocated to index.
    - In a component-based model, a set of available RGBα may comprise up to 207 colours. That is, 256 (the total colours) - 17 (the fixed colours) - 32 (the colours dependent on an implementation) = 207.

Table 5-5 ColorName and RGB value in Component-based Model

colour-name	R	G	B	α
white	255	255	255	255
black	0	0	0	255
red	255	0	0	255
lime	0	255	0	255
yellow	255	255	0	255
blue	0	0	255	255
fuchsia	255	0	255	255
aqua	0	255	255	255
transparent	0	0	0	0
grey	128	128	128	255
green	0	128	0	255

colour-name	R	G	B	α
maroon	128	0	0	255
navy	0	0	128	255
olive	128	128	0	255
purple	128	0	128	255
silver	192	192	192	255
teal	0	128	128	255

The following restriction is applied to each presentation model.

Operation of CLUT	Index-based model	Component-based model
Restriction on CLUT for text and graphics plane	The CLUT property of the body element is used to specify index colours 17 through 223. Index Colours 0 through 16 must be fixed values. Especially, Index Colour 8 must be a transparent colour and handled as a non-drawing colour.	This CLUT is internally used by a browser and is not allowed to be directly specified through a content. Colours are specified as RGB values.
Restriction on translucent colours	Any object containing translucent colours is not allowed to overlay any object rendered in a text and graphics plane (The object must not be positioned in Z order to be in front of any other object in a text and graphics plane).	No translucent colour is operated. For PNG, tRNS chunk may be used and index8 may be handled as a non-drawing colour.

A3-5.1.10 Operation of remote control keys

A remote control key is operated as defined in Section A2-5.1.8, APPENDIX 2.

A3-5.1.11 Character entry into input element and textarea element

The minimum operational scope that must be supported by remote control keys is shown in Table 5-6.

Table 5-6 Character Entry into input/textarea Element with Remote Control Keys

Remote control key	Application to input element
0,1,2,3,4,5,6,7,8,9	Used to enter numeric characters 0 through 9. May be used for KANA-KANJI conversion
"Back"	Deletes one character (Backspace)

When in a content where a “BACK” key or a numeric key is specified as an access key a focus is placed in the input element or the textarea element, the access key is disabled. In this case, removing the focus from the input element or the textarea element enables the access key.

A3-5.1.12 Operation of Focus

In each presentation model, a focus is operated as described below.

Focus		Index-based model	Component-based model
1	To move a focus	The nav-index property is operated.	The nav-index property is not operated. A receiver is responsible for navigation.

Any receiver must be equipped with a function for presenting an element that has a focus. For more information about enabling or disenabling a function of a currently focused element, refer to Section A3-4.4.1.

When the nav-index property is not operated, a receiver must automatically identify an element that needs a focus to perform a proper navigation. An element that satisfies one of the following conditions needs a focus.

(1) the input element, the select element, the textarea element, and the a element with the href attribute

(2) an element that has one of the following attributes: onclick, onfocus, onblur, onkeydown, or onkeyup

A3-5.2 Guidelines on behaviours of external characters

Behaviours of external characters are governed by the guidelines provided in Section A2-5.2, APPENDIX 2.

A3-5.3 Guidelines on behaviours of DOM

A3-5.3.1 Use of DOM

Any DOM is used based on the convention defined in Section A2-5.3.1, APPENDIX 2.

However, the hasFeature( ) method returns true when the feature attribute is "BML" and the version attribute is a version specified in an operational standard.

A3-5.3.2 Handling of control codes (blank, line break, tab)

The control codes (blank, line break, tab) are handled based on the convention on defined in Section A2-5.3.2, APPENDIX 2.

However, an element that is not the p element, the span element, nor the a element and has a child element containing a control code in a string is governed by the convention provided in Table 5-11, APPENDIX 2.

A3-5.3.3 Generation of DOM tree

This section provides the difference from the description in Section A2-5.3.4, APPENDIX 2.

- Comment node

  Whether or not comment nodes are generated depends on an implementation.

- Behaviours of the pre element

The pre element retains all whitespace characters except leading and trailing whitespace characters, as defined in Section 3.2, “User Agent Conformance”, the W3C Recommendation, “XHTML1.0“ (http://www.w3.org/TR/2000/REC-xhtml1-20000126/ - uaconf).

A3-5.3.4 Restriction on DOM work memory

The following Table 5-7 defines the maximum number of DOM objects allowed to be generated per item.

Table 5-7 Maximum value of DOM objects

Item	Maximum value	Remarks
Number of elements	512	
The total number of nodes	1024	
Size of character string set to NodeValue	1024 (bytes)	
Number of child nodes to be connected to one node	255	
Depth of tree structure	16	The depth of a node immediately under the HTMLElement node is 1. The HTMLElement node is not counted.

A3-5.3.5 Restriction on style description

The restriction provided in Table 5-8 is applied to description of styles.

Table 5-8 Rescriction on Style Description

Item	Maximum number	Remarks
CSS selectors	512	
CSS declarations	4096	

A3-5.4 Operational Guideline on ECMAScript

A3-5.4.1 Operation of script working environment

The restriction provided in Table 5-9 is applied to a script working environment.

Table 5-9 Restriction on script work memory

Item	Maximum value	Remarks
Length of a symbol name character string	255 bytes	
Function arguments	255	
Local variables	255	
Total length of all character strings	131072 bytes	The total length of strings (including evaluated values of string equations, string constants, string variables) and symbol names.

Item	Maximum value	Remarks
Instances of objects	512	Total number of instances of Object, Number, String, Boolean, Array, Date, BinaryTable and, Function.
Properties of one instance of one object	256	Maximum number of instances of Object, Number, String, Boolean, Array, Date, BinaryTable or, Function. For Array, the number of properties that correspond subscripts are excluded.
Elements of one array	1024	
Nest levels with function for invoking	32	Including functions invoked through event handlers.
Total number of properties of all objects	8192	The total number of properties to which the "properties of one instance of one object" restriction applied. Including number of properties of activation object and arguments object. Excluding built-in properties of global object (built-in functions, built-in objects, extended functions for broadcasting, extended objects for broadcasting) and properties of host object.
Global variables	256	
Function definitions	256	Any function is defined globally. Excluding event handlers.
Work memory for ECMAScript	16384 steps	Based on the computing method defined in ANNEX C, APPENDIX 2.

A3-5.4.2 Data type

Any data type is operated as defined in Section A2-5.4.2, APPENDIX 2.

A3-5.4.3 Effects on basic objects caused by data type restrictions

This section contains the difference from Section A2-5.4.3, APPENDIX 2.

A3-5.4.3.1 Generation of Date object

When the new Date() function cannot obtain the current time, the behaviour depends on an implementation.

A3-5.4.4 Operation rule of implementation-dependent portion

The operation rule defined in Section A2-5.4.4, APPENDIX 2 must be obeyed.

A3-5.5 Operational guidelines on extended objects for broadcasting

This section contains the difference from Section A2-5.2, APPENDIX 2.

A3-5.5.1 Operational guideline on BinaryTable class

The restriction shown in Table 5-10 is applied to the BinaryTable object.

Table 5-10 Operational restrictions on BinaryTable object

Item	Maximum Value	Remarks
Number of lines	8192 lines	
Length of record	2048 bytes	For a variable length record, a record length excluding a record length specification is assumed.
Fields per record	64	
Size of BinaryTable object	510 KB	
Comparison conditions	4	

A3-5.6 Guideline on Behaviour of Browser Pseudo Objects

A3-5.6.1 Operation of Ureg

This section contains the difference from Section A2-5.6.1, APPENDIX 2.

- Content identifier

  The available content identifier to Ureg[0] is described in Table 5-11.

Table 5-11 Content Identifier

	Broadcasting	Storing	Communication
Content identifier	Thi is service_id as a hexadecimal character string in the format of "0xXXXX". Whether upper-case or lower-case alphabetic characters are returned depends on an implementation.	Host	Host

Note: For more information about hosts and boundaries between content groups, refer to Section A3-6.1.

A3-5.6.2 Guideline on behaviours of EPG functions

The description in Section A2-5.4.3, APPENDIX 2 is applicable to behaviours of the EPG-related functions.

A3-5.6.3 Operation of non-volatile memory functions

The description in Section A2-5.6.3, APPENDIX 2 is applicable to operations of the non-volatile memory functions.

A3-5.6.4 Operation of subtitle presentation control

Subtitle presentation is controlled as defined in Section A2-5.6.4, APPENDIX 2.

A3-5.6.5 Operation of interaction channel function

This section provides operational guidelines on the extended functions for broadcasting used to control IP connections.

A3-5.6.5.1 Process of IP connection

It is assumed that IP connection is processed in either of the following two ways.

- Automatic connection

  By default, a receiver uses the automatic connection function to establish a connection to an IP network. In an automatic connection process, when functions sending and/or receiving data are specified, a connection to an IP network is established before the concerned data is sent and/or received.

- Script description

  When an automatic connection is not applicable to a content, script description occurs in the content. In this case, verifying which protocol is supported by a receiver, what status is an IP connection, and which preferred line type is specified is responsible for the content. The connectPPP() or connectPPPWithISPParams() function is used to establish the intended connection. No other function is used to establish an IP connection.

By default, a receiver uses an automatic connection function to establish a connection to an IP network. When an automatic connection is not desirable and not used, script description is needed instead of the default setting.

A3-5.6.5.2 Guideline on BML content description for connection/disconnection

- To use the connectPPP() or connectPPPWithISPParams() function to establish a connection, the isIPConnected() function and the getConnectionType() function must have been used to verify the current connection status, the specified preferred line type, and other configurations before the connectPPP() or connectPPPWithISPParams() function is invoked.
- To trigger the connectPPP() or connectPPPWithISPParams() function in a content, it is recommended that a broadcaster that has described the concerned content also describes the disconnectPPP() function. However, it is not necessary to describe the disconnectPPP() function in cases including the case where a transition among document groups occurs. If any trouble, failure, or problem arises during communication (while a connection remains established) in a communication channel, the link must be disconnected.
- Even if a receiver whose specified preferred line type is "Ethernet" is used, the connectPPP() function can be enabled as far as a dial-up connection is available to the receiver and the concerned invoke for a dial-up connection is verified by the end user.
- To disconnect a PPP connection that has been established by the automatic connection function in a receiver, the disconnectPPP() function must require an explicit verification by the end user to be activated.
- The connectPPP() function, the connectPPPWithISPParams() function, and any element with an automatic connection function (e.g. the a element) are governed by an automatic reconnection regulation, regardless of whether the execution of the function or the element involves an end user interaction or is automatically triggered from a content through a timer or others. An automatic reconnection regulation is defined by a broadcaster.
- To operate a content using TCP/IP to establish an interactive communication, it is recommended that an error message is presented to allow an end user to inform a customer center of the error.

- isIPConnected()

The available return values of this function are explained in Table 5-12.

Table 5-12 Return values of isIPConnected()

Return value	Semantics	Description
0	No IP connection has been established.	No IP connection has been established.
1	IP connection has been established with automatic connection function.	The specified IP connection has been established without using connectPPP()/connectPPPWithISPParams().
2	IP connection has been established through connectPPP()/connectPPPWithISPParams().	The connectPPP()/connectPPPWithISPParams() has been used to establish the specified IP connection.
NaN	Failure	No IP connection status is obtained.

The relationship among the return values of isIPConnected(), preferred line types, and connecting protocols is shown in Table 5-13.

Table 5-13 Relationship between Preferred Line Types and isIPConnected() Return Values

Return value	Preferred Line Type	PSTN	ISDN	PIAFS	PDC	PDC-P	DS-CDMA	MC-CDMA	CDMA Cellular Syetem
	Connecting Protocol	PPP							
0 (No IP connection has been established.)		O	O	O	O	O	O	O	O
1 (IP connection has been established with automatic connection function.)		O	O	O	O	O	O	O	O
2 (IP connection has been established through connectPPP()/connectPPPWithISPParams().)		O	O	O	O	O	O	O	O
NaN (Failure)		O	O	O	O	O	O	O	O

Return value	Preferred Line Type	Ethernet		
	Connecting Protocol	PPP/PPPoE	Fixed IP	DHCP
0 (No IP connection has been established.)		O	O	O
1 (IP connection has been established with automatic connection function.)		O	O	O
2 (IP connection has been established through connectPPP()/connectPPPWithISPParams().)		O	-	-
NaN (Failure)		O	O	O

- getConnectionType()

The available return values of this function are explained in Table 5-14.

Table 5-14 Return values of is getConnectionType()

Return value	Semantics	Description
1	PSTN	The preferred line type specification on a receiver has been set to PSTN.
100	ISDN	The preferred line type specification on a receiver has been set to ISDN. However, it is not applicable to the connection via an analogue port of TA.
200	PHS	The preferred line type specification on a receiver has been set to PHS and whether PIAFS 2.0 or PIAFS 2.1 is set was not recognized.
201	PHS (PIAFS2.0)	The preferred line type specification on a receiver has been set to PIAFS 2.0.
202	PHS (PIAFS2.1)	The preferred line type specification on a receiver has been set to PIAFS 2.1.
300	Mobile phone	The preferred line type specification on a receiver has been set to a mobile phone system and which connecting procedure of the concerned carrier is set was not recognized.
301	Mobile phone (PDC)	The preferred line type specification on a receiver has been set to the mobile phone system (PDC).
302	Mobile phone (PDC-P)	The preferred line type specification on a receiver has been set to the mobile phone system (PDC-P).
303	Mobile phone (DS-CDMA)	The preferred line type specification on a receiver has been set to the mobile phone system (DC-CDMA).
304	Mobile phone(MC-CDMA)	The preferred line type specification on a receiver has been set to the mobile phone system (MC-CDMA).
305	Mobile phone (CDMA CelluarSystem)	The preferred line type specification on a receiver has been set to the mobile phone system (CDMA CelluarSystem).
401	Ethernet (PPPoE)	The preferred line type specification on a receiver has been set to Ethernet and PPPoE has been selected in order to obtain IP addresses.
402	Ethernet (Fixed IP)	The preferred line type specification on a receiver has been set to Ethernet and Fixed IP has been selected in order to obtain IP addresses.
403	Ethernet (DHCP)	The preferred line type specification on a receiver has been set to Ethernet and DHCP has been selected in order to obtain IP addresses.
NaN	Failure	No preferred line type specification on a receiver has been obtained.

The relationship among the return values of getConnectionType(), preferred line types, and connecting protocols is shown in Table 5-15.

Table 5-15 Relationship between Preferred Line Types and getConnectionType() Return Values

Return value	Preferred Line Type	PSTN	ISDN	PIAFS	PDC	PDC-P	DS-CDMA	MC-CDMA	CDMA Cellular System
	Connecting Protocol	PPP							
1 (PSTN)		O	-	-	-	-	-	-	-
100 (ISDN)		-	O	-	-	-	-	-	-
200 (PHS)		-	-	O (Note)	-	-	-	-	-
201 (PIAFS2.0)		-	-	O	-	-	-	-	-
202 (PIAFS2.1)		-	-	O	-	-	-	-	-
300 (Mobile Phone)		-	-	-	O (Note)	O (Note)	O (Note)	O (Note)	O (Note)
301 (PDC)		-	-	-	O	-	-	-	-
302 (PDC-P)		-	-	-	-	O	-	-	-
303 (DS-CDMA)		-	-	-	-	-	O	-	-
304 (MC-CDMA)		-	-	-	-	-	-	O	-
305 (CDMA CelluarSystem)		-	-	-	-	-	-	-	O
401 (Ethernet:PPPoE/PPP)		-	-	-	-	-	-	-	-
402 (Ethernet: Fixed IP)		-	-	-	-	-	-	-	-
403 (Ethernet:DHCP)		-	-	-	-	-	-	-	-
NaN		O	O	O	O	O	O	O	O

Return value	Preferred Line Type	Ethernet		
	Connecting Protocol	PPP/PPPoE	Fixed IP	DHCP
1 (PSTN)		-	-	-
100 (ISDN)		-	-	-
200 (PHS)		-	-	-
201 (PIAFS2.0)		-	-	-
202 (PIAFS2.1)		-	-	-
300 (Mobile Phone)		-	-	-
301 (PDC)		-	-	-
302 (PDC-P)		-	-	-
303 (DS-CDMA)		-	-	-
304 (MC-CDMA)		-	-	-
305 (CDMA CelluarSystem)		-	-	-
401 (Ethernet:PPPoE/PPP)		O	-	-
402 (Ethernet: Fixed IP)		-	O	-
403 (Ethernet:DHCP)		-	-	O
NaN		O	O	O

Note: This return value is obtained when a connecting procedure for a mobile phones/PHS system cannot be identified.

- connectPPP()

The available return values of this function are explained in Table 5-16.

Table 5-16 Return values of connectPPP()

Return value	Semantics	Description
1	Success	The preferred line type specification on a receiver has been identified. When the specified preferred line type is PSTN (mobile phone/PHS), dial tones are detected before dialling is performed and a modem negotiation procedure (after PIAFS procedures/connecting procedures specific to each mobile system) is performed to establish the link. Note that a function for detecting dial tones is disenabled in a terminal configuration, dialling waits until three seconds have passed. When the specified preferred line type is Ethernet and PPPoE is specified to be used, PPPoE process occurs before PPP process finishes to establish the link.
-1	Parameter error	Invalid values in arguments of connectPPP().
-3	Time-out occurred	The link had not been established before the time out value for connecting configured in a receiver passed.
-4	No dial tone detected	No dial tone was detected. Note that when a function for detecting dial tones is disenabled in a terminal configuration or the specified preferred line type is Ethernet, this error does not occur.
-5	No carrier detected	No carrier detected Either of the following: No carrier was detected after dialling had been performed and a center response had been received. No center response was received after dialling had been performed. Note that this error does not occur when the specified preferred line type is Ethernet.
-6	Disconnection enforced	One of the following: The connection was disconnected by a center during link establishing process (authenticating/obtaining IP address) after a center response had been received. The connection was disconnected through a receiver. The Back key was pressed during link establishing process.
-7	Modem in use	Other software application uses the modem or a BML browser is using the modem. Note that when a PPP connection established via a high-speed modem is detected, this error does not occur.
-8	Line is busy	The specified line was busy. Note that when the preferred line type specification is Ethernet, this error does not occur.
-100	PPP connection has been established	The specified PPP connection has been established through connectPPP(), connectPPPWithISPParams(), or an automatic connection function.
-200	Receiver has been configured not to use PPP	The preferred line type specification on a receiver has been set to Ethernet and Fixed IP or DHCP has been selected in order to obtain IP addresses.
-301	Outside of the network service range	The preferred line type specification on a receiver has been set to mobile phone /PHS system and the concerned location is outside of the network service range (if this is detectable).

Return value	Semantics	Description
-302	External communication device was not available	The preferred line type specification on a receiver has been set to mobile phone /PHS system and the concerned external communication device is not powered or used for other service (if this is detectable).
NaN	Failure by other causes	A condition used to restrict an automatic retransmission was detected.

The relationship among the return values of connectPPP(), preferred line types, and connecting protocols is shown in Table 5-17.

Table 5-17 Relationship between Preferred Line Types and connectPPP() Return Values

Return value		Preferred Line Type	PSTN	ISDN	PIAFS	PDC	PDC-P	DS-CDMA	MC-CDMA	CDMA Cellular System
		Connecting Protocol	PPP							
1	(Success)		O	O	O	O	O	O	O	O
-1	(Parameter error)		O	O	O	O	O	O	O	O
-3	(Time-out occurred)		O	O	O	O	O	O	-	-
-4	(No dial tone detected)		O	-	O	O	O	O	-	-
-5	(No carrier detected)		O	-	O	O	O	O	-	-
-6	(Disconnection enforced)		O	O	O	O	O	O	-	-
-7	(Modem in use)		O	O	O	O	O	O	O	O
-8	(Line is busy)		O	O	O	O	O	O	O	O
-100	(PPP connection has been established)		O	O	O	O	O	O	O	O
-200	(Receiver has been configured not to use PPP)		-	-	-	-	-	-	-	-
-301	(Outside of the network service range)		-	-	-	O	O	O	-	-
-302	(External communication device was not available)		-	-	O	O	O	O	-	-
NaN	(Failure by other causes)		O	O	O	O	O	O	O	O

Return value		Preferred Line Type	Ethernet		
		Connecting Protocol	PPP/PPPoE	Fixed IP	DHCP
1	(Success)		O	-	-
-1	(Parameter error)		O	O	O
-3	(Time-out occurred)		O	-	-
-4	(No dial tone detected)			-	-
-5	(No carrier detected)			-	-
-6	(Disconnection enforced)		O	-	-

Return value	Preferred Line Type	Ethernet		
	Connecting Protocol	PPP/PPPoE	Fixed IP	DHCP
-7 (Modem in use)		O (Note)	O (Note)	O (Note)
-8 (Line is busy)			-	-
-100 (PPP connection has been established)		O	-	-
-200 (Receiver has been configured not to use PPP)			O	O
-301 (Outside of the network service range)			-	-
-302 (External communication device was not available)			-	-
NaN (Failure by other causes)		O	O	O

Note: This return value is not required to be returned when a receiver is capable of establishing an IP connection via Ethernet and a connection via modem at the same time.

- connectPPPWithISPParams()

The available return values of this function are explained in Table 5-18.

Table 5-18 Return values of connectPPPWithISPParams()

Return value	Semantics	Description
1	Success	The preferred line type specification on a receiver has been identified. When the specified preferred line type is PSTN (mobile phone/PHS), dial tones are detected before dialling is performed and a modem negotiation procedure (after PIAFS procedures/connecting procedures specific to each mobile system) is performed to establish the link. Note that a function for detecting dial tones is disenabled in a terminal configuration, dialling waits until three seconds have passed. When the specified preferred line type is Ethernet and PPPoE is specified to be used, PPPoE process occurs before PPP process finishes to establish the link.
-1	Parameter error	The ISP connecting configuration on a receiver contains invalid values.
-3	Time-out occurred	The link had not been established before the time out value for connecting configured in a receiver passed.
-4	No dial tone detected	No dial tone was detected. Note that when a function for detecting dial tones is disenabled in a terminal configuration or the specified preferred line type is Ethernet, this error does not occur.
-5	No carrier detected	No carrier was detected after dialling had been performed and a center response had been received. No center response was received after dialling had been performed. Note that this error does not occur when the specified preferred line type is Ethernet.
-6	Disconnection enforced	One of the following: The connection was disconnected by a center during link establishing process (authenticating/obtaining IP address) after a center response had been received. The connection was disconnected through a receiver. The Back key was pressed during link establishing process.

Return value	Semantics	Description
-7	Modem in use	Other software application uses the modem or a BML browser is using the modem. Note that when a PPP connection established via a high-speed modem is detected, this error does not occur.
-8	Line is busy	The specified line was busy. Note that when the preferred line type specification is Ethernet, this return error does not occur.
-100	PPP connection has been established	The specified PPP connection has been established through connectPPP(), connectPPPWithISPParams(), or an automatic connection function.
-200	Receiver has been configured not to use PPP	The preferred line type specification on a receiver has been set to Ethernet and Fixed IP or DHCP has been selected in order to obtain IP addresses.
-301	Outside of the network service range	The preferred line type specification on a receiver has been set to mobile phone /PHS system and the concerned location is outside of the network service range.
-302	External communication device was not available	The preferred line type specification on a receiver has been set to mobile phone /PHS system and the concerned external communication device is not powered or used for other service.
NaN	Failure by other causes	Failure by other causes

The relationship among the return values of connectPPPWithISPParams(), preferred line types, and connecting protocols is shown in Table 5-19.

Table 5-19 Relationship between Preferred Line Types and connectPPPWithISPParams()

Return value	Preferred Line Type	PSTN	ISDN	PIAFS	PDC	PDC-P	DS-CDMA	MC-CDMA	CDMA Cellular System
	Connecting Protocol	PPP							
1 (Success)		O	O	O	O	O	O	O	O
-1 (Parameter error)		O	O	O	O	O	O	O	O
-3 (Time-out occurred)		O	O	O	O	O	O	-	-
-4 (No dial tone detected)		O	-	O	O	O	O	-	-
-5 (No carrier detected)		O	-	O	O	O	O	-	-
-6 (Disconnection enforced)		O	O	O	O	O	O	-	-
-7 (Modem in use)		O	O	O	O	O	O	O	O
-8 (Line is busy)		O	O	O	O	O	O	O	O
-100 (PPP connection has been established)		O	O	O	O	O	O	O	O
-200 (Receiver has been configured not to use PPP)		-	-	-	-	-	-	-	-
-301 (Outside of the network service range)		-	-	-	O	O	O	-	-

Return value	Preferred Line Type	PSTN	ISDN	PIAFS	PDC	PDC-P	DS-CDMA	MC-CDMA	CDMA Cellular System
	Connecting Protocol	PPP							
-302 (External communication device was not available)		-	-	O	O	O	O	-	-
NaN (Failure by other causes)		O	O	O	O	O	O	O	O

Return value	Preferred Line Type	Ethernet		
	Connecting Protocol	PPP/PPPoE	Fixed IP	DHCP
1 (Success)		O	-	-
-1 (Parameter error)		O	O	O
-3 (Time-out occurred)		O	-	-
-4 (No dial tone detected)			-	-
-5 (No carrier detected)			-	-
-6 (Disconnection enforced)		O	-	-
-7 (Modem in use)		O (Note)	O (Note)	O (Note)
-8 (Line is busy)			-	-
-100 (PPP connection has been established)		O	-	-
-200 (Receiver has been configured not to use PPP)			O	O
-301 (Outside of the network service range)			-	-
-302 (External communication device was not available)			-	-
NaN (Failure by other causes)		O	O	O

Note: This return value is not required to be returend when a receiver is capable of establishing an IP connection via Ethernet and a connection via modem at the same time.

- setDelayedTransmissionData(), getDelayedTransmissionStatus(), getDelayedTransmissionResult()

The structure of a management area for the setDelayedTransmissionData(), getDelayedTransmissionStatus() and getDelayedTransmissionResult() is described below.

	Element	Description	Size (in bytes)	Format
1	regname	Registration name	256	EUC-JP
2	uri	URI representing the location to which the data is sent	256	Alphanumerics
3	start_time	Time to start the data transmission	12	YYYYMMDDhhmm
4	end_time	Time to end the data transmission	12	YYYYMMDDhhmm
5	strtext	Text data to be sent	4096	Character code specified with charset
6	-	Time at which the data is sent	12	YYYYMMDDhhmm
7	-	Result code of the data transmission. A return value of transmitTextdataOverIP() is obtained as a result code.	4	Alphanumerics
8	-	Other information (depending on a receiver)	472	

	Element	Description	Size (in bytes)	Format
9	-	Flag indicating whether a BASIC-based or IP-based function. The operation depends on a receiver and is not defined here.	1	
		Total	5121	

The size of a whole management area for setDelayedTransmissionData(), getDelayedTransmissionStatus() and getDelayedTransmissionResult() must be 51210 bytes and contain up to 10 delayed calls. If no free area exists when the setDelayedTransmissionData() function attempts to register a delayed call, the delayed call that has been sent and whose sent time is the earliest is deleted first.

- transmitTextDataOverIP()

The available return values of this function are explained in Table 5-20.

Table 5-20 Return value (Array[0]) of transmitTextDataOverIP()

Return value		Semantics	Description
Array[0]	1	Success	The text argument is successfully sent to the internet resource specified with the uri argument and the response is received.
	-1	Parameter error	Arguments contains errors.
	-2	Line was disconnected during transfer.	The line was disconnected while the text data is sent.
	-3	Time-out occurred.	The specified period of time has passed without the completion of the data transmittion and/or the response reception.
	-300	Fails to establish an automatic connection	An automatic connection was not established due to an error and the data was not sent to the internet resource specified with the uri argument. If an error occurs while a physical line connection, PPP, and/or TCP is processed, this return value is returned.
	-400	Fails to map names using DNS	The host name specified with the uri argument was not resolved to an IP address.
	-500	Fails to process TLS-based operation	A TLS operation involving an HTTPS connection was not completed.
	NaN	Failure by other causes	A condition used to restrict an automatic retransmission was detected.

A3-5.6.5.3 Process of Disconnection

An established link is disconnected in either of the following two ways.

- Disconnection caused by an expiring inactivity timer

  A disconnection can be caused by an inactivity timer that expires regardless of the fact that the connection has been established by either an automatic connection function or a script description. When a connection has been established by an automatic connection function or the connectPPPWithISPParams() function without idleTime parameter, an inactivity

timer configuration on a receiver is referenced. When the connectPPP() function has established a connection, an inactivity timer value specified with the attribute is referenced.

- Disconnection caused by a script description

  When a connection has been established by the connectPPP() or connectPPPWithISPParams(), the connection can be disconnected by using a script description. When a connection has been established by an automatic connection, if the end user gives an explicit verification, the connection can be disconnected by using a script description

- disconnectPPP()

The available return values of this function are explained in Table 5-21.

Table 5-21 Return values of disconnectPPP()

Return value	Semantics	Description
1	Success	The protocol that has been used to established the connection through connectPPP()/connectPPPWithISPParams() was successfully terminated and the occupied line was successfully released.
-1	No PPP connection has been established	The specified line is not occupied.
-200	Receiver has been configured not to use PPP	The specified preferred line type is Ethernet and the specified IP address type to be obtained is Fixed IP or DHCP.
NaN	Failure	The protocol that has been used to established the PPP connection was not successfully terminated and the occupied line was not successfully released.

The relationship among the return values of disconnectPPP(, preferred line types, and connecting protocols is shown in Table 5-22.

Table 5-22 Relationship between Preferred Line Types and disconnectPPP() Return Values

Return value		Preferred Line Type	PSTN	ISDN	PIAFS	PDC	PDC-P	DS-CDMA	MC-CDMA	CDMA Cellular System
		Connecting Protocol	PPP							
1	(Success)		O	O	O	O	O	O	O	O
-1	(No PPP connection has been established)		O	O	O	O	O	O	O	O
-200	(Receiver has been configured not to use PPP)		-	-	-	-	-	-	-	-
NaN	(Failure)		O	O	O	O	O	O	O	O

Return value	Preferred Line Type	Ethernet		
	Connecting Protocol	PPP/PPPoE	Fixed IP	DHCP
1 (Success)		O	-	-
-1 (No PPP connection has been established)		O	-	-
-200 (Receiver has been configured not to use PPP)		-	O	O
NaN (Failure)		O	O	O

A3-5.6.5.4 Guideline on a telephone number specified in connect ()

The valid characters for a telephone number are shown below.

Telephone line	Valid characters
Pulse	0 ~ 9, ”(“, ”)”, ”-“, ”T”(The character string of the rest is outputted with tone), “,”(pause designation)
Tone	0 ~ 9,”(“,”)”,”-“,”*”,”#”,“,”(pause designation)

When a specified string includes invalid characters, the telephone number is assumed to be a string that does not contain the invalid characters. A string specified as a telephone number must be up to 30 bytes, including invalid characters, if any. When a specified string is more than 30 bytes, NaN (failure) is returned.

A3-5.6.5.5 Guideline on a name server specified in connect ()

A string specified in the argument of connectPPP() as a primary name server or a secondary name server must be a valid IPv4 dotted-decimal string. To omit a name server specification, an empty string is used.

A3-5.6.5.6 Guideline on sending/receiving text data

To use the transmitTextDataOverIP() or setDelayedTransmissionData() function to send or receive text data, the following must be obeyed.

- The maximum text data size that is allowed to sent with a single execution of the function is 4096 bytes.
- The available character encoding specifications to a string to be sent or received through the function are EUC-JP and Shift-JIS.

A3-5.6.5.7 Congestion Avoidance

When an event program is expected to cause congestion, a delayed call or a time-specified call should be used to avoid congestion.

A delayed call is configured by using random numbers, the timer function, a specified time, any combination of them, or all of them.

A time-specified call is configured by setteing a time at which data for an interaction channel service is sent independently of data for an event program. In this case, no data that requires an end user to confirm the sent or received data must be sent.

A3-5.6.6 Guideline on extended functions for broadcasting used to control ISP connecting information

A3-5.6.6.1 Control of ISP connecting information

The following defines how ISP connecting information is controlled.

- Creation of ISP connecting information

  A new set of ISP connecting information may be created either by using a receiver's function or through a BML content provided that no value for ISP connecting information has been configured.

- Deletion of ISP connecting information

  A deletion operation of ISP connecting information is applicable provided that ISP connecting information has been configured in a receiver and the deletion of the information has been confirmed by an end user. This deletion operation must be performed only by using a receiver's function.

- Modification of ISP connecting information

  A modification operation of ISP connecting information is applicable provided that ISP connecting information has been configured in a receiver and the modification of the information has been confirmed by an end user. A user ID specification or other elements are allowed to be modified either by using a receiver's function or through a BML content. However, only a broadcaster that has described a BML content is allowed to use the BML content to modify ISP connecting information.

A3-5.6.6.2 Guideline on configuration status of ISP connecting information

The following defines the three configuration statuses of ISP connecting information.

ISP configuration status value

1: not configured (no ID nor password has been configured)

2: configured (permanent ID and password has been configured)

3: temporarily configured (temporary ID and password has been configured)

Note: Once a BML content or a receiver's function has been used to configure ISP connecting information, a receiver retains the corresponding ISP configuration status value.

One of the three values is obtained by the getISPParams() function in Array[9] as a current status of configured parameters. The value, 2, or the value, 3 may be contained in the status argument of the setISPParams() function. No "not configured" configuration status can be obtained by using setISPParams().

The following Figure 5-1 and Table 5-23 shows how a configuration operation of ISP connecting information using a receiver's function or through a BML content makes the configuration status of ISP connecting information transit.

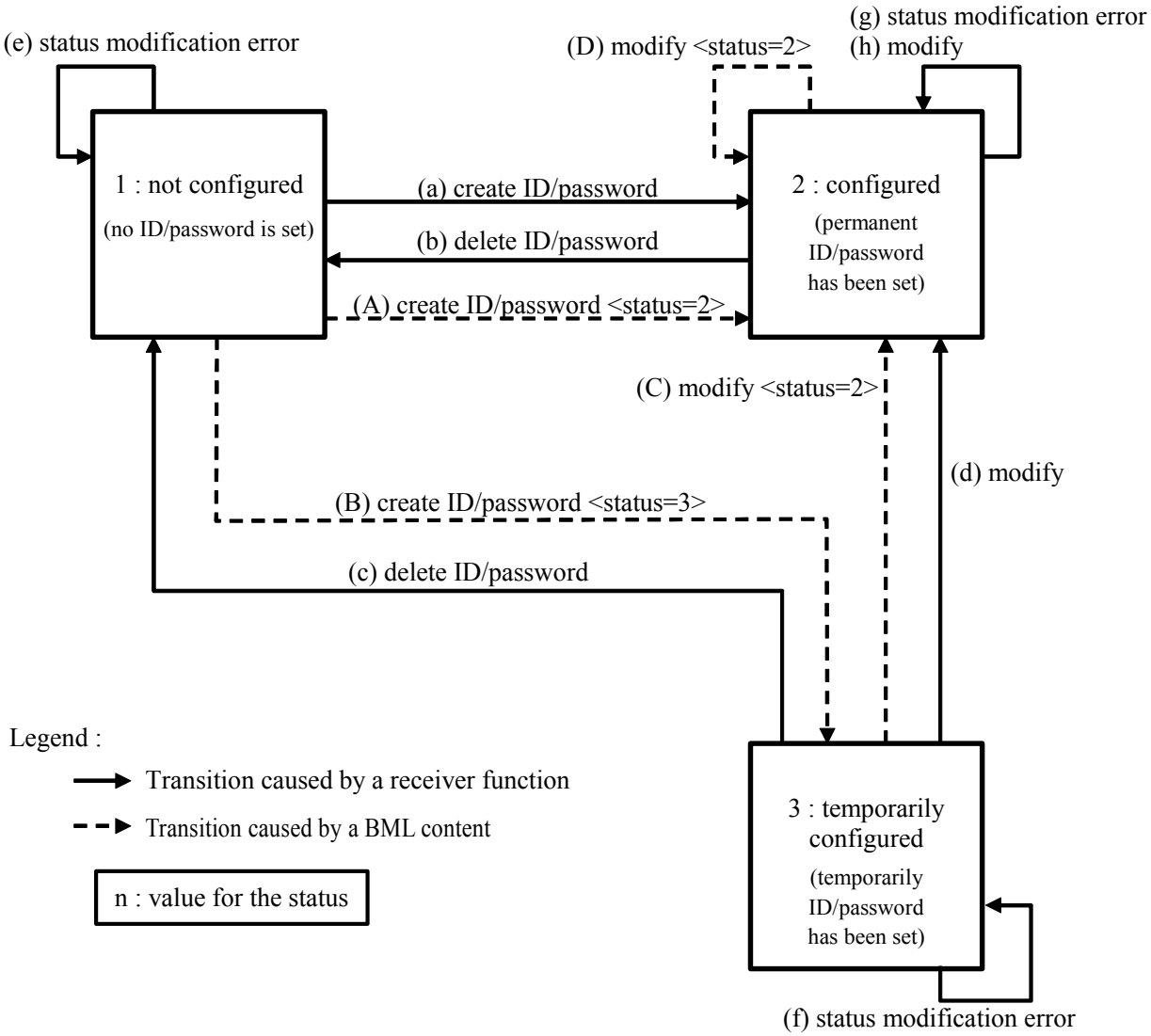


Figure 5-1 Transition of ISP connecting information configuration status

Table 5-23 Transition Status

	Transition Status	Transition invoked using	Status Before Transition status value obtained by getISPPatams()	Status After Transition
(a)	Created using receiver UI	receiver UI	1: not configured	2: configured
(b)	Deleted using receiver UI	receiver UI	2: configured	1: not configured
(c)	Deleted using receiver UI	receiver UI	3: temporarily configured	1: not configured
(d)	Modified using receiver UI	receiver UI	3: temporarily configured	2: configured
(e)	Status modification error	receiver's function	1: not configured	1: not configured
(f)	Status modification error	receiver's function	3: temporarily configured	3: temporarily configured

	Transition Status	Transition invoked using	Status Before Transition status value obtained by getISPPatams()	Status After Transition
(g)	Status modification error	receiver's function	2: configured	2: configured
(h)	Modified using receiver UI	receiver UI	2: configured	2: configured
(A)	Created through BML content	BML content	1: not configured	2: configured
(B)	Temporarily configured through BML content	BML content	1: not configured	3: temporarily configured
(C)	Modified through BML content	BML content	3: temporarily configured	2: configured
(D)	Modified through BML content	BML content	2: configured	2: configured

A3-5.6.6.3 Guideline on BML content description used to configure ISP connecting information

The following provides guidelines for describing a BML content to use extended functions for broadcasting in order to manipulate ISP connecting information.

- When a temporary ID and a temporary password has been configured, it is recommended that the ID and the password are not allowed to be presented through a BML content to prevent an unauthorized access.
- When an end user attempts to enter values through a BML content in order to create a new ISP connecting configuration or modify an existing configuration, whether or not the entered values are valid should be verified in a BML content. For example, a BML content should require an end user to enter a combination of numbers that is contained in a written agreement that has been mailed from the concerned ISP.
- When the setISPParams() function is executed, a message prompting an end user to confirm the modification should be presented.

A3-5.6.7 Channel status after execution of interaction channel function

A channel status is either "free" or "busy". Each channel status transits depending on a result (return value) obtained by an execution of a function. This transition concept is outlined in Figure 5-2.

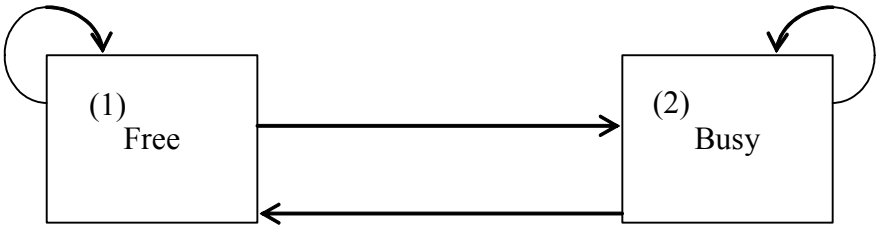


Figure 5-2 Transition of Channel Status

The following Table 5-24 describes how an execution of a function makes a channel status transit.

Table 5-24 Channel Statuses before and after Execution of Function

Extended function for broadcasting	Return value		Channel Status Status Number (Refer to Figure 5-2) Before Execution of Function	 After Execution of Function	Channel release operation is required	Remarks
connectPPP()	1	Success	(1)	(2)		
	-1	Parameter error	-	-		
	-3	Timeout occurred	(1)	(1)	O	
	-4	No dial tone detected	(1)	(1)		
	-5	No carrier detected	(1)	(1)	O	
	-6	Disconnection enforced	(1)	(1)	O	
	-7	Modem in use (Note 1)	(2) (BASIC)	(2) (BASIC)		During communication using BASIC-based procedure
			(2) (CAS)	(2) (CAS)		During communication using CAS
	-8	Line is busy	(1)	(1)		
	-100	PPP connection has been established	(2)	(2)		Including automatic connection status (Note 2)
	-200	Receiver has been configured not to use PPP	(1)	(1)		
			(2)	(2)		
	-301	Outside of the network service range	(1)	(1)		
	-302	External communication device was not available	(1)	(1)		
	NaN	Failure by other causes	(1)	(1)	O	
connectPPPWithISPParams()	1	Success	(1)	(2)		
	-1	Parameter error	-	-		
	-3	Timeout occurred	(1)	(1)	O	
	-4	No dial tone detected	(1)	(1)		
	-5	No carrier detected	(1)	(1)	O	
	-6	Disconnection enforced	(1)	(1)	O	
	-7	Modem in use (Note 1)	(2) (BASIC)	(2) (BASIC)		During communication using BASIC – based procedure
			(2) (CAS)	(2) (CAS)		During communication using CAS
	-8	Line is busy	(1)	(1)		
	-100	PPP connection has been established	(2)	(2)		Including automatic connection status (Note 2)
	-200	Receiver has been configured not to use PPP	(1)	(1)		
			(2)	(2)		

Extended function for broadcasting	Return value		Channel Status		Channel release operation is required	Remarks
			Status Number (Refer to Figure 5-2)			
			Before Execution of Function	After Execution of Function		
	-301	Outside of the network service range	(1)	(1)		
	-302	External communication device was not available	(1)	(1)		
	NaN	Failure by other causes	(1)	(1)	O	
disconnect PPP()	1	Success	(2)	(1)	O	
	-1	No connection has been established	(1)	(1)		Connection has not been established
	-200	Receiver has been configured not to use PPP	(1)	(1)		
			(2)	(2)		
	NaN	Failure	(1)	(1)	O	

Note 1: This return value is applicable for a communication established by using BASIC-based protocols.
Note 2: A receiver must be responsible for controlling a status of an automatic connection.

A3-5.6.8 Operation of operational control functions

The description in Section A2-5.6.7, APPENDIX 2 is applicable to operations of the operational control functions, excluding the getActiveDoccument() function.

- getActiveDocument()

  Any URI for a return value for a BML document transmitted as part of contents to be broadcast in real time must be an abbreviated form (a relative path that begins with "/". Any URI for a return value for a stored BML document or a BML document to be broadcast not in real time must be an absolute path.

A3-5.6.9 Operation of receiver build-in sound control

The description in Section A2-5.6.8, APPENDIX 2 is applicable to operations of receiver build-in sound control.

A3-5.6.10 Operation of timer function

The description in Section A2-5.6.9, APPENDIX 2 is applicable to operations of a timer function.

A3-5.6.11 Operation of external character function

The description in Section A2-5.6.10, APPENDIX 2 is applicable to operations of an external character function.

A3-5.6.12 Operation of Directory Management Functions, File Management Functions, and Data Carousel Storage Functions

When the "Back" key is pressed during an execution of saveDir(), saveFile(), saveModule(), or saveResource(), the save operation aborts and a value representing "Failure" is returned. When a save operation aborts, any file and directory that the operation attempted to save is deleted.

A directory specified to be copied through the saveDir() function must be a directory under the directory in which the BML document invoking the function resides. A file specified to be copied through the saveFile() function must be a file under the directory in which the BML document invoking the function resides.

A3-5.6.13 Operation of module lock among components

The lockModuleOnMemory() function is applicable only to locking a module that is transmitted through the same component as the currently displayed BML document. The lockModuleOnMemoryEx() function is applicable to locking a module that is transmitted through the same service, including the module mentioned above, as the currently displayed BML document. As such, once a module has been locked by lockModuleOnMemoryEx() in a content memory, the module stays in the content memory unless a service other than the currently presented service is selected or an update to the data event of the carousel transmitting the currently displayed BML document is detected.

A3-5.7 Operation of link element

A3-5.7.1 Operation of link element for external style sheet specification

The available attribute values to the link element to specify an external style sheet to be imported are listed in Table 5-25.

Table 5-25 External Style Sheet Specification

rel attribute value	type attribute value	href attribute value
Stylesheet	text/css	URI of an external style sheet to be imported

A3-5.8 Other Restrictions

For the purpose of scalability, the following keywords and object names, which are operated in JavaScript, are regarded as reserved. If any of the following reserved words is used as a variable name or other, it is not assured that no unexpected behavior occurs.

ECMA-262 2nd (BML1.0) keywords

break	for	new	var
continue	function	return	void
delete	if	this	while
else	in	typeof	with

ECMA-262 2nd (BML1.0) reserved words

abstract	do	import	short
boolean	double	instanceof	static
byte	enum	int	super
case	export	interface	switch
catch	extends	long	synchronized

char	final	native	throw
class	finally	package	throws
const	float	private	transient
debugger	goto	protected	try
default	implements	public	volatile

ECMA-262 3rd keywords

break	else	new	var
case	finally	return	void
catch	for	switch	while
continue	function	this	with
default	if	throw	
delete	in	try	
do	instanceof	typeof	

ECMA-262 3rd reserved words

abstract	enum	int	short
boolean	export	interface	static
byte	extends	long	super
char	final	native	synchronized
class	float	package	throws
const	goto	private	transient
debugger	implements	protected	volatile
double	import	public	

Objects

	STD-B24	ECMA-262 3rd	Core JavaScript 1.5	Client JavaScript 1.3
Built-in	Object	Object	Object	Object
	Function	Function	Function	Function
	Array	Array	Array	Array
	String	String	String	String
	Boolean	Boolean	Boolean	Boolean
	Number	Number	Number	Number
	Date	Date	Date	Date
	Math	Math	Math	Math
		RegExp	RegExp	RegExp
		Error		
		EvalError		
		RangeError		
		ReferenceError		
		SyntaxError		
		TypeError		
		URIError		
			JavaArray	JavaArray
			JavaClass	JavaClass
			JavaObject	JavaObject
			JavaPackage	JavaPackage
			netscape	netscape
			Packages	Packages

	STD-B24	ECMA-262 3rd	Core JavaScript 1.5	Client JavaScript 1.3
			sun	sun
Host object (DOM/ extended/pseudo/host)	document			document
	CSVTable			
	BinaryTable			
	browser			
	XMLDoc			
				Anchor
				Applet
				Area
				Button
				Checkbox
				event
				FileUpload
				Form
				Frame
				Hidden
				History
				Image
				Layer
				Link
				Location
				MimeType
	navigator			navigator
				Password
				Plugin
				Radio
				Reset
				screen
				Select
				Style
				Submit
				Text
				Textarea
				window

Top-Level Properties and Functions

	STD-B24 (Operation)	ECMA-262 3rd	Core JavaScript 1.5	Client JavaScript 1.3
	NaN	NaN	NaN	NaN
		Infinity	Infinity	Infinity
		undefined	undefined	undefined
		eval()	eval()	eval()
	parseInt()	parseInt()	parseInt()	parseInt()
		parseFloat()	parseFloat()	parseFloat()
		escape()	escape()	escape()
		unescape()	unescape()	unescape()
	isNaN()	isNaN()	isNaN()	isNaN()

	STD-B24 (Operation)	ECMA-262 3rd	Core JavaScript 1.5	Client JavaScript 1.3
		isFinite()	isFinite()	isFinite()
		decodeURI()	decodeURI()	
		decodeURIComponent()	decodeURIComponent()	
		encodeURI()	encodeURI()	
		encodeURIComponent()	encodeURIComponent()	

* It is not required but recommended that the properties and functions in italics are regarded as reserved.

Each object is associated with functions converting to the basic type (Number(obj), String(obj)).

A3-5.8.1 Size of monomedia data

The size of monomedia data is restricted as shown in Table 5-26.

Table 5-26 Size of Monomedia Data

Item	Maximum Value	Remarks
Buffer for monomedia, data size before decoding	1024 KB	

A3-5.9 Operation of Cookies

A3-5.9.1 Version of Cookies

Any Cookie is operated as defined in RFC2965. For the purpose of interoperability with existing HTTP servers, this section provides the additional operational convention.

A3-5.9.1.1 Operation of response headers

- Any receiver must interpret a Set-Cookie response header. Whether and how a Set-Cookie is interpreted depend on an implementation.
- The following parameters must be interpreted by any receiver.

Attribute	Inclusion in response header by server	Description
NAME=VALUE	Required	The substance of Cookie. How double quotes ("), spaces, tabs, and line breaks in the VALUE string are interpreted depends on a receiver.
domain=DOMAIN	Optional	The name of a domain for which the Cookie is valid.
path=PATH	Optional	The path to which the Cookie is applied.
secure	Optional	A Cookie with the secure attribute specified is sent only when a communication with a host is secured (e.g. when connected with a HTTPS server).
expires=DATE	Optional	Expiration date. The format of the DATE attribute must be " Wdy,dd Mth yyyy hh:mm:ss GMT" or " Wdy,dd-Mth-yyyy hh:mm:ss GMT".

- The semicolon, ";" must be interpreted as a delimiter between attributes. Whether or not the comma, "," is interpreted depends on an implementation.
- A single response header contains not more than one Cookie.

A3-5.9.1.2 Operation of request headers

- NAME=VALUE pairs are included in a HTTP request header to be sent to a request host.

  For example,

  Cookie: NAME1=VALUE1;_ NAME2=VALUE2: _NAME3=VALUE3; - - -

- A receiver needs not specify $Version, $Path, nor $Domain in a request header.

