



ARIB TR-B33
Version 2.3-E1

ENGLISH TRANSLATION

OPERATIONAL GUIDELINES FOR TERRESTRIAL MOBILE MULTIMEDIA BROADCASTING BY TRANSMISSION SYSTEM BASED ON CONNECTED SEGMENTS FOR VHF-HIGH BAND

ARIB TECHNICAL REPORT

ARIB TR-B33 Version 2.3

(Fascicle 3)

Established on March 28th, 2001

Revised on September 30th, 2015 Version 2.3

Association of Radio Industries and Businesses

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

1. Notes on Copyright

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

2. Notes on English Translation

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

Foreword

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

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

This ARIB Technical Report is developed for the terrestrial television broadcasting and the terrestrial multimedia broadcasting by means of segment-connected transmission system using terrestrial basic stations through the use of radio waves with a frequency in the range of 207.5–222 MHz (VHF high band). In order to ensure fairness and transparency in the defining stage, the report was set by consensus at the ARIB Standard Assembly with the participation of both domestic and foreign interested parties from radio equipment manufacturers, telecommunication operators, broadcasting equipment manufacturers, broadcasters and users.

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

Integrated Contents

Foreword

Volume 0 Basic Concept and Common Elements of Multimedia Broadcasting	Fascicle1
Volume 1 Provisions for Download Operation for Multimedia Broadcasting	Fascicle1
Volume 2 Multimedia Broadcasting Function Specification for the Receiver ...	Fascicle1
Volume 3 Digital Terrestrial Television Broadcasting Multimedia Coding Regulation	Fascicle1

Foreword

Volume 4 Multimedia Broadcasting Provisions for PSI/SI Operations	Fascicle2
Volume 5 Multimedia Broadcasting Conditional Access System Operational Regulations Receiver Specifications	Fascicle2
Volume 6 (vacant number)	Fascicle2
Volume 7 Multimedia Broadcasting Provisions for Carrier Operations	Fascicle2
Volume 8 Multimedia Broadcasting Provisions for Contents Protection	Fascicle2

Foreword

Volume 9 (vacant number)	Fascicle3
Volume 10 Multimedia Broadcasting Metadata Operational Standard	Fascicle3
Volume 11 Multimedia Broadcasting Operation of Storable Broadcasting ...	Fascicle3
Volume 12 Compensation of Stored Content	Fascicle3

VOLUME 9

(vacant number)

VOLUME 10

Multimedia Broadcasting Metadata Operational Standard

Contents

Chapter 1 General Terms.....	1
1.1 Introduction	1
1.2 References	1
1.3 Terminology.....	3
Chapter 2 Outline for Using Metadata	9
2.1 Definition of Metadata.....	9
2.2 Metadata Description Language.....	9
2.3 Content Model and Metadata Model	9
2.4 Reference Model Related to Metadata	12
2.4.1 About metadata	14
2.4.2 Identification information.....	15
2.4.3 Authority and publisher.....	15
Chapter 3 Use of the Description Format of Description Language-type Metadata	17
3.1 Use of Metadata Character Coding.....	17
3.1.1 Character coding for metadata	17
3.1.2 Pictographic code elements.....	19
3.1.3 Use of tab, line feed, carriage return, and space	19
3.2 Use of Metadata Namespace.....	19
3.3 Use of Metadata Documents	19
3.4 Use of Content Description Metadata.....	23
3.4.1 Use of classification scheme elements.....	24
3.4.2 Use of program information elements.....	25
3.4.3 Use of group information elements	43
3.4.4 Use of review information elements.....	59
3.4.5 Use of purchase information elements.....	61
3.5 Use of Instance Description Metadata.....	69
3.5.1 Use of broadcast event information elements.....	69
3.5.2 Use of on-demand program information elements	72
3.5.3 Use of on-demand service information elements	73
3.5.4 Reference relationship between program information elements and instance description metadata.....	74
3.5.5 Use of instance metadata IDs.....	75
3.5.6 Use of service information elements	75
3.6 Use of License Reference Information Elements.....	79