A3-5.9.1.3 Operation of cache control

Whether and how a cache control field (Cache-Control or Cache-Control2) of a response header is interpreted depend on an implementation.

A3-5.9.2 Function enabling Cookies

It is recommended that a receiver is equipped with a function that allows an end user to enable or disable Cookies.

A3-5.9.3 Restrictions on implementation of Cookies

- The maximum length of a string contained in a Set-Cookie header is 4096 bytes, excluding the leading string part "Set-Cookie:".
- The number of Cookies that must be retained in a receiver and the maximum number of Cookies per host are defined in an operational standard.

A3-5.9.4 Reference of Cookie from DOM

To reference or write a Cookie in a procedure description language, document.cookie is used.

A3-5.9.5 Reference of Cookie in broadcast/communication/stored content

- Among the contents, only contents that transmitted via interactive communication networks are capable of referencing Cookies. No stored content nor content broadcast in real time reference a Cookie.

- To share information among contents received in real time, contents stored from a broadcasting service, and contents transmitted via interactive communication networks, Greg is used.

  Note) When a mechanism that uses non-volatile memory to share information, other than Greg working with RAM, is needed, the mechanism is defined in an operational standard.

Chapter APPENDIX 3-6 Guidelines on Contents Transmission and Namespace

A3-6.1 Scope Mapping to Transmission System

A BML document switching operation is performed when:

- the a element in a content is used to switch documents.
- the launchDocument( ) function is used in a script of a content.
- other service than the currently selected service is selected.
- a signal that forces a document switching is received.
- the submit attribute of the form element represents a document switching.

If a scope identifier representing a scope level of a content becomes invalid after the above switching operation, the corresponding content gets out of the scope. The following three sections define scope mapping for contents received in real time, contents transmitted via interactive communication networks associated with broadcasting services, and stored contents.

A3-6.1.1 Scope mapping for contents received in real time

For a content received in real time, scopes, mapping among scope identifiers, and transmission signals is operated as defined in Table 6-1.

Table 6-1 Transmission and scope mapping

Scope	Scope identifier	Corresponding transmission signal
Within document	Document scope identifier	moduleId, resourceName
Within document group	Document group scope identifier	data_event_id
Within content	Contents scope identifier	component_tag
Within content group	Content group scope identifier	service_id

Figure 6-1 shows an example in which content group A has one entry content and two contents, and content group B has one content being broadcasted. The following explains this transmision example using the numbers in Figure 6-1.

A tuning operation selects a given content to be started. In the example below, content 2 begins.

[1] When documents are switched within data_event_id in content 2, the operating scope is defined as “between the documents” and “within the document group”.

[2] When a different within data_event_id specification from the current specification is used to switch documents in content 2, the operating scope is defined as “between the document groups” and “within the content”.

[3] When data_event_id is switched, content 2 is forced to terminnate by the return_to_entry_flag signal, and the entry content starts, the operating scope is defined as “between the contents” and “within the content group”.

[4] When the a element or the launchDocument( ) function in the document causes the entry content to switch to content 2, the entry content terminates and content 2 starts. This operating scope is defined as “between contents” and “within the content group”.

[5] When an end user’ s operation causes content2 to switch to another content group B, content 2 terminates and content 1 of the group starts. This operating scope is defined as “out of the content group”.

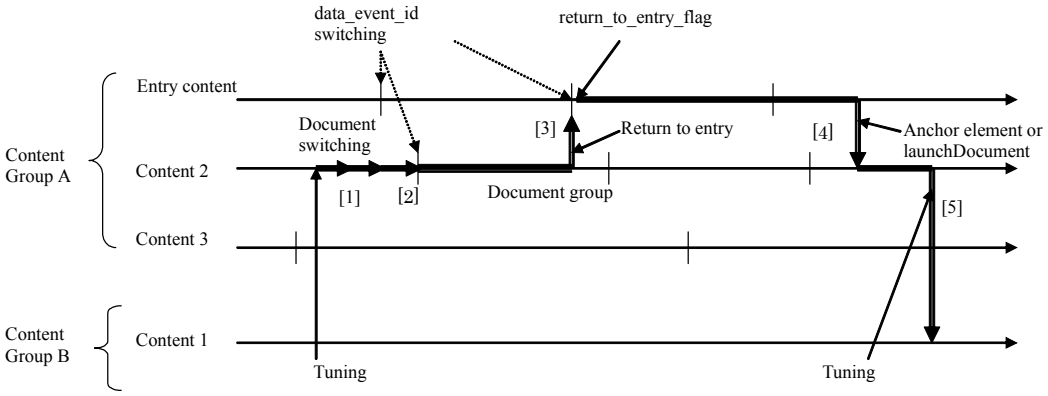


Figure 6-1 Example of contents transmission

A3-6.1.2 Scope mapping for contents transmitted via interactive communication networks associated with broadcasting services

The concepts of "document group", "content", and "content group" are not applied to contents transmitted via interactive communication networks associated with broadcasting services.

However, when a content transmitted via interactive communication networks is identified as a primary link destination from a content received in real time or a content stored from a broadcasting service, the content transmitted via interactive communication networks is regarded as a member of the same content group as the linking content. Specifically, in the following case, a content transmitted via interactive communication networks is identified as a primary link.

- A BML content transmitted via interactive communication networks is in or under the same path (directory) as a BML document that is the first destination of a linking content received in real time or a linking stored content.
- However, when a content received in real time or a stored content uses the launchDocumentRestricted() function to transit to a BML content transmitted via interactive communication networks as the first destination, no BML content obtained via interactive communication networks is identified as a primary link destination.

A3-6.1.3 Scope mapping for stored contents

Table 6-2 Stored contents and scope mapping

Scope	Scope identifier	Corresponding transmission signal
Within document	Document scope identifier	full path URI to file
Within document group	Document group scope identifier	none
Within content	Content scope identifier	none
Within content group	Content group scope identifier	contents that transmitted through the same carousel

A3-6.2 Guideline on reference across media types

A3-6.2.1 Restriction on document transmission among media types

Among contents received in real time, contents transmitted via interactive communication networks, and stored contents (Sub-profile 2 only), which content is allowed to be specified as a linked destination of a BML document is defined in Table 6-3. In this table, the following transition types are represented by the numbers 1 through 4.

(1) transition caused by epgTune()/epgTuneToComponent()

(2) transition caused by a value in the href attribute of the a element

(3) transition caused by launchDocument()

(4) transition caused by the submit function of the form element

Table 6-3 Restriction on Document Transmission across Media Types

	Transiting (Linking) Content											
	Broadcast (contents received in real time)				Stored (Note 1) (stored contents)				Communication (contents transmitted via interactive communication networks)			
Transited (Linked) Content	(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)
Broadcast	O	O (Note 2)	O (Note 2)	-	O	O (Note 3)	O (Note 3)	-	O	O (Note 3)	O (Note 3)	-
Stored (Note 1)	-	O	O	-	-	O	O	-	-	O	O	-
Communication	-	O	O	O	-	O	O	O	-	O	O	O

Note 1: This description is applicable only to Sub-profile 2 contents.
Note 2: This description is applicable to BML documents belong to the same service.
Note 3: This description is applicable to BML documents belong to the currently selected service.

Among the transition patters in Table 6-3, when a transition from a stored content or a content transmitted via interactive communication networks to a broadcast content involving a service selection operation is caused by the 2 through 4 transition types described above how a receiver behaves depends on an implementation.

A3-6.2.2 Restriction on reference of monomedia and others across media types

The following seven items are assumed as resources referenced by a BML document. The operational convention on these items are summarized in Table 6-4.

(1) A monomedia that can be specified with the data attribute of the object element (an AV stream for broadcasting services, a stored time-stamped TS file)

(2) A monomedia that can be specified with the data attribute of the object element (e.g. JPEG, PNG, MNG, MPEG-2 audio), a CLUT data specified by the clut property, and a JPEG data specified by the background-image property

(3) A DRCS file specified with an attribute of the loadDRCS() function

(4) A binary table

(5) A NPT reference message referred by the beitem element

(6) An event message referred by the beitem element

(7) A proprietary data that is designed to be stored in a bridge media or others by a saving operation through a BML document (for only sub-profile 2)

The following table contains no convention on a function transmitting messages with BASIC-based protocol or TCP/IP.

Table 6-4 Resources Referenced by BML document

			Referencing BML Document			Remarks
			Content received in real time (Broadcast)	Stored content (Stored) (sub-profile 2)	Content transmitted via interactive communication network (Communication)	
Referenced Resource	Broadcast	(1) Mainstream MPEG2-AV (Note 1)	O Within the same service	O (Note 4)	O (Note 5)	- For Broadcast to Broadcast reference, not more than one carousel specified with the same data_ event_id is allowed to be referenced at a time. - No simultaneous reference of a mainstream MPEG2-AV data and a stored MPEG2-AV data is allowed. If it occurs, the behaviour depends on an implementation. - If an update of data_ event_id occurs when a Broadcast resource is referenced, the entry component of the carousel to which the referenced resource belongs becomes active.
		(2) JPEG/PNG/MNG/ MPEG-2 audio	O Within the same carousel	O (Note 4)	O (Note 5 ,7)	
		(3) DRCS	O Within the same carousel (Note 3)	O (Note 4)	-	
		(4) Binary Table	O Within the same carousel (Note 3)	O (Note 4)	-	
		(5) NPT	O Within the same service	O (Note 4)	O (Note 5)	
		(6) Event Message	O Within the same service	O (Note 4)	O (Note 5)	
		(7) Proprietary Data	O Within the same carousel provided that through saveResource()	O (Note 4)	-	
	Stored	(1) Stored MPEG2-AV (Note 2) (Time-stamped TS)	-	O	O (Note 5)	No simultaneous reference of a mainstream MPEG2-AV data and a stored MPEG2-AV data is allowed. If it occurs, the behaviour depends on an implementation.
		(2) JPEG/PNG/MNG/ MPEG-2 audio	-	O	-	
		(3) DRCS	-	O	-	
		(4) Binary Table	-	O	-	

		(7) Proprietary Data	-	Δ Limited to saving operations	-	
	Communi-cation	(2) JPEG/PNG/MNG/ MPEG-2 audio	-	O	O	
		(4) Binary Table	-	O	O	
		(7) Proprietary Data	-	Δ Limited to saving operations	Δ Limited to saving operations	

Note 1: Video/audio streams contained in an event program that is to be received in real time
Note 2: Time-stamped TS file that have been stored
Note 3: This description is also applicable to resources contained in a module that has been transmitted in the same ES and locked, even if the concerned content does not transmitted in the same carousel.
Note 4: This description is applicable to the case where a referencing stored content has been transmitted through the same carousel as a stored content that is the first destination of a transition from a referenced broadcast content.
Note that this restriction may be ineffective depending how an individual media type implementation is operated.
Note 5: This description is only applicable when a referencing communication content is identified as a trusted content in either of the following methods. Which method, including the combination of the two, is operated is defined in an operational standard for each media.
- Identifying the referencing communication content as a primary link destination transited by a referenced broadcast content.
- Identifying the concerned content as a trusted content using a certificate (Details of certificates are defined in an operational standard.)

Note 6: This description is only applicable when a referencing communication content is identified as a trusted content in either of the following methods. Which method, including the combination of the two, is operated is defined in an operational standard for each media.
- Identifying the referencing communication content as a primary link destination transited by a referenced stored content.
- Identifying the concerned content as a trusted content using a certificate (Details of certificates are defined in an operational standard.)

Note 7: Actual operation of this referencing pattern depends on an individual media type implementation.

A3-6.3 Namespace

A3-6.3.1 Namespace for contents received in real time

- Any resource specification containing arib or arib-dc complies with the convention defined in Volume 2.
- However, in a component-base model, it is allowed that the href attribute of the a element contains the id attribute of other element in a BML document in the following format.
#<id attribute>

A3-6.3.2 Namespace for contents transmitted via interactive communication networks associated with broadcasting services

- A specification of a content transmitted via interactive communication networks associated with broadcasting services begins with http.

- A specification of a resource to be obtained via secure communication begins with https.
  - The namespace for URI complies with the definition provided in RFC 18055 and the additional restrictions described below are applied.
    - When an access to a directory is requested (the concerned URL ends with /), how a server behaves depends on the server and how a receiver behaves depends on the implementation.
    - The representation "./" or "../" is applicable only to the beginning of a relative URL.
    - It is assumed that ~ is not interpreted by a receiver.
    - The representation, ;params, is not operated.
    - The namespace for a CGI argument complies with the namespace defined in RFC 1808.
    - A directory name or file name may contain a single period, ".", unless the name begings nor ends with the period.

    The valid syntax is shown below.

    URL = (absoluteURL | relativeURL) ["#" fragment]

absoluteURL	=	scheme ":" "//" net_loc "/" rel_path
relativeURL	=	[("./" | "../" | "/")] rel_path
net_loc	=	*(pchar|";"|"?")
rel_path	=	[path] ["?" query]
path	=	fsegment *("/" fsegment)
fsegment	=	1*fpchar ["." 1*fpchar]
query	=	*(uchar | reserved)
fragment	=	1*fchar ["." 1*fchar]
fpchar	=	fchar| ":" | "@" | "&" | "="
fchar	=	alpha | digit | "-" | "_" | "+"
pchar	=	uchar | ":" | "@" | "&" | "="
uchar	=	unreserved | escape
unreserved	=	alpha | digit | safe | extra
escape	=	"%" hex hex
hex	=	digit | "A" | "B" | "C" | "D" | "E" | "F" | "a" | "b" | "c" | "d" | "e" | "f"
alpha	=	lowalpha | hialpha
lowalpha	=	"a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" |"j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z"
hialpha	=	"A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z"
digit	=	"0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
safe	=	"$" | "-" | "_" | "." | "+"
extra	=	"!" | "*" | "'" | "(" | ")" | ","
national	=	"{" | "}" | "|" | "\" | "^" | "~" | "[" | "]" | "`"
reserved	=	";" | "/" | "?" | ":" | "@" | "&" | "="

punctuation = "<" | ">" | "#" | "%" | <">

- The maximum length of a URI specification must be 255 bytes.
- A shortened form of URI is interpreted based on RFC 1808 and the additional restrictions described below.
  - Any tilde ~ character is interpreted as interpreted by a server. A receiver does not interpret the character.
  - A URI specification containing the representation "./" or "../" is interpreted as a relative URI to a file or directory in which the current BML file resides by a receiver. Note that this restriction is applicable only to a URI specification that does not begin with "./" nor "../". For example, the specification, foo/../bar is not operated.
  - The slash "/" character is interpreted as the root directory of a server by a receiver.

A3-6.3.3 Namespace for stored contents (sub-profile 2)

To identify a stored content, the content specification must contain the scheme specification "arib-file:", as defined in Section 9.2.15, Volume 2.

A3-6.4 Guidelines on transmission of contents

A3-6.4.1 Syntax of module

- Each entity-body that is part of another entity-body in the multi-part format must not be of the multi-part type as media type.
- The entity headers that any basic receiver is required to interpret are listed in Table 6-5.

Table 6-5 Entity Headers Interpreted by Any Basic Receiver

Entity header	Semantics
Content-Type	Indicates a media type of a resource contained in an entity body. Define Content-Type for multipart entity body as "multipart/mixed" and specify a boundary parameter.
Content-Length	Indicates the byte length of an entity body.
Content-Location	Indicates the identifier of an entity body.

- The Content-Base:header and the Content-Encoding field are optional.
- Any Content-Location field specification contains only <resourceName> as a relative specification to the concerned module. The namespace for <resourceName> complies with the namespace for resources, as defined in Chapter 9, Volume 2.

A3-6.4.2 Resource list in multi-part formatted module

Any module in the multi-part format that is to be received in real time must contain a resource list at the beginning.

A3-6.4.3 Resource list in multi-part formatted module assuming stored data services (sub-profile 2)

Any module in the multi-part format that is part of a content to be transmitted in a stored data services must contain a stored resource list.

A3-6.4.4 Operation of event messages

The character encoding specification for private_data_byte of a generic event message descriptor must be "EUC-JP".

A3-6.5 Operational guidelines on communication protocols

A3-6.5.1 Operational guidelines on HTTP

The version of HTTP to be support by a receiver must be 1.1. Any receiver must be able to successfully connect with any server operating under HTTP/1.0.

Which methods are operated for HTTP are listed in Table 6-6. Which and how headers are operated for HTTP are shown in Table 6-7 and Table 6-8.

Table 6-6 Operation of Methods

Method	Operation	Remarks
GET	O	
HEAD	-	
POST	O	
PUT	-	
DELETE	-	
OPTINON	-	
TRACE	-	
CONNECT	O	Required for SSL communication using Proxy

Table 6-7 Operation of Header for Request

Header \ Method		GET	POST
Request	Accept	-	-
	Accept-Charset	O	O
	Accept-Encoding	-	-
	Accept-Language	-	-
	Authorization	O	O
	From	-	-
	Host	O	O
	If-Modified-Since	O	-
	If-Unmodified-Since	-	-
	If-Match	-	-
	If-None-Match	-	-

Header	Method	GET	POST
	If-Range	-	-
	Max-Forward	-	-
	Proxy-authorization	O	O
	Range	O	-
	Referer	O	O
	User-Agent	O	O
General	Cache-Control	O	O
	Connection	Limited to close	Limited to close
	Date	-	-
	Pragma	-	-
	Transfer-Encoding	-	-
	Upgrade	-	-
	Via	-	-
Entity	Allow	-	-
	Content-Encoding	-	-
	Content-Language	-	-
	Content-Length	-	O
	Content-Location	-	-
	Content-MD5	-	-
	Content-Range	-	-
	Content-Type	-	O
	Etag	-	-
	Last-Modified	-	-

Table 6-8 Operation of Header for Response

Header	Method	GET	POST
Response	Accept-Ranges	O	O
	Authentication-info	O	O
	Location	O	O
	Proxy-Authenticate	O	O
	Proxy-Authentication-info	O	O
	Public	-	-
	Retry-After	-	-
	Server	-	-
	Vary	-	-
	Warning	-	-
	WWW-Authentication	O	O
	Age	O	O
	Expire	O	O
General	Cache-Control	O	O

Header	Method	GET	POST
	Connection	Limited to close	Limited to close
	Date	O	O
	Pragma	Limited to no-cache	Limited to no-cache
	Transfer-Encoding	Limited to chunked	Limited to chunked
	Upgrade	-	-
	Via	-	-
Entity	Allow	-	-
	Content-Encoding	Limited to none	Limited to none
	Content-Language	O	O
	Content-Length	O	O
	Content-Location	O	O
	Content-MD5	-	-
	Content-Range	O	-
	Content-Type	O	O
	Etag	-	-
	Last-Modified	O	O

A3-6.5.2 Operational Guidelines on mapping between API and protocols

To use http to make a BML content access to a communication server, the methods listed in the following table are applicable.

Access	Applicable Method
To obtain a BML document from a server by including http in the href attribute in the a element	GET method is used
To obtain a resource from a server by including http in the data attribute in the object element	GET method is used
To obtain a BML document from a server by including http in an argument in the launchDocument() function	GET method is used
To send a file to a server by using the sendHttpServerFile() function	POST method is used
To obtain a file from a server by using the saveHttpServerFileAs() function	GET method is used
To communicate with a server by using the form element	GET method or POST method is used depending on a value in the action attribute

In the cases where the Get method is used as mentioned above, the maximum length of a specified URI string must be 255 bytes.

A3-6.5.3 Operational guidelines on secure communication

To include https: in a resource specification to establish a secure communication session, TLS 1.0 must be used.

Detailed operation of TLS 1.0 is defined in an operational standard defined for each media or by each broadcaster.

Annex A Default Style Sheet

A.1 Default Style Sheet Common to Index-based Model and Component-based Model

```
@media tv{
body, h1, h2, h3, h4, h5, h6,
address, blockquote, div,
p, pre,
dl, dt, dd, ol, ul,
hr, form       { display: block }
li             { display: list-item }
table          { display: table }
tr             { display: table-row }
th, td         { display: table-cell }
caption        { display: table-caption }
head, param    { display: none }
/* display */
meta, title, script, style, head, bevent, beitem, { display : none }
input[type=hidden]{ display: none }
noscript{display:none}

address, cite, dfn, em,i
               { font-style: italic }

pre            { white-space: pre }
br:before      { content: "\A" }

q:before       { content:open-quote }
q:after        { content:close-quote }
q              { quotes:'"' '"' }
}
```

A.2 Default Style Sheet Common Specific to Index-based Model

```
/* The following rule sets are added to the default style sheet provided in A.1 for
an indec-based model.*/
@media tv{
/* margin */
div, p, input, object {margin: 0}
/* padding */
div, object {padding-top : 0; padding-right : 0; padding-bottom : 0; padding-left :
0; }
/* border */
div, p, span, a, input {border-width : 0 }
object {border-width : 0; border-style : none}
/* position */

body, h1, h2, h3, h4, h5, h6, address, blockquote,
div,p, pre, dl, ol, ul, hr, form, table, input, object { position : absolute }
br, span, a { position : static }
/* top left width height*/
p,div, input, object {top : 0; left : 0; width : 0; height : 0 }
/* visibility */
```

```
body { visibility : visible}
/* overflow */
p, div, input, object { overflow : hidden }
/* text */

p, input { font-family : "丸ゴシック"; font-size : 24px }
span, a { font-family : inherit; font-size : inherit }
p, input { text-align : left }
div, p, span, a, input { background-color-index : 8 }
object { background-color-index : 8}
/* grayscale-color-index */
p, input { grayscale-color-index : 30 15 }
}
/* line-height */
body { line-height: 1.0; }
```

A.3 Default Style Sheet Common Specific to Component-based Model

```
/* The following rule set is added to the defalut style sheet provided in A.1 for a
component-based model. */
a[href]        { text-decoration: underline;}
```

Annex B Operated DTD

B.1 BML Driver DTD

```
<!-- .................................................................... -->
<!-- BML DTD 2.1 for extended service ..................................... -->
<!-- file: bml_2_1.dtd -->

<!-- BML 2.1 DTD

     This is BML, a broadcast markup language as a modular XML application.
     Copyright 2001 ARIB, All Rights Reserved.
     Revision: $Id: bml_2_1.mod,v 1.0 2001/10/09 21:16 ARIB XML-WG BasicXML-TG

-->

<!-- This is the driver file for version 2.1 of the BML DTD.

     Please use this formal public identifier to identify it:

         "-//ARIB//DTD XHTML BML 2.1//JA"
-->
<!ENTITY % XHTML.version  "-//ARIB//DTD XHTML BML 2.1//JA" >

<!-- Use this URI to identify the default namespace:

         "http://www.w3.org/1999/xhtml"

     See the Qualified Names module for information
     on the use of namespace prefixes in the DTD.
-->
<!ENTITY % NS.prefixed "IGNORE" >
<!ENTITY % XHTML.prefix "" >

<!-- Reserved for use with the XLink namespace:
-->
<!ENTITY % XLINK.xmlns "" >
<!ENTITY % XLINK.xmlns.attrib "" >

<!-- reserved for future use with document profiles -->
<!ENTITY % XHTML.profile  "" >

<!-- Bring in any qualified name modules outside of XHTML -->
<!ENTITY % bml-qname.mod
       PUBLIC "-//W3C//ENTITIES BML Qualified Names 2.0//JA"
              "http://www.arib.or.jp/B24/DTD/bml-qname_2_0.mod" >
%bml-qname.mod;

<!-- Bidirectional Text features
     This feature-test entity is used to declare elements
     and attributes used for bidirectional text support.
-->
<!ENTITY % XHTML.bidi  "IGNORE" >
```

```
<!-- Pre-Framework Redeclaration placeholder  ..................... -->
<!-- this serves as a location to insert markup declarations
     into the DTD prior to the framework declarations.
-->
<!ENTITY % xhtml-prefw-redecl.module "IGNORE" >
<![%xhtml-prefw-redecl.module;[
%xhtml-prefw-redecl.mod;
<!-- end of xhtml-prefw-redecl.module -->]]>

<!ENTITY % xhtml-events.module "IGNORE" >

<!-- Inline Style Module  ....................................... -->
<!ENTITY % xhtml-inlstyle.module "INCLUDE" >
<![%xhtml-inlstyle.module;[
<!ENTITY % xhtml-inlstyle.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Inline Style 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-inlstyle-1.mod" >
%xhtml-inlstyle.mod;]]>

<!-- declare Document Model module instantiated in framework
-->
<!ENTITY % xhtml-model.mod
     PUBLIC "-//W3C//ENTITIES BML DOCUMENT MODEL 2.1//JA"
            "http://www.arib.or.jp/B24/DTD/bml-model_2_1.mod" >

<!ENTITY % Events.attrib
     "onclick      %Script.datatype;        #IMPLIED
      ondblclick   %Script.datatype;        #IMPLIED
      onmousedown  %Script.datatype;        #IMPLIED
      onmouseup    %Script.datatype;        #IMPLIED
      onmouseover  %Script.datatype;        #IMPLIED
      onmousemove  %Script.datatype;        #IMPLIED
      onmouseout   %Script.datatype;        #IMPLIED
      onkeypress   %Script.datatype;        #IMPLIED
      onkeydown    %Script.datatype;        #IMPLIED
      onkeyup      %Script.datatype;        #IMPLIED"
>

<!-- Modular Framework Module (required) ......................... -->
<!ENTITY % xhtml-framework.module "INCLUDE" >
<![%xhtml-framework.module;[
<!ENTITY % xhtml-framework.mod
     PUBLIC "-//W3C//ENTITIES XHTML Modular Framework 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-framework-1.mod" >
%xhtml-framework.mod;]]>

<!-- additional attributes on anchor element
-->
<!ATTLIST %a.qname;
     onfocus      %Script.datatype;         #IMPLIED
     onblur       %Script.datatype;         #IMPLIED
>
```

```
<!-- additional attributes on form element
-->
<!ATTLIST %form.qname;
      onsubmit     %Script.datatype;        #IMPLIED
      onreset      %Script.datatype;        #IMPLIED
>

<!-- additional attributes on label element
-->
<!ATTLIST %label.qname;
      onfocus      %Script.datatype;        #IMPLIED
      onblur       %Script.datatype;        #IMPLIED
>

<!-- additional attributes on input element
-->
<!ATTLIST %input.qname;
      onfocus      %Script.datatype;        #IMPLIED
      onblur       %Script.datatype;        #IMPLIED
      onselect     %Script.datatype;        #IMPLIED
      onchange     %Script.datatype;        #IMPLIED
>

<!-- additional attributes on select element
-->
<!ATTLIST %select.qname;
      onfocus      %Script.datatype;        #IMPLIED
      onblur       %Script.datatype;        #IMPLIED
      onchange     %Script.datatype;        #IMPLIED
>

<!-- additional attributes on textarea element
-->
<!ATTLIST %textarea.qname;
      onfocus      %Script.datatype;        #IMPLIED
      onblur       %Script.datatype;        #IMPLIED
      onselect     %Script.datatype;        #IMPLIED
      onchange     %Script.datatype;        #IMPLIED
>

<!-- additional attributes on body element
-->
<!ATTLIST %body.qname;
      onload       %Script.datatype;        #IMPLIED
      onunload     %Script.datatype;        #IMPLIED
>

<!-- Post-Framework Redeclaration placeholder  ..................... -->
<!-- this serves as a location to insert markup declarations
     into the DTD following the framework declarations.
-->
<!ENTITY % xhtml-postfw-redecl.module "IGNORE" >
```

```
<![%xhtml-postfw-redecl.module;[
%xhtml-postfw-redecl.mod;
<!-- end of xhtml-postfw-redecl.module -->]]>

<!ENTITY % pre.element  "INCLUDE" >
<![%pre.element;[
<!ENTITY % pre.content
     "( #PCDATA
      | %InlStruct.class;
      %InlPhras.class;
      | %tt.qname; | %i.qname; | %b.qname;
      %I18n.class;
      %Anchor.class;
      | %script.qname;
      %Inline.extra; )*"
>
<!-- end of pre.element -->]]>

<!-- Text Module (Required)  ..................................... -->
<!ENTITY % xhtml-text.module "INCLUDE" >
<![%xhtml-text.module;[
<!ENTITY % xhtml-text.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Text 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-text-1.mod" >
%xhtml-text.mod;]]>

<!-- Hypertext Module (required) ................................. -->
<!ENTITY % xhtml-hypertext.module "INCLUDE" >
<![%xhtml-hypertext.module;[
<!ENTITY % xhtml-hypertext.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Hypertext 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-hypertext-1.mod" >
%xhtml-hypertext.mod;]]>

<!-- Lists Module (required)  .................................... -->
<!ENTITY % xhtml-list.module "INCLUDE" >
<![%xhtml-list.module;[
<!ENTITY % xhtml-list.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Lists 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-list-1.mod" >
%xhtml-list.mod;]]>

<!-- ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::: -->

<!-- Edit Module  ................................................ -->
<!ENTITY % xhtml-edit.module "IGNORE" >
<![%xhtml-edit.module;[
<!ENTITY % xhtml-edit.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Editing Elements 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-edit-1.mod" >
%xhtml-edit.mod;]]>

<!-- BIDI Override Module  ....................................... -->
```

```
<!ENTITY % xhtml-bdo.module "%XHTML.bidi;" >
<![%xhtml-bdo.module;[
<!ENTITY % xhtml-bdo.mod
     PUBLIC "-//W3C//ELEMENTS XHTML BIDI Override Element 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-bdo-1.mod" >
%xhtml-bdo.mod;]]>

<!-- Ruby Module  ..................................................... -->
<!ENTITY % Ruby.common.attlists "IGNORE" >
<!ENTITY % Ruby.common.attrib "%Common.attrib;" >
<!ENTITY % xhtml-ruby.module "IGNORE" >
<![%xhtml-ruby.module;[
<!ENTITY % xhtml-ruby.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Ruby 1.0//EN"
            "http://www.w3.org/TR/ruby/xhtml-ruby-1.mod" >
%xhtml-ruby.mod;]]>

<!-- Presentation Module  ............................................. -->
<!ENTITY % xhtml-pres.module "INCLUDE" >
<![%xhtml-pres.module;[
<!ENTITY % xhtml-pres.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Presentation 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-pres-1.mod" >
%xhtml-pres.mod;]]>

<!-- Link Element Module  ............................................. -->
<!ENTITY % xhtml-link.module "INCLUDE" >
<![%xhtml-link.module;[
<!ENTITY % xhtml-link.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Link Element 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-link-1.mod" >
%xhtml-link.mod;]]>

<!-- Document Metainformation Module  ................................. -->
<!ENTITY % xhtml-meta.module "INCLUDE" >
<![%xhtml-meta.module;[
<!ENTITY % xhtml-meta.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Metainformation 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-meta-1.mod" >
%xhtml-meta.mod;]]>

<!-- Base Element Module  ............................................. -->
<!ENTITY % xhtml-base.module "INCLUDE" >
<![%xhtml-base.module;[
<!ENTITY % xhtml-base.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Base Element 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-base-1.mod" >
%xhtml-base.mod;]]>

<!-- Scripting Module  ................................................ -->
<!ENTITY % xhtml-script.module "INCLUDE" >
<![%xhtml-script.module;[
<!ENTITY % xhtml-script.mod
```

```
     PUBLIC "-//W3C//ELEMENTS XHTML Scripting 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-script-1.mod" >
%xhtml-script.mod;]]>

<!-- Style Sheets Module  ..................................... -->
<!ENTITY % xhtml-style.module "INCLUDE" >
<![%xhtml-style.module;[
<!ENTITY % xhtml-style.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Style Sheets 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-style-1.mod" >
%xhtml-style.mod;]]>

<!-- Image Module  ............................................ -->
<!ENTITY % xhtml-image.module "INCLUDE" >
<![%xhtml-image.module;[
<!ENTITY % xhtml-image.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Images 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-image-1.mod" >
%xhtml-image.mod;]]>

<!-- Client-side Image Map Module  ............................ -->
<!ENTITY % xhtml-csismap.module "IGNORE" >
<![%xhtml-csismap.module;[
<!ENTITY % xhtml-csismap.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Client-side Image Maps 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-csismap-1.mod" >
%xhtml-csismap.mod;]]>

<!-- Server-side Image Map Module  ............................ -->
<!ENTITY % xhtml-ssismap.module "IGNORE" >
<![%xhtml-ssismap.module;[
<!ENTITY % xhtml-ssismap.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Server-side Image Maps 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-ssismap-1.mod" >
%xhtml-ssismap.mod;]]>

<!-- Param Element Module  .................................... -->
<!ENTITY % xhtml-param.module "INCLUDE" >
<![%xhtml-param.module;[
<!ENTITY % xhtml-param.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Param Element 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-param-1.mod" >
%xhtml-param.mod;]]>

<!-- Embedded Object Module  .................................. -->
<!ENTITY % xhtml-object.module "INCLUDE" >
<![%xhtml-object.module;[
<!ENTITY % xhtml-object.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Embedded Object 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-object-1.mod" >
%xhtml-object.mod;]]>

<!-- Tables Module ............................................ -->
```

```
<!ENTITY % xhtml-table.module "IGNORE" >
<![%xhtml-table.module;[
<!ENTITY % xhtml-table.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Tables 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-table-1.mod" >
%xhtml-table.mod;]]>

<!-- Basic Tables Module ..................................................... -->
<!ENTITY % xhtml-basic-table.module "INCLUDE" >
<![%xhtml-basic-table.module;[
<!ENTITY % xhtml-basic-table.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Basic Tables 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-basic-table-1.mod"
>
%xhtml-basic-table.mod;]]>

<![%table.attlist;[
<!ATTLIST %table.qname;
      width        %Length.datatype;        #IMPLIED
>
<!-- end of table.attlist -->]]>

<!-- Forms Module  ........................................................... -->
<!ENTITY % xhtml-form.module "IGNORE" >
<![%xhtml-form.module;[
<!ENTITY % xhtml-form.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Forms 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-form-1.mod" >
%xhtml-form.mod;]]>

<!-- Basic Forms Module  ..................................................... -->
<!ENTITY % xhtml-basic-form.module "INCLUDE" >
<![%xhtml-basic-form.module;[
<!ENTITY % xhtml-basic-form.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Basic Forms 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-basic-form-1.mod"
>
%xhtml-basic-form.mod;]]>

<!-- Legacy Markup ........................................................... -->
<!ENTITY % xhtml-legacy.module "IGNORE" >
<![%xhtml-legacy.module;[
<!ENTITY % xhtml-legacy.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Legacy Markup 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-legacy-1.mod" >
%xhtml-legacy.mod;]]>

<!-- Frames Module ........................................................... -->
<!ENTITY % xhtml-frames.module "IGNORE" >
<![%xhtml-frames.module;[
<!ENTITY % xhtml-frames.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Frames 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-frames-1.mod" >
```

```
%xhtml-frames.mod;]]>

<!-- Target Markup ............................................ -->
<!ENTITY % xhtml-target.module "IGNORE" >
<![%xhtml-target.module;[
<!ENTITY % target-iframe.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Target 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-target-1.mod" >
%xhtml-target.mod;]]>

<!-- IFrame Module ............................................ -->
<!ENTITY % xhtml-iframe.module "IGNORE" >
<![%xhtml-iframe.module;[
<!ENTITY % xhtml-iframe.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Inline Frame Element 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-iframe-1.mod" >
%xhtml-iframe.mod;]]>

<!-- Name Identification Markup ............................... -->
<!ENTITY % xhtml-nameident.module "IGNORE" >
<![%xhtml-nameident.module;[
<!ENTITY % xhtml-nameident.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Name Identifier 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-nameident-1.mod"
%xhtml-nameident.mod;]]>

<!-- Document Structure Module (required)  ..................... -->
<!ENTITY % xhtml-struct.module "INCLUDE" >
<![%xhtml-struct.module;[
<!ENTITY % xhtml-struct.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Document Structure 1.0//EN"
            "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-struct-1.mod" >
%xhtml-struct.mod;]]>

<!-- see the Qualified Names module for information
     on how to extend XHTML using XML namespaces
-->
<!ATTLIST %html.qname;
      %BML.xmlns.attrib;
>

<!-- ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::: -->

<!-- BML Element Module ....................................... -->
<!ENTITY % bml-elements.module "IGNORE" >
<![%bml-elements.module;[
<!ENTITY % bml-elements.mod
     PUBLIC "-//ARIB//ELEMENTS XHTML BML Elements 2.0//JA"
               "http://www.arib.or.jp/B24/DTD/bml-elements_2_0.mod" >
%bml-elements.mod;]]>

<!-- Basic BML Element Module ................................... -->
<!ENTITY % bml-basic-elements.module "INCLUDE" >
```

```
<![%bml-basic-elements.module;[
<!ENTITY % bml-basic-elements.mod
     PUBLIC "-//ARIB//ELEMENTS XHTML Basic BML Elements 2.0//JA"
               "http://www.arib.or.jp/B24/DTD/bml-basic-elements_2_0.mod" >
%bml-basic-elements.mod;]]>

<!-- end of BML 2.1 Basic DTD  ..................................... -->
<!-- ................................................................ -->
```

B.2 BML Document Model Module

```
<!-- ................................................................ -->
<!-- BML Document Model Module for extended service ................. -->
<!-- file: bml-model_2_1.mod

   This is BML, a broadcast markup language as a modular XML application.
   Copyright 2001 ARIB, All Rights Reserved.
   Revision: $Id: bml-model_2_1.mod, v 1.0 2001/10/09 21:14 ARIB XML-WG BasicXML-TG

   This DTD module is identified by the PUBLIC and SYSTEM identifiers:

       PUBLIC "-//ARIB//ELEMENTS BML DOCUMENT MODEL 2.1//JA"
       SYSTEM "http://www.arib.or.jp/B24/DTD/bml-model_2_1.mod"

     ............................................................... -->

<!-- BML Document Model

     This module describes the groupings of elements that make up
     common content models for XHTML elements.

-->

<!-- Extending the Model

     While in some cases this module may need to be rewritten to
     accommodate changes to the document model, minor extensions
     may be accomplished by redeclaring any of the three *.extra;
     parameter entities to contain extension element types as follows:

         %Misc.extra;    whose parent may be any block or
                         inline element.

         %Inline.extra;  whose parent may be any inline element.

         %Block.extra;   whose parent may be any block element.

     If used, these parameter entities must be an OR-separated
     list beginning with an OR separator ("|"), eg., "| a | b | c"

     All block and inline *.class parameter entities not part
     of the *struct.class classes begin with "| " to allow for
```

```
     exclusion from mixes.
-->

<!-- ..............  Optional Elements in head  .................. -->

<!ENTITY % HeadOpts.mix
     "( %script.qname; | %style.qname; | %meta.qname;
      | %link.qname; | %object.qname; | %BML.bevent.qname; )*"
>

<!-- .................  Miscellaneous Elements  .................. -->

<!-- script and noscript are used to contain scripts
     and alternative content
-->
<!ENTITY % Script.class "| %script.qname; | %noscript.qname;" >

<!ENTITY % Misc.extra "" >

<!-- These elements are neither block nor inline, and can
     essentially be used anywhere in the document body.
-->
<!ENTITY % Misc.class
      "%Script.class;
      %Misc.extra;"
>

<!-- ....................  Inline Elements  ..................... -->

<!ENTITY % InlStruct.class "%br.qname; | %span.qname;" >

<!ENTITY % InlPhras.class
     "| %em.qname; | %strong.qname; | %dfn.qname; | %code.qname;
      | %samp.qname; | %kbd.qname; | %var.qname; | %cite.qname;
      | %abbr.qname; | %acronym.qname; | %q.qname;" >

<!ENTITY % InlPres.class
     "| %tt.qname; | %i.qname; | %b.qname; | %big.qname;
      | %small.qname; | %sub.qname; | %sup.qname;" >

<!ENTITY % I18n.class "" >

<!ENTITY % Anchor.class "| %a.qname;" >

<!ENTITY % InlSpecial.class
     "| %img.qname; | %object.qname;" >

<!ENTITY % InlForm.class
     "| %input.qname; | %select.qname; | %textarea.qname;
      | %label.qname;" >

<!ENTITY % Inline.extra "" >
```

```
<!-- %Inline.class; includes all inline elements,
     used as a component in mixes
-->
<!ENTITY % Inline.class
     "%InlStruct.class;
      %InlPhras.class;
      %InlPres.class;
      %Anchor.class;
      %InlSpecial.class;
      %InlForm.class;
      %Inline.extra;"
>


<!-- %InlNoAnchor.class; includes all non-anchor inlines,
     used as a component in mixes
-->
<!ENTITY % InlNoAnchor.class
     "%InlStruct.class;
      %InlPhras.class;
      %InlPres.class;
      %InlSpecial.class;
      %InlForm.class;
      %Inline.extra;"
>

<!-- %InlNoAnchor.mix; includes all non-anchor inlines
-->
<!ENTITY % InlNoAnchor.mix
     "%InlNoAnchor.class;
      %Misc.class;"
>

<!-- %Inline.mix; includes all inline elements, including %Misc.class;
-->
<!ENTITY % Inline.mix
     "%Inline.class;
      %Misc.class;"
>

<!-- ......................  Block Elements  ...................... -->

<!-- In the HTML 4.0 DTD, heading and list elements were included
     in the %block; parameter entity. The %Heading.class; and
     %List.class; parameter entities must now be included explicitly
     on element declarations where desired.
-->

<!ENTITY % Heading.class
     "%h1.qname; | %h2.qname; | %h3.qname;
      | %h4.qname; | %h5.qname; | %h6.qname;" >

<!ENTITY % List.class "%ul.qname; | %ol.qname; | %dl.qname;" >
```

```
<!ENTITY % Table.class "| %table.qname;" >

<!ENTITY % Form.class  "| %form.qname;" >

<!ENTITY % BlkStruct.class "%p.qname; | %div.qname;" >

<!ENTITY % BlkPhras.class
     "| %pre.qname; | %blockquote.qname; | %address.qname;" >

<!ENTITY % BlkPres.class "| %hr.qname;" >

<!ENTITY % BlkSpecial.class
     "%Table.class;
      %Form.class;"
>

<!ENTITY % Block.extra "" >

<!-- %Block.class; includes all block elements,
     used as an component in mixes
-->
<!ENTITY % Block.class
     "%BlkStruct.class;
      %BlkPhras.class;
      %BlkPres.class;
      %BlkSpecial.class;
      %Block.extra;"
>

<!-- %Block.mix; includes all block elements plus %Misc.class;
-->
<!ENTITY % Block.mix
     "%Heading.class;
      | %List.class;
      | %Block.class;
      %Misc.class;"
>

<!-- .................  All Content Elements  .................. -->

<!-- declares all content except tables -->
<!ENTITY % FlowNoTable.mix
     "%Heading.class;
      | %List.class;
      | %BlkStruct.class;
      %BlkPhras.class;
      %BlkPres.class;
      %Form.class;
      %Block.extra;
      | %Inline.class;
      %Misc.class;"
>
```

```
<!-- %Flow.mix; includes all text content, block and inline
-->
<!ENTITY % Flow.mix
     "%Heading.class;
      | %List.class;
      | %Block.class;
      | %Inline.class;
      %Misc.class;"
>
<!-- end of bml-model-2_1.mod -->
```

B.3 Description of DTD Declaration

This section provides the convention on description of DTD declaration.

- For a document whose character encoding specification is EUC-JP, the DTD declaration must be:

```
<?xml version="1.0" encoding="EUC-JP" ?>
```

- For a document whose character encoding specification is Shift-JIS, the DTD declaration must be:

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<!DOCTYPE html PUBLIC
            "-//ARIB//DTD XHTML BML 2.1//JA"
"http://www.arib.or.jp/B24/DTD/bml_2_1.dtd">
<?bml bml-version="XX.XX" ?>
```

where "XX.XX" represents a value defined in an operational standard.

APPENDIX 4

Operational Guidelines for Implementing Services for Portable Receiving System

Contents

Chapter APPENDIX 4-1 Purpose 715
Chapter APPENDIX 4-2 Scope 716
Chapter APPENDIX 4-3 Legend 717
Chapter APPENDIX 4-4 Guidelines for Coding of BML Documents 718
A4-4.1 Character Coding Schemes 718
A4-4.1.1 Character coding scheme used for BML documents 718
A4-4.1.2 External characters 718
A4-4.1.3 Operation of character reference 718
A4-4.1.4 Operation of character encoding available to contents 719
A4-4.2 Media Types and Operational Range of Monomedia 719
A4-4.2.1 Bitmap 721
A4-4.2.2 Video/Audio 721
A4-4.2.3 8-bit character code 722
A4-4.3 Operation of XML Namespace 722
A4-4.4 Operation of BML Elements 722
A4-4.4.1 XML declaration and DOCTYPE declaration 722
A4-4.4.2 Operation of BML Elements 722
A4-4.4.3 Entity 726
A4-4.4.4 Attribute 726
A4-4.5 Operation of CSS-Based Style Sheet 735
A4-4.5.1 Elements applicable to property 737
A4-4.5.2 Default style sheet 738
A4-4.5.3 Operation of selectors 738
A4-4.5.4 Media type 738
A4-4.5.5 Box model 738
A4-4.5.6 Visual formatting model 739
A4-4.5.7 Other visual effects 740
A4-4.5.8 Colors and Backgrounds 740
A4-4.5.9 Fonts 741
A4-4.5.10 Text 742
A4-4.5.11 Pseudo classes 742
A4-4.5.12 Operation of length 743
A4-4.5.13 Operation of extended properties 743
A4-4.5.14 Actual values derived from computed values 745
A4-4.6 Operational Guidelines on Procedural Description 745
A4-4.6.1 Operational guidelines on DOM 745

A4-4.6.2 Operational scope for built-in objects 759
A4-4.6.3 Operational scope for extended objects for broadcasting 762
A4-4.6.4 Operational scope for browser pseudo objects 762
A4-4.6.4.3 Operation of extended functions 767
A4-4.6.4.4 Operational guideline on functions for communication 767
A4-4.6.5 Operation of Navigator pseudo objects 767
A4-4.7 Restrictions on Description of BML Documents 768
A4-4.7.1 Restrictions on script element 768
A4-4.7.2 Positioning and restrictions 768
A4-4.7.3 Vertical writing 769
A4-4.7.4 Restrictions on arrangement of elements 769
A4-4.7.5 Operational range of object element 769
A4-4.7.5.1 Operation of presenting aGIF/MNG 769
A4-4.7.6 Operational range of p element 770
A4-4.7.6.1 Operation of p element with -wap-marquee property 770
A4-4.7.7 Operation of nesting elements 770
A4-4.7.8 Operational range of textarea element 770
A4-4.7.8.1 Handling markup in textarea element 771
A4-4.8 Operational Guidelines of Presentation 771
A4-4.8.1 Operational guidelines for presenting objects 771
A4-4.8.2 Guideline on clipping 771
A4-4.8.3 Operation of attributes regarding presentation of stream data 772
A4-4.8.4 Guidelines on presentation behavior of aGIF/MNG and operation on streamstatus attribute 773
Chapter APPENDIX 4-5 Guidelines on Behaviors of BML Browsers 774
A4-5.1 Operational Guidelines on Presentation 774
A4-5.1.1 Presentation model 774
A4-5.1.2 Operation of monomedia presentation 774
A4-5.1.3 Operation of colour map data 774
A4-5.1.4 Operation of remote control keys 775
A4-5.1.5 Character entry into input element and textarea element 775
A4-5.1.6 Operation of Focus 775
A4-5.2 Guidelines on Behaviors of External Characters 776
A4-5.3 Guidelines on Behaviors of DOM 776
A4-5.3.1 Use of DOM 776
A4-5.3.2 Handling of control codes (blank, line break, tab) 776
A4-5.3.3 Generation of DOM tree 776
A4-5.3.4 Restriction on DOM work memory 776
A4-5.4 Operational Guidelines on ECMAScript 777
A4-5.4.1 Operation of script working environment 777
A4-5.4.2 Data type 778
A4-5.4.3 Effects on basic objects caused by data type restrictions 779
A4-5.4.4 Operation rule of implementation-dependent portion 780
A4-5.5 Operational Guidelines on Broadcasting Extended Objects 782

A4-5.5.1 Operational guidelines on BinaryTable class 782
A4-5.6 Guidelines on Behavior of Browser Pseudo Objects 782
A4-5.7 Other Restrictions 782
Chapter APPENDIX 4-6 Guidelines on Contents Transmission and Namespace 785
A4-6.1 Scope Mapping to Transmission System 785
A4-6.1.1 Scope mapping for contents received in real time 785
A4-6.1.2 Scope mapping for contents transmitted via interactive communication networks associated with broadcasting services 786
A4-6.2 Guidelines on Reference across Media Types 786
A4-6.2.1 Restrictions on document transmission among media types 786
A4-6.2.2 Restrictions on reference of monomedia and others across media types 787
A4-6.3 Namespace 788
A4-6.3.1 Namespace for BML content received in real time 788
A4-6.3.2 Namespace for BML content transmitted via interactive communication networks associated with broadcasting services 788
A4-6.4 Guidelines on Transmission of BML content 789
A4-6.4.1 Syntax of module 789
A4-6.4.2 Resource list in multi-part formatted module 790
A4-6.4.3 Operation of event messages 790
A4-6.5 Operational Guidelines on Communication Protocols 790
A4-6.5.1 Operational guidelines on HTTP 790
A4-6.5.2 Operational guidelines on mapping between API and protocols 790
A4-6.5.3 Operational guidelines on secure communication 791
Annex A Operated DTD and Default Style Sheet 792
A.1 DTD Operated for Appendix 4 792
A.2 Default Style Sheet 798
Informative Explanation 800
1 Sample presentation for a receiver employing a portable component-based model 800

[BLANK]

Chapter APPENDIX 4-1 Purpose

The Appendix 4 contains the operational guidelines for implementing an integrated service that provides broadcasting services and services via the Internet for a portable receiving system, by using the XML-based multimedia coding scheme specification responsible for the data broadcasting scheme, part of the digital data broadcasting scheme specified as the standard in Japan.

Chapter APPENDIX 4-2 Scope

The guidelines contained in this appendix are applied to an integrated service that provides broadcasting services and services via the Internet for a portable receiving system, by using the XML-based multimedia coding scheme specification responsible for the data broadcasting scheme, part of the digital data broadcasting scheme specified as the standard in Japan.

Chapter APPENDIX 4-3 Legend

O: This symbol marks an item required to implement a service based on a specified sub-profile. It is desirable that a receiver designed for a concerned service should present this item normally when the corresponding signal is contained in a content. However, an item marked by "*" or "Note" may not be available due to restrictions on a media type, a specific implementation, or others.

-: This symbol marks an item not required to implement a service based on a specified sub-profile. Whether or not the item is presented depends on an implementation designed for a concerned service, when the corresponding signal contained in a content to be presented. It is desirable that no signal corresponding to a "-" marked item is contained in a content. However, this does not prevent an operational standard regulation for a specific media type from using a "-" marked item.

Chapter APPENDIX 4-4 Guidelines for Coding of BML Documents

A4-4.1 Character Coding Schemes

Any operation of character coding schemes must comply with the specification defined in Section 5.1, Volume 2.

A4-4.1.1 Character coding scheme used for BML documents

The Shift-JIS character coding scheme must be used for any BML document to be transmitted.

A4-4.1.1.1 Shift-JIS

Table 4-1 shows the operational guideline on the Shift-JIS character codes.

Table 4-1 Shift-JIS Code Set

Code set	Character set	Operating range
1-byte character Byte range 21~7F, A1~DF	JIS X 0201-1976 (JIS Roman Alphabets) Half-width (Hankaku) Katakana	O
2-byte character 1st byte range 81~9F, E0~EF 2nd byte range 40~7E, 80~FC	JIS X 0208-1997 Note that a character with the same kuten code in the Kanji character set is allocated in the corresponding location in the sections 90 ku to 94 ku (free area).	O
Control codes	Blank character (20)	O
	Delete character (7F)	-
	Line feed (0D0A)	O
	Tab (09)	O

A4-4.1.2 External characters

No external characters are operated.

Any external character, if needed, is replaced with bitmap data. Any external character is assumed to be displayed in a default size, which is defined in an operational standard regulation. Whether an external character image presented with bitmap data is fitted into a character size, which is currently presented by a receiver or not depends on the implementation.

A4-4.1.3 Operation of character reference

AA4-4.1.3.1 Shift-JIS

- Character reference by “&#” is not operated.
- HexEscapeSequence, OctalEscapeSequence, and UnicodeEscapeSequence in ECMAScript are not operated.
  And, among SingleEscapeCharacters, “¥b”, “¥t”, and “¥f” are not operated but “¥””, “¥’”, “¥r”, “¥n”, and “¥¥” are operated.
  However,

- The “¥” character must be followed by “0x5c”.
- The “¥r” and “¥n” characters must be used as the combination, “¥r¥n”.

A4-4.1.4 Operation of character encoding available to contents

- Which character encoding specification is internally used by an ECMAScript processor depends on an implementation. Which character encoding specification is applied to functions for comparing lengths of strings and sorting strings also depends on an implementation.
- The character encoding specification used in a Greg or Ureg pseudo object is Shift-JIS.
- The writePersisitentArray() function and readPersistentArray()functions writes and reads information written in Shift-JIS to and from NVRAM, respectively.
- Any binary table contains information written in Shift-JIS.
- The privateData attribute of EventMessageFired contains strings in Shift-JIS.

A4-4.2 Media Types and Operational Range of Monomedia

Table 4-2 shows the operational guideline on the media types required for basic services and the monomedia coding schemes designated by object element.

Table 4-2 Operational Guidelines on Media Types and Monomedia Coding Schemes

Schema	Media type	Operation	Operation (referenced by object element)	Operation (referenced by img element)	Remarks
arib-dc:	multipart/mixed	O	-	-	
	text/xml;charset="euc-jp"	-	-	-	
	text/xml;charset="UTF-8"	-	-	-	
	text/xml;charset="UTF-16"	-	-	-	
	text/xml;charset="Shift-JIS"	-	-	-	
	text/css	O	-	-	(Note 1, 6)
	text/xsl;charset="euc-jp"	-	-	-	
	text/xsl;charset=”Shift-JIS”	-	-	-	
	text/xsl;charset="UTF-8"	-	-	-	
	text/xsl;charset="UTF-16"	-	-	-	
	text/html;charset="euc-jp"	-	-	-	
	text/html;charset="UTF-8"	-	-	-	
	text/html;charset="UTF-16"	-	-	-	
	text/html;charset=”Shift_JIS”	O	-	-	(Note 2)
	text/X-arib-bml; charset="euc-jp"	-	-	-	
	text/X-arib-bml; charset="UTF-8"	-	-	-	
	text/X-arib-bml; charset="UTF-16"	-	-	-	
	text/X-arib-bml; charset="Shift-JIS"	-	-	-	
	text/plain; charset="euc-jp"	-	-	-	
	text/plain; charset="UTF-8"	-	-	-	
	text/plain; charset="UTF-16"	-	-	-	
	text/plain; charset="Shift-JIS"	-	-	-	
	text/X-arib-jis8text	-	-	-	
	text/X-arib-ecmascript; charset="euc-jp"	-	-	-	
	text/X-arib-ecmascript; charset= "UTF-8"	-	-	-	
	text/X-arib-ecmascript; charset= "UTF-16"	-	-	-	

Schema	Media type	Operation	Operation (referenced by object element)	Operation (referenced by img element)	Remarks
	text/X-arib-ecmascript; charset="Shift-JIS"	O	-	-	(Note 2)
	image/jpeg	O	O	O	
	image/png	O	O	O	
	image/gif	O	O	O	(Note 6)
	image/X-arib-png	-	-	-	
	image/X-arib-mng	O	O	-	
	image/X-arib-mpeg2-I	-	-	-	
	image/X-arib-mpeg4-I-simple	-	-	-	
	image/X-arib-mpeg4-I-core	-	-	-	
	audio/X-arib-mpeg2-aac	O	O	-	(Note 3)
	audio/X-arib-mpeg2-bc	-	-	-	
	audio/X-arib-aiff	O	O	-	(Note 3)
	audio/X-arib-additional	-	-	-	
	video/X-arib-mpeg1	-	-	-	
	application/xhtml+xml; charset="euc-jp"	-	-	-	
	application/xhtml+xml; charset="Shift_JIS"	-	-	-	
	application/xhtml+xml; charset="UTF-8"	-	-	-	
	application/xhtml+xml; charset="UTF-16"	-	-	-	
	application/X-arib-stream-text; charset="euc-jp"	-	-	-	
	application/X-arib-stream-text; charset="Shift_JIS"	-	-	-	
	application/X-arib-stream-text; charset="UTF-8"	-	-	-	
	application/X-arib-stream-text; charset="UTF-16"	-	-	-	
	application/X-arib-stream-jis8text	-	-	-	
	application/X-arib-stream-png	-	-	-	
	application/X-arib-stream-jpeg	-	-	-	
	application/X-arib-stream-mpeg2-I	-	-	-	
	application/X-arib-stream-mpeg4-I-simple	-	-	-	
	application/X-arib-stream-mpeg4-I-core	-	-	-	
	application/X-arib-stream-H264-I-baseline	-	-	-	
	application/X-arib-stream-H264-I-main	-	-	-	
	application/X-arib-mpeg2-tts	-	-	-	
	application/X-arib-bmlclut	-	-	-	
	application/X-arib-btable	O	-	-	
	application/X-arib-drcs	-	-	-	
	application/X-arib-PDI	-	-	-	
	application/X-arib-resourceList	O	-	-	
	application/X-arib-storedResourceList	-	-	-	
	application/X-arib-contentPlayControl	-	-	-	
arib:	audio/X-arib-mpeg2-aac	O	-	-	(Note 4)
	audio/X-arib-mpeg2-bc	-	-	-	
	audio/X-arib-mpeg4	-	-	-	
	video/X-arib-mpeg1	-	-	-	
	video/X-arib-mpeg2	-	-	-	
	video/X-arib-mpeg4-simple	O	-	-	(Note 3, 4)
	video/X-arib-mpeg4-core	-	-	-	

Schema	Media type	Operation	Operation (referenced by object element)	Operation (referenced by img element)	Remarks
	video/X-arib-H264-baseline	O	-	-	(Note 3, 4)
	video/X-arib-H264-main	O	-	-	(Note 3, 4)
arib-ic:	image/X-arib-mpeg2-I	-	-	-	
	application/X-arib-stream-mpeg2-I	-	-	-	
	application/X-arib-stream-mpeg4-I-simple	-	-	-	
	application/X-arib-stream-mpeg4-I-core	-	-	-	
	application/X-arib-stream-H264-I-baseline	-	-	-	
	application/X-arib-stream-H264-I-main	-	-	-	
romsound:	audio/X-arib-romsound	O	-	-	

Note 1: Any CSS data set (media type: text/css) can either be specified with the style attribute or be transmitted as a separate resource. When a CSS data set is transmitted as a separate monomedia resource, the CSS data set must require no other CSS data sets to have syntactical completeness.

Note 2: The charset attribute value is required. An ECMAScript data set can be either part of a BML document or be transmitted as a separate resource. When an ECMAScript data set is transmitted as a separate monomedia resource, the ECMAScript data set must require no other ECMAScript-based files to have syntactical completeness.

Note 3: Actual operation is defined in an operational standard regulation for each media type.

Note 4: This is referenced by no element. Actual presentation depends on an implementation.

Note 5: This identifies a built-in sound specified by a broadcaster.

Note 6: Any animation GIF data set, as specified in 4.2.1.1[A.T.1], must not referenced by any "img" element.

A4-4.2.1 Bitmap

A4-4.2.1.1 GIF

Any GIF-based graphics format must comply with what Section 5.5, ARIB STD-B24 Volume 1 Part I contains.

A4-4.2.1.2 Animation GIF (aGIF)

A series of two or more GIF-based still pictures is defined as animation GIF (or aGIF).

A4-4.2.1.3 JPEG

Any JPEG file must comply with the BaseLine format in ISO/IEC 10918-1 (ITU-T T.81) JPEG standard. Apply Section 7.2, ARIB STD-B24 Volume 1 Part 1[A.T.2] is to a JPEG file for proper colorimetry.

A4-4.2.2 Video/Audio

A4-4.2.2.1 MPEG-2 Video stream

No video/audio stream is referenced by a document. Any video/audio stream is presented according to an instruction described in PTE, whose actual operation is defined in an operational defined, synchronized with applicable other documents.

A4-4.2.3 8-bit character code

No 8-bit character code must be operated.

A4-4.3 Operation of XML Namespace

The XML Namespace must operated based on the conventions defined in Chapter 10, Volume 2. When two or more namespaces arise from proprietary data or other elements, these namespaces should be appropriately supported by a receiver.

A4-4.4 Operation of BML Elements

A4-4.4.1 XML declaration and DOCTYPE declaration

Any XML declaration and DOCTYPE declaration must be the following:

* 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”>

A4-4.4.2 Operation of BML Elements

The following table outlines how the BML elements applicable to this appendix must be operated. See Annex A.1 for the DTD to be operated.

Table 4-3 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

Module		BML (Operation)	Element
		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
	Bi-directional	-	bdo
Form	Basic Forms	-	form
		-	input
		-	label
		-	select
		-	option
		-	textarea
	Forms	O	form
		O	input
		-	select
		-	option
		O	textarea
		-	button
		-	fieldset
		-	label
		-	legend
		-	optgroup
Table	Basic Tables	-	caption

Module		BML (Operation)	Element
		-	table
		-	td
		-	th
		-	tr
	Tables	-	caption
		-	table
		-	td
		-	th
		-	tr
		-	col
		-	colgroup
		-	tbody
		-	thead
		-	tfoot
Image		O	img
CS Image Map		-	a&
		-	area
		-	img&
		-	Input&
		-	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

Module		BML (Operation)	Element
		O	script
Stylesheet		-	style
Style Attribute		O	
Link		O	link
Base		-	base
Name Identification (deprecated)		-	a&
		-	applet&
		-	form&
		-	frame&
		-	iframe&
		-	img&
		-	map&
Legacy (deprecated)		-	basefont
		-	center
		-	dir
		-	font
		-	isindex
		-	menu
		-	s
		-	strike
		-	u
		-	body&
		-	br&
		-	caption&
		-	div&
		-	dl&
		-	h1-h6&
		-	hr&
		-	img&
		-	input&
		-	legend&
		-	li&
		-	ol&
		-	p&
		-	pre&
		-	script&
		-	table&
		-	tr&
		-	th&
		-	td&
		-	ul&
BML	BML	-	bml:bml
		-	bml:bevent
		-	bml:beitem
		-	iframe&

Module		BML (Operation)	Element
		-	body&
		-	div&
		-	p&
		-	a&
		-	bdo&
		-	object&
BML	Basic BML	-	bml:bevent
		-	bml:beitem
		-	body&
		-	div&
		-	p&
		-	span&
		-	object&
	Portable BML	O	bml:bevent
		O	bml:beitem
		O	object&

A4-4.4.3 Entity

The following entities are used without explicit definition. Any other entity is optional.

- lt <
- gt >
- amp &
- apos '
- quot "

A4-4.4.4 Attribute

The table 4-4 outlines how the attributes of the BML elements which described to be operated in 4.4.2 (div, v, br, pre, span, a, form, input, textarea, img, object, meta, title, script, link, body, head, html, bml:bevent, bml:beitem) to be operated.

Table 4-4 Operation of Attributes of BML Elements

Legend:

O: This symbol marks an attribute that will be specified in a BML document.

⊗[A.T.3]: This symbol marks an attribute to which a fixed value will be assigned on a receiver and of which no specification will be in any BML document.

Element	Attribute	Operation	Operational Restrictions
Common Attributes			
Core Attributes			

Element	Attribute	Operation	Operational Restrictions
	id	O	String (up to 128 bytes)
	class	O	
	title	-	
I18N Attributes			
	xml:lang	-	Not operated
Events Attributes			The %Events.attrib; attribute is applicable only to the "a", "input", "textarea", and "object" elements.
	onclick	O	
	ondbclick	-	
	onmousedown	-	
	onmouseup	-	
	onmouseover	-	
	onmousemove	-	
	onmouseout	-	
	onkeypress	-	
	onkeydown	O	
	onkeyup	O	
Style Attribute			
	style	O	
Core Modules			
Structure Module			
body	%Core.attrib;	O	
	%Style.attrib;	O	
head	%I18n.attrib;	-	
	profile	-	
html	xmlns	△	xmlns="http://www.w3.org/1999/xhtml" Operated as a fixed value xmlns:bml="http://www.arib.or.jp/bml" Operated as a fixed value
	%I18n.attrib;	-	
	version	-	
title	%I18n.attrib;	-	
Text Module			
blockquote	%Common.attrib;	-	
	%Style.attrib;	-	

Element	Attribute	Operation	Operational Restrictions
	cite	-	
br	%Core.attrib;	O	
	%Style.attrib;	O	
div	%Core.attrib;	O	
	%Style.attrib;	O	
h1-h6	%Common.attrib;	-	
	%Style.attrib;	-	
p	%Core.attrib;	O	
	%Style.attrib;	O	
pre	%Core.attrib;	O	(Note 4)
	%Style.attrib;	O	
	xml:space	△	Fixed to "preserve"
span	%Core.attrib;	O	
	%Style.attrib;	O	
Text Extention Module			
hr	%Common.attrib;	-	
	%Style.attrib;	-	
Hypertext Module			
a	%Common.attrib;	O	
	(%Events.attrib;)	O	%Events.attrib; is applicable to the "a" element.
	%Style.attrib;	O	
	accesskey	O	
	charset	△	Fixed to “Shift_JIS”
	href	O	
	hreflang	-	
	rel	-	
	rev	-	
	tabindex	-	
	type	-	
List Module			
dl	%Common.attrib;	-	
	%Style.attrib;	-	
dt	%Common.attrib;	-	
	%Style.attrib;	-	

Element	Attribute	Operation	Operational Restrictions
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	△	Fixed to "application/x-www.form-urlemcpded"
input	%Common.attrib;	O	
	(%Events.attrib;);	O	%Events.attrib; is applicable to the "input" element.
	%Style.attrib;	O	
	accept	-	
	accesskey	O	
	alt	-	
	checked	-	
	disabled	-	
	maxlength	O	An integer from 1 to 40. (Note 1)
	name	O	
	readonly	O	
	size	-	
	src	-	
	tabindex	-	
	type	O	any one of the following "text", "password", or "submit"
	value	O	

Element	Attribute	Operation	Operational Restrictions
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	%Common.attrib;	O	(Notes 1, 4)
	(%Events.attrib;)	O	%Events.attrib; is applicable to the "textarea" element
	%Style.attrib;	O	
	accesskey	O	
	cols	-	An applied area is specified with CSS [width, height].
	disabled	-	
	name	O	
	readonly	O	
	rows	-	An applied area is specified with CSS [width, height].
	tabindex	-	
	xml:space		Fixed to "preserve"
button	%Common.attrib;	-	
	%Style.attrib;	-	
	accesskey	-	
	disabled	-	
	name	-	

Element	Attribute	Operation	Operational Restrictions
	tabindex	-	
	type	-	
	value	-	
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	Actual representation depends on the implementation of a receiver. Note that its representation is optional.
	longdesc	-	
	height	-	
	width	-	
Object Module			
object	%Common.attrib;	O	
	(%Events.attrib;)	O	%Events.attrib; is applicable to the "object" element.
	%Style.attrib;	O	
	archive	-	
	classid	-	
	codebase	-	
	codetype	-	
	data	O	
	declare	-	
	height	-	
	name	-	
	standby	-	

Element	Attribute	Operation	Operational Restrictions
	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	
select &	onblur	-	
	onfocus	-	
	onchange	-	
textarea&	onblur	-	
	onfocus	-	
	onselect	-	
	onchange	-	
Metainformation Module			
meta	%I18n.attrib;	-	
	http-equiv	-	
	name	O	
	content	O	
	scheme	-	
Scripting Module			
script	id	-	

Element	Attribute	Operation	Operational Restrictions
	charset	△	Fixed to "Shift_JIS"
	type	△	Fixed to "text/X-arib-ecmascript; charset=" Shift_JIS""
	src	O	No more than one script file, which require no other script files to have syntactical completeness, must be referenced.
	defer	-	
	xml:space	△	Fixed to "preserve"
Link Module			
link	%Core.attrib;	-	
	%Style.attrib;	-	
	charset	△	Fixed to "Shift_JIS"
	href	O	
	hreflang	-	
	type	△	Fixed to "text/css"
	rel	△	Fixed to "stylesheet"
	rev	-	
	media	△	Fixed to "tv"
BML Module			
bml:beven t	id	O	String (up to 128 bytes)
bml:beite m	id	O	String (up to 128 bytes)
	type	O	One of these: EventMessageFired, ModuleUpdated, ModuleLocked, TimerFired, DataEventChanged, MediaStopped, MainAudioStreamChanged
	onoccur	O	
	es_ref	O	
	message_group_i d	O	0 or 1 (Note 3)
	message_id	O	
	message_version	O	
	module_ref	O	
	time_mode	O	"absolute" or "origAbsolute"
	time_value	O	
	object_id	O	This attribute is applicable only to ID of the object

Element	Attribute	Operation	Operational Restrictions
			element of which type attribute is "audio/X-arib-mpeg2-acc" or "audio/X-arib-aiff", and that refers to what has been transmitted through a data carousel.
	subscribe	O	
object&	bml:streamstatus	O	
	bml:onfocus	O	
	bml:onblur	O	
	bml:accesskey	O	

Note 1: When a value which is greater than the maxlength value (240 for textarea) is entered for the input or textarea element, the maxlength value is effective for the element. The extra length is not rendered. A single tab (0x09) is counted as a single character. A single blank (0x20) is also counted as a single character.Whether a single line break (0x0D0A) is counted as a single character or two characters depends on the implementation of a receiver.

Note 2: The extended functions for broadcasting applicable to the onunload event handler are limited to readPersistentArray(), writePersistentArray(), unlockModuleOnMemoryEX(), and unlockAllModulesOnMemory(). For rapid document transaction, it is recommended that only functions which require not a long time - writing Ureg data, easy conditional branching, and so on - is used.

Note 3: Actual behaviors when 0 or 1 is specified are defined in an operational standard for each media type.

Note 4: Any specified tab (0x09) is treated as a tab in a DOM node. According to CSS, however, a single tab is rendered as a single blank (0x20). As for the texarea element, characters are rendered from left to right. When the length of characters exceeds the width, the characters are rendered as two or more lines, each of which goes left to right.

A4-4.4.4.1 Restrictions on order of elements in head element

In a head element,a title element, a meta element, a link element, a script element and a bml:bevent element can be used in this order. A single title element is used. No or one meta element, link element, and bml:bevent element are used. No or one script element without a src attribute can be used. No, one, or two script elements with a src attribute can be used.

A4-4.4.4.2 Operation of bml:beitem element

To use DOM to change attribute values dynamically, first, define the subscribe attribute as false. Once the value is changed, define the subscribe attribute as true.

A4-4.4.4.3 Extension of object element

For the purpose of ARIB STD-B24, in order to allow any object element to be focused, the object element is extended to support the bml:onfocus/bml:onblur attribute, an additional attribute.

A4-4.5 Operation of CSS-Based Style Sheet

The table 4-5 outlines operation of properties of CSS2-based style sheets.

Table 4-5 Operation of Properties of CSS-Based Style Sheet[A.T.4]

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
Value assignment/Inheritance	
@import	-
!important	-
Other configuration rules	
@charset	-
@font-face	-
@color-profile	-
Media type	
@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
border-color	-
border-top-style	-
border-right-style	-

Property	Operation
border-bottom-style	-
border-left-style	-
border-style	O
border-top	-
border-right	-
border-botom	-
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 effects	
visibility	O
overflow	O
clip	-
Generated content/Auto numbering/List	
content	-
quotes	-
counter-reset	-
counter-increment	-
marker-offset	-
list-style-type	-
list-style-image	-
list-style-position	-
list-style	-
Page media	
"@page"	-
size	-
marks	-
page-break-before	-
page-break-after	-
page-break-inside	-
page	-
orphans	-
widows	-
Color/ Background	
color	O
background	-

Property	Operation
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	-
Table	
caption-side	-
border-collapse	-
border-spacing	-
table-layout	-
empty-cells	-
speak-header	-
User interface	
outline-color	-
outline-width	-
outline-style	-
outline	-
cursor	-
Voice style sheet	
volume	-
speak	-
pause-before	-
pause-after	-
pause	-
cue-before	-

Property	Operation
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	-
resolution	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	-

A property with a fixed value must be specified as a "!important" rule in a default style sheet. This type of "!important" rules always take precedence over normal rules. No "!important" rules are operated in any BML document, to which only normal rules are applicable. This means that a fixed value is successfully assigned to a certain property, as specified in the default style sheet.

A4-4.5.1 Elements applicable to property

Which BML elements are applicable to each available property is shown in Table 4-6.

Table 4-6 BML Elements Applicable to CSS Property[A.T.5]

	body	div	p	pre	br	span,a	form	input,textarea	object	img
Box model										
margin	-	⊗	⊗	⊗	-	-	⊗	⊗	⊗	⊗
padding-top	-	⊗	O	O	-	-	⊗	O	⊗	⊗
padding-right	-	⊗	O	O	-	-	⊗	O	⊗	⊗
padding-bottom	-	⊗	O	O	-	-	⊗	O	⊗	⊗
padding-left	-	⊗	O	O	-	-	⊗	O	⊗	⊗
border-width	-	O	O	O	-	-	O	O	⊗	⊗
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	⊗	⊗
Visual model										
position	-	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
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	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
line-height	-	-	O	O	⊗	⊗	-	O	-	-
display	⊗	⊗	O	⊗	⊗	⊗	⊗	⊗	⊗	⊗
Other visual effects										
visibility	⊗	O	O	O	-	⊗	O	O	O	O
overflow	-	⊗	⊗	⊗	-	-	⊗	⊗	⊗	⊗
Background										
color	-	-	O	O	-	O	-	O	-	-
background-color	O	O	O	O	-	O	O	O	⊗	⊗
background-image	O	-	-	-	-	-	-	-	-	-
background-repeat	⊗	-	-	-	-	-	-	-	-	-
Font										
font-family	-	-	⊗	⊗	-	⊗	-	⊗	-	-
font-size	-	-	O	O	-	O	-	O	-	-
font-weight	-	-	O	O	-	O	-	O	-	-
Text										
text-align	-	-	O	-	-	-	-	O	-	-
white-space	-	-	⊗	⊗	-	-	-	⊗	-	-
Extended Properties										
resolution	⊗	-	-	-	-	-	-	-	-	-
used-key-list	O	-	-	-	-	-	-	-	-	-
-wap-marquee-style	-	-	O	-	-	-	-	-	-	-
-wap-marquee-loop	-	-	O	-	-	-	-	-	-	-
-wap-marquee-dir	-	-	⊗	-	-	-	-	-	-	-
-wap-marquee-speed	-	-	O	-	-	-	-	-	-	-
-wap-input-format	-	-	-	-	-	-	-	O	-	-

A4-4.5.2 Default style sheet

The default style sheet applicable to this Appendix is defined in Annex A.2.

A4-4.5.3 Operation of selectors

Any selector must be used either in the default style sheet or a CSS data set that is transmitted in a separate resource. The selectors applicable to this Annex are limited to:

* selectors for any element applicable to this Annex e.g.: * { color : white }
* type selectors e.g.: div { color : white }
* class selectors e.g.: div.class { color: white }, .class { color: white }
* ID selectors e.g.: #chap1 { color: white }

No other selector than those listed above must be operated. Note that two or more selectors must not be specified at the same time, even if "," is used to separate two or more selectors. For interfaces to which a specified selector is applicable, refer to Table 4-8 that shows which interfaces are operated.

A4-4.5.4 Media type

The operated media type is limited to tv. The media type must be specified with a @media rule in a style sheet. When a media type other than tv is specified according to a @media rule, any receiver must ignore the designated properties in the @media rule.

A4-4.5.5 Box model

This section outlines how box model properties are operated.

- margin

This property must take the following fixed value.

Fixed value : 0

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

Available value : <length> (0 or an integer in pixels)
Initial value : 0
Inheritance : no

Note that the fixed value, 0, must be assigned for the div element, the object element, the form element, and the img element.

- border-width

This property must be operated as described below.

Available value : <length> (0 or an integer in pixels)
Initial value : 0
Inheritance : no

A common width, as specified with this property, is applied to the four sides. Note that the fixed value, 0, must be assigned for the object element and the img element.

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

This property must be operated as described below.

Available value : <color> | transparent | inherit
Initial value : transparent
Inheritance : no

First, a left border is rendered, then a right border is rendered, followed by a top border and a bottom border in this order. Each corner is overwritten during this rendering process. When border-top-color, border-right- color, border-bottom- color, or border-left- color is not specified, it is assumed that the initial value, transparent is specified. This means that a value specified with the color property applicable to the intended element is not used. For color units, see the description about the color property in 4.5.8.

- border-style

This property must be operated as described below.

Available value :dotted | dashed | solid | none
Initial value : none
Inheritance : no

A common style, as specified with this property, is applied to the four sides. Actual presentation with "dotted", "dashed", or "solid" complies with the definition in the convention defined in Section 8.1, Chapter 8, Part 2, Volume 1. How what is beyond the convention is rendered depends on the implementation of a receiver.

Note that the fixed value, none, must be assigned for the object element and the img element.

A4-4.5.6 Visual formatting model

This section outlines how visual formatting model properties are operated.

- display

This property must contain a fixed value specific to a certain element, as defined in Annex A.2.

The fixed value, block, is used for these elements: div, body, input, textarea, img, object, form, and pre.

The fixed value, inline, is used for the br element, the span, element, and the a element.[A.T.6]

For the p element, the initial value, block, is used. Note that -wap-marquee can be specified for the p element. This means:

Available value :block | -wap-marquee.
Inheritance :no

Note that it is assumed that the fixed value, none, is used for these elements: meta, title, script, head, html, link, bml:bevent, and bml:beitem.

- position

This property must contain a fixed value specific to a certain element, as defined in Annex A.2.

The fixed value, absolute, is used for these elements: p, div, textarea, input, img, object, pre and form.

The fixed value, static, is used for these elements: br, span , and a.

The position property is applied to no other element than those listed above.

- left, top, width, height

This property must be operated as described below. Note that the initial value is defined in the default style sheet in Annex A.2.

Available value : <length> (0 or an integer in pixels)
Initial value : 0
Inheritance : no

- z-index

This property must take the following fixed value.

Fixed value : auto

- line-height

This property must be operated as described below.

Available value : normal | <length> (0 or an integer in pixels) | inherit
Initial value : normal
Inheritance : yes

When this property takes normal, a value that is 1.0 times as great as font-size is assumed. However, for the br element, the span element, and the a element the fixed value, inherit, shall be used. And, the line-height property can contain the value, inherit, only for above three elements.

A4-4.5.7 Other visual effects

This section outlines how other visual effect properties are operated.

- visibility

Available value : visible | hidden | inherit
Initial value : inherit
Inheritance :no

When the visibility property of a parent element is set to hidden and the visibility property of the child element is set to visible, only the parent element is not presented. The fixed value, inherit, is used for the span element and the a element. The fixed value, visible, is used for the body element.

- overflow

This property must take the following fixed value.

Fixed value : hidden

A4-4.5.8 Colors and Backgrounds

This section outlines how colors and backgrounds are operated.

- color

Available value: <color> | inherit

Initial value: black

Inheritance: yes

Color units must be operated in compliance with the convention defined in Section 4.4.1, Appendix 1. Only the six-digit format is applicable as the #rgb option. The rgb(0.0%-100.0%, …) format is not operated. As for the natural language option, the following color designations must be accepted: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow. Whether or not other designations are used is optional.

- background-color

This property must be operated as described below.

Available value : <color> | transparent | inherit
Initial value : transparent
Inheritance : no

The fixed value, transparent, is used for the object element and the img element. For color units, see the description about the color property in 4.5.8.

- background-image

This property must be operated as described below.

Available value : <uri> | none
Initial value : none
Inheritance : no

The graphic image coding specification specified with the <uri> value type must be limited to JPEG. Drow from the upper left corner of the body element.

- background-repeat

This property must take the following fixed value.

Fixed value : repeat

A4-4.5.9 Fonts

The available specifications for a font in a BML document are listed in Table 4-7. Any 1-byte character is rendered with a half width of the font size.

Which font of the three in Table 4-7 is used depends on the implementation of a receiver. It is recommended that when a receiver contains a Gothic font, the Gothic font is used.

Table 4-7 Available Fonts

Font family	Font size
round gothic	small, medium, large

This section outlines how font properties are operated.

- font-family

This property must take the following fixed value.

Fixed value : round gothic

Note that an actual rendered result on a screen depends on the implementation of a receiver.

- font-size

This property must be operated as described below.

Available value: small | medium | large | inherit

Initial value: medium

Inheritance: yes

Designers should take it into consideration that some receivers may not differentiate the small option from the medium option.

- font-weight

This property must be operated as described below.

Available value: normal | bold | inherit

Initial value: normal

Inheritance: yes

How a bold character is rendered depends on a receiver.

A4-4.5.10 Text

This section outlines how text properties are operated.

- text-align

This property must be operated as described below.

Available value: left | right | center

Initial value: left

Inheritance: yes

- white-space

This property must take the following fixed value.

The other fixed value, pre, is used for the pre element and the textarea element.

Fixed value : normal (pre for the pre and textarea elements)

A4-4.5.11 Pseudo classes

No pseudo classes must be operated for the purpose of Appendix 4.

A4-4.5.12 Operation of length

The <length> value must be an integer in pixels, for example "100px". When the property contains "0", which is acceptable and interpreted as "0px", the return value for DOM is "0px".

A4-4.5.13 Operation of extended properties

This section outlines how extended properties are operated.

- used-key-list

This property must be operated as described below. Actual operation of key-group is defined in an operational standard for each media type.

Available value : <key-group>+ | none
Initial value : basic
Inheritance : no

- resolution

This property must take the following fixed value.

Fixed value : 240x480
Inheritance : no

- -wap-marquee-style

This property must be operated as described below.

Available Value: scroll | slide
Apply to: p element
Initial value: scroll
Inheritance: no

How many times the specified action occurs is defined with -wap-marquee-loop.

- -wap-marquee-loop

This property must be operated as described below.

Available Value: <integer> | infinite
Apply to: p element
Initial value: 1
Inheritance: no

The maximum value for a finite loop must be 16. For an infinite loop, "infinite" must be specified. It is acceptable that a receiver may treat the "infinite" option as a certain finite number, doing only a limited times of renderings. When this property contains "0", it is treated as if the specified marquee action has been done.

- -wap-marquee-dir

This property must take the following fixed value.

Fixed value: rtl

This property specifies a direction in which a scroll occurs. The available direction is limited to rtl.

rtl: a scroll from right to left.

- -wap-marquee-speed

This property must be operated as described below.

Available Value: slow | normal | fast

Available Value: p element

Initial value: normal

Inheritance: no

This property specifies how fast a scroll action occurs. Actual speed for each value depends on the implementation of a receiver.

- -wap-input-format

This property must be operated as described below.

Available Value: <format>

Apply to: input, textarea elements

Initial value: "*M"

Inheritance: no

* The options applicable to <format> are the following.

A : Uppercase alphabetic characters (including symbols and punctuations, excluding numerical characters)

a : Lowercase alphabetic characters (including symbols and punctuations, excluding numerical characters)

N : Numerical characters (excluding symbols and punctuations)

n : Numerical characters (including symbols and punctuations)

X : Uppercase alphabetic characters and numerical characters (including symbols and punctuations)

x : Lowercase alphabetic characters and numerical characters (including symbols and punctuations)

M : All characters (Set to uppercase characters as the initial value)

m : All characters (Set to lowercase characters as the initial value)

* The number of characters to be used can be specified. For example, "MMM" or "3M" can be used to specify a three-character format. When there is no need to specify the number of characters, place * before the intended option, for example "*M". Note that this format accepts no combination of two or more options.

* For the available characters, refer to Section 7.3 in Chapter 7, Part 2, Volume 1 of ARIB STD-B24. Note that the characters in the ARIB STD B5 Kanji set (Sections 90 to 94), the tab character (09), and the delete character (7F) are not operated.

* Actual fonts[A.T.7] for numerical characters, alphabetic characters, and symbols are specified in an operational standard.

* The -wap-input-format property does not care the line break (0x0D0A) control code. The value attribute for the input element, however, must not contain a line break. Each receiver is responsible for proper handling of line breaks.

A4-4.5.14 Actual values derived from computed values

An actual value derived from a computed value that is a decimal fraction depends on an implementation.

A4-4.6 Operational Guidelines on Procedural Description

A4-4.6.1 Operational guidelines on DOM

This section provides tables that indicate the available interface, attribute, and method is applicable and use the following symbols:

The O mark means that the interface, attribute, or method must be operated.

The - mark means that the interface, attribute, or method must not be operated.

The R character means that any access to the interface, attribute, or method is limited to a reading access.

The RW string means that any access to the interface, attribute, or method may be a reading access or a writing access.

A4-4.6.1.1 Operation of DOM core interfaces

The table 4-8 outlines how DOM core interfaces are operated. For the purpose of this Appendix, any mechanism that allows a tree structure of documents to change dynamically is not operated.

Table 4-8 Operation of DOM core interfaces

Interface	Operation
Core 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 table 4-9 outlines how attributes of DOM core interfaces are operated.

Table 4-9 Operation of attributes and methods of DOM core interfaces

Interface	Attribute/Method	Operation	Restriction
DOMImplementation	hasFeature()	O	
Document	doctype	-	
	implementation	O	R
	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*
	length	O	R *
	substringData()	-	
	appendData()	-	
	insertData()	-	
	deleteData()	-	
	replaceData()	-	
Element	tagName	O	R
	getAttribute()	-	

Interface	Attribute/Method	Operation	Restriction
	setAttribute()	-	
	removeAttribute()	-	
	getAttributeNode()	-	
	setAttributeNode()	-	
	removeAttributeNode()	-	
	getElementsByTagName()	-	
	normalize()	-	
Text	splitText()	-	
CDATASection			

*: This property is not allowed to access any child node of the script element and the textarea element. This property is not allowed to write on any other nodes, but the p element, and a child node of the span, a, and pre elements. A single tab (0x09) is counted as a single character. A single blank (0X20) is also counted as a single character. Whether a single line break (0x0D0A) is counted as a single character or two characters depends on the implementation of a receiver.

A4-4.6.1.2 Operation of DOM HTML interfaces

Operation of DOM HTML interfaces applicable to events

This section provides the table that indicates which DOM HTML interfaces are applicable to events. This section also contains the table that describes what and how values of the type attribute and the target attribute are obtained by using the BMLEvent interface.

Table 4-10 Operation of attributes and methods of BML interfaces

Interface	Attribute/Method	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 a value obtained through this attribute must be /<component_tag>, as defined in Section 9.2, Volume 2.

Note 2: The format of a value obtained through this attribute must be /<component_tag>/<module_id>, as defined in Section 9.2, Volume 2.

The table 4-11 outlines which value the type element of BMLEvent contains depending on an event.

Table 4-11 Relationship between event and type element of BMLEvent

Event	Value of type
A key, as specified in used-key-list, has been pressed	"keydown" (Note 3)
A key, as specified in used-key-list, has been released	"keyup" (Note 3)
The "Enter" or "Access" key has been pressed to verify a focused element	"click" (Note 3)
The current focus has been turned on	"focus"
The current focus has been turned off	"blur"
A document has been loaded	"load"

Event	Value of type
A document is going to be unloaded	"unload"
A change in the value attribute of the input element has been detected, when the current focus was off the input element	"change"
The submit button specified in the form element is pressed or the submit method of the form element is called	"submit"
An event message has been received	"EventMessageFired"
An updated module has been detected	"ModuleUpdated"
A module has been locked	"ModuleLocked"
The timer, as specified with beitem, has been fired	"TimerFired"
A presentation by a monomedia decoder has stopped (Note 2)	"MediaStopped"
An updated data_even_event has been detected	"DataEventChanged"
Another selection of the main audio stream has been verified	"MainAudioStreamChanged"
A set of global code has started to be executed or setInterval() was executed to trigger the specified function	undefined (Note 1)

Note 1: In this case, the target attribute is set to null.

Note 2: The monomedia types that can cause this event must be limited to audio/x-arib-mpeg2-acc and audio/X-arib-aiff.

Note 3: Whether or not the "Enter" key can cause events applicable to the input element (excluding the input element with the type attribute containing "submit") and the textarea element depends on a receiver.

Table 4-12 Relationship between executed event handlers and type/target values

Condition for reading access			Value of type attribute	Value of target attribute
Event handler onkeydown is executed			"keydown"	Element specifying the event handler
Event handler onkeyup is executed			"keyup"	Element specifying the event handler
Event handler onclick is executed			"click"	Element specifying the event handler
Event handler onload is executed			"load"	Element specifying the event handler
Event handler onunload is executed			"unload"	Element specifying the event handler
Event handler onchange is executed			"change"	Element specifying the event handler
Event handler onfocus is executed			"focus"	Element specifying the event handler
Event handler onblur is executed			"blur"	Element specifying the event handler
Event handler onsubmit is executed			"submit"	Element specifying the event handler
Event handler onoccur is executed	Event type	EventMessageFired	"EventMessageFired"	Element specifying the event handler
		ModuleUpdated	"ModuleUpdated"	
		ModuleLocked	"ModuleLocked"	
		TimerFired	"TimerFired"	
		DataEventChanged	"DataEventChanged"	
		MediaStopped	"MediaStopped"	
		MainAudioStreamChanged	"MainAudioStreamChanged"	
Global code is executed			undefined	null
Specified procedure description is executed upon invoking setInterval(), extended function for broadcasting			undefined	null

Operation of interfaces applicable to document nodes

The table 4-13 indicates which and how DOM HTML interface and DOM BML interface is applicable to document nodes.

Table 4-13 Operation of interfaces applicable to document nodes

Interface	Attribute/Method	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 (*)

* When a value obtained through each attribute in the above table is assigned to a global variable, how the interface behaves is implementation dependent.

Operation of DOM HTML interfaces applicable to element nodes

The table 4-14 describes which and how DOM HTML interfaces are applicable to element nodes.

Table 4-14 Operation of DOM HTML interfaces applicable to element nodes

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
html	HTMLElement	id	-	
		className	-	
		title	-	
		lang	-	
		dir	-	
	HTMLHtmlElement	version	-	
bml:bevent	HTMLElement	id	O	R
		className	-	
		title	-	
		lang	-	
		dir	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
	BMLBeventElement			
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 represented by Node	Applicable Interface	Attribute/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	-	
br	HTMLElement	id	O	R
		className	O	R
		title	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		lang	-	
		dir	-	
	HTMLBRElement			
	BMLBRElement	style	-	
		normalStyle	O	RW (Note 2)
		focusStyle	-	
		activeStyle	-	
span	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLSpanElement			
	BMLSpanElement	style	-	
		normalStyle	O	RW (Note 2)
		focusStyle	-	
		activeStyle	-	
		accessKey	-	
		focus()	-	
		blur()	-	
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()	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		focus()	-	
	BMLAnchorElement	style	-	
		normalStyle	O	RW (Note 2)
		focusStyle	-	
		activeStyle	-	
		effect	-	
form	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	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

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		size	-	
		src	-	
		tabIndex	-	
		type	O	R
		useMap	-	
		value	O	RW
		blur()	-	
		focus()	-	
		select()	-	
		click()	-	
	BMLInputElement	style	-	
		normalStyle	O	RW (Note 2)
		focusStyle	-	
		activeStyle	-	
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

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		height	-	
		isMap	-	
		longDesc	-	
		src	O	RW (Note 1)
		useMap	-	
		width	-	
	BMLImageElement	style	-	
		normalStyle	O	RW (Note 2)
		focusStyle	-	
		activeStyle	-	
object	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLObjectElement	form	-	
		code	-	
		archive	-	
		codeBase	-	
		codeType	-	
		data	O	RW (Note 1)
		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	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		streamStatus	O	RW
		streamLooping	-	
		streamSpeedNumerator	-	
		streamSpeedDenominator	-	
		streamLevel	-	
		setSpeed()	-	
		movePosition()	-	
		hasAssociatedIndex()	-	
		assignToLocalEvnet()	-	
		assignToNodePlayMode()	-	
		getMainAudioStream()	-	
		setMainAudioStream()	-	
		focus()	-	
		blur()	-	
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	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		dir	-	
	HTMLLinkElement	disabled	-	
		charset	-	
		href	-	
		hreflang	-	
		media	-	
		rel	-	
		rev	-	
		target	-	
		type	-	

Note 1 : To allow a resource presented by the img or object element to be rewritten, the resulting effective presentation area must be identical with the previous effective presentation area. Otherwise, how the img or object element is presented depends on the implementation of a receiver. To allow a resource presented by the img element to be rewritten, the resulting media type must be identical with the previous media type.

Note 2: This attribute can be accessed through the BMLCSS2Properties interface. When this attribute is read directly, an object must be returned. Otherwise, this attribute must not be read directly.

Operation of BMLCSS2Properties interface

The table 4-15 outlines how the BMLCSS2Properties interface is operated.

Table 4-15 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	-	

Attribute	Operation	Remarks
borderRightStyle	-	
borderBottomStyle	-	
borderLeftStyle	-	
borderStyle	O	R
borderTop	-	
borderRight	-	
borderBottom	-	
borderLeft	-	
border	-	
Visual formatting model		
position	-	
left	O	R
top	O	R
width	O	R
height	O	R
zIndex	-	
lineHeight	-	
verticalAlign	-	
display	-	
bottom	-	
right	-	
cssFloat	-	

Attribute	Operation	Remarks
clear	-	
direction	-	
unicodeBidi	-	
maxHeight	-	
minHeight	-	
maxWidth	-	
minWidth	-	
Other visual effects		
visibility	O	RW
overflow	-	
clip	-	
Generated content/Auto numbering/List		
content	-	
quotes	-	
counterReset	-	
counterIncrement	-	
markerOffset	-	
listStyleType	-	
listStyleImage	-	
listStylePosition	-	
listStyle	-	
Page media		
size	-	
marks	-	
pageBreakBefore	-	
pageBreakAfter	-	
pageBreakInside	-	
page	-	
orphans	-	
widows	-	
Background		
background	-	
backgroundColor	O	RW
backgroundImage	-	
backgroundRepeat	-	
backgroundPosition	-	
backgroundAttachment	-	
Font		
color	O	RW
fontFamily	-	
fontStyle	-	
fontSize	-	
fontVariant	-	
fontWeight	O	RW
font	-	
fontStretch	-	
fontSizeAdjust	-	
Text		
textIndent	-	
textAlign	O	R
textDecoration	-	
textShadow	-	
letterSpacing	-	
wordSpacing	-	
textTransform	-	
whiteSpace	-	
Table		
captionSide	-	
borderCollapse	-	
borderSpacing	-	

Attribute	Operation	Remarks
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 extension		
borderColorIndex	-	
borderTopColorIndex	-	
borderRightColorIndex	-	
borderrLeftColorIndex	-	
borderBottomColorIndex	-	
backgroundColorIndex	-	
colorIndex	-	
grayscaleColorIndex	-	
outlineColorIndex	-	
clut	-	
resolution	-	
displayAspectRatio	-	
navIndex	-	
navUp	-	
navDown	-	
navLeft	-	
navRight	-	
usedKeyList	O	RW
WapMarqueeStyle	O	R
WapMarqueeLoop	O	R
WapMarqueeDir	-	
WapMarqueeSpeed	O	R
WapAccesskey	-	
WapInputFormat	O	R
WapInputRequired	-	

The above DOM attribute values to be operated conform to the conventions on operation of the CSS2 properties.

A4-4.6.2 Operational scope for built-in objects

Table 4-16 shows an operational guideline on the operational scope for the ECMAScript built-in objects.

Table 4-16 Operational Guidelines on the ECMAScript Built-in Objects

Built-in Object	Method/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 operated	
	prototype	O	
	Object([value])	O	
	new Object([value])	O	
Object.prototype		All operated	
	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 operated	
	constructor	O	
	toString()	O	(Note 3)
Array		All operated	
	prototype	O	
	length	O	
	Array(item0, item1, . . .)	O	
	new Array(item0, item1, . . .)	O	
	new Array([len])	O	
Array.prototype		All operated	
	constructor	O	
	toString()	O	
	join([separator])	O	
	reverse()	O	
	sort([comparefn])	O	
String		All operated	
	prototype	O	
	length	O	
	String([value])	O	

Built-in Object	Method/Property	Operation	Remarks
	new String([value])	O	
	String.fromCharCode(char0[, char1, . . .])	O	
String.prototype		All operated	
	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 operated	
	prototype	O	
	Boolean([value])	O	
	new Boolean([value])	O	
Boolean.prototype		All operated	
	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 operated	
	constructor	O	
	toString([radix])	O	
	valueOf()	O	
Math		None operated	
	E	-	
	LN10	-	
	LN2	-	
	LOG2E	-	
	LOG10E	-	
	PI	-	
	SQRT1_2	-	
	SQRT2	-	
	abs(x)	-	
	acos(x)	-	
	asin(x)	-	
	atan(x)	-	
	atan2(y, x)	-	
	cos(x)	-	
	exp(x)	-	
	floor(x)	-	
	log(x)	-	

Built-in Object	Method/Property	Operation	Remarks
	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)
	setMonth(mon [, date])	O	(Note 6)

Built-in Object	Method/Property	Operation	Remarks
	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	
	toUTCString()	O	
	toGMTString()	-	(Note 5)

Note 1: This is not operated, because it is related to Float.
Note 2: This is not operated, because it is related to Unicode.
Note 3: How this behaves is defined in Appendix 2, Volume 2.
Note 4: This is not operated, because it is related to Number.
Note 5: This is not operated, because it is specified to maintain backward compatibility with source codes written ealier.
Note 6: This is operated with restricted specification, because it is related to Number.
Note 7: The value of radix for parseInt() must be "8, 10, or 16". A value "0" is interpreted as "10".

A4-4.6.3 Operational scope for extended objects for broadcasting

Table 4-17 shows an operational guideline on extended objects for broadcasting with respect to ECMAScript.

Any defined extended object for broadcasting that is not listed in Table 4-9 is assumed that its Operation is “-”.

Table 4-17 Operation of Extended Objects for Broadcasting

Object	Method/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	

A4-4.6.4 Operational scope for browser pseudo objects

A4-4.6.4.1 Script description

Any document must not contain more than one Global Object.

A4-4.6.4.2 Operational Scope for Browser Pseudo Objects

The table 4-18 provides operational guidelines on browser pseudo objects.

Legend for the Operation column.

O: This symbol marks a basic function required to conform to this Appendix.

O (*1): This symbol marks an optional function for this Appendix. To invoke this type of function by a content, the getBrowserSupport() function must be used to verify whether or not a receiver supports the function to allow the function to be invoked only when the receiver support the function.

O (*2): This symbol marks a basic function for this Appendix. This means that this type of function is usually required, except in the case where a receiver does not support any communication function via a BML browser. In this case, when the function is called, the return value "Failure" must be returned.

O (*3): This symbol marks a function required for a receiver with a function to support the Partial TS format. To invoke this type of function by a content, the getBrowserSupport() function must be used to verify whether or not a receiver supports the function to allow the function to be invoked only when the receiver support the function.

-: This symbol marks a function that is not a basic function nor an optional function. When a BML document containing the function is used, an error occurs in a receiver.

Table 4-18 Operational Scope for Browser Pseudo Objects

Function		Operation	Remarks
Ureg functions			
	Ureg[]	O	
Greg functions			
	Greg[]	O	
EPG functions			
	epgGetEventStartTime()	-	
	epgGetEventDuration()	-	
	epgTune()	O	
	epgTuneToComponent()	-	
	epgTuneToDocument()	-	
	epgIsReserved()	O	
	epgReserve()	O(*1)	
	epgCancelReservation()	O(*1)	
	epgRecIsReserved()	-	
	epgRecReserve()	-	
	epgRecCancelReservation()	-	
Program group index functions			
	grpIsReserved()	-	
	grpReserve()	-	
	grpCancelReservation()	-	
	grpRecIsReserved()	-	
	grpRecReserve()	-	
	grpRecCancelReservation()	-	

Function		Operation	Remarks
	grpGetNodeEventList()	-	
	grpGetERTNodeName()	-	
	grpGetERTNodeDescription()	-	
	epgXTune()	-	
Series reservation functions			
	seriesIsReserved()	-	
	seriesReserve()	-	
	seriesCancelReservation()	-	
	seriesRecIsReserved()	-	
	seriesRecReserve()	-	
	seriesRecCancelReservation()	-	
Non-volatile memory functions			
	readPersistentString()	-	
	readPersistentNumber()	-	
	readPersistentArray()	O	
	writePersistentString()	-	
	writePersistentNumber()	-	
	writePersistentArray()	O	
	copyPersistent()	-	
	getPersistentInfoList()	-	
	deletePersistent()	-	
	getFreeSpace()	-	
Functions for access controlled non-volatile memory			
	isSupportedPersistentType()	-	
	setAccessInfoOfPersistentArray()	-	
	checkAccessInfoOfPersistentArray()	-	
	writePersistentArrayWithAccessCheck()	-	
	readPersistentArrayWithAccessCheck()	-	
Extended APIs for storing - Directory management functions			
	saveDirAs()	-	
	saveDir()	-	
	createDir()	-	
	getParentDirName()	-	
	getDirNames()	-	
	isDirExisting()	-	
Extended APIs for storing - File management functions			
	saveFileAs()	-	
	saveFile()	-	
	getFileNames()	-	
	isFileExisting()	-	
Extended APIs for storing - File input/output functions			
	writeArray()	-	
	readArray()	-	
Extended APIs for storing - Inquiry Functions			
	getDirInfo()	-	
	getFileInfo()	-	
	getContentSource()	-	
	getStorageInfo()	-	
	getCarouselInfo()	-	
	getModuleInfo()	-	
Extended APIs for storing - Data carousel storage functions			
	saveCarouselAs()	-	
	saveCarousel()	-	

Function		Operation	Remarks
	saveModuleAs()	-	
	saveModule()	-	
	saveResourceAs()	-	
	saveResource()	-	
Interaction channel functions			
Interaction channel functions - Communication Functions assuming Basic procedure			
	connect()	-	
	disconnect()	-	
	sendBinaryData()	-	
	receiveBinaryData()	-	
	sendTextData()	-	
	receiveTextData()	-	
Interaction channel functions - Delayed call functions			
	registerTransmission()	-	
	registerTransmissionStatus()	-	
	getTransmissionStatus()	-	
	setDelayedTransmissionDataOverBasic()	-	
Interaction channel function - Communication Function using the Mass call reception service			
	vote()	-	
Interaction channel functions - Functions for encrypted communication using CAS			
	startCASEncryption()	-	
	endCASEncryption()	-	
	transmitWithCASEncryption()	-	
Interaction channel functions - Functions for communication with public key encryption not using CAS			
	setEncryptionKey()	-	
	beginEncryption()	-	
	endEncryption()	-	
Interaction channel functions - Communication functions assuming TCP/IP			
	setISPParams()	-	
	getISPParams()	-	
	connectPPP()	-	
	connectPPPWithISPParams()	-	
	disconnectPPP()	-	
	getConnectionType()	O (*2)	
	isIPConnected()	O (*2)	
	saveHttpServerFileAs()	-	
	saveHttpServerFile()	-	
	sendHttpServerFileAs()	-	
	saveFtpServerFileAs()	-	
	saveFtpServerFile()	-	
	sendFtpServerFileAs()	-	
	sendTextMail()	-	
	trasmitTextDataOverIP()	O (*2)	
	setDelayedTransmissionData()	-	
	setCacheResourceoverIP()	-	
Interaction channel function - For retrieving delayed call status common to BASIC-based procedures and IP connections			
	getDelayedTransmissionStatus()	-	
	getDelayedTransmissionResult()	-	
Interaction Channel functions - Function for obtaining line connection status			
	getPrefixNumber()	-	
Operational control functions			

Function		Operation	Remarks
	reloadActiveDocument()	O	
	getNPT()	-	
	getProgramRelativeTime()	O	
	isBeingBroadcast()	-	
	lockExecution()	-	
	unlockExecution()	-	
	lockModuleOnMemory()	-	
	unlockModuleOnMemory()	-	
	setCachePriority()	-	
	getLinkSourceServiceStr()	-	
	getLinkSourceEventStr()	-	
	getIRDID()	O	
	getBrowserVersion()	O	
	getProgramID()	O	
	getActiveDocument()	O	
	lockScreen()	O	
	unlockScreen()	O	
	getBrowserSupport()	O	
	launchDocument()	O	
	launchDocumentRestricted()	-	
	quitDocument()	O	
	launchExApp()	-	
	getFreeContentsMemory()	O	
	isSupportedMedia()	-	
	detectComponent()	O	
	lockModuleOnMemoryEx()	O	
	unlockModuleOnMemoryEx()	O	
	unlockAllModulesOnMemory ()	O	
	getLockedModuleInfo()	O	
	getBrowserStatus()	O	
	getResidentAppVersion()	O	
	isRootCertificateExisting()		
	getRootCertiricateInfo()		
	startResidentApp()		
	getDataDisplayrAreaSize()		
	setFullDataDisplayArea()	O (*1)	
Receiver sound control			
	playRomSound()	O	
Timer functions			
	sleep()	-	
	setTimeout()	-	
	setInterval()	O	
	clearTimer()	O	
	pauseTimer()	-	
	resumeTimer()	-	
	setCurrentDateMode()	O (*3)	
External character functions			
	loadDRCS()	-	
	unloadDRCS()	-	
Other functions			
	random()	O	
	subDate()	O	
	addDate()	O	

Function		Operation	Remarks
	formatNumber()	O	
Subtitle presentation control functions			
	setCCStreamReference()	-	
	getCCStreamReference()	-	
	setCCDisplayStatus()	-	
	getCCDisplayStatus()	-	
	getCCLanguageStatus()	-	
Functions for controlling bookmark areas			
	writeBookmarkArray()	-	
	readBookmarkArray()	-	
	deleteBookmark()	-	
	lockBookmark()	-	
	unlockBookmark()	-	
	getBookmarkInfo()	-	
Functions for printing			
	getPrinterStatus()	-	
	printFile()	-	
	printTemplate()	-	
	printUri()	-	
	printStaticScreen()	-	
	saveImageToMemoryCard()	O (*1)	
	saveStaticScreenToMemoryCard()	O (*1)	
	saveHttpServerImageToMemoryCard()	-	

A4-4.6.4.3 Operation of extended functions

* Operation of lockScreen()

To use the lockScreen() function, the contents author should be aware of the execution of unlockScreen() in a following process.

* Operation of unlockScreen()

To use the lockScreen() function, the contents author should be aware of the execution of unlockScreen() in a following process .

* Operation of reloadActiveDocument()

Even if a connection for communication fails to be established, the reloadActiveDocument() must not be invoked repeatedly.

* Operation of setInterval()

It is recommended that invoking the setInterval() does not overload a receiver. No more than four timers must be specified at a time.

A4-4.6.4.4 Operational guideline on functions for communication

To use any function that involves communication, the function must not be invoked without an explicit interaction of end users in the concerned content.

A4-4.6.5 Operation of Navigator pseudo objects

No Navigator pseudo-objects are operated.

A4-4.7 Restrictions on Description of BML Documents

A4-4.7.1 Restrictions on script element

* The applicable operations of browser pseudo-objects regarding to script descriptions are limited to describing procedures in a CDATA section node that is the sole child element of the script element and reading an external script file via the src attribute.
* The script element may be contained only in a header object.
* A header object must contain no or one script element without the src attribute and no, one, or two script elements with the src attribute. The description part involving the script element is enclosed with <![CDATA[and]> to be a separate CDATA section.
* Any function used as an event handler must contain no arguments. The resulting valid syntax for specifying an event handler is:

  attr="func();" (where attr is an attribute designation of an event handler and func is a function name)
* Operating automatic semicolon insertion is optional. Refer to ECMA-262 (page 20).

A4-4.7.2 Positioning and restrictions

This section provides guidelines on designation of position for the elements located in the body element. Any BML document, as defined in ARIB STD-B24, must be associated with a style sheet. Although this standard does not require that each element sticks to a presentation unique to the element, any element specified in a BML document must be presented to end users in a proper fashion.

The restrictions described below cover only the elements that occupy two-dimensional space by placing them in the body element. This does not mean that including other elements in the body element is inhibited.

In the sentences of restrictions, "the size and position of a box is specified by absolute positioning" is referred to simply as "a position is specified." The positioning method uses relative coordinates from a parent element (equivalent to containing block of CSS2) as follows:

left:	Distance from the left of box of parent element (number of pixels);
top:	Distance from top of box of parent element (number of pixels);
width:	Width (number of pixels);
height:	Height (number of pixels);
visibility:	"visible", "hidden" or "inherit";

The position attribute must be fixed to absolute. Therefore, it is not specified in the style part.

In addition, the following terms are used in the sentences below:

boxed element: The followings are named generically boxed elements:

a) elements having two-dimensional space (input, textarea)
b) elements having intrinsic dimension (img,object)
c) div elements
d) form elements
d) p elements
d) pre elements

normal flow element: The elements that can be used in operation other than boxed elements (br, span, a elements) are named normal flow elements.

Note:

The following elements are neither boxed elements nor normal flow elements: html, head, body, meta, script, link, title, bml:bevent, bml:beitem.

Restriction 1: A position of any boxed element must be specified.

Restriction 2: A position of any normal flow element must not be specified.

Restriction 3: A child element of any body element must be either a div, form, p or pre element.

Restriction 4: The div element and the form element can include only boxed elements. Nesting div element(s) allows nest of position designation.

Restriction 5: The p element and the pre element are allowed to have one of normal flow elements or a text line as its child element.

Restriction 6: No nesting is applied to the span element and the a element.

Note that the restrictions listed above supersede what a DTD contains.

Example:

abcdefghij

As defined in CSS2, absolute positioning or normal flow can be specified for each element. The concept underlying the aforementioned rule is, however, to completely separate the line of position-specified elements from the line of normal flow elements.

A4-4.7.3 Vertical writing

Specifying a writing direction in a BML document is not operated. No vertical writing is used.

A4-4.7.4 Restrictions on arrangement of elements

When a div element has as its child element the object element associated with aGIF/MNG graphic items, the background color (background-color) of the div element must be a color other than the colors to be rendered. In this case, no other element than the div element must be presented as the background of the aGIF/MNG gfapic items, with an exception in which an area overlapping with the aGIF/MNG graphic items consists of pixels in the non-drawing colours. There must be no overlapping area among the associated aGIF/MNG graphic items.

A4-4.7.5 Operational range of object element

A4-4.7.5.1 Operation of presenting aGIF/MNG

This section defines how an attribute is interpreted when aGIF/MNG graphic items are applied to the object and how an aGIF/MNG graphic item is presented on a screen.

* Making any change in the data attribute is allowed only while noaGIF/MNG graphic item is rendered.
* It depends on the implementation of a receiver whether a rerendering the presentation frame of an aGIF/MNG graphic item is suspended or is continued during the period when rerendring the screen is prohibited by the execution of the lockScreen() function.

A4-4.7.6 Operational range of p element

A4-4.7.6.1 Operation of p element with -wap-marquee property

This section defines how the p element with the -wap-marquee property set is operated and how it is presented on a screen.

* Updating what the p element with the -wap-marquee property set contains is allowed only when visibility is set to hidden.
* A child element applicable to the p element is limited to a single PCDATA node. The span, a, and br elements cannot be a child element of the p element.
* What a p element with the -wap-marquee property set can contain is limited to PCDATA, which has no line break resulting in a single line and of which length is no more than 128 bytes. Extra characters are not presented.
* Any single BML document must not contain two or more elements with the -wap-marquee property set. When a BML document contains one -wap-marquee property, the BML document must have no object element referencing an aGIF/MNG file.
* The background color of the p element with the -wap-marquee property set must not be transparent.
* It depends on the implementation of a receiver whether repositioning of what the p element contains is suspended or is continued during the period when rerendring the screen is prohibited by the execution of the lockScreen() function.
* No overlapping area must exist between the presentation area defined by the p element and an background-image item. No foreground object must exist on the presentation area defined by the p element.
* When the value of visibility is switched from hidden to visible, rendering starts at the initial state of an object to be presented.
* Any single BML document must not contain two or more elements with the -wap-marquee property set.
* No line break is applied, resulting in a single line of the characters.

A4-4.7.7 Operation of nesting elements

No nesting is applied to the span element and the a element. This means any span element and a element contain no other elements but the br element, the sole exception.

A4-4.7.8 Operational range of textarea element

Reading/writing a string entered into a textarea element is implemented by using the value attribute of the textarea element, as defined in Table 4-14. Note that no string entered into a textarea element is contained in a CDATA node that is a child element of the textarea element.Take it into consideration that reading/writing does not applied to a child node of the textarea element.

A string that is the initial value of a child element of the textarea can be read by using the defaultValue attribute.

Whether or not a DOM has textAreaElement.firstChild.data depends on whether or not first child node exists. When the textarea element has no child elements, no child node is generated. This means even if characters are entered into the textarea element, no dynamic generation of a child node occurs. Therefore, no BML document that results reading/writing textAreaElement.firstChild.data must be made. In case of reading/writing textAreaElement.firstChild.data is executed, the following behavior depends on the implementation of a receiver.

A4-4.7.8.1 Handling markup in textarea element

Any string specified as the initial value of the textarea must contain no markup elements that are defined in Section 2.4 Character Data and Markup in XML 1.0. For example, it depedns on a receiver how the textarea element behaves for initialization when comment lines including Example 1 and Exmaple 2 are specified or an initial value is set with a CDATA section.

Example 1:

<textarea>abc<!--d-->efg</textarea>

Example 2:

<textarea>abc<![CDATA[d]]>efg</textarea>

As the sole exception, referencing any of the entities, as defined in Section A4-4.4.3, is allowed to consist of the initial value of the textarea element. The example 3 is a sample valid descrition.

Example 3:

<textarea><test></textarea>

A4-4.8 Operational Guidelines of Presentation

A4-4.8.1 Operational guidelines for presenting objects

The operation rule for object presentation is given below. Objects are presented in the appearance order. In other words, the elements that appeared first in a BML document are presented at the inner part (position far from the viewer) of the screen. The relationship between the DOM object structure and presentation is illustrated below in Fig. 4-1. DOM objects generate a tree structure in the order of element presentation. Objects 1 to 9 are presented in the order of 1, 2, 4, 5, 7, 8, 9, 3 and 6 if having the structure shown below.

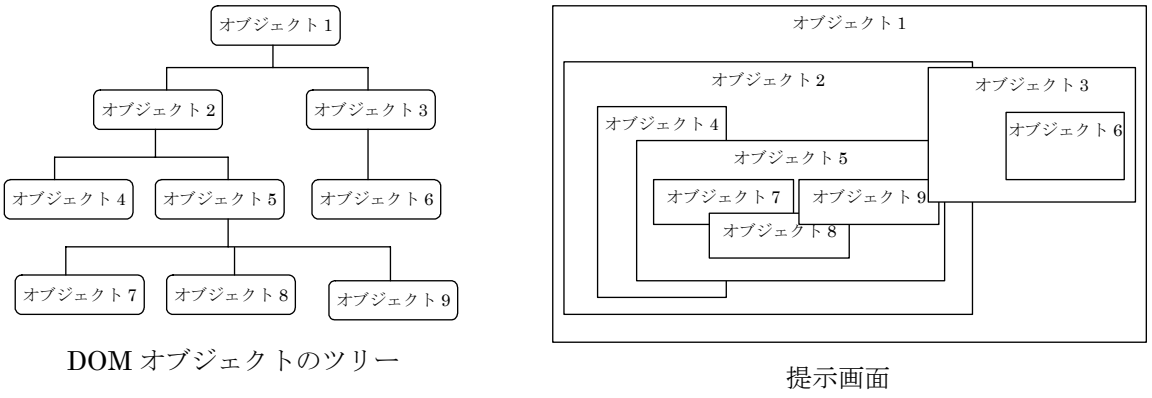


Figure 4-1 The DOM object structure and presentation

A4-4.8.2 Guideline on clipping

This guideline allows a child element to be clipped by a parent element in order to fix the overflow property to hidden. As shown left in Figure 4-2, the content is clipped if sticking out of the rectangular area of the specified element (object 1). Moreover, as shown right below, if the child element (object 3) is clipped by the parent object (object 2), any areas other than the rectangular area specified by the parent element are hidden.

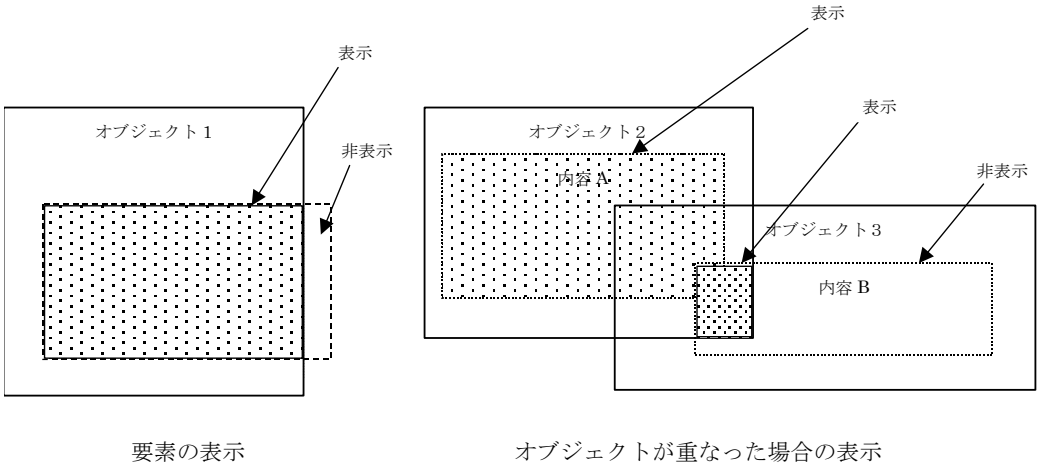


Figure 4-2 Example of Clipping

The above clipping operation is not performed by the parent div element when a still picture, an aGIF image or a MNG image is specified by the object or img element. In other words, monomedia image specified by the object or img element must not be specified out of area with the parent div element. The above clipping operation also is not performed by the parent div element when the p element contains marquee. Clipping that involves two or more generations behaves in other ways than the above. No description must instruct an ancestor element to perform a clipping.

A4-4.8.3 Operation of attributes regarding presentation of stream data

The table 4-19 outlines an operational guideline on the streamstatus attribute of the object element.

Table 4-19 Operation of attributes regarding presentation of stream data

type attribute	streamstatus
audio/X-arib-mpeg2-aac	play, stop (Note 2)(Note 4)
audio/X-arib-aiff	play, stop (Note 2)(Note 4)
image/jpeg	play (Note 1)
image/png	play (Note 1)
image/gif	play, stop, pause (Note 2)(Note 3)
image/X-arib-mng	play, stop, pause (Note 2)

Note 1: The initial value of streamstatus for an object element referencing media of type attribute which takes 'play' only is 'play'.

Note 2: The initial value of streamstatus for an object element referencing media of type attribute which can take 'stop' is 'stop'.No MediaStopped event arises in the initial state. Note that dynamically changing type attributes and dynamically changing schemas by changing data attributes for sound are not applicable to the object element.

Note 3: The initial value of streamstatus is 'play'.When this attribute is not applied to an aGIF file, no descrbing the streamstatus attribute nor describing any other script to modify the status. No modifying the data attribute to switch between GIF and aGIF is allowed.

Note 4: When the playback ends, streamstatus is automatically set to 'stop'.

A4-4.8.4 Guidelines on presentation behavior of aGIF/MNG and operation on streamstatus attribute

This section contains guidelines on the interpretation of the attribute and the presentation behavior when aGIF/MNG is specified in the object element. The table 4-20 explains how aGIF/MNG behaves depending onf the initial value of the streamstatus attribute.

Table 4-20 Inital values applicable to streamstatus attribute

Initial value of the streamstatus attribute	Behavior
play	A playback starts at the first frame.
stop	No frame is displayed. The frame to be played first is reset to the first frame.
pause	The first frame is presented. The current playback is assumed to be tha first round.

Also, table 4-21 explains how aGIF/MNG is presented when the streamstatus attribute is modified using DOM API.

Table 4-21 Access to streamstatus attribute

The former value of the streamstatus	Set for play	Set for stop	Set for pause
play	-	* The current playback is stopped and no frame is displayed.	It depends on the implementation of a receiver which frame is displayed.
stop	* A playback starts at the first frame.	-	* The first frame is presented
pause	* A playback starts. It depends on the implementation of a receiver at which frame the playback starts.	* No frame is displayed.	-

Chapter APPENDIX 4-5 Guidelines on Behaviors of BML Browsers

A4-5.1 Operational Guidelines on Presentation

A4-5.1.1 Presentation model

This section defines a component-based model for a portable receiving system. This component-based model uses an area for presenting informative content to present still pictures, characters, and graphics. The requirements for the component-based model are the following.

(1) Uniqueness of presentation is not required.

- There are no specifications on the number of pixels of an area for presenting informative content nor font size.

(2) A component-based model must have two or more standard presentation patterns and develop rules and assumptions for presentation.

- Any informative content must be produced to ensure that it complies with these rules and supports these assumptions to be successfully presented in these presentation patterns.
- A receiver is expected to be capable of optimizing presentation as far as operation of a concerned content complies with the specifications defined in this section. If a presentation is not optimized, at least, a presentation must work.

(3) A receiver may employ more display formats than formats required to support assumed presentation patterns.

- This implies certain consideration on implementing a receiver is needed to ensure that presentation works regardless of the existence of display formats that do not support assumed presentation patterns.

(4) Any presentation of video data and audio data is controlled by functions in a receiver, not by any informative content. This implies that a receiver is expected to optimize position and size for presentation.

A4-5.1.2 Operation of monomedia presentation

Actual operation of monomedia presentation is defined in an operational standard.

A4-5.1.3 Operation of colour map data

The color names reserved for this Appendix are: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, yellow, and transparent which means that the applied area is made transparent.

The table 5-1 provides an example showing how 16 reserved color names (excluding transparent) relate to RGB values. This is an informative example, which allows other relationships to be operated.

Table 5-1 Sample Relationship between Color Names and RGB Values

color-name	R	G	B
white	255	255	255
black	0	0	0
red	255	0	0
lime	0	255	0
yellow	255	255	0

color-name	R	G	B
blue	0	0	255
fuchsia	255	0	255
aqua	0	255	255
grey	128	128	128
green	0	128	0
maroon	128	0	0
navy	0	0	128
olive	128	128	0
purple	128	0	128
silver	192	192	192
teal	0	128	128

A4-5.1.4 Operation of remote control keys

The mapping among remote control keys, key codes, and characters specified as access keys are shown in the Table 5-3.

Table 5-2 Remote control keys, key codes, and characters for access keys

Remote control keys	Key codes	Characters for access keys
↑	1	N/A
↓	2	N/A
←	3	N/A
→	4	N/A
0, 1, 2, 3, 4, 5, 6, 7, 8, 9,	5-14	N/A
10,11,12 (Note 1)	15-17	N/A
"Enter" (Note 2)	18	N/A
"Back" (Note 2)	19	N/A

Note 1: When these additional numeral keys are equipped with a remote control, how they are operated is defined in an operational standard.
Note 2: Actual keys should be defined in an operational standard.

The above key codes can be taken out by the keyCode attribute of the BMLIntrinsicEvent interface.

A4-5.1.5 Character entry into input element and textarea element

Characters applicable to the input and textarea element are defined in an operational standard.

A4-5.1.6 Operation of Focus

The nav-index property is not operated. A receiver is responsible for proper navigation.

It is recommended that a receiver is equipped with a function for presenting an element that has a focus.

Because the nav-index property is not operated, a receiver must automatically identify an element that needs a focus to perform a proper navigation. An element that satisfies one of the following conditions needs a focus.

(1) the input element, the select element, the textarea element, and the a element with the href attribute

(2) an element that has one of the following attributes: onclick, onfocus, onblur, onkeydown, or onkeyup

Whether or not the input element (excluding the input element with the type attribute specified as "submit") with the readonly attribute and without the onclick, onkeydown, nor onkeyup attribute has a focus depends on the implementation of a receiver. Whether or not the textarea element with the readonly attribute and without the onclick, onkeydown, nor onkeyup attribute has a focus depends on the implementation of a receiver. Whether or not an element with the accesskey attribute, failing to satisfy the two conditions listed above has a focus depends on the implementation of a receiver.

A4-5.2 Guidelines on Behaviors of External Characters

No coding of external characters is operated. When an external character is needed, it is represented as bitmap data.

A4-5.3 Guidelines on Behaviors of DOM

A4-5.3.1 Use of DOM

Any DOM is used based on the convention defined in Section A2-5.3.1, APPENDIX 2.

However, the hasFeature( ) method returns true when the feature attribute is "BML" and the version attribute is a version specified in an operational standard.

A4-5.3.2 Handling of control codes (blank, line break, tab)

The control codes (blank, line break, tab) used for the pre element and the textarea element are handled based on the convention, as defined in Section 3.2, "User Agent Conformance", the W3C Recommendation, "XHTML1.0" (http://www.w3.org/TR/2000/REC-xhtml1-20000126#uaconf), being left intact (with the exception of leading and trailing control codes). Whether or not control codes in a child element of the pre element are left intact depends on the implementation of a receiver. Any control code used for an element other than the pre element are handled based on the convention, as defined in Section A2-5.3.2, Appendix 2. Note that an element that is not the p element, the span element, nor the a element and has a child element containing a control code in a string is governed by the convention provided in Table 5-11, APPENDIX 2.

A4-5.3.3 Generation of DOM tree

Any DOM tree is generated according to the description in Section A2-5.3.4, APPENDIX 2.

A4-5.3.4 Restriction on DOM work memory

The following Table 5-4 defines the maximum number of DOM objects allowed to be generated per item.

Table 5-3 Maximum Value of DOM Objects

Item	Maximum value	Remarks
Number of elements	512	
The total number of nodes	1024	
Size of character string set to NodeValue	1024 (bytes)	

Number of child nodes to be connected to one node	255	
Depth of tree structure	16	The depth of a node immediately under the HTMLElement node is 1.The HTMLElement node is not counted.

A4-5.4 Operational Guidelines on ECMAScript

A4-5.4.1 Operation of script working environment

The restriction provided in Table 5-5 is applied to a script working environment.

Table 5-4 Restrictions on Script Work Memory

Item	Maximum value	Remarks
Length of a symbol name character string	255 bytes	
Function arguments	16	
Local variables	64	
Total length of all character strings	32768 bytes	The total length of strings (including evaluated values of string equations, string constants, string valiables) and symbol names.
Instances of objects	128	Total number of instances of Object, Number, String, Boolean, Array, Date, BinaryTable and, Function.
Properties of one instance of one object	64	Maximum number of instances of Object, Number, String, Boolean, Array, Date, BinaryTable or, Function. For Array, the number of properties that correspond subscripts are excluded.
Elements of one array	1024	
Nest levels with function for invoking	16	Including functions invoked through event handlers.
Total number of properties of all objects	2048	The total number of properties to which the "properties of one instance of one object" restriction applied. Including number of properties of activation object and arguments object. Excluding built-in properties of global object (built-in functions, built-in objects, extended functions for broadcasting, extended objects for broadcasting) and properties of host object for DOM.
Global variables	64	
Function definitions	64	Any function is defined globally. Excluding event handlers.
Work memory for ECMAScript	16384 steps	Based on the computing method defined in ANNEX C, APPENDIX 2.

A4-5.4.2 Data type

It is assumed that any data type has the following restrictions:

- Number must be of single precision (32 bits). Note that any Number object must also be of single precision (32 bits).
- Float must not be supported.
- Math built-in object must not be supported.
- Dynamic type conversion must be restricted.
- The run-time interpretation of a script character string must not be supported.
- Shift-JIS must be used as a character code of character string data.
- Use of Unicode Value must be restricted.
- Functions for compatibility with old codes must be restricted.
- The argument and the operation result of addDate() and subDate() must be restricted from 1900/3/31 to 2100/2/28.

A4-5.4.2.1 Character string

For operation of character string data, the following restrictions are assumed:

- Shift-JIS must be used for description of script itself.
- UnicodeEscapeSequence (\uxxxx) in literal description must not be supported.

A4-5.4.2.2 Type conversion

The restrictions on dynamic type conversion are assumed below.

- A type of method argument is defined to restrict dynamic type conversion in DOM API and broadcast extended classes (The method of Browser quasi-object and extended object for broadcasting).
- Since ECMAScript has no variable type, there is no structure of examining a type on calling a function. Any data type can be assigned in the case of calling an extended function for broadcasting as argument having value of the other type of definition. Calling a function is prohibited if type can not be converted.
- Four basic types have been defined and regulated as mentioned below according to the operating restrictions.
  - [1] Numeric type (Number) must be a 32-bit single precision integer.
  - [2] Character string type (String) must be a character string coded by Shift-JIS.
  - [3] Logical type (Boolean) has no particular restrictions.
  - [4] Object type (Object) has no particular restrictions.
- If the argument is specified as NaN where it must be a Number type and there is no particular description, it must be taken as a failure instead of 0.

The rules of conversions are listed in Table 5-6.

Table 5-5 Rule of Type Conversion

Value	Character string	Number	Logic	Object
Character string	-	Numeric value of	true	String

		character string		
Empty character string	-	0	false	String
0	"0"	-	false	Number
NaN	"NaN"	-	false	Number
Number	Numeric character string	-	true	Number
True	"true"	1	-	Boolean
False	"false"	0	-	Boolean
Object	toString()	valueOf()	true	-
Null	"null"	0	false	-
Undefined value	"undefined"	NaN	false	(error)

A4-5.4.3 Effects on basic objects caused by data type restrictions

Data restriction causes the following language effects:

A4-5.4.3.1 Effects caused by number object of signed 32-bit integer

- Behavior of Date object
  - Calculation of Date object is returned in milli- second order as Number object. However overflow or underflow can be assumed if Number object is under 64bit. subDate() and addDate() are used in calculation of Date object.
  - The precision of Date object must be kept while internally processed (not being affected by the restriction on Number object).
- Operation range of DOM object
  - valueOf() method is not supported.
  - getTime() method is not supported. (A method for individually acquiring hours, minutes and seconds is used instead.)
  - setTime() method is not supported. (A method for individually acquiring hours, minutes and seconds is used instead.)
  - Date.parse() method and Date.UTC() method are not supported. (Numeration of time representation is restricted.)
  - Operations on Date object are not supported. Object the guidelines when operating the Number object of less than 64 bits. (subDate() function or addDate() function is used instead.)
  - [[Value]] characteristics of date object instance depend on an implementation.
- Number object
  - MAX_VALUE attribute must be 2147483647. (Maximum value of signed 32-bit integer)
  - MIN_VALUE attribute must be 1. (Minimum value of signed 32-bit integer)
  - If the value exceeds 2˜31, the process depends on the implementation.
  - If the string "Infinity" is not treated particularly and converted by ToNumber, the return value is NaN.
  - The exponential expression is not used in the operation.

A4-5.4.3.2 Behavior in the case of not using Float

- Number object

    - NaN attribute is not expressed with IEEE754 but depend on an implementation.
    - MAX_VALUE attribute and MIN_VALUE attribute are defined with integer number which can be expressed by Number object.
    - NaN is returned in the case of division by 0.
    - NEGATIVE_INFINITY attribute is not supported.
    - POSITIVE_INFINITY attribute is not supported.
    - NaN attribute must be able to be distinguished from an integer which can be expressed by Number object. (Internal representation must depend on an implementation)
    - The result of a division by 0 returns NaN.
- No Math objects are supported.
- Global object
    - The infinity attribute is not supported.
    - The parseFloat() function is not supported.
    - The isFinite() function is not supported.

A4-5.4.3.3 Effects by restrictions on run-time interpretation of script character string

- The eval() function of Global object is not supported.
- Constructor new Function() of Function object is not supported.

A4-5.4.3.4 Restrictions of functions left for compatibility with old codes

- Date object
    - The getYear() method is not supported. (getFullYear() method is used.)
    - The setYear() method is not supported. (setFullYear() method is used.)
    - The toGMTString() method is not supported. (toUTCString() method is used.)

A4-5.4.3.5 Creation of Data object

When newData() is executed but the current time is not available, then the behavior depends on the implementation.

A4-5.4.4 Operation rule of implementation-dependent portion

The operation rule of the implementation-dependent portion is shown below.

A4-5.4.4.1 Main syntax

- Sequence of the properties taken out by "for (variable in Expression) statement"(ECMA-262 p.58)

  It must remain implementation-dependent. The "for (variable in Array object)" sequence is also implementation-dependent.

A4-5.4.4.2 Host object

- Host object range
  - DOM object
  - Browser pseudo object
  - Ureg pseudo object
  - Greg pseudo object
- Result of typeof for host object (ECMA-262 p.43)

  Returns "hostobject."
- Result of new Object(hostobject) for host object (ECMA-262 p.66)

  Returns reference to hostobject in the same manner as when using a normal object as an argument.
- Results of Array.prototype.join(), Array.prototype.reverse() and Array.prototype.sort() for host object (ECMA-262 p.71,72,73, Section 15.4.4.3)

  Inhibits addition of any property to the host object.

  Remarks: There is no need to consider this behavior because it becomes impossible to insert Array.prototype.join, etc. as a new property of the host object.
- [[Class]] of host object

  "hostobject."

A4-5.4.4.3 Built-in object

- [[prototype]] (ECMA-262 p.62) of Global object

  null
- Result of Function.prototype.toString() (ECMA-262 p.69)

  "function FUNCTIONNAME() {}"(FUNCTIONNAME is the name of a specified function.)
- Result of Array.prototype.sort() (ECMA-262 p.72)

  An array element that is not influenced on sorting may not be automatically generated as an undefined. The internal comparison sequence of sort() may depend on an implementation.
- Result of Number.prototype.toString(radix) without radix = 10 (ECMA-262 p.81)

  Only radix = 8,10, or 16 is applicable. Otherwise, operation depends on an implementation.

  radix = 8: ["0"-"7"]*

  radix = 16: ["0"-"9","a"-"f"]*
- Result of Date.prototype.toString() (ECMA-262 p.95)

  Must be in the format of "DateThoursminutesseconds."

  Date must be YYYY-MM-DD. (Ex. 1999-01-01)

  Hours, minutes and seconds must be hh:mm:ss. (Ex.: 23:01:34)

  'T' (character code 0x54) must be used as a delimiter between the date and the hours, minutes and seconds. (Ex.: 1999-01-01T23:01:34)

  If the result is a negative value, the low-order four digits are used and the sign (d.c. or a.c.) is ignored.

- Results of Date.prototype.toLocaleString() and Date.prototype.toUTCString() (ECMA-262 p.100, 101)

  Must be of the same output format as Date.prototype.toString().

- LocalTZA of Date object must be fixed to +9 hours of Japan Standard Time.
- TimeClip() clip range (ECMA-262 15.9.1.14)

  Must be within the range of signed 64-bit integers.

A4-5.4.4.4 Implementation of event handler

The code type of the event handler must be an implementation supplied code. Also, neither ImplicitThis nor ImplicitParent must be set. Even if an event handler return value is false, the next processing is performed continuously regardless of its value.

A4-5.5 Operational Guidelines on Broadcasting Extended Objects

A4-5.5.1 Operational guidelines on BinaryTable class

The size of the BinaryTable object is defined in an operational standard.

A4-5.6 Guidelines on Behavior of Browser Pseudo Objects

How Browser pseudo objects behave is defined in an operational standard.

A4-5.7 Other Restrictions

For the purpose of scalability, the following keywords and object names, which are operated in JavaScript, are regarded as reserved. If any of the following reserved words is used as a variable name or other, it is not assured that no unexpected behavior occurs.

ECMA-262 2nd keywords

break	for	new	var
continue	function	return	void
delete	if	this	while
else	in	typeof	with

ECMA-262 2nd reserved words

abstract	do	import	short
boolean	double	instanceof	static
byte	enum	int	super
case	export	interface	switch
catch	extends	long	synchronized
char	final	native	throw
class	finally	package	throws
const	float	private	transient
debugger	goto	protected	try

default	implements	public	volatile

ECMA-262 3rd keywords

break	else	new	var
case	finally	return	void
catch	for	switch	while
continue	function	this	with
default	if	throw	
delete	in	try	
do	instanceof	typeof	

ECMA-262 3rd reserved words

abstract	enum	int	short
boolean	export	interface	static
byte	extends	long	super
char	final	native	synchronized
class	float	package	throws
const	goto	private	transient
debugger	implements	protected	volatile
double	import	public	

Objects

	STD-B24	ECMA-262 3rd	Core JavaScript 1.5	Client JavaScript 1.3
Built-in	Object	Object	Object	Object
	Function	Function	Function	Function
	Array	Array	Array	Array
	String	String	String	String
	Boolean	Boolean	Boolean	Boolean
	Number	Number	Number	Number
	Date	Date	Date	Date
	Math	Math	Math	Math
		RegExp	RegExp	RegExp
		Error		
		EvalError		
		RangeError		
		ReferenceError		
		SyntaxError		
		TypeError		
		URIError		
			JavaArray	JavaArray
			JavaClass	JavaClass
			JavaObject	JavaObject
			JavaPackage	JavaPackage
			netscape	netscape
			Packages	Packages
			sun	sun
Host object (DOM/extended/pseudo/host)	document			document
	CSVTable			
	BinaryTable			
	browser			

	STD-B24	ECMA-262 3rd	Core JavaScript 1.5	Client JavaScript 1.3
	XMLDoc			
				Anchor
				Applet
				Area
				Button
				Checkbox
				event
				FileUpload
				Form
				Frame
				Hidden
				History
				Image
				Layer
				Link
				Location
				MimeType
	navigator			navigator
				Password
				Plugin
				Radio
				Reset
				screen
				Select
				Style
				Submit
				Text
				Textarea
				window

Top-Level Properties and Functions

	STD-B24 (Operation)	ECMA-262 3rd	Core JavaScript 1.5	Client JavaScript 1.3
	NaN	NaN	NaN	NaN
		Infinity	Infinity	Infinity
		undefined	undefined	undefined
		eval()	eval()	eval()
	parseInt()	parseInt()	parseInt()	parseInt()
		parseFloat()	parseFloat()	parseFloat()
		escape()	escape()	escape()
		unescape()	unescape()	unescape()
	isNaN()	isNaN()	isNaN()	isNaN()
		isFinite()	isFinite()	isFinite()
		decodeURI()	decodeURI()	
		decodeURIComponent()	decodeURIComponent()	
		encodeURI()	encodeURI()	
		encodeURIComponent()	encodeURIComponent()	

* It is not required but recommended that the properties and functions in italics are regarded as reserved.

Each object is associated with functions converting to the basic type (Number(obj), String(obj)).

Chapter APPENDIX 4-6 Guidelines on Contents Transmission and Namespace

A4-6.1 Scope Mapping to Transmission System

A BML document switching operation is performed when:

- the a element in a BML document is used to switch documents.
- the launchDocument( ) function is used to switch documents in a script of a BML document.
- other service than the currently selected service is selected.
- a signal that forces a document switching is received.
- the submit function of the form element represents a document switching.

If a scope identifier representing a scope level of a BML document becomes invalid after the above switching operation, the corresponding BML document gets out of the scope. The following three sections define scope mapping for BML contents received in real time, BML contents transmitted via interactive communication networks associated with broadcasting services, and stored BML contents.

A4-6.1.1 Scope mapping for contents received in real time

For a content received in real time, scopes, mapping among scope identifiers, and transmission signals is operated as defined in Table 6-1.

Table 6-1 Transmission and Scope Mapping

Scope	Scope identifier	Corresponding transmission signal
Within document	Document scope identifier	moduleId, resourceName
Within document group	Document group scope identifier	data_event_id
Within content	Content scope identifier	component_tag
Within content group	Content group scope identifier	service_id

Figure 6-1 shows an example in which content group A has one entry content and two contents, and content group B has one content being broadcasted. The following explains this transmision example using the numbers in Figure 6-1.

A tuning operation selects a given BML document to be started. In the example below, content 2 begins.

[1] When documents are switched within the same data_event_id in content 2, the operating scope is defined as "between the documents" and "within the document group".

[2] When more than one data_event_id value is used to switch documents in content 2, the operating scope is defined as "between the document groups" and "within the content".

[3] When data_event_id is switched, content 2 is forced to terminnate by the return_to_entry_flag signal, and the entry content starts, the operating scope is defined as "between the contents" and "within the content group".

[4] When the a element or the launchDocument( ) function in the document causes the entry content to switch to content 2, the entry content terminates and content 2 starts. This operating scope is defined as "between contents" and "within the content group".

[5] When an end user’ s operation causes content 2 to switch to another content group B, content 2 terminates and content 1 of the group starts. This operating scope is defined as "out of the content group".

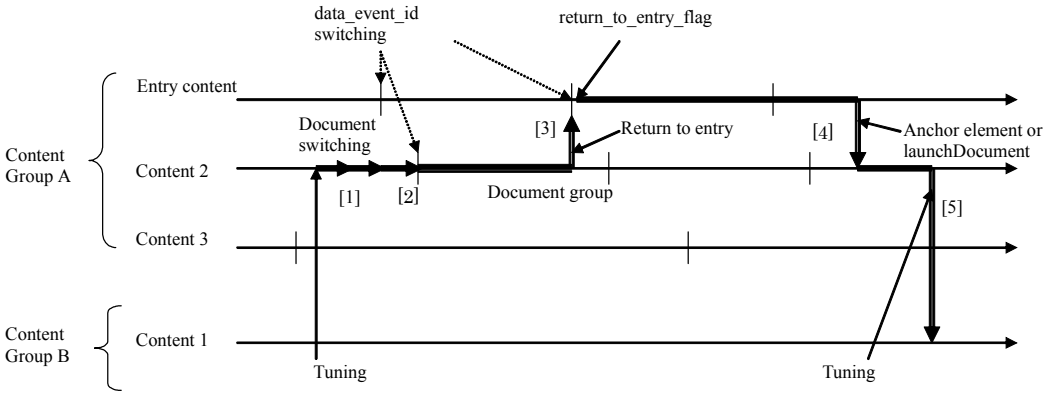


Figure 6-1 Example of Contents Transmission

A4-6.1.2 Scope mapping for contents transmitted via interactive communication networks associated with broadcasting services

The concepts of "document group", "content", and "content group" are not applied to contents transmitted via interactive communication networks associated with broadcasting services.

However, when a content transmitted via interactive communication networks is identified as a primary link destination from a content received in real time or a content stored from a broadcasting service, the content transmitted via interactive communication networks is regarded as a member of the same content group as the linking content. Specifically, in the following case, a content transmitted via interactive communication networks is identified as a primary link.

- A BML document transmitted via interactive communication networks is in or under the same path (directory) as a BML document that is the first destination of a linking BML document received in real time or a linking stored BML document.

A4-6.2 Guidelines on Reference across Media Types

A4-6.2.1 Restrictions on document transmission among media types

Among contents received in real time and contents transmitted via interactive communication networks which content is allowed to be specified as a linked destination of a BML document is defined in Table 6-2. In this table, the following transition types are represented by the numbers 1 through 4.

[1] transition caused by epgTune()

[2] transition caused by a value in the href attribute of the a element

[3] transition caused by launchDocument()/launchDocumentRestriced()

[4] transition caused by the submit function of the form element

Table 6-2 Restrictions on Document Transmission across Media Types[JI8]

	Transiting (Linking) Content	
	Broadcast (contents received in real time)	Communication (contents transmitted via interactive communication network)

Transited (Linked) Content	[1]	[2]	[3]	[4]	[5]	[1]	[2]	[3]	[4]	[5]
Broadcast	O	O (Note 1)	O (Note 1)	-	O	O	O (Note 2)	O (Note 2)	-	O (Note 2)
Communi -cation	-	O	O	O		-	O	O	O	

Note 1: This description is applicable to BML documents belong to the same service.

Note 2: This description is applicable to BML documents belong to the currently selected service.

Among the transition patters in Table 6-2, when a transition from a content transmitted via interactive communication networks to a broadcast content involving a service selection operation is caused by the 2 through 5 transition types described above how a receiver behaves depends on an implementation. Note that this will be superseded by an operational standard defined outside of ARIB STD-B24 specific to a media type.

A4-6.2.2 Restrictions on reference of monomedia and others across media types

The following three items are assumed as resources referenced by a BML document. The operational convention on these items are summarized in Table 6-3.

[1] A monomedia that can be specified with the src attribute of the img element or the data attribute of the object element (e.g. JPEG, PNG, MNG, GIF, aGIF)

[2] A binary table

[3] An event message referred by the beitem element

The following table contains no convention on a function transmitting messages with TCP/IP.

Table 6-3 Resources Referenced by BML Document

			Referencing BML Document		Remarks
			Content received in real time (Broadcast)	Content transmitted via interactive communication network (Communication)	
Referenced Resource	Broadcast	[1] JPEG/PNG/MNG /GIF/aGIF	Within the same carousel (Note 1)	-	For Broadcast to Broadcast reference, not more than one carousel specified with the same data_event_id is allowed to be referenced at a time.
		[2] Binary Table	O Within the same carousel (Note 1)	-	
		[3] Event Message	O Within the same service	O (Note 2)	
	Communi -cation	[1] JPEG/PNG/MNG/GI F/aGIF	-	O	
		[2] Binary Table	-	O	

Note 1: This description is also applicable to resources contained in a module that has been transmitted in the same ES and locked, even if the content does not transmitted in the same carousel.

Note 2: This description is only applicable when a referencing communication content is identified as a trusted content in either of the following methods. Which method, including the combination of the two, is operated is defined in an operational standard for each media.
- Identifying the referencing communication content as a primary link destination transited by a referenced broadcast content.
- Identifying the content as a trusted content using a certificate (Details of certificates are defined in an operational standard.)

A4-6.3 Namespace

A4-6.3.1 Namespace for BML content received in real time

- Any resource specification containing arib or arib-dc complies with the convention defined in Volume 2.
- However, in a component-base model, it is allowed that the href attribute of the element contains the id attribute of other element in a BML document in the following format.
  #<id attribute>

A4-6.3.2 Namespace for BML content transmitted via interactive communication networks associated with broadcasting services

- A specification of a content transmitted via interactive communication networks associated with broadcasting services begins with http.
- A specification of a resource to be obtained via secure communication begins with https.
- The namespace for URI complies with the definition provided in RFC 1808[1] and the additional restrictions described below are applied.
  - When an access to a directory is requested (the URL ends with /), how a server behaves depends on the server and how a receiver behaves depends on the implementation.
  - The representation "./" or "../" is applicable only to the beginning of a relative URL.
  - It is assumed that ~ is not interpreted by a receiver.
  - The representation,;params, is not operated.
  - The namespace for a CGI argument complies with the namespace defined in RFC 1808.
  - A directory name or file name may contain a single period,".", unless the name begins nor ends with the period.

The valid syntax is shown below.