3.6.1 License reference information elements	79
3.6.2 Information elements of license reference information elements	80
3.6.3 Use of license reference information elements	80
3.7 Use of Segmentation Metadata	85
3.7.1 Use of segment information elements	85
3.7.2 Use of segment group information elements	93
3.7.3 Reference relationship between program information elements and segmentation metadata	100
3.7.4 Use of segmentID and segment groupID	100
3.8 Use of Coupon Description Information	101
3.8.1 Coupon description information	101
3.8.2 Use of coupon description information	101
3.8.3 Reference relationship among the program information elements, item information elements, segment information elements, and coupon metadata	105
3.9 Others	107
3.9.1 Use of credit information elements	107
3.9.2 Correspondence between EIT and metadata	109
Chapter 4 Use of Metadata Encoding for Transmission	119
4.1 Coding Scheme of Metadata	119
4.2 Metadata Description Unit	119
4.3 File Format of Metadata	120
4.4 Use of Fragment Identification	120
4.4.1 Use of fragmentId	120
4.4.2 Use of fragmentVersion	121
4.4.3 Reuse of fragmentId	121
Chapter 5 Use of Metadata Transmission and Storage Control Method in Broadcasting	123
5.1 Use of Modules to Transmit Metadata	123
5.1.1 Metadata file format	123
5.2 Transmission at the Partial Reception Layer (Layer A)	124
5.3 Transmission Using a Layer Other Than a Partial Reception Layer (Layer B)	124
5.4 Guidelines for Organizing Metadata	124
5.4.1 Guidelines for organizing metadata at the partial reception layer (Layer A)	124
5.4.2 Guidelines for organizing metadata at a layer other than the partial reception layer (Layer B)	125
Chapter 6 Use of Metadata Distribution Methods via Communication	127
6.1 Overview of Metadata Distribution Methods	127

6.2 Protocols Used.....	127
6.3 Search Request	127
6.3.1 Overview of a search request.....	127
6.3.2 Search request details.....	128
6.3.3 Supplementary notes on search request description.....	138
6.4 Reception of the Search Result	145
6.4.1 Normal response	145
6.4.2 Error response.....	151
Chapter 7 Metadata Storage Control.....	153
7.1 Metadata Storage Control Method	153
7.1.1 Metadata storage guidelines.....	153
7.2 Adding/updating Metadata.....	153
7.3 Deleting Metadata	153
7.3.1 Expiration date of metadata.....	154
7.3.2 Notes on the expiration date of metadata.....	154
Chapter 8 Use of Content and Metadata Identification Information	155
8.1 Use of a Content Reference Identifier (CRID).....	155
8.1.1 Description format of the content reference identifier (CRID)	155
8.1.2 Use of authority (<authority>)	155
8.1.3 Use of data (<data>).....	156
8.2 Use of Location Resolution	157
8.2.1 Description format of locator	157
8.2.2 Transferring a locator	158
8.2.3 Guidelines for using location resolution.....	158
8.2.4 Layer resolution	161
Annex 1 Use of Classification Schemes.....	163
1 Common Items.....	163
2 Use of the ARIBParentalRatingCS Dictionary.....	165
3 Use of the HowRelatedCS Dictionary	170
4 Use of the FileFormatCS Dictionary	179
5 Use of the ISDBTMMContentTypeCS Dictionary.....	180
6 Use of the ISDBTMMDeliveryStatusCS Dictionary	182
7 Use of the ISDBTMMItemCategoryCS Dictionary	183
8 Use of the ISDBTMMIntendedAudienceCS Dictionary.....	185
9 Use of the ISDBTMMStereoVideoFormatCS Dictionary.....	186
10 Acquisition of a Classification Scheme and Management Guidelines.....	187

Annex 2 Metadata Usage Guidelines	189
1 Summary of Metadata Related Identifiers	189
Annex 3 Guidelines for Creating the EPG/ECG Using Metadata	191
1 Displaying EPG/ECG.....	191
1.1 EPG.....	191
1.2 ECG.....	192
2 Example of a Metadata Search	194
2.1 Example of retrieving metadata in a receiver without a large-capacity cache	195
2.2 Example of retrieving metadata in a receiver with a large-capacity cache.....	198
Annex 4 Metadata Operation Guidelines in Multimedia Broadcasting	201
1 Use of Furigana	201
2 Other Usage Details	202
2.1 Embedding links/images in an article.....	202
2.2 Description of various control information using Genre [<i>@type</i> ="other"]	202
2.3 Description of various control information using Keyword [<i>@type</i> ="other"]	204
2.4 Operation when multiple resources (multiple video image files) are included	213
Annex 5 BiM Encoding Rules	215

Chapter 1 General Terms

1.1 Introduction

Metadata services for multimedia broadcasting are provided based on the Ordinance and Notification of the Ministry of Internal Affairs and Communications (including the Ordinance and Notification of the former Ministry of Posts and Telecommunications) and the applicable standards of the Association of Radio Industries and Businesses (hereinafter referred to as “ARIB”). However, in order to widely use the applicable standards, detailed operational standards need to be specified separately. This is the reason for the creation of this volume.

The operational standard defined in this volume is intended to be the guideline for the signal transmission specification and receiver specification to correctly provide metadata services in multimedia broadcasting, premised on ensuring the extensibility to develop future metadata services.