```
URL          = (absoluteURL | relativeURL ) [ "#" fragment ]
absoluteURL  = scheme ":" "//" net_loc "/" rel_path
relativeURL  = [ ("./" | "../" | "/" ) ] rel_path
rel_path     = [ path ] [ "?" query ]
path         = fsegment *( "/" fsegment )
fsegment     = 1*fpchar [ "." 1*fpchar ]
```

---

[1] Note that this description does not exclude the possibility that defining a URI may involve referring to other standards than RFC 1808.

query = *(uchar | reserved)

fragment = 1*fchar ["." 1*fchar]

fpchar = fchar| ":" | "@" | "&" | "="

fchar = alpha | digit | "-" | "_" | "+"

pchar = uchar | ":" | "@" | "&" | "="

uchar = unreserved | escape

unreserved = alpha | digit | safe | extra

escape = "%" hex hex

hex = digit | "A" | "B" | "C" | "D" | "E" | "F" | "a" | "b" | "c" | "d" | "e" | "f"

alpha = lowalpha | hialpha

lowalpha = "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" |"j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z"

hialpha = "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z"

digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

safe = "$" | "-" | "_" | "." | "+"

extra = "!" | "*" | "'" | "(" | ")" | ","

national = "{" | "}" | "|" | "\" | "^" | "~" | "[" | "]" | "`"

reserved = ";" | "/" | "?" | ":" | "@" | "&" | "="

punctuation = "<" | ">" | "#" | "%" | <">

- The maximum length of a URI specification must be 255 bytes.
- A shortened form of URI is interpreted based on RFC 1808 and the additional restrictions described below.
  - Any tilde ~ character is interpreted as interpreted by a server. A receiver does not interpret the character.
  - A URI specification containing the representation "./" or "../" is interpreted as a relative URI to a file or directory in which the current BML file resides by a receiver. Note that this restriction is applicable only to a URI specification that does not begin with "./" nor "../". For example, the specification, foo/../bar is not operated.
  - The slash "/" character is interpreted as the root directory of a server by a receiver.

A4-6.4 Guidelines on Transmission of BML content

A4-6.4.1 Syntax of module

- Each entity-body that is part of another entity-body in the multi-part format must not be of the multi-part type as media type.
- The entity headers that any basic receiver is required to interpret are listed in Table 6-4.

Table 6-4 Entity Headers Interpreted by Any Basic Receiver

Entity header	Semantics

Content-Type	Indicates a media type of a resource contained in an entity body. Define Content-Type for multipart entity body as "multipart/mixed" and specify a boundary parameter.
Content-Length	Indicates the byte length of an entity body.
Content-Location	Indicates the identifier of an entity body.

- The Content-Base:header and the Content-Encoding field are optional.
- Any Content-Location field specification contains only <resourceName> as a relative specification to the module. The namespace for <resourceName> complies with the namespace for resources, as defined in Chapter 9, Volume 2.

A4-6.4.2 Resource list in multi-part formatted module

Any module in the multi-part format that is to be received in real time must contain a resource list at the beginning.

A4-6.4.3 Operation of event messages

The character encoding specification for private_data_byte of a generic event message descriptor must be "Shift_JIS".

A4-6.5 Operational Guidelines on Communication Protocols

A4-6.5.1 Operational guidelines on HTTP

The version of HTTP to be supported by a receiver must be 1.1. Any receiver must be able to successfully connect with any server operating under HTTP/1.0. Note that this will be superseded by an operational standard defined outside of ARIB STD-B24 specific to a media type.

A4-6.5.2 Operational guidelines on mapping between API and protocols

To use http to make a BML content access to a communication server, the methods listed in the following table are applicable.

Table 6-5 Methods applicable to accesses using http

Access	Applicable Method
To obtain a BML document from a server by including http in the href attribute in the a element	GET method is used
To obtain a resource from a server by including http in the data attribute in the object element	GET method is used
To obtain a BML document from a server by including http in an argument in the launchDocument() function	GET method is used
To communicate with a server by using the form element	GET method or POST method is used depending on a value in the action attribute
To communicate with a server by using the transmitTextDataOverIP() function	POST method is used

A4-6.5.3 Operational guidelines on secure communication

How secure communication is operated is defined in an operational standard.

Annex A Operated DTD and Default Style Sheet

A.1 DTD Operated for Appendix 4