This operational standard is applied to the specifications for sending metadata in multimedia broadcasting.

Sufficient attention should be paid to a multimedia broadcasting receiver so as to be able to receive the signals sent based on this operational standard and to prevent failure and other errors due to signals other than the specified signals being received.

1.2 References

The following documents are related to Volume 10.

- (1) ARIB STD-B10, “Service Information for Digital Broadcasting System” ARIB Standard
- (2) ARIB STD-B24, “Data Coding and Transmission Specification for Digital Broadcasting” ARIB Standard
- (3) ARIB STD-B38, “Coding, Transmission and Storage Specification for Broadcasting System Based on Home Servers” ARIB Standard
- (4) ARIB TR-B14, “Operational Guidelines for Digital Terrestrial Television Broadcasting” ARIB Technical Report
- (5) ARIB TR-B15, “Operating Guidelines for Digital Satellite Broadcasting” ARIB Technical Report
- (6) IETF RFC-1034, “Domain Names - Concepts and Facilities”
- (7) IETF RFC-1035, “Domain Names - Implementation and Specification”
- (8) IETF RFC-1738, “Uniform Resource Locators (URL)”
- (9) IETF RFC-2396, “Uniform Resource Identifiers (URI): Generic Syntax”
- (10) IETF RFC-2616, “Hypertext Transfer Protocol -- HTTP/1.1”
- (11) IETF RFC-2818, “HTTP Over TLS”
- (12) W3C-HTML 4.01, “HTML 4.01 Specification”

- (13) ETSI TS 102 822-3-1, “Phase-1 - Metadata Schema”
- (14) ETSI TS 102 822-3-3, “Phase-2 - Extended Metadata Schema”

1.3 Terminology

The terminology used in this standard is defined as follows.

ARIB	Association of Radio Industries and Business: The ARIB is an organization that standardizes technologies in relation to the use of radio in Japan, with participation by broadcasters, telecommunications operators, and equipment manufacturers.
AV resource	Resources that include video, audio and other data
BiM	Binary format for MPEG-7 XML binary coding scheme defined in MPEG-7
CRID	Content Reference Identifier: Name used to uniquely identify the available unit(s) of content
ECG	Electronic Content Guide; the method used to enable the presentation of the content information provided in multimedia broadcasting as well as the selection of content
EPG/ECG metadata	Description language-type metadata stipulated in ARIB STD-B38; this is used for performing content guides and content navigations.
EPG	Electronic Program Guide; method used to enable the presentation of program information, such as the timetable, and the selection of programs for real-time broadcasting services
HTTP	HyperText Transfer Protocol: An application layer protocol, which is used to transfer World Wide Web data (RFC2616)
HTTPS	Hyper Text Transfer Protocol Security: HTTP with the encryption function of SSL (Secure Sockets Layer)
MJD	Modified Julian Date: Calculated based on the method provided in “ARIB STD-B10 Part 2 Annex C” expressing the date in Japanese standard time
RMPI	Rights Management and Protection Information: This is the condition under which content is used.
URI	Uniform Resource Identifier; description system used to indicate the resource location; this concept includes the URL.
URL	Uniform Resource Locator: Description format indicating the location of a resource
W3C	World Wide Web Consortium: International standardization

	organization that develops protocols for providing and sharing information on the Internet
XML	Extensible Markup Language: One of the description languages established by W3C
XML Schema	A schema language that defines the structure of an XML document based on W3C standards
Event	A set of streams provided for the same service (organization channel) within the same time range (start and end time) in real-time broadcasting service, such as news and drama programs
Entry resource	A resource that needs to be started first when playing back an available unit of content
Authority	The unit of a namespace that ensures the uniqueness of the name of a reference identifier of an available unit of content, resource, group of available unit of content, segment, or segment group
On-demand program information element	A type of program location information element in program metadata; this is an element of the description language-type metadata that describes (stores) the location of an available unit of content obtained through communication.
On-demand service information element	A type of program location information element in program metadata; this element exists in the description language-type metadata that describes (stores) the service information related to the on-demand program information element.
Description language-type metadata	Metadata represented in XML format specified in ARIB STD-B38
Coupon information element	An element of description language-type metadata that describes (stores) the information related to coupons in program metadata
Conditional access system for playback	Conditional access system for playback in digital broadcasting
Conditional access system for reception	Conditional access system for reception in digital broadcasting
Service information element	An element of description language-type metadata that describes (stores) service-related information in program metadata
Business entities identifier	Code used to identify the service business entity for the operation of the Conditional Access System; this is included in the related information.