```
<!-- 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
    %Core.attrib;
    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 -->
```

A.2 Default Style Sheet

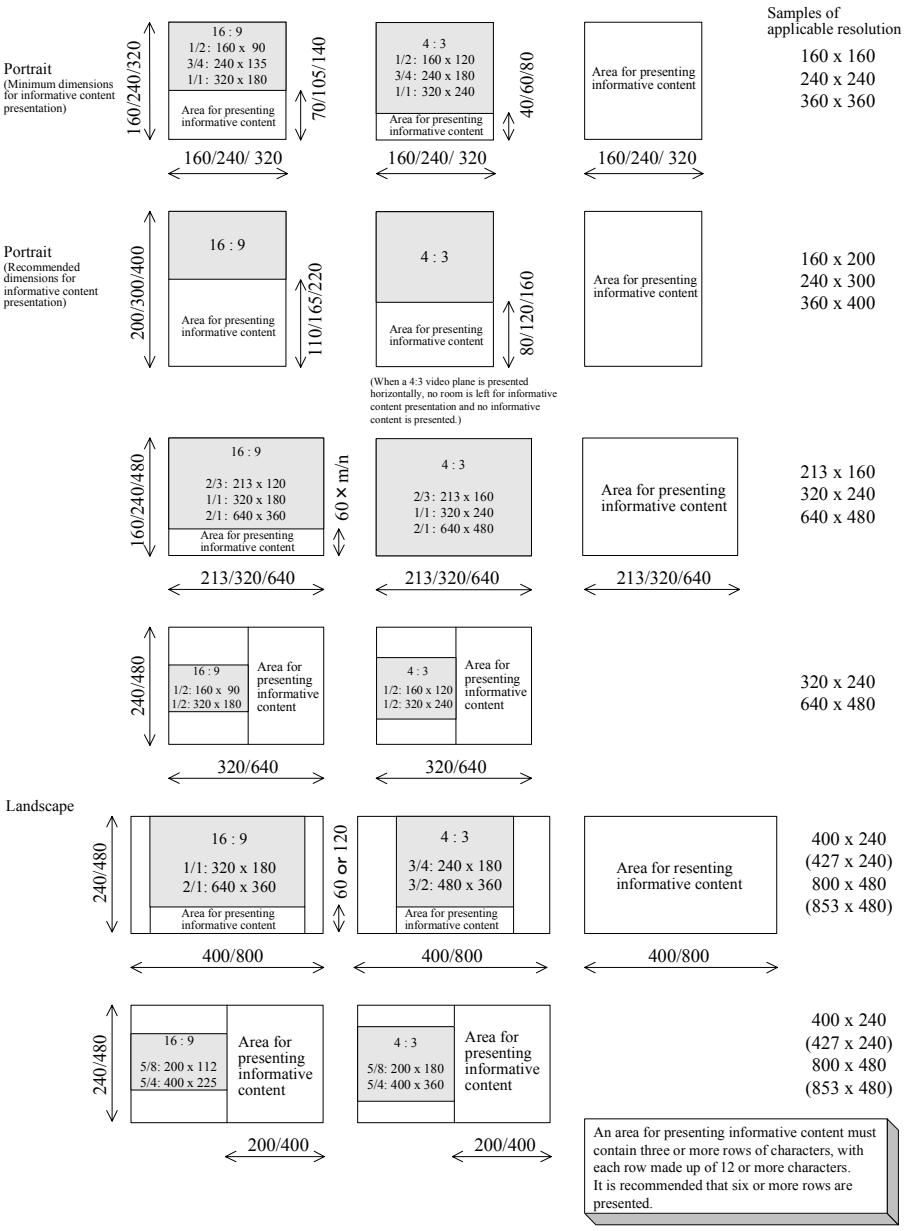
```
@media tv {
/* margin */
div, p, pre, form, input, textarea, object, img { margin: 0 !important }
/* padding */
div, form, object, img { padding-top: 0 !important; padding-right: 0 !important; padding-bottom: 0 !important; padding-left: 0 !important }
/* border */
div, p, pre, form, input, textarea { 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 }
/* background-color */
object, img { background-color: transparent !important }
body { background-color: white !important }
/* background-repeat */
body { background-repeat: repeat !important }
```

```
/* font-family */
p, pre, span, a, input, textarea { font-family: "丸ゴシック" !important }
/* text-align */
p, input, textarea { text-align: left }
/* white-space */
p, input, textarea { white-space: normal !important }
pre { white-space: pre !important }
/* resolution */
body { resolution: 240x480 !important }
/* marquee */
p { -wap-marquee-loop: 1; -wap-marquee-dir: rtl !important }
}
```

Informative Explanation

1 Sample presentation for a receiver employing a portable component-based model

This section contains a sample presentation patterns for a receiver that employs a portable component-based model.



Sample presentation for a portable receiver
(These figures are designed to illustrate how an "area for presenting informative content" is located to a video plane.)

APPENDIX 5

Operational Guidelines for Implementing Extended Services for Mobile Receiving System

Contents

Chapter APPENDIX 5-1 Purpose 801

Chapter APPENDIX 5-2 Scope 802

Chapter APPENDIX 5-3 Legend 803

Chapter APPENDIX 5-4 Guidelines for Coding of BML Documents 804
A5-4.1 Character Coding Schemes 804
A5-4.1.1 Character coding scheme used for BML documents 804
A5-4.1.2 External characters 806
A5-4.1.3 Operation of character reference 807
A5-4.1.4 Operation of character encoding available to contents 808
A5-4.2 Media Types and Operational Range of Monomedia 808
A5-4.2.1 Bitmap 810
A5-4.2.2 Video/Audio 811
A5-4.2.3 8-bit character code 812
A5-4.3 Operation of XML Namespace 812
A5-4.4 Operation of PI 812
A5-4.4.1 Switching presentation model 812
A5-4.5 Operation of BML Elements 813
A5-4.5.1 Operational convention on BML elements for XHTML modules 813
A5-4.5.2 Entity 823
A5-4.5.3 Attribute 824
A5-4.6 Operation of CSS-Based Style Sheet 830
A5-4.6.1 Elements applicable to property 832
A5-4.6.2 Operation of selectors 834
A5-4.6.3 Media type 835
A5-4.6.4 !important rule 835
A5-4.6.5 Box model 835
A5-4.6.6 Visual formatting model 837
A5-4.6.7 Other visual effects 839
A5-4.6.8 Generated content, autonumbering, list 839
A5-4.6.9 Colours and backgrounds 840
A5-4.6.10 Fonts 841
A5-4.6.11 Text 842

A5-4.6.12 Tables 842
A5-4.6.13 Pseudo classes 843
A5-4.6.14 Extended properties 845
A5-4.6.15 Default style sheet 846
A5-4.6.16 Operation of length 846
A5-4.6.17 Actual values derived from computed values 846
A5-4.6.18 <color-index> 846
A5-4.7 Operational Guidelines on Procedural Description 846
A5-4.7.1 Operational guidelines on DOM 846
A5-4.7.2 Operational scope for built-in objects 869
A5-4.7.3 Operational scope for extended objects for broadcasting 873
A5-4.7.4 Operational scope for browser pseudo objects 873
A5-4.7.5 Operation of Navigator pseudo objects 877
A5-4.8 Restrictions on Description of BML Documents and Operational Guidelines on Presentation Control 878
A5-4.8.1 Positioning and restrictions for index-based model 878
A5-4.8.2 Vertical writing 879
A5-4.8.3 Restrictions on arrangement of elements 879
A5-4.8.4 Operation of object element and img element 880

Chapter APPENDIX 5-5 Guidelines on Behaviours of Browsers 885
A5-5.1 Operational Guidelines on Presentation 885
A5-5.1.1 Presentation model 885
A5-5.1.2 Coordinate system for text and graphics plane 885
A5-5.1.3 Operational guidelines on Z-index 885
A5-5.1.4 Presentation plane 886
A5-5.1.5 Operation of monomedia presentation 887
A5-5.1.6 Guidelines on clipping 890
A5-5.1.7 Values applicable to CSS2 properties 891
A5-5.1.8 Operation of colour map data 891
A5-5.1.9 Operation of remote control keys 892
A5-5.1.10 Character entry into input element and textarea element 893
A5-5.1.11 Operation of Focus 893
A5-5.2 Guidelines on Behaviours of External Characters 894
A5-5.2.1 Sharing of external characters among documents 894
A5-5.2.2 Presentation of external characters 894
A5-5.3 Guidelines on Behaviours of DOM 894
A5-5.3.1 Handling of control codes (blank, line break, tab) 895
A5-5.3.2 Generation of DOM tree 896
A5-5.3.3 Restriction on DOM work memory 896
A5-5.3.4 Restrictions on style description 897
A5-5.4 Operational Guidelines on ECMAScript 897
A5-5.4.1 Operation of script working environment 897
A5-5.4.2 Data type 898
A5-5.4.3 Effects on basic objects caused by data type restrictions 899

A5-5.4.4 Operational general rule of implementation-dependent behaviour 900
A5-5.5 Operational Guidelines on Broadcasting Extended Objects 902
A5-5.5.1 Operational guideline on BinaryTable class 902
A5-5.6 Guidelines on Behaviour of Browser Pseudo Objects 905
A5-5.6.1 Guidelines on behaviours of EPG functions 905
A5-5.6.2 Operation of non-volatile memory functions 905
A5-5.6.3 Operation of interaction channel function 906
A5-5.6.4 Operation of operational control functions 907
A5-5.6.5 Operation of receiver build-in sound control 908
A5-5.6.6 Operation of timer function 908
A5-5.6.7 Operation of external character function 908
A5-5.6.8 Operation of data carousel storage functions 908
A5-5.6.9 Operation of module lock among components 908
A5-5.7 Operation of link Element 909
A5-5.7.1 Operation of link element for external style sheet specification 909
A5-5.8 Other Restrictions 909
A5-5.9 Operation of Extended Functions 911
A5-5.9.1 Operation of High-resolution Presentation Function 911
A5-5.10 Operation of Cookies 912
A5-5.10.1 Version of Cookies 912
A5-5.10.2 Function enabling Cookies 913
A5-5.10.3 Restrictions on implementation of Cookies 913
A5-5.10.4 Reference of Cookie from DOM 913
A5-5.10.5 Reference of Cookie in broadcast/communication/stored content 913

Chapter APPENDIX 5-6 Guidelines on Contents Transmission and Namespace 914
A5-6.1 Scope Mapping to Transmission System 914
A5-6.1.1 Scope mapping for contents received in real time 914
A5-6.2 Guidelines on Reference across Media Types 915
A5-6.2.1 Restrictions on document transmission among media types 915
A5-6.2.2 Restrictions on reference of monomedia and others across media types 916
A5-6.3 Namespace 918
A5-6.3.1 Namespace for BML content received in real time 918
A5-6.3.2 Namespace for BML content transmitted via interactive communication networks associated with broadcasting services 918
A5-6.4 Guidelines on Transmission of BML content 919
A5-6.4.1 Syntax of module 919
A5-6.4.2 Resource list in multi-part formatted module 920
A5-6.4.3 Operation of event messages 920
A5-6.5 Operational Guidelines on Communication Protocols 920
A5-6.5.1 Operational guidelines on HTTP 920
A5-6.5.2 Operational Guidelines on mapping between API and protocols 920
A5-6.5.3 Operational guidelines on secure communication 920

Annex A Default Style Sheet 921
A.1 Default Style Sheet Common to Index-based Model and Component-based Model 921
A.2 Default Style Sheet Specific to Index-based Model 921
A.3 Default Style Sheet Specific to Component-based Model 921

Annex B Operated DTD 923
B.1 BML Driver DTD 923
B.2 BML Document Model Module 923

Chapter APPENDIX 5-1 Purpose

The Appendix 5 contains the operational guidelines for implementing an integrated service that provides broadcasting services and services via the Internet for a mobile receiving system, by using the XML-based multimedia coding scheme specification responsible for the data broadcasting scheme, part of the digital data broadcasting scheme specified as the standard in Japan.

Chapter APPENDIX 5-2 Scope

The guidelines contained in this appendix are applied to an integrated service that provides broadcasting services without stored services and services via the Internet for a mobile receiving system, by using the XML-based multimedia coding scheme specification responsible for the data broadcasting scheme, part of the digital data broadcasting scheme specified as the standard in Japan.

Chapter APPENDIX 5-3 Legend

O: This symbol marks an item required to implement a service based on a specified sub-profile. It is desirable that a receiver designed for a concerned service should present this item normally when the corresponding signal is contained in a content. However, an item marked by "*" or "Note" may not be available due to restrictions on a media type, a specific implementation, or others.

-: This symbol marks an item not required to implement a service based on a specified sub-profile. Whether or not the item is presented depends on an implementation designed for a concerned service, when the corresponding signal contained in a content to be presented. It is desirable that no signal corresponding to a "-" marked item is contained in a content. However, this does not prevent an operational standard regulation for a specific media type from using a "-" marked item.

Chapter APPENDIX 5-4 Guidelines for Coding of BML Documents

A5-4.1 Character Coding Schemes

Any operation of character coding schemes must comply with the specification defined in Section 5.1, Volume 2.

A5-4.1.1 Character coding scheme used for BML documents

The EUC-JP or JIS character coding scheme must be used for any BML document to be transmitted.

A5-4.1.1.1 EUC-JP

The EUC-JP character codes are used based on Section 4.1.1, Appendix 2.

Table 4-1 shows the operational guideline on the EUC-JP character codes.

Table 4-1 EUC-JP Code Set

Code set	Character set	Operating range
Code set 0 Byte range 21~7E	JIS X 0201-1976 (JIS Roman Alphabets) (Presentation of the bits combination 7/15(DEL) depends on an implementation.)	O
Code set 1 1st byte range A1~FE 2nd byte range A1~FE	JIS X 0208-1990 (Additional symbols defined for the sections 90 to 94 on page 101 and 102 (table 10-11 and 10-12) in the ARIB-STD-B5 are specified for the reserved area of the sections 90 to 94. The sections 87 to 88 are used for the area defined by external characters. And the point 13 to 18 of the section 1 and the point 94 of the section 2 are not used.	O
Code set 2 1st byte 8E 2nd byte range A1~DF	Half-size katakana characters	-
Code set 3 1st byte 8F 2nd byte range A1~FE 3rd byte range A1~FE	JIS X 0212-1990	-
Control codes	Blank character (20)	O
	Delete character (7F)	-
	Line feed (0D0A)	O
	Tab (09)	O
	SS2 (8E)	-
	SS3 (8F)	-

A5-4.1.1.2 Shift-JIS

Table 4-2 shows the operational guideline on the Shift-JIS character codes.

Table 4-2 Shift-JIS Code Set

Code set	Character set	Operating range
1-byte character Byte range 21~7F, A1~DF	JIS X 0201-1976 (JIS Roman Alphabets) Half-width (Hankaku) Katakana	O (Note)
2-byte character 1st byte range 81~9F, E0~EF 2nd byte range 40~7E, 80~FC	JIS X 0208-1997 Note that a character with the same kuten code in the Kanji character set is allocated in the corresponding location in the sections 90 ku to 94 ku (free area). The sections 87 ku to 88 ku are reserved for defining external characters.	O
Control codes	Blank character (20)	O
	Delete character (7F)	-
	Line feed (0D0A)	O
	Tab (09)	O

Note: Using half-size Katakana characters and full-size alphanumerical characters should be avoided.

A5-4.1.1.3 Conversions of character codes

This section describes operational guidelines for converting a character coding for BML documents into the other.

- Converting EUC-JP into Shift-JIS

An EUC-JP character code is converted into a Shift-JIS code as described below.

(1) When an EUC-JP character code belongs to the code set 0:

The EUC-JP character code is converted into a 1-byte character in Shift-JIS.

Provided that an EUC-JP character code to be converted is E1 and the resulting Shift-JIS character code is S1, this conversion is expressed as below.

S1 = E1

(2) When an EUC-JP character code belongs to the code set 1:

The EUC-JP character code is converted into a 2-byte character in Shift-JIS.

Provided that the first byte of an EUC-JP character code to be converted is E1, the second byte of the EUC-JP code is E2, the first byte of the resulting Shift-JIS character code is S1, the second byte of the Shift-JIS code is S2, and div is a integer division (discarding all digits to the right of the decimal point) this conversion is expressed as below.

a) S1 is obtained as described below.

a-1) When 0xA1 <= E1 <= 0xDE, S1 = (E1- 1) div 2 + 0x31.

a-2) When 0xDF <= E1 <= 0xFE, S1 = (E1 - 1) div 2 + 0x71.

b) S2 is obtained as described below.

b-1) Provided that E2 is an odd number:

b-1.1) When 0xA1 <= E2 <= 0xDF, S2 = E2 - 0x61.

b-1.2) When 0xE0 <= E2 <= 0xFE, S2 = E2 - 0x60.

b-2) Provided that E2 is an even number:

S2 = E2 - 0x2

(3) When an EUC-JP character code is a control code (a blank, a line break, or a tab):

The EUC-JP character code is converted into a corresponding character code in Shift-JIS.

Provided that an EUC-JP control code to be converted is Ec and the resulting Shift-JIS control code is Sc, this conversion is expressed as below.

Sc = Ec

- Converting Shift-JIS into EUC-JP

(1) When a Shift-JIS character code is a 1-byte character:

The Shift-JIS character code is converted into an EUC-JP character code that belongs to the code set 0.

Provided that a Shift-JIS character code to be converted is S1 and the resulting EUC-JP character code is E1, this conversion is expressed as below.

E1 = S1

(2) When a Shift-JIS character code is a 2-byte character:

The Shift-JIS character code is converted into an EUC-JP character code that belongs to the code set 1.

Provided that the first byte of a Shift-JIS character code to be converted is S1, the second byte of the Shift-JIS code is S2, the first byte of the resulting EUC-JP character code is E1, and the second byte of the EUC-JP code is E2, this conversion is expressed as below.

a) E1 is obtained as described below.

a-1) Provided that 0x81 <= S1 <= 0x9F,

a-1.1) When 0x40 <= S2 <= 0x9E, E1 = (S1 - 0x30) x 2 - 0x1.

a-1.2) When 0x9F <= S2 <= 0xFC, E1 = (S1 - 0x30) x 2.

a-2) Provided that 0xE0 <= S1 <= 0xEF,

a-2.1) When 0x40 <= S2 <= 0x9E, E1 = (S1 - 0x70) x 2 - 0x1.

a-2.2) When 0x9F <= S2 <= 0xFC, E1 = (S1 - 0x70) x 2.

b) E2 is obtained as described below.

b-1) When 0x40 <= S2 <= 0x7E, E2 = S2 + 0x61.

b-2) When 0x80 <= S2 <= 0x9E, E2 = S2 + 0x60.

b-3) When 0x9F <= S2 <= 0xFC, E2 = S2 + 0x2.

(3) When a Shift-JIS character code is a control code (a blank, a line break, or a tab):

The Shift-JIS character code is converted into a corresponding character code in EUC-JP.

Provided that a Shift-JIS control code to be converted is Sc and the resulting EUC-JP control code is Ec, this conversion is expressed as below.

Ec = Sc

A5-4.1.2 External characters

A DRCS can be used to present external characters. This section provides the operational guideline for coding and using external characters. The operational guideline for managing external character is defined in Section A2-5.2, APPENDIX 2. The operational guideline on the loadDRCS( ) function is described in Section A2-5.6.10, APPENDIX 2.

The code area available to the external characters is the sections 87 through 88. A pattern data invoked by loadDRCS() is referenced by a BML document. To configure a pattern data in an area for managing external characters in a content and to make external characters referenced, the following conditions [1] must be satisfied:

- The external characters referenced by a BML document must have been set before the BML document is presented. The loadDRCS( ) function must be executed at the time when global codes of ECMAScript are executed. It is not executed in event handlers of ECMAScript. The loadDRCS function may be executed in global codes not more than twice.
- The external characters referenced by a BML document must be in a string presented by the referencing BML document, a message transmitted via an interaction channel, and a string representing privateData transmitted in an event message.
- Two or more sets of pattern data may be set for a single character code by executing loadDRCS(), provided that each set of pattern data is a different combination of a font family and a font size. When a character code is associated with two or more sets of pattern data and each set is the same combination of a font family and a font size, how the redundant sets are treated depends on an implementation.

A structure of DRCS data that is read by loadDRCS( ) is an extension to a data structure defined in ARIB STD-B24 Volume 1 Annex D, to which the following restrictions are also applied:

- The range of the numberOfCode value is from 1 to 188.
- The available values to characterCode correspond to the characters from the point 1 of the section 87 to the point 94 of the section 88 in the character code table. A characterCode value is a section and point code. For example, to transmit the character in the point 1 of section 1, 0x2121 is transmitted.
- The following values can be specified for the fontID property. The fontID property must not set to 0.

Value of fontId	Typeface
1	Round gothic

- The mode must be 0001. That is, the pattern data is multi-scaled and is not compressed. It is not compressed with geometric coding.
- The depth must be 2. That is, the pattern data has 4-step scaled values.
  The same restrictions as the pixel value for font-size property are applied to width and height.

A5-4.1.3 Operation of character reference

A5-4.1.3.1 EUC-JP

- Character reference by “&#” is not operated.
- When external characters are set, codes applicable to a character coding scheme that is used for a concerned BML document must be specified directly.
- HexEscapeSequence, OctalEscapeSequence, and UnicodeEscapeSequence in ECMAScript are not operated.

[1] The purposes of these conditions are to prevent a redundant pattern data associated with the same external character code from causing any unexpected miss-presentation of the character and to avoid any disordered presentation of contents due to a receiver that cannot ensure that a redundant pattern data is presented at an intended timing.

And, among SingleEscapeCharacters, “¥b”, “¥t”, and “¥f” are not operated but “¥””, “¥’”, “¥r”, “¥n”, and “¥¥” are operated. However,

- The “¥” character must be followed by “0x5c”.
- The “¥r” and “¥n” characters must be used as the combination, “¥r¥n”.

A5-4.1.4 Operation of character encoding available to contents

- Which character encoding specification is internally used by an ECMAScript processor depends on an implementation. Which character encoding specification is applied to functions for comparing lengths of strings and sorting strings also depends on an implementation.
- Any character encoding specification used in a Greg or Ureg pseudo object is converted into a character encoding used by a BML document that is presented at the time when the pseudo object is read out.
- The writePersisitentArray() function and readPersistentArray()functions writes and reads information written in EUC-JP to and from NVRAM, respectively.
- Any binary table contains information written in EUC-JP.
- The privateData attribute of EventMessageFired contains strings in EUC-JP.
- To transmit information via IP:
  - The sendTextMail() function sends or receives information written in ISO 2022-JP.
  - Any information sent to or from a server via the form element is written the same character encoding specification as used in a document that contains the form element.
  - The transmitTextDataOverIP() function sends or receives messages written in a character encoding specification specified in the function.

A5-4.2 Media Types and Operational Range of Monomedia

Table 4-3 shows operational guidelines on the media types required for basic services and the monomedia coding schemes designated by object element.

Table 4-3 Operational Guidelines on Media Types and Monomedia Coding Schemes

Schema	Media type	Operation	Operation (referenced by object element)	Operation (referenced by ing element)	Remarks
arib-dc:	multipart/mixed	O	-	-	
	text/xml;charset="euc-jp"	-	-	-	
	text/xml;charset="UTF-8"	-	-	-	
	text/xml;charset="UTF-16"	-	-	-	
	text/xml;charset="Shift_JIS"	-	-	-	
	text/css	O	-	-	(Note 1)
	text/xsl;charset="euc-jp"	-	-	-	
	text/xsl;charset="Shift_JIS”	-	-	-	
	text/xsl;charset="UTF-8"	-	-	-	
	text/xsl;charset="UTF-16"	-	-	-	
	text/html;charset="euc-jp"	O	-	-	
	text/html;charset="UTF-8"	-	-	-	
	text/html;charset="UTF-16"	-	-	-	
	text/html;charset="Shift_JIS”	O	-	-	
	text/X-arib-bml; charset="euc-jp"	O	-	-	(Note 2)

Schema	Media type	Operation	Operation (referenced by object element)	Operation (referenced by ing element)	Remarks
	text/X-arib-bml; charset="UTF-8"	-	-	-	
	text/X-arib-bml; charset="UTF-16"	-	-	-	
	text/X-arib-bml; charset="Shift_JIS"	O	-	-	(Note 2)
	text/plain; charset="euc-jp"	-	-	-	
	text/plain; charset="UTF-8"	-	-	-	
	text/plain; charset="UTF-16"	-	-	-	
	text/plain; charset="Shift_JIS"	-	-	-	
	text/X-arib-jis8text	-	-	-	
	text/X-arib-ecmascript; charset="euc-jp"	O	-	-	(Note 2)
	text/X-arib-ecmascript; charset= "UTF-8"	-	-	-	
	text/X-arib-ecmascript; charset= "UTF-16"	-	-	-	
	text/X-arib-ecmascript; charset="Shift_JIS"	O	-	-	(Note 2)
	image/jpeg	O	O	O	
	image/png	O	O	O	
	image/X-arib-png	O	O	O	
	image/X-arib-mng	O	O	-	
	image/X-arib-mpeg2-I	-	-	-	
	image/X-arib-mpeg4-I-simple	O	O	-	
	image/X-arib-mpeg4-I-core	-	-	-	
	image/X-arib-H264-I-baseline	O	-	-	(Note 4)
	image/X-arib-H264-I-main	O	-	-	(Note 4)
	audio/X-arib-mpeg2-aac	O	O	-	
	audio/X-arib-mpeg2-bc	-	-	-	
	audio/X-arib-aiff	O	O	-	
	audio/X-arib-additional	-	-	-	
	video/X-arib-mpeg1	-	-	-	
	application/xhtml+xml; charset="euc-jp"	-	-	-	
	application/xhtml+xml; charset="Shift_JIS"	-	-	-	
	application/xhtml+xml; charset="UTF-8"	-	-	-	
	application/xhtml+xml; charset="UTF-16"	-	-	-	
	application/X-arib-stream-text; charset="euc-jp"	-	-	-	
	application/X-arib-stream-text; charset="UTF-8"	-	-	-	
	application/X-arib-stream-text; charset="UTF-16"	-	-	-	
	application/X-arib-stream-text; charset="Shift_JIS"	-	-	-	
	application/X-arib-stream-jis8text	-	-	-	
	application/X-arib-stream-png	-	-	-	
	application/X-arib-stream-jpeg	-	-	-	
	application/X-arib-stream-mpeg2-I	-	-	-	
	application/X-arib-stream-mpeg4-I-simple	-	-	-	
	application/X-arib-stream-mpeg4-I-	-	-	-	

Schema	Media type	Operation	Operation (referenced by object element)	Operation (referenced by ing element)	Remarks
	core				
	application/X-arib-stream-H264-I-baseline	-	-	-	
	application/X-arib-stream-H264-I-main	-	-	-	
	application/X-arib-mpeg2-tts	O	O	-	
	application/X-arib-bmlclut	O	-	-	
	application/X-arib-btable	O	-	-	
	application/X-arib-drcs	O	-	-	
	application/X-arib-PDI	-	-	-	
	application/X-arib-resourceList	O	-	-	
	application/X-arib-storedResourceList	-	-	-	
	application/X-arib-contentPlayControl	-	-	-	
arib:	audio/X-arib-mpeg2-aac	O	O	-	
	audio/X-arib-mpeg2-bc	-	-	-	
	audio/X-arib-mpeg4	-	-	-	
	video/X-arib-mpeg1	-	-	-	
	video/X-arib-mpeg2	-	-	-	
	video/X-arib-mpeg4-simple	O	O	-	
	video/X-arib-mpeg4-core	-	-	-	
	video/X-arib-H264-baseline	O	O	-	(Note 4)
	video/X-arib-H264-main	O	O	-	(Note 4)
arib-ic:	image/X-arib-mpeg2-I	-	-	-	
	application/X-arib-stream-mpeg2-I	-	-	-	
	application/X-arib-stream-mpeg4-I-simple	O	O	-	
	application/X-arib-stream-mpeg4-I-core	-	-	-	
	application/X-arib-stream-H264-I-baseline	O	O	-	(Note 4)
	application/X-arib-stream-H264-I-main	O	O	-	(Note 4)
romsound:	audio/X-arib-romsound	O	-	-	(Note 3)

Note 1: When a referencing document is a BML document, the same character encoding specification as the BML document is used for a CSS and the charset attribute for the CSS is omitted. When a referencing document is an XHTML document for printing, a character encoding specification designated by the @charset rule. With the @charset rule omitted, the same character encoding specification as the XHTML document for printing is used.

Note 2: The charset attribute must not be omitted.

Note 3: This specifies a built-in sound that is a broadcaster.

Note 4: Actual operation depends on an operational standard regulation defined for each media type.

A5-4.2.1 Bitmap

A5-4.2.1.1 PNG

The conventions on coding and transmitting PNG defined in Section 8.2.3, Volume 2 must be obeyed. Note that in order to use a component-based model (refer to Section A5-5.1.1), a PLTE chunk is required. When a component-based model is used, a tRNS chunk is employed.

The length of Chunk Data in a tRNS chunk must be not less than nine bytes. The ninth byte must contain 00 and the other bytes contain a value other than 00.

A5-4.2.1.2 MNG

The conventions on coding and transmitting MNG defined in Section 8.2.4, Volume 2 must be obeyed. Note that in order to use a component-based model (refer to Section A5-5.1.1), a PLTE chunk is required. When a component-based model is used, a tRNS chunk is employed.

The length of Chunk Data in a tRNS chunk must be not less than nine bytes. The ninth byte must contain 00 and the other bytes contain a value other than 00.

A5-4.2.2 Video/Audio

A5-4.2.2.1 MPEG-2 Video stream and H.264|MPEG-4 AVC stream

The available sizes to present scaled video in a video plane is described in Section A5-5.1.5.1.

A5-4.2.2.2 Time-stamped TS file

The convention on transmitting a time-stamped TS file defined in Section 8.1.2.2, Volume 2 must be obeyed. Note that only transmitting at a fixed bit rate is allowed and the fixed bit rate must be contained in TSbitrate of the TimestampedTS descriptor.

To transmit a time-stamped TS file in a module in the multipart format, TimestampedTSdescriptor() must be contained in additionalFileInfo() of storedFileInfo() indicating a time-stamped TS resource by using X-arib-storedResourceList() in the multipart module. TimestampedTSdescriptor() is operated based on Table 4-4.

Table 4-4 Operation of TimestampedTSdescriptor()

Data	Operation
TSbitrate	Required
PCR_PID	Required
default_video_flag	0 (Required when a video stream exists) 1 (When no video stream exist)
default_audio_flag	0 (Required when an audio stream exists) 1 (When no audio stream exist)
timestampedTS_PBctrl_info_flag	Optional (A receiver is not required to interpret a time-stamped TS playback control information.)
videoPID	Required when a video stream exists
audioPID	Required when an audio stream exists

A time-stamped TS file contains a single program, video streams, or vide streams as listed in Table 4-5.

Table 4-5 Operation of Time-stamped TS File

Property	Operation
the number of programs	Limited to one
the number of video streams	One or zero
the number of audio streams	Zero through two

In order to play a time-stamped TS file, the object element is operated as described in Table 4-6. When the data attribute value of the object element that specifies a video stream or an audio stream is -1, a stream specified with videoPID or audioPID of TimestampedTSdescriptor() is assumed.

Table 4-6 Operation of Object for Time-stamped TS Playback

Property	Operation
value of the value attribute of the object specifying a time-stamped TS	Limited to “-1”
value of the data attribute of the object specifying a video stream	Limited to “-1”
value of the data attribute of the object specifying an audio stream	“-1” or /<component_tag>[;<channel_id>]

A5-4.2.3 8-bit character code

No 8-bit character code must be operated.

A5-4.3 Operation of XML Namespace

The XML Namespace must operated based on the conventions defined in Chapter 10, Volume 2.

A5-4.4 Operation of PI

A5-4.4.1 Switching presentation model

Presentation models are switched by using PI. To operate a component-based model, PI is used to control how a focused frame is presented. The definition format for bml-model processing instruction is shown below.

- Definition

```
BmlModelPI      ::=   '<?bml-model' (S PseudoAtt)* S? '?>'
PseudoAtt       ::=   Name S? '=' S? PseudoAttValue
PseudoAttValue  ::=   ("" ([^"<&] | CharRef)* ""
                      | '' ([^'<&] | CharRef)* '')
                        - (Char* '?>' Char*)
```

- PseudoAtt, PseudoAttValue

  presentation-model(index-base|component-base) "component-base"

  focus-visibility (UAcontrol | invisible) "UAcontrol"

  The focus-visibility described above is applicable only to a component-based model. When an index-based model is presented, no focus is presented regardless of whether or not a browser supports the focus presentation.

- Semantics

Table 4-7 PI to Switch Presentation Models

PseudoAtt	PseudoAttValue	Semantics
presentation-model	index-base	Presentation is based on an index-based model.
	component-base	Presentation is based on a component-based model.

PseudoAtt	PseudoAttValue	Semantics
focus-visibility	UAcontrol	Presentation of a focus is controlled by a browser.
	invisible	A browser does not present a focus.

Only prolog of a BML document must contain a bml- model processing instruction.

Example1: To use an index-based model for presentation,

<?bml-model presentation-model="index-base"?>

Example2: To use a component-based model for presentation,

<?bml-model presentation-model="component-base"?>

A5-4.5 Operation of BML Elements

A5-4.5.1 Operational convention on BML elements for XHTML modules

Any BML element must be operated to ensure consistency with XHTML 1.0. Any BML element is operated by selecting a module based on the W3C Recommendation, Modularization of XHTML except that the behaviours of the the ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout events, each of those defined with Events as described Section 5.1 Attribute Collection, the Modularization of XHTML recommendation, depend on an implementation.

Table 4-8 Operation of XHTML modules

Module		Operation	Available elements to module	Attributes applicable to element
Core	Structure (required)	O	body	%Common;
			head	%I18N;, profile
			html	%I18N;, version, xmlns
			title	%I18N;
	Text (required)	O	abbr	%Common;
			acronym	%Common;
			address	%Common;
			blockquote	%Common;, cite
			br	%Core;
			cite	%Common;
			code	%Common;
			dfn	%Common;
			div	%Common;
			em	%Common;
			h1 ~ h6	%Common;
			kbd	%Common;
			p	%Common;
			pre	%Common;, xml:space="preserve"
			q	%Common;, cite
			samp	%Common;
			span	%Common;
			strong	%Common;

Module		Operation	Available elements to module	Attributes applicable to element
			var	%Common;
	Hypertext (required)	O	a	%Common;, accesskey, charset, href, hreflang, rel, rev, tabindex, [2] type
	List (required)	O	dl	%Common;
			dt	%Common;
			dd	%Common;
			ol	%Common;
			ul	%Common;
			li	%Common;
Applet (deprecated)		-	applet	%Core;, alt*, archive, code, codebase, height*, object, width*
			param	id, name*, type, value, valuetype
Text Extension	Presentation [3]	O	b	%Common;
			big	%Common;
			hr	%Common;
			i	%Common;
			small	%Common;
			sub	%Common;
			sup	%Common;
			tt	%Common;
	Edit	-	del	%Common;, cite, datetime
			ins	%Common;, cite, datetime
	Bi-directional	-	bdo	%Core;, dir*
Form	Basic Forms	O	form	%Common;, action*, method, enctype

2 How tabindex is interpreted depends on a receiver.
3 How it is presented depends on a receiver. Presenting it is not required.

Module		Operation	Available elements to module	Attributes applicable to element
			input	%Common; accesskey, checked, maxlength, name, size, src, tabindex, type, value
			label	%Common;, accesskey, for
			select	%Common;, multiple, name, size, tabindex
			option	%Common;, selected, value
			textarea	%Common;, accesskey, cols*, name, rows*, tabindex
	Forms	-	form	%Common;, accept, accept-charset, action*, method, enctype
			input	%Common;, accept, accesskey, alt, checked, disabled, maxlength, name, readonly, size, src, tabindex, type, value
			select	%Common;, disabled, multiple, name, size, tabindex

Module		Operation	Available elements to module	Attributes applicable to element
			option	%Common;, disabled, label, selected, value
			textarea	%Common;, accesskey, cols*, disabled, name, readonly, rows*, tabindex
			button	%Common;, accesskey, disabled, name, tabindex, type, value
			fieldset	%Common;
			label	%Common;, accesskey, for
			legend	%Common;, accesskey
			optgroup	%Common;, disabled, label*
Table	Basic Tables	O	caption	%Common;
			table	%Common;, summary, width
			td	%Common;, abbr, align, axis, colspan, headers, rowspan, scope, valign
			th	%Common;, abbr, align, axis, colspan, headers, rowspan, scope, valign

Module		Operation	Available elements to module	Attributes applicable to element
			tr	%Common;, align, valign
	Tables	-	caption	%Common;
			table	%Common;, border, cellpadding, cellspacing, datapagesize, frame, rules, summary, width
			td	%Common;, abbr, align, axis, char, charoff, colspan, headers, rowspan, scope, valign
			th	%Common;, abbr, align, axis, char, charoff, colspan, rowspan, scope, valign
			tr	%Common;, align, char, charoff, valign
			col	%Common;, align, char, cahroff, span, valign, width
			colgroup	%Common;, align, char, charoff, span, valign, width

Module		Operation	Available elements to module	Attributes applicable to element
			tbody	%Common;, align, char, charoff, valign
			thead	%Common;, align, char, charoff, valign
			tfoot	%Common;, align, char, charoff, valign
Image		O	img	%Common;, alt*, height, longdesc, src*, width
CS Image Map		-	*a&*	coords, shape
			area	%Common;, accesskey, alt*, coords, href, nohref, shape, tabindex
			img&	usemap
			Input&	usemap
			map	%I18N;, %Events;, class, id*, title
			object&	usemap
SS Image Map		-	img&	ismap
			input&	ismap

Module		Operation	Available elements to module	Attributes applicable to element
Object		O	object	%Common;, archive, classid, codebase, codetype, data, declare, height, name, standby, tabindex, type, width
			param	id, name*, type, value, valuetype
Frames		-	frameset	%Core;, cols, rows
			frame	%Core;, frameborder, longdesc, marginheight, marginwidth, noresize, scrolling, src
			noframes	%Common;
Target		-	*a&*	target
			area&	target
			base&	target
			link&	target
			form&	target
Iframe		-	iframe	%Core;, frameborder, height, longdesc, marginheight, marginwidth, scrolling, src, width
Intrinsic Events		O [4]	*a&*	onblur, onfocus
			area&	onblur, onfocus

4 The behaviours of an event controlled by a mouse or other input devices whose movement directions are not predefined depend on an implementation.

Module		Operation	Available elements to module	Attributes applicable to element
			frameset&	onload, onunload
			form&	onreset, onsubmit
			body&	onload, onunload
			label&	onblur, onfocus
			input&	onblur, onchange, onfocus, onselect
			select&	onblur, onchange, onfocus
			textarea&	onblur, onchange, onfocus, onselect
			button&	onblur, onfocus
Metainformation		O	meta	%I18N;, content*, http-equiv, name, scheme
Scripting		O	noscript	%Common;
			script	charset, defer, src, type*, xml:space="preserve"
Stylesheet		O	style	%I18N;, media, title, type*, xml:space="preserve"
Style Attribute		O		%Style (activate)
Link		O	link	%Common;, charset, href, hreflang, media, rel, rev, type
Base		O	base	href*
Name Identification		-	*a&*	name
			applet&	name

Module		Operation	Available elements to module	Attributes applicable to element
(deprecated)			*form&*	name
			frame&	name
			iframe&	name
			img&	name
			map&	name
Legacy (deprecated)		-	basefont	color, face, id, size
			center	%Common;
			dir	%Common;, compact
			font	%Core;, %I18N;, color, face, size
			isindex	%Core;, %I18N;, prompt
			menu	%Common;, compact
			s	%Common;
			strike	%Common;
			u	%Common;
			body&	alink, background, bgcolor, link, text, vlink
			br&	clear
			caption&	align
			div&	align
			dl&	compact, type
			h1-h6&	align
			hr&	align, noshade, size, width
			img&	align, border, hspace, vspace
			input&	align
			legend&	align
			li&	type, value
			ol&	compact, start, type

Module		Operation	Available elements to module	Attributes applicable to element
			p&	align
			pre&	width
			script&	language
			table&	align, bgcolor
			tr&	bgcolor
			th&	bgcolor, height, nowrap, width
			td&	bgcolor, height, nowrap, width
			ul&	compact, type
BML	BML	-	bml:bml	%I18N;, version, xmlns
			bml:bevent	id
			bml:beitem	id*, type*, onoccur*, es_ref, message_group_id, message_id, message_version, module_ref, language_tag, register_id, service_id, event_id, time_mode, time_value, object_id, subscribe
			body&	bml:invisible
			div&	bml:onfocus, bml:onblur, bml:accesskey
			p&	bml:onfocus, bml:onblur, bml:accesskey
			a&	bml:effect
			bdo&	bml:orientation

Module		Operation	Available elements to module	Attributes applicable to element
			object&	bml:remain, bml:streamstatus, bml:streamposition, bml:streamlooping, bml:streamspeednumerator, bml:streamspeeddenominator , bml:streamlevel, bml:onfocus, bml:onblur, bml:accesskey
BML	Basic BML	O	bml:bevent	id
			bml:beitem	id*, type*, onoccur*, es_ref, message_group_id, message_id, message_version, module_ref, language_tag, peripheral_ref, (Note) time_mode, time_value, object_id, subscribe,
			body&	bml:invisible,
			div&	bml:onfocus, bml:onblur, bml:accesskey
			p&	bml:onfocus, bml:onblur, bml:accesskey
			span&	bml:onfocus, bml:onblur, bml:accesskey
			object&	bml:remain, bml:streamstatus, bml:streamposition, bml:streamlooping, bml:onfocus, bml:onblur, bml:accesskey

Note: How this attribute is interpreted depends on an implementation.

A5-4.5.2 Entity

The following entities are used without explicit definition. Any other entity is optional.

- lt <
- gt >
- amp &

- apos '
- quot "

A5-4.5.3 Attribute

For a module to which operational restrictions are applied , the attributes applicable to the elements of the module must be operated based on Table 4-9. Other operational Modules must comply with the reference specifications.

Table 4-9 Operational Restrictions on Attributes

Module Classification	Element Entity	Attribute	Operational Restrictions
Basic Entities	coreattrs	id	String (up to 128 bytes)
		class	
		style	
		title	Whether or not it is rendered depends on an implementation.
	i18n	xml:lang	Must be set to “ja”
		lang	Must be set to “ja”
		dir	Must be set to “ltr”
		orientation	Must be set to “horiz”
	events	onclick	
		ondblclick	When it is applied, the behaviour depends on an implementation.
		onmousedown	When it is applied, the behaviour depends on an implementation.
		onmouseup	When it is applied, the behaviour depends on an implementation.
		onmouseover	When it is applied, the behaviour depends on an implementation.
		onmousemove	When it is applied, the behaviour depends on an implementation.
		onmouseout	When it is applied, the behaviour depends on an implementation.
		onkeypress	When it is applied, the behaviour depends on an implementation.
		onkeydown	
		onkeyup	
Forms			
Basic Forms	form	%coreattrs;	
		%i18n;	
		%events;	
		action	
		method	
		enctype	The method attribute must be “post”, and the input element must be “application/x-www-form-urlencoded” or “multipart/form-data”.
	input	%coreattrs;	
		%i18n;	
		%events;	
		type	Must be set to one the following : “text”, “password”, “checkbox”, “radio”, “submit”, “reset”, “hidden”
		name	
		value	
		checked	

Module Classification	Element Entity	Attribute	Operational Restrictions
		size	This element is available only when the type attribute is “text” or “password”. Must be an integer from 1 to 40.
		maxlength	Must be set to an integer from 1 to 40
		src	
		tabindex	
		accesskey	
	label	%coreattrs;	
		%i18n;	
		%events;	
		for	
		accesskey	When it is applied, the behaviour depends on an implementation.
	select	%coreattrs;	
		%i18n;	
		%events;	
		name	
		size	Must be set to an integer from 1 to 16
		multiple	
		tabindex	
	option	%coreattrs;	
		%i18n;	
		%events;	
		selected	
		value	
	textarea	%coreattrs;	
		%i18n;	
		%events;	
		name	
		rows	Must be set to an integer from 1 to 16
		cols	Must be set to an integer from 1 to 32
		tabindex	
		accesskey	
Table			
Basic Tables	caption	%coreattrs;	
		%i18n;	
		%events;	
	table	%coreattrs;	
		%i18n;	
		%events;	
		summary	
		width	The range applicable to a pixel specification is 0 through 960. The range applicable to a percentage specification is 0 % through 100%.
	tr	%coreattrs;	
		%i18n;	
		%events;	

Module Classification	Element Entity	Attribute	Operational Restrictions
		align	Mustr be set to on one of he following: “left”, “center”, “right”
		valign	Mustr be set to on one of he following: “top”, “middle”, “bottom”
	th	%coreattrs;	
		%i18n;	
		%events;	
		abbr	
		axis	
		headers	
		scope	
		rowspan	Must be an integer from 1 to 32
		colspan	Must be set to an integer from 1 to 32
		align	Must be set to one of the following: “left”, “center”, “right”
		valign	Must be set to one of the following: “top”, “middle”, “bottom”
	td	%coreattrs;	
		%i18n;	
		%events;	
		abbr	
		axis	
		headers	
		scope	
		rowspan	Must be set to an integer from 1 ro 32
		colspan	Must be set to an integer from 1 to 32
		align	Must be set to one of the following: “left”, “center”, “right”
		valign	Must be set to one of the following: “top”, “middle”, “bottom”
Image			
	img	%coreattrs;	
		%i18n;	
		%events;	
		src	
		alt	
		longdesc	
		height	The range applicable to a pixel specification is 0 through 540. No percentage specification is operated.
		width	The range applicable to a pixel specification is 0 through 960. No percentage specification is operated.
Object			
	object	%coreattrs;	
		%i18n;	
		%events;	
		declare	
		classid	

Module Classification	Element Entity	Attribute	Operational Restrictions
		codebase	
		data	
		type	
		codetype	
		archive	
		standby	
		height	The range applicable to a pixel specification is 0 through 540. No percentage specification is operated.
		width	The range applicable to a pixel specification is 0 through 960. No percentage specification is operated.
		tabindex	
	param	id	
		name	Must be one of the following: “program_number”,“stream_ref”
		value	
		valuetype	Must be set to one of the following: “data”,“object”
		type	
Intrinsic Events			
	a&	onblur	
		onfocus	
	area&	onblur	Invalid because no CS Image Map module is used
		onfocus	Invalid because no CS Image Map module is used
	frameset&	onload	Invalid because no Frams module is used
		onunload	Invalid because no Frams module is used
	form&	onreset	
		onsubmit	
	body&	onload	
		onunload	
	label&	onblur	When it is applied, the behavior depends on an implementation.
		onfocus	When it is applied, the behavior depends on an implementation.
	input&	onblur	
		onchange	
		onfocus	
		onselect	When it is applied, the behavior depends on an implementation.
	select&	onblur	
		onchange	
		onfocus	
	textarea&	onblur	
		onchange	
		onfocus	
		onselect	When it is applied, the behavior depends on an implementation.
	button&	onblur	Invalid because no Forms module is used.
		onfocus	Invalid because no Forms module is used.
Scripting			
	noscript	%coreattr;	

Module Classification	Element Entity	Attribute	Operational Restrictions
		%i18n;	
		%events;	
	script	charset	Must be set to “euc-jp” or “Shift-JIS”
		type	Must be set to 'text/X-arib-ecmascript; charset=”euc-jp”' or 'text/X-arib-ecmascript; charset=”Shift_JIS”
		src	
		defer	
		event	
		xml:space	
Stylesheet			
	style	%i18n;	
		type	Must be set to "text/css"
		media	
		title	
		xml:space	
Link			
	link	%coreattrs;	
		%i18n;	
		%events;	
		charset	
		href	
		hreflang	
		type	
		rel	Must be set to “Stylesheet”
		rev	
		media	
Base			
	base	href	
Basic BML			
	bevent	id	
	beitem	id	
		type	Must be set to one of the following: “EventMessageFired”, “ModuleUpdated”, “ModuleLocked”, “TimerFired”, “DataEventChanged”, “CCStatusChanged”, “MainAudioStreamChanged”, “NPTReferred”, “MediaStopped”, “DataButtonPressed”, "PeripheralEventOccured"
		onoccur	
		es_ref	
		message_group_id	Must be set to “0”
		message_id	
		message_version	
		module_ref	
		peripheral ref	
		language_tag	

Module Classification	Element Entity	Attribute	Operational Restrictions
		time_mode	Must be set to one of the following: “absolute”, “origAbsolute”, “NPT”
		time_value	
		object_id	
		subscribe	
	body&	invisible	
	div&	onfocus	
		onblur	
		accesskey	
	p&	onfocus	
		onblur	
		accesskey	
	span&	onfocus	
		onblur	
		accesskey	
	object&	remain	
		streamstatus	
		streamposition	
		streamlooping	
		onfocus	
		onblur	
		accesskey	

A5-4.6 Operation of CSS-Based Style Sheet

Operational guidelines on properties of CSS-based style sheets are shown in Table 4-10.

Table 4-10 Operational Guidelines on Properties of CSS-Based Style Sheet

Property	Operation
Box model	
margin-top	O
margin-right	O
margin-bottom	O
margin-left	O
margin	O
padding-top	O
padding-right	O
padding-bottom	O
padding-left	O
padding	O
border-top-width	O
border-right-width	O
border-bottom-width	O
border-left-width	O
border-width	O
border-top-color	O
border-right-color	O
border-bottom-color	O
border-left-color	O
border-color	O
border-top-color-index	O
border-right-color-index	O
border-bottom-color-index	O
border-left-color-index	O
border-color-index	O
border-top-style	O
border-right-style	O
border-bottom-style	O
border-left-style	O
border-style	O
border-top	O
border-right	O
border-bottom	O
border-left	O
border	O
Visual formatting model	
display	-
position	O
top	O
right	-
bottom	-
left	O
float	O
clear	O
z-index	-
direction	-
unicode-bidi	-
width	O
height	O
min-width	-
max-width	-
min-height	-
max-height	-
line-height	O
vertical-align	O
Other visual effects	
overflow	O
clip	-
visibility	O
Generated content/Autonumbbering/List	
content	-
quotes	-
counter-reset	-
counter-increment	-
marker-offset	-
list-style-type	O
list-style-image	-
list-style-position	-
list-style	O
Page media	
size	-
marks	-
page-break-before	-
page-break-after	-
page-break-inside	-
page	-
orphans	-
widows	-
Colours and Background	

Property	Operation
clut	O
color	O
color-index	O
background-color	O
background-color-index	O
background-image	O
background-repeat	-
background-attachment	-
background-position	-
background	O
Fonts	
font-family	O
font-style	O
font-variant	-
font-weight	O
font-stretch	-
font-size	O
font-size-adjust	-
font	O
Text	
text-indent	-
text-align	O
text-decoration	O
text-shadow	-
letter-spacing	O
word-spacing	-
text-transform	-
white-space	-
Tables	
caption-side	O
table-layout	-
border-collapse	-
border-spacing	O
empty-cells	-
speak-header	-
User Interface	
cursor	-
outline-color	-
outline-color-index	-
outline-width	-
outline-style	-
outline	-
Aural style sheets	
volume	-
speak	-

Property	Operation
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	-
speak-numeral	-
Extended properties	
resolution	O
display-aspect-ratio	O
greyscale-color-index	O
used-key-list	O
nav-index	O
nav-up	O
nav-down	O
nav-right	O
nav-left	O
-wav-marquee-style	O
-wav-marquee-loop	O
-wav-marquee-loop	O
-wav-marquee-speed	O
-wav-marquee	O

A5-4.6.1 Elements applicable to property

Which BML elements are applicable to each available property is shown in Table 4-11.

Table 4-11 BML elements applicable to CSS property

Legend
- O Refer to Chapter A5-3.
- I This symbol marks a BML element that must be operated only in a BML content to which an index-based model is specified. (If any restriction exists, it is expressed in the corresponding operational guideline.)
- C This symbol marks a BML element that must be operated only in a BML content to which a component-based model is specified. (If any restriction exists, it is expressed in the corresponding operational guideline.)
- Δ This symbol marks a BML element that must be described in a BML content operated under the assumption which fixed-value predetermined to the property may be specified. If the different value is explicitly specified in a BML document, how the BML content behaves is implementation dependent.
- \- Refer to Chapter A5-3.

Property	body	h1-h6, address, blockquote, pre	div, p	em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, b, big, i, small, sub, sup, tt	q	br	span, a	ol, ul	dl , dt, dd	li	hr	form	input	label	select	option	textarea	caption	table	tr	th, td	img	object
Box model																							
margin-top	Δ	O	O	Δ	Δ	Δ	Δ	O	O	O	O	O	O	Δ	O	Δ	O	O	O	-	-	O	O
margin-right	Δ	O	O	Δ	Δ	Δ	Δ	O	O	O	O	O	O	Δ	O	Δ	O	O	O	-	-	O	O
margin-bottom	Δ	O	O	Δ	Δ	Δ	Δ	O	O	O	O	O	O	Δ	O	Δ	O	O	O	-	-	O	O
margin-left	Δ	O	O	Δ	Δ	Δ	Δ	O	O	O	O	O	O	Δ	O	Δ	O	O	O	-	-	O	O
margin	Δ	O	O	Δ	Δ	Δ	Δ	O	O	O	O	O	O	Δ	O	Δ	O	O	O	-	-	O	O
padding-top	Δ	O	O	Δ	Δ	Δ	Δ	O	O	O	Δ	O	Δ	Δ	Δ	Δ	Δ	O	O	Δ	O	Δ	Δ
padding-right	Δ	O	O	Δ	Δ	Δ	Δ	O	O	O	Δ	O	Δ	Δ	Δ	Δ	Δ	O	O	Δ	O	Δ	Δ
padding-bottom	Δ	O	O	Δ	Δ	Δ	Δ	O	O	O	Δ	O	Δ	Δ	Δ	Δ	Δ	O	O	Δ	O	Δ	Δ
padding-left	Δ	O	O	Δ	Δ	Δ	Δ	O	O	O	Δ	O	Δ	Δ	Δ	Δ	Δ	O	O	Δ	O	Δ	Δ
padding	Δ	O	O	Δ	Δ	Δ	Δ	O	O	O	Δ	O	Δ	Δ	Δ	Δ	Δ	O	O	Δ	O	Δ	Δ
border-top-width	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O
border-right-width	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O
border-bottom-width	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O
border-left-width	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O
border-width	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O
border-top-color	Δ	C	C	C	C	Δ	C	C	C	C	C	C	C	C	C	Δ	C	C	C	-	C	C	C
border-right-color	Δ	C	C	C	C	Δ	C	C	C	C	C	C	C	C	C	Δ	C	C	C	-	C	C	C
border-bottom-color	Δ	C	C	C	C	Δ	C	C	C	C	C	C	C	C	C	Δ	C	C	C	-	C	C	C
border-left-color	Δ	C	C	C	C	Δ	C	C	C	C	C	C	C	C	C	Δ	C	C	C	-	C	C	C
border-color	Δ	C	C	C	C	Δ	C	C	C	C	C	C	C	C	C	Δ	C	C	C	-	C	C	C
border-top-color-index	Δ	I	I	I	I	Δ	I	I	I	I	I	I	I	I	I	Δ	I	I	I		I	I	I
border-right-color-index	Δ	I	I	I	I	Δ	I	I	I	I	I	I	I	I	I	Δ	I	I	I		I	I	I
border-bottom-color-index	Δ	I	I	I	I	Δ	I	I	I	I	I	I	I	I	I	Δ	I	I	I		I	I	I
border-left-color-index	Δ	I	I	I	I	Δ	I	I	I	I	I	I	I	I	I	Δ	I	I	I		I	I	I
border-color-index	Δ	I	I	I	I	Δ	I	I	I	I	I	I	I	I	I	Δ	I	I	I		I	I	I
border-top-style	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O
border-right-style	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O
border-bottom-style	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O

Property	body	h1-h6, address, blockquote, pre	div, p	em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, b, big, i, small, sub, sup, tt	q	br	span, a	ol, ul	dl , dt, dd	li	hr	form	input	label	select	option	textarea	caption	table	tr	th, td	img	object
border-left-style	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O
border-style	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O
border-top	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O
border-right	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O
border-bottom	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O
border-left	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O
border	Δ	O	O	O	O	Δ	O	O	O	O	O	O	O	O	O	Δ	O	O	O	-	O	O	O
Visual formatting model																							
display	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
position	Δ	O	O	Δ	Δ	Δ	Δ	O	O	Δ	O	O	O	Δ	Δ	Δ	Δ	Δ	O	Δ	Δ	Δ	O
top	-	O	O	-	-	-	-	O	O	-	O	O	O	-	-	-	-	-	O	-	-	-	O
right	-	Δ	Δ	-	-	-	-	Δ	Δ	-	Δ	Δ	Δ	-	-	-	-	-	Δ	-	-	-	Δ
bottom	-	Δ	Δ	-	-	-	-	Δ	Δ	-	Δ	Δ	Δ	-	-	-	-	-	Δ	-	-	-	Δ
left	-	O	O	-	-	-	-	O	O	-	O	O	O	-	-	-	-	-	O	-	-	-	O
float	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	O	O
clear	-	O	O	-	-	-	-	O	O	-	O	O	-	-	-	-	-	-	O	-	-	-	-
z-index	-	Δ	Δ	-	-	-	-	Δ	Δ	-	Δ	Δ	Δ	-	-	-	-	-	Δ	-	-	-	Δ
width	Δ	O	O	-	-	-	-	O	O	Δ	O	O	O	-	Δ	Δ	Δ	Δ	O	-	Δ	O	O
height	Δ	O	O	-	-	-	-	O	O	Δ	O	O	O	-	Δ	Δ	Δ	Δ	O	Δ	Δ	O	O
line-height	O	O	O	Δ	Δ	Δ	Δ	O	O	Δ	-	O	O	Δ	Δ	Δ	Δ	Δ	O	Δ	Δ	-	-
vertical-align	-	-	-	Δ	Δ	Δ	Δ	-	-	Δ	-	-	Δ	Δ	Δ	Δ	Δ	Δ	O	O	O	O	O
Other visual effects																							
overflow	Δ	O	O	-	-	-	-	O	O	O	-	O	-	-	-	-	Δ	O	O	O	O	O	O
clip	Δ	Δ	Δ	-	-	-	-	Δ	Δ	Δ	-	Δ	-	-	-	-	Δ	Δ	Δ	Δ	Δ	Δ	Δ
visibility	Δ	O	O	Δ	Δ	Δ	Δ	O	O	O	O	O	O	O	O	Δ	O	O	O	O	O	O	O
Generated content/Auto-numbering/List																							
content	-	-	-	-	Δ	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
quotes	-	-	-	-	Δ	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
list-style-type	-	-	-	-	-	-	-	O	-	Δ	-	-	-	-	-	-	-	-	-	-	-	-	-
list-style-image	-	-	-	-	-	-	-	Δ	-	Δ	-	-	-	-	-	-	-	-	-	-	-	-	-
list-style-position	-	-	-	-	-	-	-	Δ	-	Δ	-	-	-	-	-	-	-	-	-	-	-	-	-
list-style	-	-	-	-	-	-	-	O	-	Δ	-	-	-	-	-	-	-	-	-	-	-	-	-
Colours and Background																							
clut	I	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
color	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
color-index	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
background-color	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
background-color-index	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
background-image	O	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
background-repeat	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
background	C	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Fonts																							
font-family	Δ	Δ	Δ	Δ	Δ	-	Δ	Δ	Δ	Δ	-	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	-	-
font-style	Δ	Δ	Δ	Δ	Δ	-	Δ	Δ	Δ	Δ	-	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	-	-

Property	body	h1-h6, address, blockquote, pre	div, p	em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, b, big, i, small, sub, sup, tt	q	br	span, a	ol, ul	dl , dt, dd	li	hr	form	input	label	select	option	textarea	caption	table	tr	th, td	img	object
font-weight	O	O	O	O	O	-	O	O	O	O	-	O	O	O	O	O	O	O	O	O	O	-	-
font-size	O	O	O	O	O	-	O	O	O	O	-	O	O	O	O	O	O	O	O	O	O	-	-
font	O	O	O	O	O	-	O	O	O	O	-	O	O	O	O	O	O	O	O	O	O	-	-
Text																							
text-align	O	O	O	-	-	-	-	O	O	O	-	O	O	-	-	-	-	O	O	O	O	-	-
text-decoration	O	O	O	O	O	-	O	O	O	O	-	O	O	O	O	O	O	O	O	O	O	-	-
letter-spacing	O	O	O	Δ	Δ	Δ	Δ	O	O	O	-	O	O	Δ	Δ	Δ	Δ	O	O	O	O	-	-
white-space	Δ	Δ	Δ	-	-	-	-	Δ	Δ	Δ	-	Δ	Δ	-	-	-	-	Δ	Δ	Δ	Δ	-	-
Tables																							
caption-side	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	O	-	-	-	-	-
table-layout	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Δ	-	-	-	-
border-collapse	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Δ	-	-	-	-
border-spacing	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	O	-	-	-	-
Extended properties																							
resolution	O	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
display-aspect-ratio	O	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
greyscale-color-index	I	I	I	I	I	-	I	I	I	I	-	I	I	I	I	I	I	I	I	I	I	I	I
nav-index	-	-	I	-	-	-	I	-	-	-	-	-	I	-	I	-	I	-	-	-	-	-	I
nav-up	-	-	I	-	-	-	I	-	-	-	-	-	I	-	I	-	I	-	-	-	-	-	I
nav-right	-	-	I	-	-	-	I	-	-	-	-	-	I	-	I	-	I	-	-	-	-	-	I
nav-down	-	-	I	-	-	-	I	-	-	-	-	-	I	-	I	-	I	-	-	-	-	-	I
nav-left	-	-	I	-	-	-	I	-	-	-	-	-	I	-	I	-	I	-	-	-	-	-	I
used-key-list	O	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-wap-marquee-style	-	-	-	-	-	-	O	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-wap-marquee-loop	-	-	-	-	-	-	O	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-wap-marquee-dir	-	-	-	-	-	-	O	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-wap-marquee-speed	-	-	-	-	-	-	O	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-wap-marquee	-	-	-	-	-	-	O	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

A5-4.6.2 Operation of selectors

This section lists the available selectors to an external style sheet or the style element among the selectors defined in CSS2. The * symbol represents a universal selector. The term, element, represents a type selector. The term, .class, represents a class selector. The term, #id, represents an ID selector. The term, :pseudoClass, represents an :active pseudo-class or a :focus pseudo-class. Any order in describing #id, :pseudoClass, and .class in a single selector is accepted. To operate a selector, the available combinations of a CSS property and a BML element are limited to the combinations marked by O in Table 4-11. How a signal that is not operable is processed is implementation dependent.

*

element

.class

#id

:pseudoClass

*.class

*:pseudoClass

*#id

element.class

element:pseudoClass

.class:pseudoClass

#id:pseudoClass

*.class:pseudoClass

element.class:pseudoClass

*#id:pseudoClass

Note that this section does not restrict any usage nor operation of a default style sheet.

A5-4.6.3 Media type

Among the @ rules, only @media rules must be operated. The other @ rules, @import rules and the @page rules must not operated. The available media type to @media rules is limited to tv. When a media type other than tv is specified according to a @media rule, any receiver must ignore the designated properties in the @media rule.

A5-4.6.4 !important rule

No !important rule must be operated.

A5-4.6.5 Box model

This section provides operational guidelines on box model properties.

- margin-top, margin-right, margin-bottom, margin-left

  Available values <margin-width>

  Initial value: 0

  Inherited: no

  The <margin-width> value type may take one of the following values: <length> (not a negative value), <percentage> (not a negative value), or auto. However, the auto value is not operated in a BML document based on an index-based model.

- margin

  Available value: <margin-width>{1,4}

  Initial value: not defined because of shorthand properties

  Inherited: no

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

  Available value: <padding-width>

  Initial value: 0

  Inherited: no

The <padding-width> value type may take one of the following values: <length> (not a negative value that is not supported by CSS2) or <percentage> (not a negative value that is not supported by CSS2). However, the auto value is not operated in a BML document based on an index-based model.

- padding

  Available value: <padding-width>{1,4}

  Initial value: not defined because of shorthand properties

  Inherited: no

- border-top-width, border-right-width, border-bottom-width, border-left-width

  Available value: <border-width>

  Initial value: medium

  Inherited: no

  The <border-width> value type may take one of the following values: thin, medium, thick, or <length> (not a negative value that is illegal). The thin value, the medium value, the thick value must be interpreted as 1px, 2px, 5px, respectively.

- border-width

  Available value: <border-width>{1,4}

  Initial value: refer to individual properties

  Inherited: no

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

  Available value: <color> | transparent | inherit

  Initial value: the value of the colour property

  Inherited: no

  This property is applicable only to a component-based model and not operated in an index-based model. An order in rendering borders is implementation dependent.

- border-color

  Available value: <color>{1,4} | transparent | inherit

  Initial value: refer to individual properties

  Inherited: no

  This property is applicable only to a component-based model and not operated in an index-based model.

- border-top-color-index, border-right-color-index, border-bottom-color-index,

  border-left-color-index

  Available value: <color-index> | inherit

  Initial value: 0

  Inherited: no

  This property is applicable only to an index-based model and not operated in a component-based model. An order in rendering borders is implementation dependent. When this property is omitted, the initial value, 0, is assumed instead of the color-index value of the element.

- border-color-index

Available value: <color-index>{1,4} | inherit

Initial value: refer to individual properties

Inherited: no

This property is applicable only to an index-based model and not operated in a component-based model.

- border-top-style, border-right-style, border-bottom-style, border-left-style

Available value: none | dotted | dashed | solid

Initial value: none

Inherited: no

How the specified dotted, dashed, or solid line style is presented is implementation dependent.

- border-style

Available value: [none | dotted | dashed | solid]{1,4}

Initial value: refer to individual properties

Inherited: no

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

Available value: <'border-width'> || [none | dotted | dashed | solid] | | [<color> | transparent] | inherit

Initial value: refer to individual properties

Inherited: no

This property is applicable only to a component-based model and not operated in an index-based model.

- border

Available value: <'border-width'> | | [none | dotted | dashed | solid] | | [<color> | transparent] | inherit

Initial value: refer to individual properties

Inherited: no

This property is applicable only to a component-based model and not operated in an index-based model.

A5-4.6.6 Visual formatting model

This section provides operational guidelines on visual formatting models.

- display

This property must contain a fixed value specific to an element. Refer to Annex A.

- position

Available value: static | absolute

Initial value: static

Inherited: no

A parent element whose child element takes the absolute value must take the absolute value or must be the body element. However, depending on an element, a fixed value may be applied. See Table 4-11 for elements that may take fixed values.

- left, top

  Available value: <length> | <percentage> | auto

  Initial value: auto

  Inherited: no

- right, bottom

  This property must take the following fixed value.

  Fixed value: auto

- float

  Available value: left | right | none

  Initial value: none

  Inherited: no

- clear

  Available value: none | left | right | both

  Initial value: none

  Inherited: no

- z-index

  This property must take the following fixed value.

  Fixed value: auto

- width, height

  Available value: <length> | <percentage> | auto

  Initial value: auto

  Inherited: no

- line-height

  Available value: normal | <number> | <length> | <percentage>

  Initial value: normal

  Inherited: yes

  When this property takes normal, a value that is 1.2 times as great as font-size is assumed. However, depending on an element, the fixed value, inherit, may be applied. Any negative value is illegal for <number> and <percentage>, as defined in CSS2.

- vertical-align

  Available value: baseline | top | middle | bottom

  Initial value: baseline

  Inherited: no

  The baseline value is not applicable to the table related elements except the caption element. The values, sub, super, and inherit, are applicable only to a default style sheet. How the element is rendered when sub or super is specified is implementation dependent.

A5-4.6.7 Other visual effects

This section provides operational guidelines on other visual effects.

- overflow

  Available value: visible | hidden

  Initial value: visible

  Inherited: no

- clip

  This property must take the following fixed value.

  Fixed value: auto

- visibility

  Available value: visible | hidden | inherit

  Initial value: inherit

  Inherited: no

  When the visibility property of a parent element is set to hidden and the visibility property of the child element is set to visible, only the parent element is not presented. It is allowed that this property contains the fixed value, inherit, depending on an element. For more information, refer to Table 4-11. Note that the visibility property of the body element must contain the fixed value, visible.

A5-4.6.8 Generated content, autonumbering, list

This section provides operational guidelines on generated content, autonumbering, and lists.

- content

  This property must contain a fixed value specific to an element. Refer to Annex A.

- quotes

  This property must contain a fixed value specific to an element. Refer to Annex A.

- list-style-type

  Available value: disc | circle | square | decimal | lower-roman | upper-roman | lower-alpha | lower-Latin | upper-alpha | upper-Latin | inherit

  Initial value: disc

  Inherited: yes

- list-style-image

  This property must take the following fixed value.

  Fixed value: none

- list-style-position

  This property must take the following fixed value.

  Fixed value: outside

- list-style

Available value: <'list-style-type'> | inherit

Initial value: not defined because of shorthand properties

Inherited: yes

A5-4.6.9 Colours and backgrounds

This section provides operational guidelines on colours and backgrounds.

- clut

  This property must be operated as described below.

  Available value: <uri>

  Initial value: variable

  Inherited: yes

  For detailed operational specification on this property, refer to Section A5-5.1.8. This property is applicable only to an index-based model and not operated in a component-based model.

- color

  Available value: <color> | inherit

  Initial value: black

  Inherited: yes

  This property is applicable only to a component-based model and not operated in an index-based model.

- color-index

  This property must be operated as described below.

  Available value: <color-index> | inherit

  Initial value: 0

  Inherited: yes

  This property is applicable only to an index-based model and not operated in a component-based model.

- background-color

  Available value: <color> | transparent | inherit

  Initial value: transparent

  Inherited: no

  This property is applicable only to a component-based model and not operated in an index-based model.

- background-color-index

  Available value: <color-index> | inherit

  Initial value: 0

  Inherited: yes

  This property is applicable only to an index-based model and not operated in a component-based model.

- background-image

Available value: <uri> | none

Initial value: none

Inherited: no

The graphic image coding specification specified with the <uri> value type must be limited to JPEG. This property is applicable only to the body element. The upper left corner of the presented area must be rendered first.

- background-repeat

  This property must take the following fixed value.

  Fixed value: repeat

- background

  Available value: <'background-color'> || <'background-image'>

  Initial value: not defined for shorthand properties

  Inherited: no

  This property is applicable only to a component-based model and not operated in an index-based model.

A5-4.6.10 Fonts

The available specifications for a font in a BML document are listed in Table 4-12. Any character of the code set 0 in EUC-JP and the 1-byte characters in Shift-JIS is rendered with a half width of the font size. Any external character specified by using DRCS is a full-width character and rendered with the corresponding font size.

Table 4-12 Available Fonts

Font family	Font size (Note 1)	Gradation (Note 2)
round gothic	12 px, 16 px, 24 px, 32 px, 48 px, 64 px	4

Note 1: The sizes 48px and 64px are operated only in a receiver that supports a high-resolution presentation function defined in Section A5-5.9.1.
Note 2: The gradation value for a component-based model is implementation dependent.

Operational guidelines on properties related to fonts are described below.

- font-family

  This property must take the fixed value, "round gothic".

  Which font-family is actually presented depends on an implementation.

- font-style

  This property must contain a fixed value, as defined in a default style sheet. Which font family is applied to the value "italic" is implementation dependent.

- font-weight

  Available value: normal | bold | inherit

  Initial value: normal

  Inherited: yes

  How a bold character is rendered complies with the definition of text with a control code (8-bit character code) specified with the text/X-arib-jis8text media type.

- font-size

Available value: <length> (0 or an integer in pixels. Refer to Table 4-12 for the available values.) | inherit

Initial value: 16 px

Inherited: yes

A5-4.6.11 Text

This section provides operational guidelines on properties related to text.

- text-align

  Available value: left | right | center

  Initial value: left

  Inherited: yes

- text-decoration

  Available value: none | blink | underline | inherit

  Initial value: none

  Inherited: no

- letter-spacing

  Available value: normal | <length> | inherit

  Initial value: normal

  Inherited: yes

  When normal is specified, 0 is assumed as a computed value. The <length> value type must not take a negative value. This property specifies the length of a space between characters in the specified elements. This property may take the fixed value, inherit depending on an element.

- white-space

  This property must take a fixed value. Refer to Annex A.

A5-4.6.12 Tables

This section provides operational guidelines on properties related to tables.

- caption-side

  Available value: top | bottom

  Initial value: top

  Inherited: yes

- table-layout

  This property must take the following fixed value.

  Fixed value auto

- border-collapse

  This property must take the following fixed value.

  Fixed value separate

- border-spacing

Available value: <length> <length>?

Initial value: 0

Inherited: yes

A5-4.6.13 Pseudo classes

The CSS properties available to the :active and :focus classes are listed in Table 4-13.

Table 4-13 Operational Guideline on CSS Properties Available to :focus and :active Classes

Legend O: Refer to Chapter A5-3. If any restriction exists, it is expressed in the corresponding operational guideline.
-: Refer to Chapter A5-3.

Property	body	h1-h6, address, blockquote, pre	div	p	em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, b, big, i, small sub, sup, tt	q	br	span, a	ol, ul	dl, ,dt, dd	li	hr	form	input	label	select	option	textarea	caption	table	tr	th, td	img	object
Box model																								
margin-top	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
margin-right	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
margin-bottom	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
margin-left	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
margin	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
padding-top	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
padding-right	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
padding-bottom	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
padding-left	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
padding	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-top-width	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-right-width	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-bottom-width	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-left-width	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-width	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-top-color	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
border-right-color	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
border-bottom-color	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
border-left-color	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
border-color	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-top-color-index	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
border-right-color-index	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
border-bottom-color-index	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
border-left-color-index	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
border-color-index	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-top-style	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-right-style	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-bottom-style	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-left-style	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-style	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Property	body	h1-h6, address, blockquote, pre	div	p	em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, b, big, i, small sub, sup, tt	q	br	span, a	ol, ul	dl, ,dt, dd	li	hr	form	input	label	select	option	textarea	caption	table	tr	th, td	img	object
border-top	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-right	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-bottom	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-left	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Visual formatting model																								
display	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
position	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
top	-	-	O	O	-	-	-	-	-	-	-	-	-	O	-	-	-	-	-	-	-	-	-	O
right	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bottom	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
left	-	-	O	O	-	-	-	-	-	-	-	-	-	O	-	-	-	-	-	-	-	-	-	O
float	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
clear	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
z-index	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
width	-	-	O	O	-	-	-	-	-	-	-	-	-	O	-	-	-	-	-	-	-	-	-	O
height	-	-	O	O	-	-	-	-	-	-	-	-	-	O	-	-	-	-	-	-	-	-	-	O
line-height	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
vertical-align	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Other visual effects																								
overflow	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
clip	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
visibility	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Generated content/Autonumbering/List																								
content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
quotes	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
list-style-type	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
list-style-image	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
list-style-position	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
list-style	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Colors and Backgrounds																								
clut	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
color	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
color	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
background-color	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
background-color-index	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	O
background-image	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
background-repeat	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
background	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Fonts																								
font-family	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
font-style	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
font-weight	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	-
font-size	-	-	O	O	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	-	-	-	-	-
font	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Text																								

Property	body	h1-h6, address, blockquote, pre	div	p	em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, b, big, i, small sub, sup, tt	q	br	span, a	ol, ul	dl, ,dt, dd	li	hr	form	input	label	select	option	textarea	caption	table	tr	th, td	img	object
text-align	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
text-decoration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
letter-spacing	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
white-space	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Tables																								
caption-side	-	-	-		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
table-layout	-	-	-		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-collapse	-	-	-		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border-spacing	-	-	-		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
empty-cells	-	-	-		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Extended properties																								
resolution	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
display-aspect-ratio	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
grayscale-color-index	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
nav-index	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
nav-up	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
nav-right	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
nav-down	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
nav-left	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
used-key-list	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-wap-marquee-style	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-wap-marquee-loop	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-wap-marquee-dir	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-wap-marquee-speed	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-wap-marquee	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

A5-4.6.14 Extended properties

This section provides operational guidelines on extended properties.

- resolution

  Available value: 960x540 | 640x480 | 480x270 | 320x240

  Initial value: 320x240

  Inherited: no

- display-aspect-ratio

  Available value: 16v9 | 4v3

  Initial value: 4v3

  Inherited: no

- grayscale-color-index

  Available value: <color-index> <color-index> | inherit

  Initial value: the value of the color-index property

Inherited: yes

The four gradation colours are, the foreground colour, the first <color-index> value, the second <color-index> value and the background colour. When 8 is specified, the background colour is displayed.

This property is applicable only to an index-based model and not operated in a component-based model.

Note that when DRCS is used for a BML content based on a component-based model, which colours are applied to the medium grey scale specifications in DRCS depends on an implementation.

- used-key-list

Available value: <key-group>+ | none

Initial value: basic data-button

Inherited: no

A5-4.6.15 Default style sheet

The available default style sheets are shown in Annex A. Each default style sheet is assumed to be virtually implemented in a receiver. There is no need to transmit any of the default style sheets. However, depending on an individual receiver, additional specifications may be contained in the default style sheet.

A5-4.6.16 Operation of length

The <length> value type must be operated in order to specify a length in em, ex, or px units. Which unit is used as a basic unit when em or ex is specified is implementation dependent. When 0 is specified, 0px is assumed as a computed value.

A5-4.6.17 Actual values derived from computed values

How is an actual value derived from a computed value that is a decimal fraction is implementation dependent.

A5-4.6.18 <color-index>

The <color-index> value type must contain an integer, 0 through 223.

A5-4.7 Operational Guidelines on Procedural Description

A5-4.7.1 Operational guidelines on DOM

This section provides tables that indicate the available interface, attribute, and method is applicable and use the following symbols:

The O mark means that the interface, attribute, or method must be operated.

The - mark means that the interface, attribute, or method must not be operated.

The R character means that any access to the interface, attribute, or method is limited to a reading access.

The RW string means that any access to the interface, attribute, or method may be a reading access or a writing access.

A5-4.7.1.1 Operation of DOM core interfaces

This section provides the table that indicates which DOM core interfaces are allowed to be operated. This section also provides the table that describes which and how attributes and methods of a DOM core interface are allowed to be operated.

Table 4-14 Operational Guidelines on DOM Core Interfaces Part 1

Interface Classification	Interface	Operation
Core 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	-

Table 4-15 Operational Guidelines on DOM Core Interfaces Part 2

Interface	Attribute/Method	Operation	Restriction
DOMImplementation	hasFeature()	O	
Document	doctype	-	
	implementation	O	R
	documentElement	O	R
	createElement()	-	
	createDocumentFragment()	-	
	createTextNode()	-	
	createComment()	-	
	createCDATASection()	-	
	createProcessingInstruction()	-	
	createAttribute()	-	
	createEntityReference()	-	
	getElementsByTagName()	-	
Node	nodeName	-	
	nodeValue	-	
	nodeType	-	
	parentNode	O	R

Interface	Attribute/Method	Operation	Restriction
	childNodes	-	
	firstChild	O	R
	lastChild	O	R
	previousSibling	O	R
	nextSibling	O	R
	attrbutes	-	
	ownerDocument	-	
	insertBefore()	-	
	replaceChild()	-	
	removeChild()	-	
	appendChild()	-	
	hasChildNodes()	-	
	cloneNode()	-	
CharacterData	data	O (Note 1)	RW (Note 2)
	length	O (Note 1)	R
	substringData()	-	
	appendData()	-	
	insertData()	-	
	deleteData()	-	
	replaceData()	-	
Element	tagName	O	R
	getAttribute()	-	
	setAttribute()	-	
	removeAttribute()	-	
	getAttributeNode()	-	
	setAttributeNode()	-	
	removeAttributeNode()	-	
	getElementsByTagName()	-	
	normalize()	-	
Text	splitText()	-	
CDATASection			

Note 1: No attribute in the above table is applicable to any CDATA section node that is a child node of the style element node or the script element node.

Note 2: The restriction level R is applicable to any CDATA section node that is a child node of the option element node or any text node that is a child node of the option element node.

A5-4.7.1 2 Operation of DOM HTML interfaces

Operation of DOM HTML interfaces applicable to events

This section provides the table that indicates which DOM HTML interfaces are applicable to events. This section also contains the table that describes what and how values of the type attribute and the target attribute are obtained by using the BMLEvent interface.

Table 4-16 Operational Guidelines on Interfaces Applied to Events

Interface	Attribute/Method	Operation	Restriction
BMLEvent	type	O	R
	target	O	R
BMLIntrinsicEvent	keyCode	O	R
BMLBeventEvent:	status	O	R
	privateData	O	R

Interface	Attribute/Method	Operation	Restriction
	esRef	O (Note 1)	R
	messageId	O	R
	messageVersion	O	R
	messageGroupId	O	R
	moduleRef	O (Note 2)	R
	languageTag	O	R
	registerId	-	
	serviceId	-	
	eventId	-	
	peripheralRef	O (Note 3)	R
	object	O	R

Note 1: The format of a value obtained through this attribute must be /<component_tag>, as defined in Section 9.2, Volume 2.
Note 2: The format of a value obtained through this attribute must be /<component_tag>/<module_id>, as defined in Section 9.2, Volume 2.
Note 3: Whether the attribute is operated or not depends on an operational standard regulation for each media type.

Table 4-17 Events and Attributes

Condition for reading access			Value of type attribute	Value of target attribute
Event handler onkeydown is executed			"keydown"	Element specifying the event handler
Event handler onkeyup is executed			"keyup"	Element specifying the event handler
Event handler onclick is executed			"click"	Element specifying the event handler
Event handler onfocus is executed			"focus"	Element specifying the event handler
Event handler onblur is executed			"blur"	Element specifying the event handler
Event handler onload is executed			"load"	Element specifying the event handler
Event handler onunload is executed			"unload"	Element specifying the event handler
Event handler onchange is executed			"change"	Element specifying the event handler
Event handler onsubmit is executed			"submit"	Element specifying the event handler
Event handler onreset is executed			"reset"	Element specifying the event handler
Event handler Onoccur is executed		EventMessageFired	"EventMessageFired"	
		ModuleUpdated	"ModuleUpdated"	
		ModuleLocked	"ModuleLocked"	
		TimerFired	"TimerFired"	
		NPTReferred	"NPTReferred"	Element specifying the handler
		MediaStopped (Note)	"MediaStopped"	
		DataEventChanged	"DataEventChanged"	
		CCStatusChanged	"CCStatusChanged"	
		MainAudioStreamChanged	"MainAudioStreamChanged"	
		DataButtonPressed	"DataButtonPressed"	
		PeripheralEventOccured	"PeripheralEventOccured"	
Global code is executed			undefined	null
Specified procedure description is executed upon invoking setInterval(), extended function for broadcasting			undefined	null

Note: An event MediaStopped arises from monomedia: audio/X-arib-mpeg2-aac, audio/X-arib-aiff, and application/X-arib-mpeg2-tts.

Operation of interfaces applicable to document nodes

The following table indicates which and how DOM HTML interface and DOM BML interface is applicable to document nodes.

Table 4-18 Operational Guidelines on Interfaces Applicable to Document Nodes

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

Note: When a value obtained through each attribute in the above table is assigned to a global variable, how the interface behaves depends on an implementation.

Operation of DOM HTML interfaces applicable to element nodes

The following table describes which and how DOM HTML interfaces are applicable to element nodes.

Table 4-19 Operational Guidelines on Interfaces Applicable to Element Nodes

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
html	HTMLElement	id	-	
		className	-	
		title	-	
		lang	-	
		dir	-	
	HTMLHtmlElement	version	-	
	BMLBmlElement	style	-	
		normalStyle	-	
		focusStyle	-	
		activeStyle	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
bml:bevent	HTMLElement	id	O	R
		className	-	
		title	-	
		lang	-	
		dir	-	
	BMLBeventElement			
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
		languageTag	O	RW
		registerId	-	
		serviceId	-	
		eventId	-	
		timeMode	O	R
		timeValue	O	RW
		objectId	O	RW
		subscribe	O	RW
		peripheralRef (Note 3)	O	RW
body	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLBodyElement			
	BMLBodyElement	invisible	O	RW
		style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
head	HTMLElement	id	-	
		className	-	
		title	-	
		lang	-	
		dir	-	
	HTMLHeadElement	profile	-	
title	HTMLElement	id	-	
		className	-	
		title	-	
		lang	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		dir	-	
	HTMLTitleElement	text	O	R
h1 h2 h3 h4 h5 h6	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLHeading Element			
	BMLHeadingElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
address	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
blockquote	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLBlockquote Element	cite	-	
	BMLBlockquote Element	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
div	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLDivElement			
	BMLDivElement	style	-	
		normalStyle	O	R
		focusStyle	O	R
		activeStyle	O	R
		accessKey	O	R
		focus()	O	
		blur()	O	
p	HTMLElement	id	O	R
		className	O	R
		title	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		lang	-	
		dir	-	
	HTMLParagraphElement			
	BMLParagraphElement	style	-	
		normalStyle	O	R
		focusStyle	O	R
		activeStyle	O	R
		accessKey	O	R
		focus()	O	
		blur()	O	
pre	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLPreElement			
	BMLPreElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
abbr	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
acronym	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
br	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLBRElement			
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		activeStyle	-	
cite	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
code	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
dfn	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
em	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
kbd	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
q	HTMLElement	id	O	R

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLQuoteElement	cite	-	
	BMLQuoteElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
samp	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
span	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLSpanElement	style	-	
		normalStyle	O	R
		focusStyle	O	R
		activeStyle	O	R
		accessKey	O	R
		focus()	O	
		blur()	O	
strong	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
var	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		activeStyle	-	
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()	O	
		focus()	O	
	BMLAnchorElement	style	-	
		normalStyle	O	R
		focusStyle	O	R
		activeStyle	O	R
		effect	-	
dl	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLDListElement			
	BMLDListElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
dt	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
dd	HTMLElement	id	O	R
		className	O	R
		title	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
ol	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLOListElement			
	BMLOListElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
ul	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLUListElement			
	BMLUListElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
li	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLLIElement			
	BMLLIElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
b	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
big	HTMLElement	id	O	R
		className	O	R

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
hr	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLHRElement			
	BMLHREElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
i	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
small	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
sub	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
sup	HTMLElement	id	O	R
		className	O	R
		title	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
tt	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	BMLElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
form	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLFormElement	elements	-	
		length	-	
		name	-	
		acceptCharset	-	
		action	O	RW
		enctype	O	RW
		method	O	R
		submit()	O	
		reset()	O	
	BMLFormElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
		accept	O	R
input	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLInputElement	defaultValue	O	R
		defaultChecked	O	R
		form	O	R
		accept	-	
		accessKey	O	R
		alt	O	RW
		checked	O	RW
		disabled	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		maxLength	O	R
		name	-	
		readOnly	-	
		size	O	R
		src	O	RW (Note 1)
		tabIndex	-	
		type	O	R
		useMap	-	
		value	O	RW
		blur()	O	
		focus()	O	
		select()	-	
		click()	-	
	BMLInputElement	style	-	
		normalStyle	O	R
		focusStyle	O	R
		activeStyle	O	R
label	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLLabelElement	form	O	R
		accessKey	-	
		htmlFor	O	R
	BMLLabelElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
select	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLSelectElement	type	-	
		selectedIndex	O	R
		value	-	
		length	-	
		form	O	R
		options	-	
		disabled	-	
		multiple	O	R
		name	O	R
		size	O	R
		tabIndex	-	
		add()	-	
		remove()	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		blur()	O	
		focus()	O	
	BMLSelectElement	style	-	
		normalStyle	O	R
		focusStyle	O	R
		activeStyle	O	R
option	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLOptionElement	form	O	R
		defaultSelected	O	R
		text	O	R
		index	O	R
		disabled	-	
		label	-	
		selected	O	RW
		value	O	R
	BMLOptionElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
textarea	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLTextAreaElement	defaultValue	O	R
		form	O	R
		accessKey	O	R
		cols	O	R
		disabled	-	
		name	O	R
		readOnly	-	
		rows	O	R
		tabIndex	-	
		type	-	
		value	O	RW
		blur()	O	
		focus()	O	
		select()	-	
	BMLTextAreaElement	style	-	
		normalStyle	O	R
		focusStyle	O	R
		activeStyle	O	R
caption	HTMLElement	id	O	R

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLTableCaptionElement			
	BMLTableCaptionElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
table	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLTableElement	caption	-	
		tHead	-	
		tFoot	-	
		rows	-	
		tBodies	-	
		border	-	
		cellPadding	-	
		cellSpacing	-	
		frame	-	
		rules	-	
		summary	-	
		width	-	
		createTHead()	-	
		deleteTHead()	-	
		createTFoot()	-	
		deleteTFoot()	-	
		createCaption()	-	
		deleteCaption()	-	
		insertRow()	-	
		deleteRow()	-	
	BMLTableElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
tr	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLTableRowElement	rowIndex	-	
		sectionRowIndex	-	
		cells	-	
		align	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		ch	-	
		chOff	-	
		vAlign	-	
		insertCell()	-	
		deleteCell()	-	
	BMLTableRowElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
th td	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLTableCellElement	cellIndex	-	
		abbr	-	
		align	-	
		axis	-	
		ch	-	
		chOff	-	
		colSpan	-	
		headers	-	
		rowSpan	-	
		scope	-	
		vAlign	-	
	BMLTableCellElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	
img	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLImageElement	lowSrc	-	
		name	-	
		alt	-	
		height	-	
		isMap	-	
		longDesc	-	
		src	-	
		useMap	-	
		width	-	
	BMLImageElement	style	-	
		normalStyle	O	R
		focusStyle	-	
		activeStyle	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
object	HTMLElement	id	O	R
		className	O	R
		title	-	
		lang	-	
		dir	-	
	HTMLObjectElement	form	-	
		code	-	
		archive	-	
		codeBase	-	
		codeType	-	
		data	O	RW
		declare	-	
		height	-	
		name	-	
		standby	-	
		tabIndex	-	
		type	O	R
		useMap	-	
		width	-	
	BMLObjectElement	style	-	
		normalStyle	O	R
		focusStyle	O	R
		activeStyle	O	R
		classId	-	
		accessKey	O	R
		remain	O	RW
		streamPosition	O (Note 2)	RW
		streamStatus	O	RW
		streamLooping	-	
		streamSpeedNumerator	-	
		streamSpeedDenominator	-	
		streamLevel	-	
		setSpeed()	-	
		movePosition()	-	
		hasAssociatedIndex()	-	
		assignToLocalEvnet()	-	
		assignToNodePlayMode()	-	
		getMainAudioStream()	O	
		setMainAudioStream()	O	
		focus()	O	
		Blur()	O	
param	HTMLElement	Id	-	
		className	-	
		title	-	
		lang	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
		dir	-	
	HTMLParamElement	name	-	
		type	-	
		value	-	
		valueType	-	
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	-	
style	HTMLElement	id	-	
		className	-	
		title	-	
		lang	-	
		dir	-	
	HTMLStyleElement	disabled	-	
		media	-	
		type	-	
link	HTMLElement	id	-	
		className	-	
		title	-	
		lang	-	
		dir	-	
	HTMLLinkElement	disabled	-	
		charset	-	
		href	-	
		hreflang	-	
		media	-	
		rel	-	
		rev	-	
		type	-	

Element represented by Node	Applicable Interface	Attribute/Method	Operation	Restriction
	BMLLinkElement	style	-	
		normalStyle	-	
		focusStyle	-	
		activeStyle	-	
base	HTMLElement	id	-	
		className	-	
		title	-	
		lang	-	
		dir	-	
	HTMLBaseElement	href	-	

Note 1: When this attribute is allowed to be rewritten, the type of a referenced monomedia must retain before and after the rewriting operation.
Note 2: This attribute is allowed to be applicable only to a node whose type attribute is set to “image/X-arib-mng”.
Note 3: Whether this attribute is operated or not depends on an operational standard regulation for each media type.

Operation of BMLCSS2Properties interface

The BMLCSS2Properties interface is allowed to be operated based on a presentation model (an index-based model or a component-based model) described in Section A5-5.1.1. Table 4-20 summarizes operational guidelines on the BMLCSS2Properties interface.

The following table contains semantics of each symbol used in Table 4-20.

Symbol	Semantics
RW	Operated as readable and rewritable
cRW	In a component-based model, operated as readable and rewritable. In an index-based model, not operated.
iRW	In an index-based model a component-based model, operated as readable and rewritable. In a component-based model, not operated.
R	Operated only as readable
cR	In a component-based model, operated only as readable. In an index-based model, not operated.
iR	In an index-based model a component-based model, operated only as readable. In a component-based model, not operated.

Table 4-20 Operational Guidelines on BMLCSS2Properties Interface

Upper Row: via the normalStyle attribute Lower Row: via the focusStyle or activeStyle attribute	body	h1, h2, h3, h4, h5, h6, address, blockquote, pre	div, p	em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, b, big, i, small, sub, sup, tt	q	br	span, a	ol, ul, dl	dt, dd	li	hr	form	input	label	select	option	textarea	caption	table	tr	th, td	img	object
marginTop	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
marginRight	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
marginBottom	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
marginLeft	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
margin	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Upper Row: via the normalStyle attribute Lower Row: via the focusStyle or activeStyle attribute	body	h1, h2, h3, h4, h5, h6, address, blockquote, pre	div, p	em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, b, big, i, small, sub, sup, tt	q	br	span, a	ol, ul, dl	dt, dd	li	hr	form	input	label	select	option	textarea	caption	table	tr	th, td	img	object
paddingTop	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
paddingRight	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
paddingBottom	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
paddingLeft	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
padding	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderTopWidth	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	R	R	R
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderRightWidth	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	R	R	R
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderBottomWidth	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	R	R	R
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderLeftWidth	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	R	R	R
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderWidth	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderTopColor	cR	cRW	cRW	cRW	cRW	cR	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cR	cRW	cRW	cRW	-	cRW	cRW	cRW
	-	-	cRW	-	-	-	cRW	-	-	-	-	-	cRW	-	cRW	-	cRW	-	-	-	-	-	cRW
borderRightColor	cR	cRW	cRW	cRW	cRW	cR	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cR	cRW	cRW	cRW	-	cRW	cRW	cRW
	-	-	cRW	-	-	-	cRW	-	-	-	-	-	cRW	-	cRW	-	cRW	-	-	-	-	-	cRW
borderBottomColor	cR	cRW	cRW	cRW	cRW	cR	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cR	cRW	cRW	cRW	-	cRW	cRW	cRW
	-	-	cRW	-	-	-	cRW	-	-	-	-	-	cRW	-	cRW	-	cRW	-	-	-	-	-	cRW
borderLeftColor	cR	cRW	cRW	cRW	cRW	cR	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cR	cRW	cRW	cRW	-	cRW	cRW	cRW
	-	-	cRW	-	-	-	cRW	-	-	-	-	-	cRW	-	cRW	-	cRW	-	-	-	-	-	cRW
borderColor	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderTopColorIndex	iR	iRW	iRW	iRW	iRW	iR	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iR	iRW	iRW	iRW	-	iRW	iRW	iRW
	-	-	iRW	-	-	-	iRW	-	-	-	-	-	iRW	-	iRW	-	iRW	-	-	-	-	-	iRW
borderRightColorIndex	iR	iRW	iRW	iRW	iRW	iR	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iR	iRW	iRW	iRW	-	iRW	iRW	iRW
	-	-	iRW	-	-	-	iRW	-	-	-	-	-	iRW	-	iRW	-	iRW	-	-	-	-	-	iRW
borderBottomColorIndex	iR	iRW	iRW	iRW	iRW	iR	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iR	iRW	iRW	iRW	-	iRW	iRW	iRW
	-	-	iRW	-	-	-	iRW	-	-	-	-	-	iRW	-	iRW	-	iRW	-	-	-	-	-	iRW
borderLeftColorIndex	iR	iRW	iRW	iRW	iRW	iR	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iR	iRW	iRW	iRW	-	iRW	iRW	iRW
	-	-	iRW	-	-	-	iRW	-	-	-	-	-	iRW	-	iRW	-	iRW	-	-	-	-	-	iRW
borderColorIndex	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderTopStyle	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	R	R	R
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderRightStyle	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	R	R	R
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderBottomStyle	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	R	R	R
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderLeftStyle	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	R	R	R
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderStyle	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderTop	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderRight	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderBottom	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderLeft	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
border	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Upper Row: via the normalStyle attribute Lower Row: via the focusStyle or activeStyle attribute	body	h1, h2, h3, h4, h5, h6, address, blockquote, pre	div, p	em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, b, big, i, small, sub, sup, tt	q	br	span, a	ol, ul, dl	dt, dd	li	hr	form	input	label	select	option	textarea	caption	table	tr	th, td	img	object
display	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
position	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
top	-	RW	RW	-	-	-	-	RW	RW	-	RW	RW	RW	-	-	-	-	-	RW	-	-	-	RW
	-	-	RW	-	-	-	-	-	-	-	-	-	RW	-	-	-	-	-	-	-	-	-	RW
right	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bottom	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
left	-	RW	RW	-	-	-	-	RW	RW	-	RW	RW	RW	-	-	-	-	-	RW	-	-	-	RW
	-	-	RW	-	-	-	-	-	-	-	-	-	RW	-	-	-	-	-	-	-	-	-	RW
float	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
clear	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
width	R	RW	RW	-	-	-	-	RW	RW	R	RW	RW	RW	-	R	R	R	R	RW	-	R	RW	RW
	-	-	RW	-	-	-	-	-	-	-	-	-	RW	-	-	-	-	-	-	-	-	-	RW
height	R	RW	RW	-	-	-	-	RW	RW	R	RW	RW	RW	-	R	R	R	R	RW	R	R	RW	RW
	-	-	RW	-	-	-	-	-	-	-	-	-	RW	-	-	-	-	-	-	-	-	-	RW
lineHeight	R	R	R	R	R	R	R	R	R	R	-	R	R	R	R	R	R	R	R	R	R	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
verticalAlign	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
overflow	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
clip	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
visibility	R	RW	RW	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	R	RW	RW	RW	RW	RW	RW	RW
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
quotes	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
listStyleType	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
listStyleImage	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
listStylePosition	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
listStyle	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
clut	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
color	cRW	cRW	cRW	cRW	cRW	-	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW
	-	-	cRW	-	-	-	cRW	-	-	-	-	-	cRW	-	cRW	-	cRW	-	-	-	-	-	cRW
colorIndex	iRW	iRW	iRW	iRW	iRW	-	iRW	iRW	iRW	iRW	-	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW
	-	-	iRW	-	-	-	iRW	-	-	-	-	-	iRW	-	iRW	-	iRW	-	-	-	-	-	iRW
backgroundColor	cRW	cRW	cRW	cRW	cRW	-	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW	cRW
	-	-	cRW	-	-	-	cRW	-	-	-	-	-	cRW	-	cRW	-	cRW	-	-	-	-	-	cRW
backgroundColorIndex	iRW	iRW	iRW	iRW	iRW	-	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW	iRW
	-	-	iRW	-	-	-	iRW	-	-	-	-	-	iRW	-	iRW	-	iRW	-	-	-	-	-	iRW
backgroundImage	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
backgroundRepeat	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
background	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
fontFamily	RW	RW	RW	RW	RW	-	RW	RW	RW	RW	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	-	-
	-	-	RW	-	-	-	RW	-	-	-	-	-	RW	-	RW	-	RW	-	-	-	-	-	-

Upper Row: via the normalStyle attribute Lower Row: via the focusStyle or activeStyle attribute	body	h1, h2, h3, h4, h5, h6, address, blockquote, pre	div, p	em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, b, big, i, small, sub, sup, tt	q	br	span, a	ol, ul, dl	dt, dd	li	hr	form	input	label	select	option	textarea	caption	table	tr	th, td	img	object
fontStyle	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
fontWeight	RW	RW	RW	RW	RW	-	RW	RW	RW	RW	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	-	-
	-	-	RW	-	-	-	RW	-	-	-	-	-	RW	-	RW	-	RW	-	-	-	-	-	-
fontSize	RW	RW	RW	RW	RW	-	RW	RW	RW	RW	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	-	-
	-	-	RW	-	-	-	RW	-	-	-	-	-	RW	-	RW	-	RW	-	-	-	-	-	-
font	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
textAlign	R	R	R	-	-	-	-	R	R	R	-	R	R	-	-	-	-	R	R	R	R	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
textDecoration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
letterSpacing	R	R	R	R	R	-	R	R	R	R	-	R	R	R	R	R	R	R	R	R	R	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
whiteSpace	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
captionSide	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
tableLayout	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderCollapse	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
borderSpacing	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
resolution		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
displayAspectRatio		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
navIndex	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
navUp	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
navRight	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
navDown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
navLeft	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
usedKeyList	RW	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

A5-4.7.2 Operational scope for built-in objects

Table 4-21 shows operational guidelines on ECMAScript built-in objects.

Table 4-21 Operational guidelines on ECMAScript built-in objects

Built-in object	Method/Property	Operation	Remarks
(global)			
	NaN	O	
	Infinity	-	(Note 1)
	eval(x)	-	
	parseInt(string, radix)	O	(Note 3)
	parseFloat(string)	-	(Note 1)

Built-in object	Method/Property	Operation	Remarks
	escape(string)	-	(Note 2)
	unescape(string)	-	(Note 2)
	isNaN(number)	O	
	isFinite(number)	-	(Note 1)
Object		All operated	
	prototype	O	
	Object([value])	O	
	new Object([value])	O	
Object.prototype		All operated	
	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 operated	
	constructor	O	
	toString()	O	
Array		All operated	
	prototype	O	
	length	O	
	Array(item0, item1, . . .)	O	
	new Array(item0, item1, . . .)	O	
	new Array([len])	O	
Array.prototype		All operated	
	constructor	O	
	toString()	O	
	join([separator])	O	
	reverse()	O	
	sort([comparefn])	O	
String		All operated	
	prototype	O	
	length	O	
	String([value])	O	
	new String([value])	O	
	String.fromCharCode(char0[, char1, . . .])	O	
String.prototype		All operated	
	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 operated	

Built-in object	Method/Property	Operation	Remarks
	prototype	O	
	Boolean([value])	O	
	new Boolean([value])	O	
Boolean.prototype		All operated	
	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 operated	
	constructor	O	
	toString([radix])	O	
	valueOf()	O	
Math		None operated	
	E	-	
	LN10	-	
	LN2	-	
	LOG2E	-	
	LOG10E	-	
	PI	-	
	SQRT1_2	-	
	SQRT2	-	
	abs(x)	-	
	acos(x)	-	
	asin(x)	-	
	atan(x)	-	
	atan2(y, x)	-	
	cos(x)	-	
	exp(x)	-	
	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 [,	O	

Built-in object	Method/Property	Operation	Remarks
	seconds [, ms]]]]]]])		
	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	
	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)
	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	
	toUTCString()	O	
	toGMTString()	-	(Note 5)

Note 1: This is not operated, because it is related to Float.
Note 2: This is not operated, because it is related to Unicode.
Note 3: The value of radix for parseInt() must be "8, 10, or 16". A value "0" is interpreted as "10".
Note 4: This is not operated, because it is related to Number.

Note 5: This is not operated, because it is specified to maintain backward compatibility with older source codes.
Note 6: This is operated with restricted specification, because it is related to Number.

A5-4.7.3 Operational scope for extended objects for broadcasting

Table 4-22 shows operational guidelines on extended objects for broadcasting with respect to ECMAScript. Any defined extended object for broadcasting that is not listed in Table 4-22 is assumed that it is not operated, represented by the Operation symbol "-".

Table 4-22 Operation of Extended Objects for Broadcasting

Object	Method/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	
XMLDoc.prototype			
	constructor	O	(Note)
	close()	O	(Note)
	create()	O	(Note)
	read ()	O	(Note)
	getDocument()	O	(Note)
	write()	O	(Note)

Note: Verify if this function is supported in a receiver when the function is needed.

A5-4.7.4 Operational scope for browser pseudo objects

A5-4.7.4.1 Script Description

Any document must not contain more than one Global Object.

The applicable operations of browser pseudo objects regarding to script descriptions are limited to describing procedures in a CDATA section node that is the sole child element of the script element and reading an external script file via the src attribute.

The script element may be contained only in a header object. Two ore more script elements may exist. However, any document may not more than one script element that directly describes procedures.

Refer to Chapter A5-3 for what the O and - symbols mean.

A5-4.7.4.2 Operational scope for browser pseudo objects

The following table provides operational guidelines on browser pseudo objects.

Table 4-23 Operational Scope for Browser Pseudo Objects

	Function	Operation	Remarks
Ureg functions			
	Ureg[]	O	
	Greg[]	O	
EPG functions			
	epgGetEventStartTime()	O	
	epgGetEventDuration()	O	
	epgTune()	O	
	epgTuneToComponent()	O	
	epgTuneToDocument()	-	
	epgIsReserved()	O	
	epgReserve()	O	
	epgCancelReservation()	O	
	epgRecIsReserved()	O	
	epgRecReserve()	O	
	epgRecCancelReservation()	O	
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 functions			
	readPersistentString()	-	
	readPersistentNumber()	-	
	readPersistentArray()	O	
	writePersistentString()	-	
	writePersistentNumber()	-	
	writePersistentArray()	O	
	copyPersistent()	-	
	getPersistentInfoList()	-	
	deletePersistent()	-	

	Function	Operation	Remarks
	getFreeSpace()	-	
Extended APIs for storing			
Extended APIs for storing - Directory management functions			
	saveDirAs()	-	
	saveDir()	-	
	createDir()	-	
	getParentDirName()	-	
	getDirNames()	-	
	isDirExisting()	O	(Note)
Extended APIs for storing - File management functions			
	saveFileAs()	-	
	saveFile()	-	
	getFileNames()	O	(Note)
	isFileExisting()	O	(Note)
Extended APIs for storing- File input/output functions			
	writeArray()	-	
	readArray()	-	
Extended APIs for storing - Inquiry functions			
	getDirInfo()	O	(Note)
	getFileInfo()	O	(Note)
	getContentSource()	O	(Note)
	getStorageInfo()	O	(Note)
	getCarouselInfo()	O	(Note)
	getModuleInfo()	O	(Note)
Extended APIs for storing - Data carousel storage functions			
	saveCarouselAs()	-	
	saveCarousel()	-	
	saveModuleAs()	-	
	saveModule()	O	(Note)
	saveResourceAs()	-	
	saveResource()	O	(Note)
Interaction channel functions			
Interaction channel functions - communication functions assuming BASIC procedure			
	connect()	-	
	disconnect()	-	
	sendBinaryData()	-	
	receiveBinaryData()	-	
	sendTextData()	-	
	receiveTextData()	-	
Interaction channel functions - Delayed call functions			
	registerTransmission()	-	
	registerTransmissionStatus()	-	
	getTransmissionStatus()	-	
	setDelayedTransmissionDataOverBasic()	-	
Interaction channel function - Communication function using the Mass call reception service			
	vote()	-	
Interaction channel functions - Functions for encrypted communication using CAS			
	startCASEncryption()	-	
	endCASEncryption()	-	

	Function	Operation	Remarks
	transmitWithCASEncryption()	-	
Interaction channel functions - Functions for communication with public key encryption not using CAS			
	setEncryptionKey()	-	
	beginEncryption()	-	
	endEncryption()	-	
Interaction channel functions - Communication functions assuming TCP/IP			
	setISPParams()	-	
	getISPParams()	-	
	connectPPP()	O	(Note)
	connectPPPWithISPParams()	-	
	disconnectPPP()	O	(Note)
	getConnectionType()	-	
	isIPConnected()	-	
	saveHttpServerFileAs()	-	
	saveHttpServerFile()	-	
	sendHttpServerFileAs()	-	
	saveFtpServerFileAs()	-	
	saveFtpServerFile()	-	
	sendFtpServerFileAs()	-	
	sendTextMail()	-	
	trasmitTextDataOverIP()	O	(Note)
	setDelayedTransmissionData()	-	
	getDelayedTransmissionStatus()	-	
	getDelayedTransmissionResult()	-	
Operational control functions			
	reloadActiveDocument()	O	
	getNPT()	O	
	getProgramRelativeTime()	O	
	isBeingBroadcast()	O	
	lockExecution()	-	
	unlockExecution()	-	
	lockModuleOnMemory()	O	
	unlockModuleOnMemory()	O	
	setCachePriority()	-	
	getLinkSourceServiceStr()	-	
	getLinkSourceEventStr()	-	
	getIRDID()	O	
	getProgramID()	O	
	getActiveDocument()	O	
	lockScreen()	O	
	unlockScreen()	O	
	getBrowserSupport()	O	
	launchDocument()	O	
	launchDocumentRestricted()	O	(Note)
	quitDocument()	O	(Note)
	launchExApp()	O	(Note)
	getFreeContentsMemory()	-	
	isSupportedMedia()	-	
	detectComponent()	O	(Note)
	lockModuleOnMemoryEx()	O	(Note)

	Function	Operation	Remarks
	unlockModuleOnMemoryEx()	O	(Note)
	unlockAllModulesOnMemory ()	O	(Note)
	getLockedModuleInfo()	O	(Note)
Receiver sound control function			
	playRomSound()	O	
Timer functions			
	sleep()	O	
	setTimeout()	-	
	setInterval()	O	
	clearTimer()	O	
	pauseTimer()	-	
	resumeTimer()	-	
	setCurrentDateMode()	-	
External character functions			
	loadDRCS()	O	
	unloadDRCS()	-	
Functions for controlling external devices			
	enumPeripherals()	O	(Note)
	passXMLDocToPeripheral()	O	(Note)
	getArrayFromPeripheral()	O	(Note)
Other functions			
	random()	O	
	subDate()	O	
	addDate()	O	
	formatNumber()	O	
Subtitle presentation control functions			
	setCCStreamReference()	-	
	getCCStreamReference()	-	
	setCCDisplayStatus()	-	
	getCCDisplayStatus()	-	
	getCCLanguageStatus()	-	

Note: Verify if this function is supported in a receiver before the function is invoked by a document.

A5-4.7.5 Operation of Navigator pseudo objects

Navigator pseudo objects are operated. Refer to Chapter A5-3 for what the O and - symbols mean.

A5-4.7.5.1 Operational Scope for Navigator pseudo objects

The following table summarizes operational guidelines on Navigator pseudo objects.

Table 4-24 Operational Scope for Navigator Pseudo Objects

Function	Operation	Remarks
appName	O	
appVersion	O	

A5-4.7.5.2 Operational guidelines on Navigator pseudo objects

As a Navigator pseudo object, navigator.appName and navigator.appVersion are allowed to be operated. When a BML-compatible browser receives navigator.appName or navigator.appVersion, the following value is returned.

- The navigator.appName pseudo object returns a string containing the ”BMLBrowser” string.
- What navigator.appVersion pseudo object returns is implementation dependent.

A5-4.8 Restrictions on Description of BML Documents and Operational Guidelines on Presentation Control

The restrictions on description of BML documents are described below.

- The style element, the bevent element and the script element must appear only once in the head element part. The script element and the style element must be in a single CDATA section, which must be enclosed with “<![CDATA” and “]]>“.
- No arguments are written in the functions used for event handlers.
  That is, the sentence for setting an event handler is as follows:
  attr = “func( );”
  Note that attr is an attribute name of the interrupt event and func is a function name.
- Automatic semicolon insertion is not mandatory for the operation.

A5-4.8.1 Positioning and restrictions for index-based model

In this section, a guideline concerning designation of position for the elements located in the body element is specified.

The restrictions described below cover only the elements that occupy two-dimensional space by placing them in the body element. This does not mean that including other elements in the body element is inhibited.

In the sentences of restrictions, “the size and position of a box is specified by absolute positioning” is referred to simply as “a position is specified.” The positioning method uses relative coordinates from a parent element (equivalent to containing block of CSS2) as follows:

<Element name style= ’

left: Distance from the left of box of parent element (number of pixels);
top: Distance from top of box of parent element (number of pixels);
width: Width (number of pixels);
height: Height (number of pixels);
visibility: “visible”, “hidden” or “inherit”;’ ...>

The position attribute must be fixed to absolute. Therefore, it is not specified in the style part.

In addition, the following terms are used in the sentences below:

boxed element: The followings are named generically ‘boxed elements’:

a) elements having two-dimensional space (input)
b) elements having intrinsic dimension (object)
c) ‘div’ element
d) ‘p’ element

normal flow element: The elements that can be used in operation other than ‘boxed elements’ are named ‘normal flow elements’.

Restriction 1: A position of boxed elements must be always specified.

Restriction 2: A position of normal flow elements must not be specified.

Restriction 3: The child element of body elements must be either div or p element.

Restriction 4: The div element can include only boxed elements. Nesting div element(s) allows nest of position designation.

Restriction 5: The p element is allowed to have one of normal flow elements or a text line as its child element.

Example:

```
<p style="...box designation...">
    abcde<span style="...">fghij</span>
</p>
```

In CSS2, absolute positioning or normal flow can be specified for each element. However, the basic concept of the aforementioned rule is to completely separate the line of position-specified elements from the line of normal flow elements.

A5-4.8.2 Vertical writing

No vertical writing is operated.

A5-4.8.3 Restrictions on arrangement of elements

This section defines the following restrictions on arrangement of elements.

- For a component-based model, no element is allowed to have translucent pixels.
- The object element that presents a video plane or a still picture plane should be the first element specified in a concerned document.
- No video plane is allowed to be overlapped with any other presentation object.
- The background colour (background-colour) of the div element that holds the object element associated with MNG graphics as a child element must be a non-drawing colour.
- Only the div element that holds the object element associated with MNG graphics as a child element, a video plane, and a still picture plane are allowed to be presented in the back of an MNG graphic object, except that any element that has non-drawing colour pixels that are to be overlapped with the MNG graphic object.
- No element is allowed to be presented in front of any MNG graphic plane.
- No operation resulting in a change in presentation of the element that is positioned to be overlapped with the MNG object whose streamstatus attribute is specified as play is allowed.
- The background colour of the MNG object whose streamstatus attribute is specified as play must not be changed.

A5-4.8.4 Operation of object element and img element

How an individual monomedia that is specified with the object element or the img element is presented is governed by the convention in Section A5-5.1.5. How the input element whose type attribute is set to image is presented is also governed by the convention in Section A5-5.1.5.

The available monomedia type to the src attribute for both the img element and the input element is limited to image/jpeg, image/png, and image/X-arib-png. No operation that modifies a type of a monomedia, that is specified with the src attribute of the img element and the src attribute of the input element, is allowed.

A5-4.8.4.1 Application of remain attribute for each type

The following Table 4-25 summarizes guidelines for applying the remain attribute of the object element.

Table 4.25 Application of the remain Attribute

type attribute	Applicability of remain attribute	Remarks
video/X-arib-mpeg4-simple	O	Applicable only when a broadcasting service is referenced
video/X-arib-H264-baseline	O	Applicable only when a broadcasting service is referenced
video/X-arib-H264-main	O	Applicable only when a broadcasting service is referenced
audio/X-arib-mpeg2-aac	Δ	Applicable only when the scheme specification is arib:(PES)
audio/X-arib-aiff	-	
image/jpeg	O	
image/png	-	
image/X-arib-png	-	
image/X-arib-mng	-	
application/X-arib-mpeg2-tts	-	

The following conditions are assumed to operate the remain attribute of the object element.

[Condition 1]

The remain attribute of the object element can be made valid only in the case where the object element is used by fulfilling either of the conditions of (1) to (3) listed below.

(1) The value of the type attribute is "video/X-arib-mpeg4-simple", "video/X-arib-H264-baseline", or "video/X-arib-H264-main" and belongs to the same content group as the object element.

(2) The value of the type attribute is "image/jpeg". In addition, the values of the width property and the height property are equal to the horizontal and vertical resolution specified by the resolution property of the body element, which is an ancestor element, respectively

(3) The scheme specified by the data attribute is "arib", the data attribute belongs to the same contents group as the object element, and the value of the type attribute is "audio/X-arib-mpeg2-aac".

Note that, for the purpose of Condition 1, to make the remain attribute valid is assumed as one of the following.

a) To make a BML document specify the value "remain" for the remain attribute.

b) To set the remain attribute of the BMLObjectElement interface of DOM API to true.

c) To let the remain attribute of the object element in a BML document before a transition be applicable to the object element in the BML document after the transition.

When the remain attribute of the object element is made valid based on any other assumption, how the remain attribute is operated depends on an implementation.

[Condition 2]

The object element whose remain attribute is made valid must be a child element of the div element which is the first child element of the body element.

When the remain attribute of the object element is made valid without meeting this condition, the behaviour depends on an implementation.

[Condition 3]

When the div element has at least one object element whose remain attribute is made valid as a child element, the div element must have not more than one child element meeting the (4) condition described below, not more than one child element meeting the (5) condition described below, and not more than one child element meeting the (4) condition described below. Otherwise, the behaviour depends on an implementation.

(4) Not more than one object element meeting the (1) condition described above.

(5) Not more than one object element meeting the (2) condition described above.

(6) Note more than one object element meeting the (3) condition described above.

[Condition 4]

To make a document transition occur in the case where the remain attribute of the object element satisfying the above condition (2) is made valid, the lockModuleOnMemory() or lockModuleOnMemoryEx() method of a browser pseudo object must have been applied to the resource specified with the data attribute of the object element. Otherwise, the behaviour depends on an implementation.

[Condition 5]

Even if the object element satisfies the condition (2), its remain attribute must not be made valid when the background-image property of the body element which is an ancestor element of the object element has been specified.

Otherwise, the behaviour depends on an implementation.

[Condition 6]

The remain attribute for a still picture that belongs to two or more document groups must not be made valid.

Otherwise, the behaviour depends on an implementation.

A5-4.8.4.2 Operation of attributes responsible for stream presentation

The following Table 4-26 summarizes operational guidelines on the streamposition attribute, the streamstatus attribute, and the streamlooping attribute of an object element.

Table 4-26 Operation of Attributes Responsible for Stream Presentation

type attribute	streamposition	streamstatus	streamlooping
video/X-arib-mpeg4-simple	Fixed to 0	play (Note 1)	Fixed to 1
video/X-arib-H264-baseline	Fixed to 0	play (Note 1)	Fixed to 1
video/X-arib-H264-main	Fixed to 0	play (Note 1)	Fixed to 1

type attribute	streamposition	streamstatus	streamlooping
audio/X-arib-mpeg2-aac	Fixed to 0	play, stop (Note 2)	Fixed to 1
image/jpeg	Fixed to 0	play (Note 1)	Fixed to 1
image/png	Fixed to 0	play (Note 1)	Fixed to 1
image/X-arib-png	Fixed to 0	play (Note 1)	Fixed to 1
audio/X-arib-aiff	Fixed to 0	play,stop (Note 2, 3)	Fixed to 1
image/X-arib-mng	Indidates frame number	play,stop,pause (Note 2)	Fixed to 1
application/X-arib-mpeg2-tts	Fixed to 0	play,stop,pause (Note 4, 5)	Fixed to 1

Note 1: The initial value of streamstatus for an object element referencing media of type attribute which takes 'play' only is 'play'.

Note 2: The initial value of streamstatus for an object element referencing media of type attribute which can take 'stop' is 'stop'. In this case, actually the corresponding sound is not stopped but muted. Note that dynamically changing type attributes and dynamically changing schemas by changing data attributes for sound are not applicable to the object element.

Note 3: When the playback ends, streamstatus is automatically set to "stop".

Note 4: When the playback ends, streamstatus is automatically set to "pause".

Note 5: If streamstatus for an object element is pause when the BML document is read and streamstatus for the referenced video object is play, a still picture in a frame near the beginning of the stream is presented. Once streamstatus for the object element has been reconfigured to play, the playback starts at the position immediately following the still picture or begins with the very beginning of the stream.

A5-4.8.4.3 Guidelines on MNG presentations

This section contains guidelines on how the attribute is interpreted and presented when MNG (the value of the type attribute is "image/X-arib-mng") is specified in the object element.

Table 4-27 shows how MNG behaves depending on the initial values of the streamstatus attribute and the streamposition attribute.

Table 4-27 Application of the initial values of the streamstatus attribute and the streamposition attribute

Initial value of the streamstatus attribute	Behaviour
play	It plays back from the frame specified by the initial value the streamposition attribute. Playback is repeated in accordance with the repetition number of times calculated with (IterationMax of TERM chunk) x streamlooping. However, it is interpreted with the infinity time repetition at the time of IterationMax = 0x7fffffff of TERM chunk.
stop	A frame is not displayed. Even if the initial value of the streamposition attribute is set to any other than 0, it will be reset in 0.
pause	The frame specified in the initial value of the streamposition attribute is displayed. As for the repetition, it is interpreted with the first round.

Table 4-28 shows how MNG behaves depending on the value of the visibility property.

Table 4-28 MNG Presentation Depending on the visibility Property Value

Value of the streamstatus attribute	visibility="visible"	visibility="hidden"
play	Normal play back	Not displayed. It depends on receivers whether to proceed play back process (renewal of a frame to be displayed and the update of the repetition number of times) without presentation. And, it is indefinite from which frame the play back starts when the value of the visibility property is set "visible" next time.
stop	Not displayed	Not displayed
pause	The frame specified in the streamposition attribute is displayed.	Not displayed

Table 4-29 outlines how the streampsition attribute is accessed via DOM API.

Table 4-29 Access to streamposition Attribute

Value of the streamstatus attribute	Read the streamposition attribute	Write the streamposition attribute
play	Not used in the operation	Not used in the operation
stop	Value is fixed to 0	Even if the value except for 0 is set, it is ignored, and the value is fixed to 0.
pause	The number which corresponds to the frame that is set for display.	A frame corresponding to the specified value is displayed. When a bigger value than the number of the frames of MNG is specified, the presentation depends on receivers.

Table 4-30 shows how MNG is presented when the streamstatus attribute is modified using DOM API.

Table 4-30 Access to streamstatus Attribute

The former value of the streamstatus	Set for play	Set for stop	Set for pause
play	-	- Play back is stopped and the frame is not displayed. - The streamposition attribute is reset in 0. - The repetition number of times is reset in 0 and it is played back from the first round when the streamstatus attribute is set for play next time.	Play back is suspended, and one frame is displayed. It depends on receivers which frame is displayed. The streamposition attribute is set for the value which corresponds to the displayed frame. The repetition number of times is reset in 0 and it is played back from the first round when the streamstatus attribute is set for play next time.
stop	Play back is started from the frame which corresponds to the condition 'streamposition attribute =0'.	-	Play back is started from the frame which corresponds to the condition 'streamposition attribute =0'.
pause	- Play back is started from the frame set up in the streamposition attribute.	A frame is not displayed. The streamposition attribute is reset in 0. The repetition number of times is reset in 0 and it is played back from the first round when the streamstatus attribute is set for play next time.	-

- The data attribute can be modified only when the streamstatus attribute is stop.
- The streamstatus attribute is set to pause automatically when the play back has been repeated as many times as specified.
  It depends on receivers whether a renewal of the presentation frame of MNG is suspended or is continued by the execution of lockScreen() in the period when the renewal of the screen is prohibited.

Chapter APPENDIX 5-5 Guidelines on Behaviours of Browsers

A5-5.1 Operational Guidelines on Presentation

A5-5.1.1 Presentation model

This section defines two presentation models for a mobile receiving system, an index-based model for a mobile receiving system and a component-based model for a mobile receiving system. Each model must be operated based on the specifications for the three-plane-based configuration (a video plane, a still picture plane, a text and graphics plane) defined in Chapter 7, Part 1, Volume 1, except the definitions for resolution.

Unless otherwise noted, "an index-based model" refers to "an index-based model for a mobile receiving system" and "a component-based model" means "a component-based model for a mobile receiving system" in this Appendix.

- Index-based model for a mobile receiving system

  This presentation model directly specifies the CLUT data of a text and graphics plane. In an index-based model, the whole area of a screen is used to present a content and no scrolling occurs.

- Component-based model for a mobile receiving system

  This presentation model does not directly specifies the CLUT data of a text and graphics plane. In a component-based model, only part of a screen is used to present a video content and an area for presenting informative content is scrolled to make the whole informative content available to an end user.

Background colours, characters, and borders of each element other than the body element behave as if they are rendered in a text and graphics plane, and take colours that specified with colour names or RGB values. A browser automatically allocates the specified colours to CLUT of the text and graphics plane. When PNG, MNG, or JPEG is to be presented, it behaves as if it is rendered in a still picture plane. However, no translucent colour is applicable.

A5-5.1.2 Coordinate system for text and graphics plane

The default coordinate system for a text and graphics plane is the 320 x240 coordinate system. Operational guidelines are summarized in Table 5-1. For operational guidelines on using other coordinate system than the 320 x240 coordinate system, refer to Section 5.9.1.

Table 5-1 Restrictions on Text and Graphics Plane

	Presentation	Index-based	Component-based
1	Screen Resolution	320 x240	320 x240
2	Effective Presentation Area	Entire screen is used for presentation.	Size of displayed area and how surrounding area is processed depend on an implementation.

A5-5.1.3 Operational guidelines on Z-index

Objects are presented in the appearance order. In other words, the elements that appeared first in a BML document are presented at the inner part (position far from the viewer) of the screen. The relationship between the DOM object structure and presentation is illustrated below in Figure 5-1. DOM objects generate a tree structure in the order of element presentation. Objects 1 to 9 are presented in the order of 1, 2, 4, 5, 7, 8, 9, 3 and 6 if having the structure shown below.

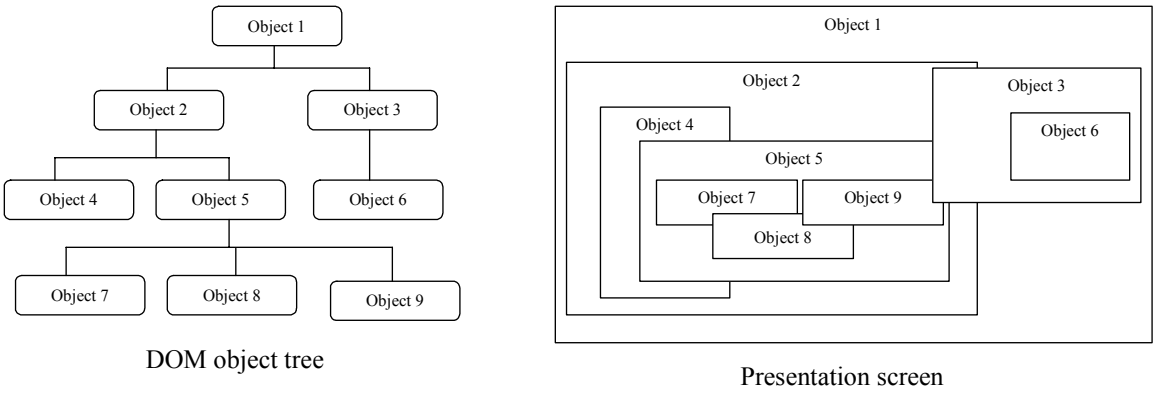


Figure 5-1 DOM Object Structure and Presentation

A5-5.1.4 Presentation plane

Table 5-2 specifies how presentation planes behave on the assumption of a model for drawing planes.

Table 5-2 Operational Restrictions on Presentation Planes

	Index-based	Component-based
To switch from a video plane to a still picture plane	When a focus is taken off from a part or the whole of a rectangle video presentation area in a currently selected video plane (to a still picture plane to select it), how the resulting presentation behaves depends on an implementation.	When a focus is taken off from a part or the whole of a rectangle video presentation area in a currently selected video plane (to a still picture plane to select it), how the resulting presentation behaves depends on an implementation.
To draw into a text and graphics plane	This appendix contains no restrictions.	When a video plane and a text and graphics plane overlaps, how the resulting presentation of the overlapped area behaves depends on an implementation.
When the invisible attribute of the body element is made valid	The whole video plane is used to play video and no text and graphics plane nor still picture plane are presented.	The whole video plane is used to present video data and no text and graphics plane nor still picture plane are presented.

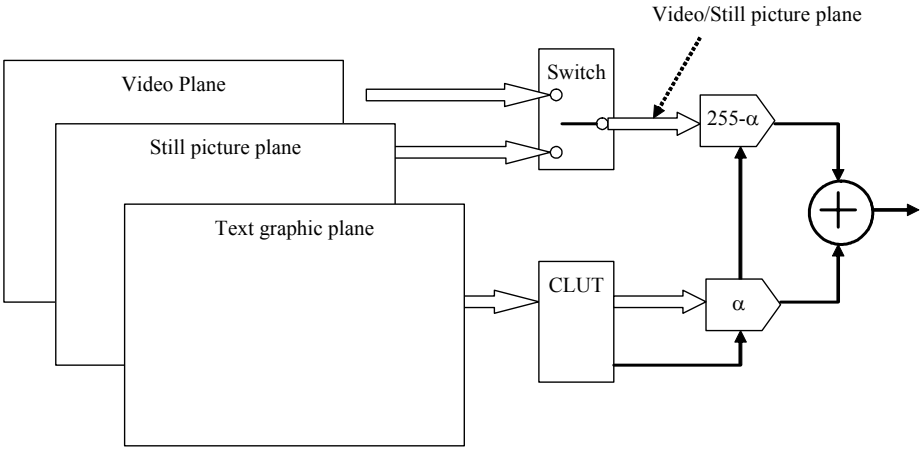


Figure 5-2 Model for drawing planes

A5-5.1.5 Operation of monomedia presentation

Table 5-3 specifies how monomedia presentation is operated for each presentation model

Table 5-3 Restrictions on Operated Monomedia

	Operation of monomedia	Index-based	Component-based
1	Presentation of PNG	CLUT data specified with the clut property of the body element is used. A text and graphics plane is used for the presentation. The presentation size is defined based on the coordinate system of the text and graphics plane.	PLTEchunk associated with PNG data is used. The presentation size is defined based on the coordinate system of the text and graphics plane.
2	Presentation of MNG	CLUT data specified with the clut property of the body element is used. The presentation behaves as if it is rendered in a text and graphics plane. The presentation size is defined based on the coordinate system of the text and graphics plane. PLTEchunk associated with MNG data is not used.	PLTEchunk associated with MNG data is used. The presentation size is defined based on the coordinate system of the text and graphics plane.
3	Presentation of JPEG	The presentation size is defined based on the coordinate system of the text and graphics plane.	The presentation size is defined based on the coordinate system of the text and graphics plane.
4	Presentation of Video	QVGA is a default resolution. Other resolutions: SIF, CIF, QCIF may be used. Details for apect ratio (4:3, 16:9) and frame rate are defined in an operational standard regulation.	QVGA is a default resolution. Other resolutions: SIF, CIF, QCIF may be used. Details for apect ratio (4:3, 16:9) and frame rate are defined in an operational standard regulation.

Sample behaviours of a text and graphics plane in which a monomedia is presented are provided below.

Table 5-4 Sample Behaviours of Text and Graphics Plane

	Sample Behaviours of Presentation	Index-based	Component-based
1	Sample Presentation of JPEG	An image object is presented in a still picture plane while a solid rectangle in the colour (R,G,B,α)=(0,0,0,0) with the same size as the image object is rendered at the same position as the image object in a text and graphics plane.	An image object is presented in a still picture plane while a solid rectangle in the colour (R,G,B,α)=(0,0,0,0) with the same size as the image object is rendered at the same position as the image object in a text and graphics plane.

	Sample Behaviours of Presentation	Index-based	Component-based
2	Sample Presentation of PNG	An image object is rendered in a text and graphics plane.	An image object excluding the non-drawing colours is presented in a still picture plane while the corresponding graphic excluding the pixels associated with non-drawing colours is also rendered in the colour (R,G,B, α)=(0,0,0,0) at the same position of the image object in a text and graphics plane.
3	Sample Presentation of Video	A solid rectangle in the colour (R,G,B, α)=(0,0,0,0) with the same size as a video object is rendered at the same position as the video object in a text and graphics plane.	A solid rectangle in the colour (R,G,B, α)=(0,0,0,0) with the same size as a video object is rendered at the same position as the video object in a text and graphics plane.

A5-5.1.5.1 Positioning by an object element

The following shows an operation example of top, left, width, and height , which are CSS2 properties applied to an object element.

When a video, a still picture or graphic form is presented using an object element, a scaling operation shown below is performed. In the diagram, W indicates the number of horizontal pixels of an image, H indicates the number of vertical pixels of an image. Also, x and y indicate horizontal and vertical coordinates on the presentation screen, respectively. Sx and Sy indicate a horizontal scaling ratio and a vertical scaling ratio, respectively. No value that places a video outside of a corresponding video plane must be specified. Also, no value that places a still picture outside of a corresponding still picture plane must be specified.

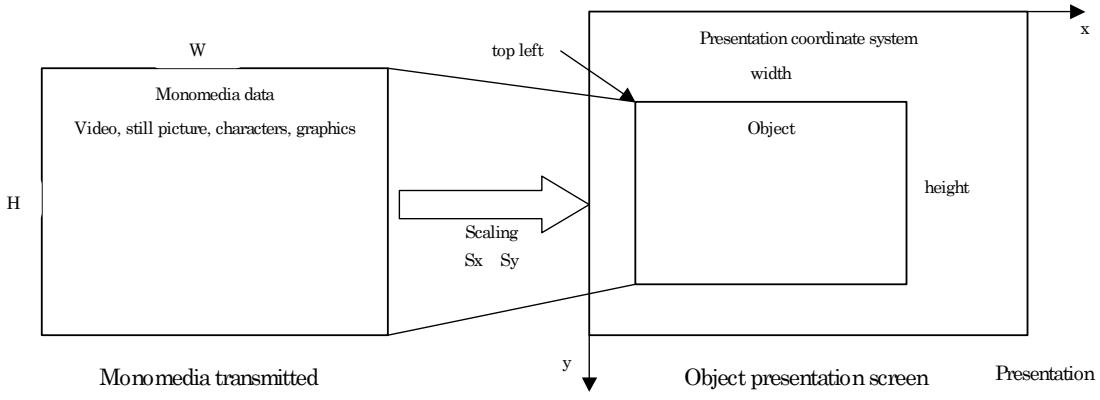


Figure 5-3 Relation between Presented Coordinates and Images

- Combinations of Planes

  The presentation coordinate systems specified by an object element must be the 320 x 240 coordinate system (QVGA). The combinations of plane resolutions for video, still picture and text and graphics are listed below in Table 5-5.

Table 5-5 Combination of Plane Resolutions

Presentation coordinate system	Text and graphics plane resolution	Still Picture plane resolution	Video plane resolution
320 x 240 coordinate system Horizontal direction(0-319) Vertical direction (0-239)	 Horizontal 320 pixels Vertical 240 pixels	 Horizontal 320 pixels Vertical 240 pixels	 Horizontal 320 pixels Vertical 240 pixels

- width and height properties

  Scaling operation is restricted depending on a media type or a scaling ratio. The expressions for calculating width and height properties are listed in Table 5-6, provided that the decimals are rounded down. If the values of the width property and the height property which do not satisfy the relationship in the table are specified, the representation on the display depends on an implementation. And, if the scaling ratio of the object element is dynamically changed, the width property and the height property must be changed when the update of the representation is locked by the lockscreen() function.

Table 5-6 Relationship between width/height Properties and Scaling Ratio

type attribute	320 x 240 coordinate system	
	width property	height property
video/X-arib-mpeg4-simple	W*Sx	H*Sy
video/X-arib-H264-baseline	W*Sx	H*Sy
video/X-arib-H264-main	W*Sx	H*Sy
image/jpeg	W	H
image/png	W	H
image/X-arib-png	W	H
image/X-arib-mng	W	H

- Scaling ratio

  The following values are applicable to Sx and Sy, where Sx=Sy.

Table 5-7 Values Applicable to Sx and Sy

Sx	256/128	128/128	96/128	64/128
Sy	256/128	128/128	96/128	64/128

- Number of horizontal and vertical pixels

  The following values are applicable to W and H.

Table 5-8 Values Applicable to W and H

	320 x 240 coordinate system	
type attribute	W	H
video/X-arib-mpeg4-simple	320	240
video/X-arib-H264-baseline	320	240
video/X-arib-H264-main	320	240
image/jpeg	1 ~ 320	1 ~ 240
image/png	1 ~ 320	1 ~ 240
image/X-arib-png	1 ~ 320	1 ~ 240
image/X-arib-mng	1 ~ 320	1 ~ 240

A5-5.1.5.2 Operation of video presentation

Video pictures are mapped once when performing the above presentation because those having the number of pixels other than the above W and H must be allowed for presentation. A video with W' having the number of horizontal pixels and H' having the number of vertical pixels is mapped once and converted into that of an intermediate format with W having the number of horizontal pixels and H having the number of vertical pixels. The converted video is scaled as mentioned above. The mapping and scaling processes are shown in Figure 5-4.

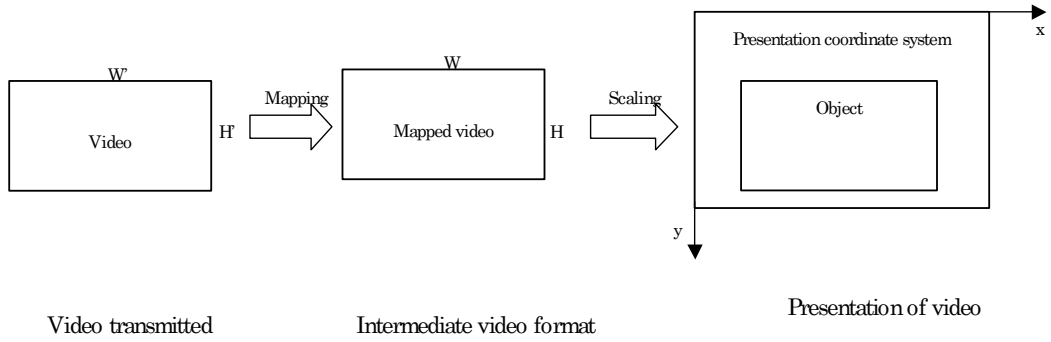


Figure 5-4 Mapping of Video

Mapping from W to W' and from H to H' must be provided separately as an operational standard regulation defined by a broadcaster.

A5-5.1.6 Guidelines on clipping

This guideline allows a child element to be clipped by a parent element in order to fix the overflow property to hidden. As shown left in Figure 5-5, the content is clipped if sticking out of the rectangular area of the specified element (object 1). Moreover, as shown right below, if the child element (object 3) is clipped by the parent object (object 2), any areas other than the rectangular area specified by the parent element are hidden.

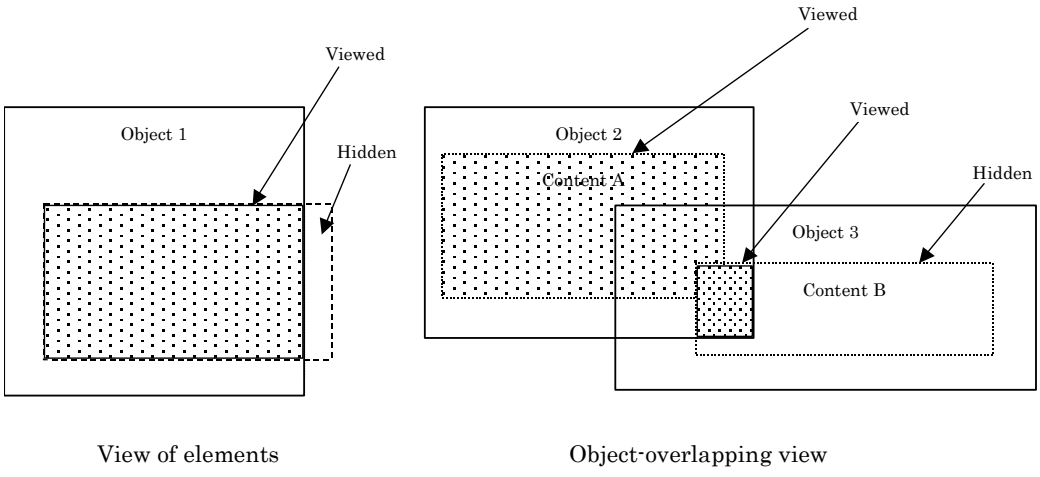


Figure 5-5 Example of Clipping

The above clipping operation is not performed by the parent div element when a video, still picture or MNG image is specified by the object element. In other words, monomedia image specified by object element must not be specified out of area with parent div element. This rule is also applicable to clipping that involves two or more generations. No description must instruct an ancestor element to perform a clipping.

A5-5.1.7 Values applicable to CSS2 properties

Table 5-9 shows the values applicable to a CSS2 property holding the value "<length>".

Table 5-9 Values Applicable to CSS2 Properties

Property	Values applicable to the property
top property	-240px - 240px
left property	-320px - 320px
width property	0 - 320px
height property	0 - 240px
border-width property	0 - 100px
padding-top, bottom property	0 - 240px
padding-right, left properties	0 - 320px
font-size property	Refer to Table 4-12
line-height property	16px - 128px
letter-spacing property	0 - 320px

A5-5.1.8 Operation of colour map data

This section provides operational restrictions in order to make a text and graphics plane available on a hardware device with a 256-colour CLUT in each presentation model.

- Operation of CLUT in an index-based model
  - The default colours of CLUT comply with the default setting defined in APPENDIX 2.
  - Index colours 0 through 16 must be fixed to default CLUT colours.
  - Index colours 224 through 255 depend on an implementation.
  - Index colours 17 through 223 may be configured as required.
  - A colour map data is allowed to be operated in an index-based model as defined in Table 5-10.

Table 5-10 Operation of Colour Map Data

Data	Value	Description
clut_type	0	Among the colour spaces, only YCbCr colour space is allowed.
depth	'01'	Limited to 8-bit CLUT (up to 256 colours)
region_flag	0	No region is specified.
start_end_flag	1	Start/end index and some elements of CLUT are specified.
start_index	17 or 128	Defines a set of 207 colours from index number 17 to 223 or a set of 96 colours from index number 128 to 223.
end_index	223	
alpha	Except 0	αvalue=0 is not allowed

- Operation of CLUT in a component-based model
  - The 17 colours in Table 5-11 are reserved. It is allowed to specify ColorName in a content.
  - Each colours may be allocated index as required provided that only one translucent colour may be allocated to index.

- In a component-based model, a set of available RGBα may comprise up to 207 colours. That is, 256 (the total colours) - 17 (the fixed colours) - 32 (the colours dependent on an implementation) = 207.

Table 5-11 ColorName and RGB value in Component-based Model

color-name	R	G	B	α
white	255	255	255	255
black	0	0	0	255
red	255	0	0	255
lime	0	255	0	255
yellow	255	255	0	255
blue	0	0	255	255
fuchsia	255	0	255	255
aqua	0	255	255	255
transparent	0	0	0	0
grey	128	128	128	255
green	0	128	0	255
maroon	128	0	0	255
navy	0	0	128	255
olive	128	128	0	255
purple	128	0	128	255
silver	192	192	192	255
teal	0	128	128	255

The following table outlines restrictions on CLUT for each presentation model.

Table 5-12 Operational Restrictions on CLUT

Operation of CLUT	Index-based model	Component-based model
Restriction on CLUT for text and graphics plane	The CLUT property of the body element is used to specify index colours 17 through 223. Index Colours 0 through 16 must be fixed values. Especially, Index Colour 8 must be a transparent colour and handled as a non-drawing colour.	This CLUT is internally used by a browser and is not allowed to be directly specified through a content. Colours are specified as RGB values.
Restriction on translucent colours	Any object containing translucent colours is not allowed to overlay any object rendered in a text and graphics plane (The object must not be positioned in Z order to be in front of any other object in a text and graphics plane).	No translucent colour is operated. For PNG, tRNS chunk may be used and index may be handled as a non-drawing colour.

A5-5.1.9 Operation of remote control keys

Table 5-13 shows the relationship among a remote control key, a key code and a characters specified as an access key.

Table 5-13 Relationship among Remote Control Keys, Key Codes and Access Keys

Remote control key	Key code	Access key character
↑	1	N/A
↓	2	N/A
←	3	N/A
→	4	N/A
0,1,2,3,4,5,6,7,8,9	5-14	N/A
“Enter”	18	N/A
“Back”	19	‘X’
“Data” (Note)	20	N/A
Reserved for ARIB data broadcast standard	29-99	reserved
“Bookmark” key	100	N/A
Reserved for future extended features	101-149	reserved
Vendor-dependent	150-	Not defined in this appendix.

Note: When the “Data” button is pressed, an event DataButtonPressed occurs and no keydown nor keyup event occurs.

Each key code in Table 5-13 can be obtained via the keyCode attribute for the BMLIntrinsicEvent interface.

A5-5.1.10 Character entry into input element and textarea element

The minimum operational scope that must be supported by remote control keys is shown in Table 5-14.

Table 5-14 Character Entry into input/textarea Element with Remote Control Keys

Remote control key	Application to input element
0,1,2,3,4,5,6,7,8,9	Used to enter numeric characters 0 through 9. May be used for KANA-KANJI conversion
"Back"	Deletes one character (Backspace)

When in a content where a “BACK” key or a numeric key is specified as an access key, and a focus is placed in the input element or the textarea element, the access key is disabled. In this case, removing the focus from the input element or the textarea element enables the access key.

Alphanumerical characters and Japanese characters can be entered using remote control keys. Details are defined in an operational standard regulation.

A5-5.1.11 Operation of Focus

In each presentation model, a focus is operated as described below.

	Focus	Index-based model	Component-based model
1	To move a focus	The nav-index property is operated.	The nav-index property is not operated. A receiver is responsible for navigation.

Any receiver must be equipped with a function for presenting an element that has a focus. For more information about enabling or disenabling a function of a currently focused element, refer to Section A5-4.4.1.

When the nav-index property is not specified, a receiver must automatically identify an element that needs a focus to perform a proper navigation. An element that satisfies one of the following conditions needs a focus.

(1) the input element, the select element, the textarea element, and the a element with the href attribute

(2) an element that has one of the following attributes: onclick, onfocus, onblur, onkeydown, or onkeyup

A5-5.2 Guidelines on Behaviours of External Characters

A5-5.2.1 Sharing of external characters among documents

How external characters are shared between documents is illustrated in Figure 5-6.

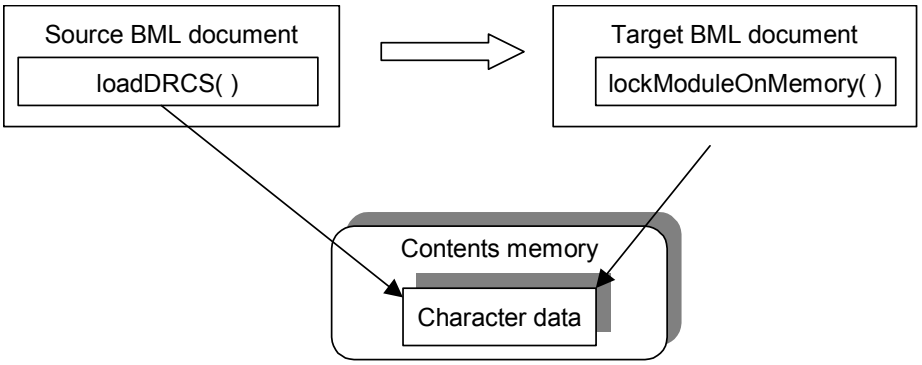


Figure 5-6 Sharing of External Characters between Documents

[1] External character data is locked on content memory by lockModuleOnMemory( ).

[2] External character data is made available by loadDRCS( ).

[3] One document is switched to the other without starting unloadDRCS( ).

[4] The loadDRCS( ) function is restarted from the jump document to make external character data available.

[5] When unloadDRCS( ) is started or when the scope comes out of the document group, external character data is released.

External character data locked by lockModuleOnMemory( ) is valid within the document group. Therefore, the data can be shared only within the document group.

A5-5.2.2 Presentation of external characters

When the width or height value of an external character do not comply with font-size property which is applied to the p, a, span, and input elements, how the external character is presented depends on an implementation.

A5-5.3 Guidelines on Behaviours of DOM

In using the DOM, it is usual to use the X interface after creating a document structure by the create X() method of Document class if some interface X is utilized. However, generating an explicit structure by the createX() method must be omitted in describing a content conforming to this specification. Therefore, if the DOM is used in the received contents, it is preferable for the receiver to develop the DOM tree in memory before using the DOM.

If an element not defined in the DTD is entered as well, its node is retained as an element with unknown contents but attributes must be allowed to be discarded.

The hasFeature( ) method of the DOMImplementation interface returns true when the feature is "BML" and the version is as described in the operation.

A5-5.3.1 Handling of control codes (blank, line break, tab)

If control codes (blank, line break, and tabs) are included in a BML document, DOM is generated with the rule described below.

- Text node is not generated for character strings of control codes.
- If control codes are contained in the character strings of child nodes of p element, span elements, a elements and input element, the rule in table 5-15 is applied.

Table 5-15 Control Codes for Character Strings of Child Node of Element

Display pattern of control code	Handling when generating DOM	Example
The row of control code just behind of start tag.	All ignored	 abc Character strings data of child node of p element is “abc”
The row of control code just before end tag.	All ignored	defg Character strings data of child node of p element is “defg”.
The row of control code between 2-bytes characters	All ignored	日本語 日本語 Character strings data of child node of p element is “日本語日本語”.
The row of control code between 2-bytes character and 1-byte character.	Blank with 1 byte(0x20)	日本語 Japanese Character strings data of child node of p element is “日本語 Japanese”.
The row of control code between 1-byte characters	Blank with 1 byte(0x20)	abc def Character strings data of child node of p element is “abc def”.

- When span element, a element, and br element are contained in p element, the rules in table 5-16 are applied.

Table 5-16 Control Codes in the Case of Other Elements in p Element

Display pattern of control code	Handling when generating DOM	Example
The row of control code just after end tag of br element.	All ignored	 abc def All blanks from to “d” are deleted and the character strings of child node of last p element is “def”

Display pattern of control code	Handling when generating DOM	Example
The row of control code just before start tag of span element, an element and br element.	Blank with 1 byte(0x20)	 abc def the character strings of fist child node of p element is “abc ” because 0x20 is contained in the end of “abc”
The row of control code just after end tag of span element, an element.	Blank with 1 byte(0x20)	abcdef ghi def the character strings of last child node of p element is “ def” because 0x20 is contained in the end of “ def”.

- When the characters of one line exceed the width of a block when representing blanks and line breaks of a CDATA section, the exceeded characters of the line will be fold back at the right end.
- Tab is processed as one-byte character code. That is, the width is same as that of one-byte character codes. Actual representation depends on each implemented receiver.

Note that when there are other elements than the p, span, and a elements that have a string with a control code, the rules in Table 5-16 are applied.

A5-5.3.2 Generation of DOM tree

Whether or not comment nodes are generated depends on an implementation.

- Behaviours of the pre element

  The pre element retains all whitespace characters except leading and trailing whitespace characters, as defined in Section 3.2, “User Agent Conformance”, the W3C Recommendation, “XHTML1.0“ (http://www.w3.org/TR/2000/REC-xhtml1-20000126/ - uaconf).

A5-5.3.3 Restriction on DOM work memory

Table 5-17 defines the maximum number of DOM objects allowed to be generated per item.

Table 5-17 Maximum Value of DOM Objects

Item	Maximum value	Remarks
Number of elements	512	
The total number of nodes	1024	
Size of character string set to NodeValue	1024 (bytes)	
Number of child nodes to be connected to one node	255	
Depth of tree structure	16	The depth of a node immediately under the HTMLElement node is 1. The HTMLElement node is not counted.

A5-5.3.4 Restrictions on style description

The restriction provided in Table 5-18 is applied to description of styles.

Table 5-18 Restrictions on Style Description

Item	Maximum number	Remarks
CSS selectors	512	
CSS declarations	4096	

A5-5.4 Operational Guidelines on ECMAScript

A5-5.4.1 Operation of script working environment

The restriction provided in Table 5-19 is applied to a script working environment.

Table 5-19 Restrictions on Script Work Memory

Item	Maximum value	Remarks
Length of a symbol name character string	255 bytes	
Function arguments	255	
Local variables	255	
Total length of all character strings	65536 bytes	The total length of strings (including evaluated values of string equations, string constants, string variables) and symbol names.
Instances of objects	256	Total number of instances of Object, Number, String, Boolean, Array, Date, BinaryTable and, Function.
Properties of one instance of one object	256	Maximum number of instances of Object, Number, String, Boolean, Array, Date, BinaryTable or, Function. For Array, the number of properties that correspond subscripts are excluded.
Elements of one array	1024	
Nest levels with function for invoking	16	Including functions invoked through event handlers.
Total number of properties of all objects	4096	The total number of properties to which the “properties of one instance of one object” restriction applied. Including number of properties of activation object and arguments object. Excluding built-in properties of global object (built-in functions, built-in objects, extended functions for broadcasting, extended objects for broadcasting) and properties of host object.
Global variables	64	
Function definitions	64	Any function is defined globally. Excluding event handlers.
Work memory for ECMAScript	8192 steps	Based on the computing method defined in ANNEX C, APPENDIX 2.

A5-5.4.2 Data type

It is assumed that any data type has the following restrictions:

- Number must be of single precision (32 bits). Note that any Number object must also be of single precision (32 bits).
- Float must not be supported.
- Math built-in object must not be supported.
- Dynamic type conversion must be restricted.
- The run-time interpretation of a script character string must not be supported.
- EUC-JP or Shift-JIS must be used as a character coding scheme of character string data.
- Use of Unicode Value must be restricted.
- Functions for compatibility with old codes must be restricted.
- The argument and the operation result of addDate() and subDate() must be restricted from 1900/3/31 to 2100/2/28.

A5-5.4.2.1 Character string

For operation of character string data, the restrictions in Section 4.1.4 are assumed.

UnicodeEscapeSequence (\uxxxx) in literal description must not be supported.

A5-5.4.2.2 Type conversion

The restrictions on dynamic type conversion are assumed below.

- A type of method argument is defined to restrict dynamic type conversion in DOM API and broadcast extended classes (The method of Browser quasi-object and extended object for broadcasting).
- Since ECMAScript has no variable type, there is no structure of examining a type on calling a function. Any data type can be assigned in the case of calling an extended function for broadcasting as argument having value of the other type of definition. Calling a function is prohibited if type can not be converted.
- Four basic types have been defined and regulated as mentioned below according to the operating restrictions.
  - [1] Numeric type (Number) must be a 32-bit single precision integer. This restriction is also applied to the number objects.
  - [2] Character string type (String) must be a character string coded by EUC-JP or Shift-JIS.
  - [3] Logical type (Boolean) has no particular restrictions.
  - [4] Object type (Object) has no particular restrictions.
- If the argument is specified as NaN where it must be a Number type and there is no particular description, it must be taken as a failure instead of 0.

The rules of conversions are listed in Table 5-20.

Table 5-20 Type Conversion Rules

Value	Character string	Number	Logic	Object
Character string	-	Numeric value of character string	true	String
Empty character string	-	0	false	String
0	“0”	-	false	Number
NaN	“NaN”	-	false	Number
Number	Numeric character string	-	true	Number
True	“true”	1	-	Boolean
False	“false”	0	-	Boolean
Object	toString()	valueOf()	true	-
Null	“null”	0	false	-
Undefined value	“undefined”	NaN	false	error

A5-5.4.3 Effects on basic objects caused by data type restrictions

Data restriction causes the following language effects:

A5-5.4.3.1 Effects caused by number object of signed 32-bit integer

- Behaviour of Date object
  - Calculation of Date object is returned in milli-second order as Number object. However overflow or underflow can be assumed if Number object is under 64 bits. subDate() and addDate() are used in calculation of Date object.
  - The precision of Date object must be kept while internally processed (not being affected by the restriction on Number object).
- Operation range of Date object
  - valueOf() method is not supported.
  - getTime() method is not supported. (A method for individually acquiring hours, minutes and seconds is used instead.)
  - setTime() method is not supported. (A method for individually acquiring hours, minutes and seconds is used instead.)
  - Date.parse() method and Date.UTC() method are not supported. (Numeration of time representation is restricted.)
  - Operations on Date object are not supported. Object the guidelines when operating the Number object of less than 64 bits. (subDate() function or addDate() function is used instead.)
  - Actual [[Value]] property of date object instance depends on an implementation.
- Number object
  - MAX_VALUE attribute must be 2147483647. (Maximum value of signed 32-bit integer)
  - MIN_VALUE attribute must be 1. (Minimum value of signed 32-bit integer)
  - If the value exceeds 2˜31, the process depends on the implementation.
  - If the string “Infinity” is not treated particularly and converted by ToNumber, the return value is NaN.

- The exponential expression is not used in the operation.

A5-5.4.3.2 Behaviours without Float

- Number object
  - NaN attribute is not expressed with IEEE754 but depends on an implementation.
  - MAX_VALUE attribute and MIN_VALUE attribute are defined with integer number which can be expressed by Number object.
  - NaN is returned in the case of division by 0.
  - NEGATIVE_INFINITY attribute is not supported.
  - POSITIVE_INFINITY attribute is not supported.
  - NaN attribute must be able to be distinguished from an integer which can be expressed by Number object. (Internal representation must depend on an implementation)
  - The result of a division by 0 returns NaN.
- Math object
  - No Math objects are supported.
- Global object
  - The infinity attribute is not supported.
  - The parseFloat() function is not supported.
  - The isFinite() function is not supported.

A5-5.4.3.3 Effects by restrictions on run-time interpretation of script character string

- The eval() function of the Global object is not supported.
- new Function() of the Function object constructor is not supported.

A5-5.4.3.4 Restrictions of functions left for compatibility with older codes

- Date object
  - The getYear() method is not supported. (The getFullYear() method is used.)
  - The setYear() method is not supported. (The setFullYear() method is used.)
  - The toGMTString() method is not supported. (The toUTCString() method is used.)

A5-5.4.3.5 Generation of Date object

When the new Date() function is invoked and fails to obtain the current, the following behaviour depends on an implementation.

A5-5.4.4 Operational general rule of implementation-dependent behaviour

This section contains operational general rules of implementation-dependent behaviours.

A5-5.4.4.1 Major syntax

- Sequence of the properties taken out by “for (variable in Expression) statement”
  It must remain implementation-dependent. The “for (variable in Array object)” sequence is also implementation-dependent.

A5-5.4.4.2 Host object

- Host object range
  - DOM object
  - Browser pseudo object
  - Ureg pseudo object
- Result of typeof for host object
  Returns "hostobject."
- Result of new Object(hostobject) for host object
  Returns reference to hostobject in the same manner as when using a normal object as an argument.
- Results of Array.prototype.join(), Array.prototype.reverse() and Array.prototype.sort() for host object
  Inhibits addition of any property to the host object.
  Remarks: There is no need to consider this behaviour because it becomes impossible to insert Array.prototype.join, etc. as a new property of the host object.
- [[Class]] of host object
  "hostobject"

A5-5.4.4.3 Built-in object

- [[prototype]]of Global object
  null
- Result of Function.prototype.toString()
  "function FUNCTIONNAME() {}"(FUNCTIONNAME is the name of a specified function.)
- Result of Array.prototype.sort()
  An array element that is not influenced on sorting may not be automatically generated as an undefined. The internal comparison sequence of sort() may depend on an implementation.
- Result of Number.prototype.toString(radix) without radix = 10
  Only radix = 8,10, or 16 is applicable. Otherwise, operation depends on an implementation.
  radix = 8: ["0"-"7"]*
  radix = 16: ["0"-"9","a"-"f"]*
- Result of Date.prototype.toString()
  Must be in the format of "DateThoursminutesseconds."
  Date must be YYYY-MM-DD. (Ex. 1999-01-01)
  Hours, minutes and seconds must be hh:mm:ss. (Ex.: 23:01:34)
  'T' (character code 0x54) must be used as a delimiter between the date and the hours, minutes and seconds. (Ex.: 1999-01-01T23:01:34)
  If the result is a negative value, the low-order four digits are used and the sign (d.c. or a.c.) is ignored.
- Results of Date.prototype.toLocaleString() and Date.prototype.toUTCString() must be of the same output format as Date.prototype.toString().
- LocalTZA of Date object must be fixed to +9 hours of Japan Standard Time.

- TimeClip() clip range
  Must be within the range of signed 64-bit integers.

A5-5.4.4.4 Implementation of event handler

The code type of the event handler must be an implementation supplied code. Also, neither ImplicitThis nor ImplicitParent must be set. Even if an event handler return value is false, the next processing is performed continuously regardless of its value.

A5-5.5 Operational Guidelines on Broadcasting Extended Objects

A5-5.5.1 Operational guideline on BinaryTable class

A-5.5.1.1 Operation of BinaryTable object

Actual size of the BinaryTable object is defined in an operational standard reguration.

A5-5.5.1.2 Behavior of BinaryTable

As shown in Figure 5-7, constructor BinaryTable allows reference of table data which resides in content memory. This reference is valid until close( ) method is invoked or the document is finished.

- If a binary table and the BML document to be referenced are in the same module, the binary data cannot be released from content memory by the close( ) method.
- If a binary table and the BML document to be referenced are in different module, the constructor continues to operate until receiving operation is completed, under the condition that table data does not reside in content memory when the constructor is invoked. If a binary table data is mapped as one module when close( ) is invoked, the data can be released from content memory. If data other than binary table data is contained in one module and the module is locked, it must not be released.
  The rows and columns which indicate record length as shown in Figure 5-7 must not be specified.
- When a BinaryTable object is referenced through a variable, in order to read an updated module that contains the BinaryTable object, the close () must be applied to the Binarytable object.
  The steps are described below.

  1) Use a beitem element to detect an update of the module
  2) A module update is detected.
     Execute the following in this order through handling a ModuleUpdated event.
     - Apply close() to the binary table.
     - Execute unlockModuleOnMemory().
     - Use lockModuleOnMemory() to load a new module in content memory.
  3) A ModuleLocked event is detected.
     Execute the following in this order through handling a ModuleLocked event.
     - Execute new BinaryTable().

In process (2), close() and lockModuleOnMemory() must be executed in the described order. If they are executed in wrong order, BinaryTable continues to refer to an existing module, it is not ensured that a new module is loaded, as intended.

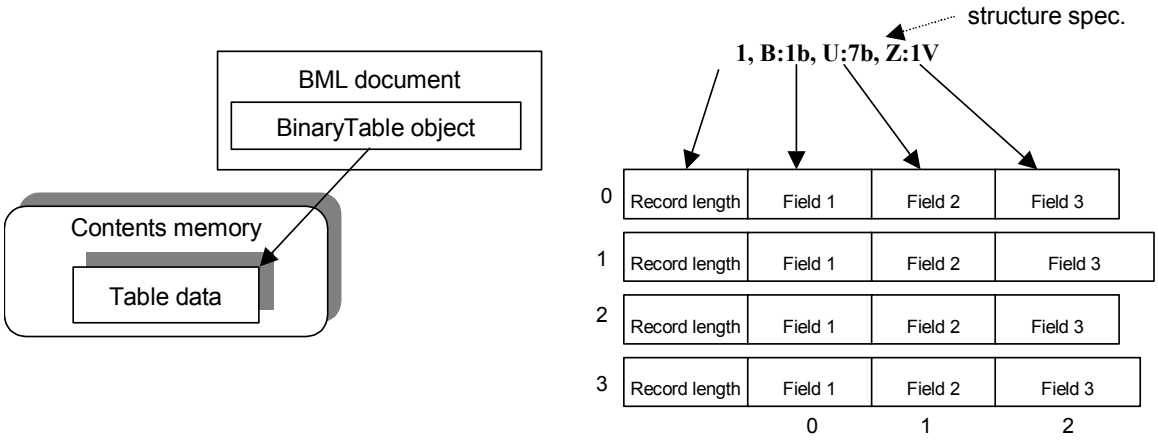


Figure 5-7 Behaviour of BinaryTable

A5-5.5.1.3 Inter-document sharing

The following shows an operation example of sharing a binary table between documents in a document group. The binary table and the BML document to be referenced are transmitted in another modules. The flow of sharing binary table data is shown below with Figure 5-8.

[1] Using lockModuleOnMemory( ), the binary table data is locked in content memory.

[2] The locked binary table data is retained in the document group.

[3] BinaryTable constructor generates a binary table object.

[4] The document is changed over without invoking the close method.

[5] The binary table data is referenced by the BinaryTable constructor of the target document again.

[6] When the scope comes out of the document group or when close( ) is done, the binary table data is released.

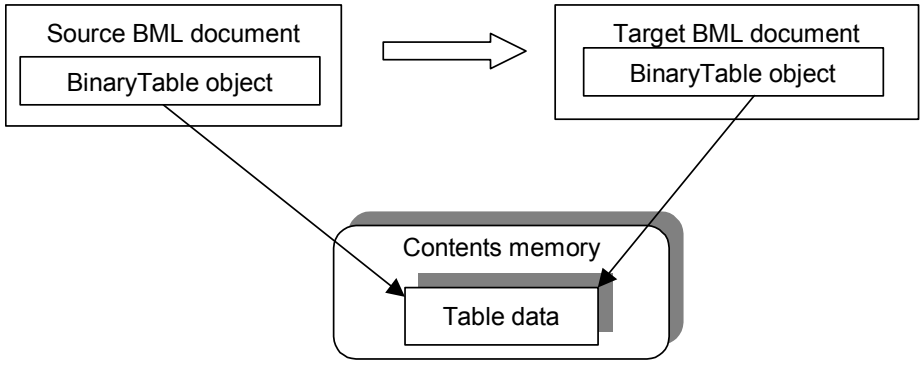


Figure 5-8 Inter-document sharing of binary table

A5-5.5.1.4 Operational restrictions on the structure argument

- The unit of the record length specified with lengthByte is byte. The record length specified here does not contain the size of lengthByte itself.
- The value of lengthByte is 0, 1, or 2. 0 is specified if and only if all records have the fixed length field.

- A decimal character string to be specified to the field length doesn't begin with 0.
- When unit is "V" (a variable length field), the length field does not include its own field. For example, when a character string "ABC" is stored as a variable length character string, ‘3’ is stored in the length field.
- When String-type data are specified to a variable length field, the length data must be either one byte or two bytes.
- The designation of a postal code is "Z : 1V".
- The padding is employed to make the data byte-aligned, and the length must be from 1 bit to 7 bits. Therefore, the padding with the specified number of bytes ("P:? B") is not employed.
- Even when a BinaryTable is composed of the fixed length field only, the start position of the record is byte-aligned. For example, when the structure is specified with "0, B : 1b", the second byte does not start from the second bit of the first byte, but from the second byte.
- When the field type is "U" (Unsigned Integer) and the length is specified with less than seven bits, it is not necessary to be byte-aligned. For example, when it was specified with U : 1b, U:1b, U:3b, U:4b, U:6b, U2b", the data specified by four bits and two bits extend over the byte alignment.
- When the data length of a String type data is 0-byte, this data is assumed to be an empty string.
- When the data length of a ZipCode type data is 0-byte, this data is assumed not to contain a specified zip code.

A5-5.5.1.5 The property of BinaryTable

- The nrow and ncolumn properties are in Number type.

A5-5.5.1.6 BinaryTable.prototype.search()

- The maximum number of the comparison condition is 4.
- When the number of the retrievals is below limitCount, the number of the elements of resultArray is the number of the retrieval result.
- The number of the retrieval result is always set up in length of resultArray. In other words, it is 0 (zero) when no more than 1 case is found, and the number of the results is set up in length of resultArray when any matched items are found.
- When a limitCount-th item is found by the last record, the return value is the position of the last record.
- It judges " The retrieval condition holds." by "The result of = = ToBoolean (operation result) is true." For which, an operation result is false for Undefined, Null, false, 0 (zero), NaN, the empty character string, and it is true except for them.
- For an operator 6 to 8, the operation result is the data masked by the size of the field.
- For an operator 9 to 11 (bit operation with the 1’s complement), the operation result is obtained by 1’s complement of the data in the table, bit-operated with comparison, and masked with the range of the field size.
- An error occurs when any object except for the Array object is specified in resultArray.
- When two retrieval expressions are specified with OR for a postal code field, ‘true’ is returned if at least one retrieval condition is satisfied.
- For example, when a table data is "1230001" and the retrieval expression is "postal code == 1230001 | | postal code = = 1230002", ‘true’ is returned.

A5-5.5.1.7 The behaviour after close()

- When the nrow or ncolumn property is referenced, or the toNumber()method is invoked after close(), the behaviour afterwards is implementation dependent.

A5-5.6 Guidelines on Behaviour of Browser Pseudo Objects

A5-5.6.1 Guidelines on behaviours of EPG functions

Operation guidekines on EPG-related methods are shown below.

- epgGetEventStartTime( )

  The event which can be designated by event_ref must be the one currently displayed.

  This function acquires a start_time value by referencing EIT.

- epgGetEventDuration( )

  The event which can be designated by event_ref must be the one currently displayed.

  This function acquires a duration value by referencing EIT.

- epgTune( )

  Once the event handler that invoked the API finished, the browser causes an onunload event immediately and places it into the event queue. After the event handler started by the given onunload finished, the browser displays a given service.

- epgReserve( )

  The startTime must be designated.

  It is implementation dependent how to handle overlaps of view reservations caused by this method.

- epgRecReserve()

  The startTime must be given.

  When a view reserved by this method overlaps with another at the specified time, handling depends on an implementation.

- epgIsReserved( )

  The startTime must be given.

- epgRecIsReserved( )

  The startTime must be given.

A5-5.6.2 Operation of non-volatile memory functions

A rule of correspondence between the file name specified with the argument filename of the non-volatile memory functions and the area of NVRAM is defined by a broadcaster.

- writePersistentArray( )

  The period, an optional argument, is not essential for operation. Written data must be allowed to be overwritten without its life controlled.

  Designation of structure is the same as the binaryTable object but has the following restrictions in Table 5-21:

Table 5-21 Restriction on Structure Specified in writePersistentArray()

Available field type	Corresponding data type	Operational restrictions
B	Boolean	An error results if byte alignment causes a mismatch.
I	Number	The upper bits are cut off if less than 32 bits are specified. An error results if a value of more than 32 bits is set.
U	Number	The upper bits are cut off if less than 32 bits are specified. An error results if a value of more than 31bits is set. An error results if a negative value is set.
S	String	The latter half of a character string is rounded off if longer than specified character string length; otherwise, a blank (0 x 20) is added. A variable-length character string is specified by S:1V.
P	-	

When actual data type is different from binaryTable data type, the type conversion rule mentioned in 5.4.2.2 of this Appendix is used. If the rule is not applicable, an error is returned. And, if the array data does not reach the number of fields specified by structure, the result must be an error. Actual data type used when 'P' is specified for the field format depends on an implementation. When an array of two or more dimensions, no successful operation is guaranteed.

- readPersistentArray( )

  The operation of structure must be the same as writePersistentArray( ). When a structure value that is not what is specified with writePersistentArray( ) is read, no successful operation is guaranteed.

A5-5.6.3 Operation of interaction channel function

- transmitTextDataOverIP()

  The return values available for this function are listed in Table 5-22.

Table 5-22 Return values (Array[0]) of transmitTextDataOverIP()

Return value		Semantics	Description
Array[0]	1	Success	The text argument is successfully sent to the internet resource specified with the uri argument and the response is received.
	-1	Parameter error	Arguments contains errors.
	-2	Line was disconnected during transfer.	The line was disconnected while the text data is sent.
	-3	Time-out occurred.	The specified period of time has passed without the completion of the data transmission and/or the response reception.
	-300	Fails to establish an automatic connection	An automatic connection was not established due to an error and the data was not sent to the internet resource specified with the uri argument. If an error occurs while a physical line connection, PPP, and/or TCP is processed, this return value is returned.
	-400	Fails to map names using DNS	The host name specified with the uri argument was not resolved to an IP address.

	-500	Fails to process TLS-based operation	A TLS operation involving an HTTPS connection was not completed.
	NaN	Failure by other causes	A condition used to restrict an automatic retransmission was detected.

A5-5.6.4 Operation of operational control functions

The following shows an operation example of functions for receiver behaviours controlling functions.

- getProgramRelativeTime()

  Returns a difference between the start time (starttime) of a broadcast program described in PresentEIT and the current time acquired by the receiver. The difference is calculated on a unit of second and shown in the relative time format.

- lockModuleOnMemory( )

  The module locked by this method must be valid within a document group. The basic operation is shown below.

  [1] lockModuleOnMemory( ) is issued to start acquisition of a module. If the specified module is not found, an error is returned.

  [2] Once the specified module is locked, a moduleLocked event occurs and the module is maintained in content memory.

  [3] When unlockModuleOnMemory( ) is invoked or when the scope is out of document group, the module is unlocked and deleted from content memory.

  [4] If an attempt is made to lock the locked module using lockModuleOnMemory( ), a ModuleLocked event is regenerated.

  The operation where an operationally locked module is deleted from the content memory is shown in Figure 5-9.

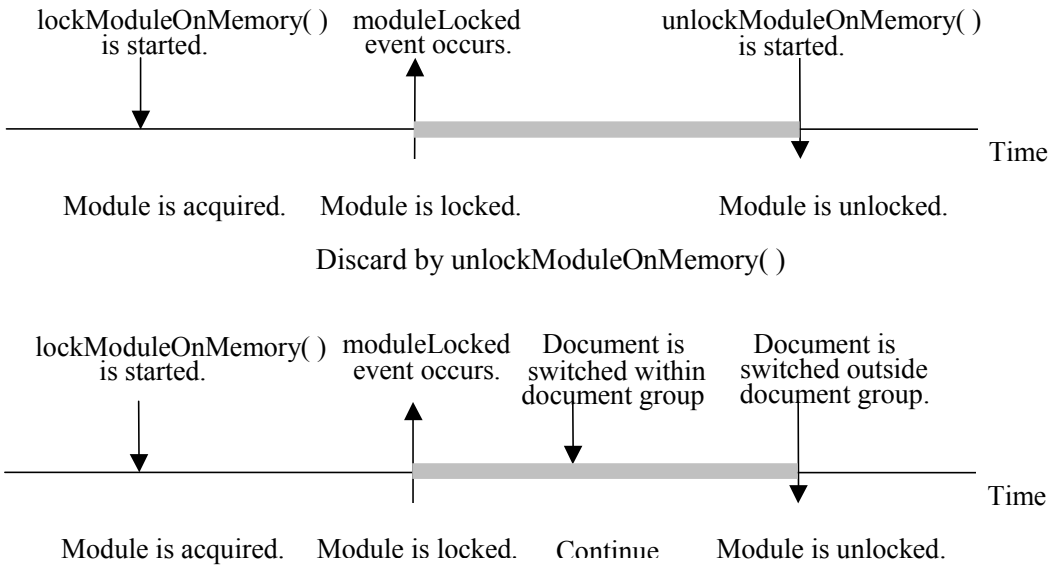


Figure 5-9 Module Lock Behavior

Note: Although an update of the version of the module that is locked in the memory by the execution of lockModuleOnMemory() can be observed, in order to get the new version of

data after detecting the update of the version (after ModuleUpdated is occurred), the lock of the old version must be released. When lockModuleOnMemory( ) is issued to lock a module that is not locked in content memory, invoking unlockModuleOnMemory() before the module is successfully locked results in cancelling the lock request. When unlockModuleOnMemory() is issued to unlock a module that is not locked in content memory and lockModuleOnMemory( ) has not been issued to lock the module, unlockModuleOnMemory() returns an error.

- getActiveDocument( )

  The return value is a URI of shortened form (a relative path starting with "/").

- launchDocument( )

  Actual process of a transition effect specified by the second argument "transitionStyle" depends on an implementation

A5-5.6.5 Operation of receiver build-in sound control

- playRomSound( )

  A sound_id(romsound://<sound_id>) value corresponding to a receiver build-in sound is defined by a rule developed by a broadcaster.

A5-5.6.6 Operation of timer function

The following shows an operation of timer function:

- setInterval( )

  The first argument is restricted to be only in the "func();" format, which is the same format used in calling the function as an event handler. Up to 8 timers can be set simultaneously in units of 100 msec. The timer precision depends on an implementation.

A5-5.6.7 Operation of external character function

External characters are operated by using the DRCS scheme.

A5-5.6.8 Operation of data carousel storage functions

When the "Back" key is pressed during an execution of saveModule() or saveResource(), the save operation aborts and a value representing "Failure" is returned. When a save operation aborts, any file and directory that the operation attempted to save is deleted.

A5-5.6.9 Operation of module lock among components

The lockModuleOnMemoryEx() function is designed to enable locking in a service to which a currently presented BML document belongs. This means locking is also applicable to the component to which the BML document belongs. Once the function is triggered, the intended object is locked in an appropriate memory area until another service is selected or the currently presented BML document detects an update on a data event in a transmitted data carousel.

A5-5.7 Operation of link Element

A5-5.7.1 Operation of link element for external style sheet specification

The attribute values available to the link element to specify an external style sheet to be imported are listed in Table 5-23.

Table 5-23 External Style Sheet Specification

rel attribute value	type attribute value	href attribute value
Stylesheet	text/css	URI of an external style sheet to be imported

A5-5.8 Other Restrictions

For the purpose of scalability, the following keywords and object names, which are operated in JavaScript, are regarded as reserved. If any of the following reserved words is used as a variable name or other, it is not assured that no unexpected behaviour occurs.

ECMA-262 2nd (BML1.0) keywords

break	for	new	var
continue	function	return	void
delete	if	this	while
else	in	typeof	with

ECMA-262 2nd (BML1.0) reserved words

abstract	do	import	short
boolean	double	instanceof	static
byte	enum	int	super
case	export	interface	switch
catch	extends	long	synchronized
char	final	native	throw
class	finally	package	throws
const	float	private	transient
debugger	goto	protected	try
default	implements	public	volatile

ECMA-262 3rd keywords

break	else	new	var
case	finally	return	void
catch	for	switch	while
continue	function	this	with
default	if	throw	
delete	in	try	
do	instanceof	typeof	

ECMA-262 3rd reserved words

abstract	enum	int	short
boolean	export	interface	static
byte	extends	long	super
char	final	native	synchronized
class	float	package	throws
const	goto	private	transient
debugger	implements	protected	volatile
double	import	public	

Objects

	STD-B24	ECMA-262 3rd	Core JavaScript 1.5	Client JavaScript 1.3
Built-in	Object	Object	Object	Object
	Function	Function	Function	Function
	Array	Array	Array	Array
	String	String	String	String
	Boolean	Boolean	Boolean	Boolean
	Number	Number	Number	Number
	Date	Date	Date	Date
	Math	Math	Math	Math
		RegExp	RegExp	RegExp
		Error EvalError RangeError ReferenceError SyntaxError TypeError URIError		
			JavaArray	JavaArray
			JavaClass	JavaClass
			JavaObject	JavaObject
			JavaPackage	JavaPackage
			netscape	netscape
			Packages	Packages
			sun	sun
Host object (DOM/ extended/pseudo/host)	document			document
	CSVTable			
	BinaryTable			
	browser			
	XMLDoc			
				Anchor
				Applet
				Area
				Button
				Checkbox
				event
				FileUpload
				Form
				Frame
				Hidden
				History
				Image
				Layer
				Link
				Location
				MimeType
	navigator			navigator

	STD-B24	ECMA-262 3rd	Core JavaScript 1.5	Client JavaScript 1.3
				Password
				Plugin
				Radio
				Reset
				screen
				Select
				Style
				Submit
				Text
				Textarea
				window

Top-Level Properties and Functions

	STD-B24 (Operation)	ECMA-262 3rd	Core JavaScript 1.5	Client JavaScript 1.3
	NaN	NaN	NaN	NaN
		Infinity	Infinity	Infinity
		undefined	undefined	undefined
		eval()	eval()	eval()
	parseInt()	parseInt()	parseInt()	parseInt()
		parseFloat()	parseFloat()	parseFloat()
		escape()	escape()	escape()
		unescape()	unescape()	unescape()
	isNaN()	isNaN()	isNaN()	isNaN()
		isFinite()	isFinite()	isFinite()
		decodeURI()	decodeURI()	
		decodeURIComponent()	decodeURIComponent()	
		encodeURI()	encodeURI()	
		encodeURIComponent()	encodeURIComponent()	

* It is not required but recommended that the properties and functions in italics are regarded as reserved.

Each object is associated with functions converting to the basic types (Number(obj) and String(obj)).

A5-5.9 Operation of Extended Functions

A5-5.9.1 Operation of High-resolution Presentation Function

The following table lists resolution description excluding QVGA and their resolutions.

	Resolution description	Resolution	Remarks
1	VGA	640 x 480	
2	SQVGA	160 x 120	
3	SIF	352 x 240	
4	QSIF	176 x 120	
5	CIF	352 x 288	
6	QCIF	176 x 144	
7	HD	960 x 540	
8	QHD	480 x 270	

Details for operating each resolution are defined in an operational standard regulation for each media type.

A5-5.10 Operation of Cookies

A5-5.10.1 Version of Cookies

Any Cookie is operated as defined in RFC2965. For the purpose of compatibility with existing HTTP servers, this section provides the additional operational convention.

A5-5.10.1.1 Operation of response headers

- Any receiver must interpret a Set-Cookie response header. Whether and how a Set-Cookie is interpreted depend on an implementation.
- The following parameters must be interpreted by any receiver.

Attribute	Inclusion in response header by server	Description
NAME=VALUE	Required	The substance of Cookie. How double quotes ("), spaces, tabs, and line breaks in the VALUE string are interpreted depends on an implementation.
domain=DOMAIN	Optional	The name of a domain for which the Cookie is valid.
path=PATH	Optional	The path to which the Cookie is applied.
secure	Optional	A Cookie with the secure attribute specified is sent only when a communication with a host is secured (e.g. when connected with a HTTPS server).
expires = DATE	Optional	Expiration date. The format of the DATE attribute must be " Wdy,dd Mth yyyy hh:mm:ss GMT" or " Wdy,dd-Mth-yyyy hh:mm:ss GMT".

- The semicolon, ";" must be interpreted as a delimiter between attributes. Whether or not the comma, "," is interpreted depends on an implementation.
- A single response header contains not more than one Cookie.

A5-5.10.1.2 Operation of request headers

- NAME=VALUE pairs are included in a HTTP request header to be sent to a host.

  For example,

  Cookie: NAME1=VALUE1;_ NAME2=VALUE2: _NAME3=VALUE3; .....

- A receiver needs not specify $Version, $Path, nor $Domain in a request header.

A5-5.10.1.3 Operation of cache control

Whether and how a cache control field (Cache-Control or Cache-Control2) of a response header is interpreted depend on an implementation.

A5-5.10.2 Function enabling Cookies

- It is recommended that a receiver is equipped with a function that allows an end user to enable or disable Cookies.

A5-5.10.3 Restrictions on implementation of Cookies

- The maximum length of a string contained in a Set-Cookie header field is 4096 bytes, excluding the leading string part "Set-Cookie:".
- The number of Cookies that must be retained in a receiver and the maximum number of Cookies per host are defined in an operational standard regulation.

A5-5.10.4 Reference of Cookie from DOM

- To reference or write a Cookie in a procedure description language, document.cookie is used.

A5-5.10.5 Reference of Cookie in broadcast/communication/stored content

- Among the contents, only contents that transmitted via interactive communication networks are capable of referencing Cookies. No stored content nor content broadcast in real time reference a Cookie.
- To share information among contents received in real time, contents stored from a broadcasting service, and contents transmitted via interactive communication networks, Greg is used.

  Note: When a mechanism that uses non-volatile memory to share information, other than Greg working with RAM, is needed, the mechanism is defined in an operational standard regulation.

Chapter APPENDIX 5-6 Guidelines on Contents Transmission and Namespace

A5-6.1 Scope Mapping to Transmission System

A BML document switching operation is performed when:

- the a element in a BML document is used to switch documents.
- the launchDocument( ) or the launchDocumentRestriced() function is used to switch documents in a script of a content.
- other service than the currently selected service is selected.
- a signal that forces a document switching is received.
- the submit attribute of the form element represents a document switching.

If a scope identifier representing a scope level of a BML document becomes invalid after the above switching operation, the corresponding BML document gets out of the scope. The following three sections define scope mapping for BML contents received in real time, BML contents transmitted via interactive communication networks associated with broadcasting services, and stored BML contents.

A5-6.1.1 Scope mapping for contents received in real time

For a content received in real time, scopes, mapping among scope identifiers, and transmission signals is operated as defined in Table 6-1.

Table 6-1 Transmission and scope mapping

Scope	Scope identifier	Corresponding transmission signal
Within document	Document scope identifier	moduleId, resourceName
Within document group	Document group scope identifier	data_event_id
Within content	Contents scope identifier	component_tag
Within content group	Content group scope identifier	service_id

Figure 6-1 shows an example in which content group A has one entry content and two contents, and content group B has one content being broadcasted. The following explains this transmission example using the numbers in Figure 6-1.

A tuning operation selects a given BML document to be started. In the example below, content 2 begins.

[1] When documents are switched within the same data_event_id in content 2, the operating scope is defined as “between the documents” and “within the document group”.

[2] When more than one data_event_id value is used to switch documents in content 2, the operating scope is defined as “between the document groups” and “within the content”.

[3] When data_event_id is switched, content 2 is forced to terminate by the return_to_entry_flag signal, and the entry content starts, the operating scope is defined as “between the contents” and “within the content group”.

[4] When the a element or the launchDocument( ) function in the document causes the entry content to switch to content 2, the entry content terminates and content 2 starts. This operating scope is defined as “between contents” and “within the content group”.

[5] When an end user’ s operation causes content 2 to switch to another content group B, content 2 terminates and content 1 of the group starts. This operating scope is defined as “out of the content group”.

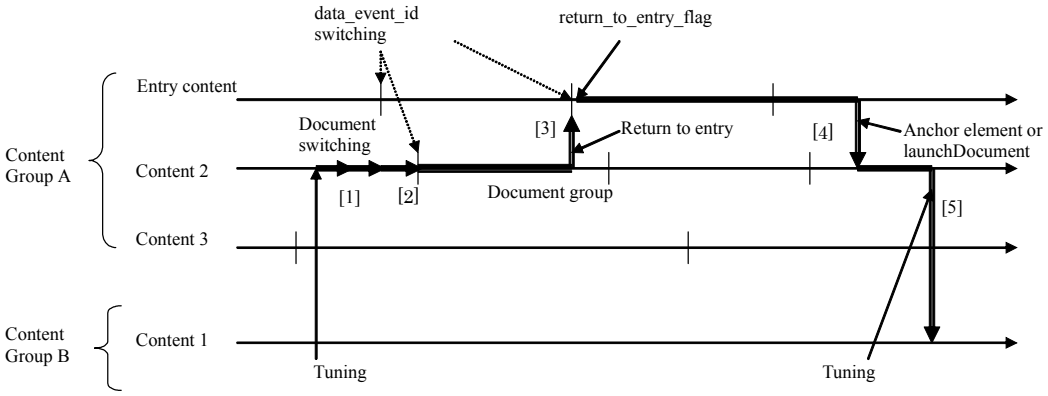


Figure 6-1 Example of Contents Transmission

A5-6.1.2 Scope mapping for contents transmitted via interactive communication networks associated with broadcasting services

The concepts of “document group”, “content”, and “content group” are not applied to contents transmitted via interactive communication networks associated with broadcasting services.

However, when a content transmitted via interactive communication networks is identified as a primary link destination from a content received in real time or a content stored from a broadcasting service, the content transmitted via interactive communication networks is regarded as a member of the same content group as the linking content. Specifically, in the following case, a content transmitted via interactive communication networks is identified as a primary link.

- A BML document transmitted via interactive communication networks is in or under the same path (directory) as a BML document that is the first destination of a linking BML document received in real time or a linking stored BML document.
- However, when a BML document received in real time or a stored content uses the launchDocumentRestricted() function to transit to a BML document transmitted via interactive communication networks as the first destination, no BML document obtained via interactive communication networks is identified as a primary link destination.

A5-6.2 Guidelines on Reference across Media Types

A5-6.2.1 Restrictions on document transmission among media types

Among contents received in real time and contents transmitted via interactive communication networks, which content is allowed to be specified as a linked destination of a BML document is defined in Table 6-2. In this table, the following transition types are represented by the numbers (1) through (4).

(1) transition caused by epgTune()/epgTuneToComponent()

(2) transition caused by a value in the href attribute of the a element

(3) transition caused by launchDocument() or launchDocumentRestriced()

(4) transition caused by the submit function of the form element

Table 6-2 Restrictions on Document Transmission across Media Types

	Transiting (Linking) Content							
	Broadcast (contents received in real time)				Communications (contents transmitted via interactive communication networks)			
Transited (Linked) Content	(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)
Broadcast	O	O (Note 1)	O (Note 1)	-	O	O (Note 2)	O (Note 2)	-
Communication	-	O	O	O	-	O	O	O

Note 1: This description is applicable to BML documents belong to the same service.
Note 2: This description is applicable to BML documents belong to the currently selected service.

Among the transition patters in Table 6-2, when a transition from a content transmitted via interactive communication networks to a broadcast content involving a service selection operation is caused by the (2) through (4) transition types described above how a receiver behaves depends on an implementation.

A5-6.2.2 Restrictions on reference of monomedia and others across media types

The following six items are assumed as resources referenced by a BML document. The operational convention on these items are summarized in Table 6-3.

(1) A monomedia that can be specified with the data attribute of the object element (an AV stream for broadcasting services)

(2) A monomedia that can be specified with the data attribute of the object element (e.g. JPEG, PNG, MNG, MPEG-2 audio), a CLUT data specified by the clut property, and a JPEG data specified by the background-image property

(3) A DRCS file specified with an attribute of the loadDRCS() function

(4) A binary table

(5) An event message referred by the beitem element

(6) A proprietary data that is designed to be stored in a bridge media or others by a saving operation through a BML document

The following table contains no convention on a function transmitting messages with TCP/IP.

Table 6-3 Resources Referenced by BML document

			Referencing BML Document		Remarks
			Content received in real time (Broadcast)	Content transmitted via interactive communication network (Communication)	
Referenced Resource	Broadcast	(1)MPEG4-V (Note 1)	O Within the same service	O (Note 3)	For Broadcast to Broadcast reference, not more than one carousel specified with the same data_ event_id is allowed to be referenced at a time.
		(1)H.264| MPEG-4 AVC (Note 1)	O Within the same service	O (Note 3)	
		(1)MPEG2-AAC (Note 1)	O Within the same service	O (Note 3)	
		(2)JPEG/PNG/ MNG	O Within the same carousel (Note 2)	-	
		(3)DRCS	O Within the same carousel (Note 2)	-	
		(3)Binary Table	O Within the same carousel (Note 2)		
		(4)Event Message	O Within the same carousel	O (Note 3)	
	Communi cation	(2)JPEG/PNG/ MNG	-	O	
		(3)Binary Table	-	O	
		(5)Proprietary Data	-	Δ Limited to saving operations	

Note 1: Video/audio streams contained in an event program that is to be received in real time

Note 2: This description is also applicable to resources contained in a module that has been transmitted in the same ES and locked, even if the content does not transmitted in the same carousel.

Note 3: This description is only applicable when a referencing communication content is identified as a trusted content in either of the following mehods. Which method, including the combination of the two, is operated is defined in an operational standard regulation for each media.

- Identifying the referencing communication content as a primary link destination transited by a referenced broadcast content
- Identifying the content as a trusted content using a certificate (Details of certificates are defined in an operational standard regulation.)

A5-6.3 Namespace

A5-6.3.1 Namespace for BML content received in real time

- Any resource specification containing arib or arib-dc complies with the convention defined in Volume 2.
- However, in a component-base model, it is allowed that the href attribute of the a element contains the id attribute of other element in a BML document in the following format.
  #<id attribute>

A5-6.3.2 Namespace for BML content transmitted via interactive communication networks associated with broadcasting services

- A specification of a content transmitted via interactive communication networks associated with broadcasting services begins with http.
- A specification of a resource to be obtained via secure communication begins with https.
  - The namespace for URI complies with the definition provided in RFC 18055 and the additional restrictions described below are applied.
    - When an access to a directory is requested (the URL ends with /), how a server behaves depends on the server and how a receiver behaves depends on the implementation.
    - The representation “./” or “../” is applicable only to the beginning of a relative URL.
    - It is assumed that ~ is not interpreted by a receiver.
    - The representation, ;params, is not operated.
    - The namespace for a CGI argument complies with the namespace defined in RFC 1808.
    - A directory name or file name may contain a single period, ”.”, unless the name begins nor ends with the period.

The valid syntax is shown below.

```
URL = (absoluteURL | relativeURL ) [ "#" fragment ]
    absoluteURL  = scheme ":" "//" net_loc "/" rel_path
    relativeURL  = [ ("./" | "../" | "/" ) ] rel_path
    net_loc      = *(pchar | ";" | "?")
    rel_path     = [ path ] [ "?" query ]
    path         = fsegment *( "/" fsegment )
    fsegment     = 1*fpchar [ "." 1*fpchar ]
    query        = *( uchar | reserved )
    fragment     = 1*fchar [ "." 1*fchar ]
    fpchar       = fchar | ":" | "@" | "&" | "="
    fchar        = alpha | digit | "-" | "_" | "+"
    pchar        = uchar | ":" | "@" | "&" | "="
    uchar        = unreserved | escape
    unreserved   = alpha | digit | safe | extra
    escape       = "%" hex hex
```

hex = digit | "A" | "B" | "C" | "D" | "E" | "F" | "a" | "b" | "c" | "d" | "e" | "f"

alpha = lowalpha | hialpha

lowalpha = "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z"

hialpha = "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z"

digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

safe = "$" | "-" | "_" | "." | "+"

extra = "!" | "*" | "'" | "(" | ")" | ","

national = "{" | "}" | "|" | "\" | "^" | "~" | "[" | "]" | "`"

reserved = ";" | "/" | "?" | ":" | "@" | "&" | "="

punctuation = "<" | ">" | "#" | "%" | <">

- The maximum length of a URI is specified in an operational standard regulation.
- A shortened form of URI is interpreted based on RFC 1808 and the additional restrictions described below.
  - Any tilde ~ character is interpreted as interpreted by a server. A receiver does not interpret the character.
  - A URI specification containing the representation "./" or "../" is interpreted as a relative URI to a file or directory in which the current BML file resides by a receiver. Note that this restriction is applicable only to a URI specification that does not begin with "./" nor "../". For example, the specification, foo/../bar is not operated.
  - The slash "/" character is interpreted as the root directory of a server by a receiver.

A5-6.4 Guidelines on Transmission of BML content

A5-6.4.1 Syntax of module

- Each entity-body that is part of another entity-body in the multi-part format must not be of the multi-part type as media type.
- The entity headers that any basic receiver is required to interpret are listed in Table 6-4.

Table 6-4 Entity Headers Interpreted by Any Basic Receiver

Entity header	Semantics
Content-Type	Indicates a media type of a resource contained in an entity body. Define Content-Type for multipart entity body as "multipart/mixed" and specify a boundary parameter.
Content-Length	Indicates the byte length of an entity body.
Content-Location	Indicates the identifier of an entity body.

- The Content-Base:header and the Content-Encoding field are optional.
- Any Content-Location field specification contains only <resourceName> as a relative specification to the module. The namespace for <resourceName> complies with the namespace for resources, as defined in Chapter 9, Volume 2.

A5-6.4.2 Resource list in multi-part formatted module

Any module in the multi-part format that is to be received in real time must contain a resource list at the beginning.

A5-6.4.3 Operation of event messages

The character encoding specification for private_data_byte of a generic event message descriptor must be "EUC-JP".

A5-6.5 Operational Guidelines on Communication Protocols

A5-6.5.1 Operational guidelines on HTTP

The version of HTTP to be supported by a receiver must be 1.1. Any receiver must be able to successfully connect with any server operating under HTTP/1.0.

A5-6.5.2 Operational Guidelines on mapping between API and protocols

To use http to make a BML content access to a communication server, the methods listed in the following table are applicable.

Access	Valid Method
To obtain a BML document from a server by including http in the href attribute in the a element	GET method is used
To obtain a resource from a server by including http in the data attribute in the object element	GET method is used
To obtain a BML document from a server by including http in an argument in the launchDocument() function	GET method is used
To communicate with a server by using the form element	GET method or POST method is used depending on a value in the action attribute

A5-6.5.3 Operational guidelines on secure communication

How secure communication is operated is defined in an operational standard regulation.

Annex A Default Style Sheet

A.1 Default Style Sheet Common to Index-based Model and Component-based Model

```
@media tv{
body, h1, h2, h3, h4, h5, h6,
address, blockquote, div,
p, pre,
dl, dt, dd, ol, ul,
hr, form          { display: block }
li             { display: list-item }
table          { display: table }
tr             { display: table-row }
th, td         { display: table-cell }
caption        { display: table-caption }
head, param    { display: none }
/* display */
meta, title, script, style, head, bevent, beitem, { display : none }
input[type=hidden]{ display: none }
noscript{display:none}

address, cite, dfn, em,i
           { font-style: italic }

pre        { white-space: pre }
br:before      { content: "\A" }

q:before       { content:open-quote }
q:after        { content:close-quote }
q          { quotes:'"' '"' }
```

A.2 Default Style Sheet Specific to Index-based Model

```
/* The following rule sets are added to the default style sheet provided in A.1 for
an index-based model.*/
@media tv{
/* visibility */
body { visibility : visible}
/* text */

div, p, span, a, input { background-color-index : 8 }
object { background-color-index : 8}
/* grayscale-color-index */
p, input { grayscale-color-index : 30 15 }
}
```

A.3 Default Style Sheet Specific to Component-based Model

```
/* The following rule set is added to the default style sheet provided in A.1 for a
component-based model. */
```

```
    a[href]         { text-decoration: underline;}
```

Annex B Operated DTD

B.1 BML Driver DTD

The BML driver DTD, as containd in Annex B.1 Section, Annex 3, Volume 2 must be used.

B.2 BML Document Model Module

The BML document model module, as containd in Annex B.2 Section, Annex 3, Volume 2 must be used.

[BLANK]

DATA CODING AND TRANSMISSION SPECIFICATIONS
FOR DIGITAL BROADCASTING

ARIB STANDARD

ARIB STD-B24 VERSION 5.1-E1
VOLUME2 (2/2)
(March 14, 2007)

This Document is based on the ARIB standard of “Data Coding and Transmission Specifications for Digital Broadcasting” in Japanese edition and translated into English in May, 2007.

Published by

Association of Radio Industries and Businesses

Nittochi Bldg. 11F
1-4-1 Kasumigaseki, Chiyoda-ku, Tokyo 100-0013, Japan

TEL 81-3-5510-8590
FAX 81-3-3592-1103

Printed in Japan
All rights reserved