Merchandise information element	An element of the description language-type metadata that describes (stores) a billing package price, sales pattern, the information for referencing a corresponding license, and other information in program metadata
Real-time broadcasting content	Content that is distributed via broadcast waves in multimedia broadcasting mainly for viewing on a real-time basis; EMM allows whether or not real-time broadcasting content can be viewed. If the viewing of real-time broadcasting content is permitted by license, then the content can be viewed within the permission range. The content is decoded with a scramble key (Ks).
Segmentation metadata	Description language-type metadata that describes (stores) the segment information element and segment group information element in program metadata
Segment	A section that is divided by time in one video/audio resource that is included in an available unit of content; for example, a segment represents one news topic in a news video.
Segment group	A group of segments that are put together based on a specific purpose or common characteristic
Segment group information element	An element of description language-type metadata that describes (stores) segment group information in program metadata
Segment playback	Playback of content based on the description in segmentation metadata; a part or all of the content is played repeatedly.
Segment information element	An element of description language-type metadata that describes (stores) segment information in program metadata
Package	A minimum set of content or licenses that can be used in a single procedure, such as a service application
Program information element	An element of the description language-type metadata that describes (stores) the information related to an available unit of content in program metadata
Program metadata	Description language-type metadata that describes (stores) various information elements related to an available unit of content
Program location information element	An element of description language-type metadata that describes (stores) the information related to the location of an available unit of content in program metadata
Storable broadcasting	One of the terrestrial multimedia broadcasting services based on

	connected segment transmission, which is provided when downloading
Storable broadcasting content	Content that is distributed via broadcast waves in multimedia broadcasting mainly for use of the content after storing such in a terminal; the content can be used within the range permitted by the license granted to the stored content. The content is encrypted with a content key (Kc).
Transmission control metadata	An XML document that includes the necessary information for receiving and downloading storable broadcasting content
Fragment	A part of metadata description
Broadcast event information element	A type of program location information element in program metadata; this is an element of the description language-type metadata that describes (stores) the location of an available unit of content transmitted in broadcasting.
Metadata	The term “metadata” in this volume refers to the aforementioned “EPG/ECG metadata” (the description language-type metadata stipulated in ARIB STD-B38).
Multimedia broadcasting receiver	Receivers that can receive current digital broadcasting as well as multimedia broadcasting; unless otherwise specifically stated in this volume, “receiver” in this volume refers to a multimedia broadcasting receiver.
License reference information element	An element of description language-type metadata that describes (stores) license related information in program metadata
License	The information regarding usage conditions (RMPI) and the exploitation rights of the available unit of content; this also refers to the copyright data containing the usage conditions (RMPI) and the decryption key used to decode the encrypted content. The license ID is configured as information used to uniquely identify the license.
Available unit of content	Content for exploitation by users; unit content is comprised of one or more resources. This volume may simply describe the unit content as “content.”
Resource	The smallest reference unit and usage unit, which comprise the available unit of content; the same concept as the object, described in Volume 11; however, the description is not unified across different parts, toward understanding the related STD and TR in the same technical field.
Location resolution	To obtain the location (locator) of a substance of an available unit of

	content from the CRID (content reference identifier)
Locator	Temporal (e.g., broadcasting date and time) and spatial (e.g., a URL) information that signifies when an available unit of content is provided
Entrusted broadcaster	Same as “certified infrastructural broadcaster”
Entrusting broadcaster	Same as “infrastructural broadcasting provider”

<Blank Page>

Chapter 2 Outline for Using Metadata

2.1 Definition of Metadata

Description language-type metadata as defined in “3.1 in ARIB STD-B38” is used for metadata in multimedia broadcasting. Hereinafter, metadata described in this volume refers to description language-type metadata. “EPG/ECG metadata” described in other volumes also refers to description language-type metadata written in this volume.

Metadata in this operational standard is provided as data for referencing from a receiver application without any constraints. It is not assumed that metadata is protected when sent as a part of the content or that it is protected with the same encryption used for content when sent.

2.2 Metadata Description Language

The description language of metadata conforms to “3.1 Description Language-type Metadata in ARIB STD-B38.”

2.3 Content Model and Metadata Model

The content model indicates the processing unit of a resource or resource group to process the stored content for multimedia broadcasting in a receiver. The elements that make up the content model are available units of content, content groups that group multiple available units of content for a specific purpose, such as a package, AV resources (which are included in available units of content), segments, segment groups, purchase information, licenses, and coupons, etc.

On the other hand, the metadata model indicates a processing unit of metadata to process the stored content for multimedia broadcasting in a receiver. The description language-type metadata defined in “3.1 in ARIB STD-B38” is used for the metadata elements of the metadata model.

Figure 2-1 shows the classification of description language-type metadata information elements.

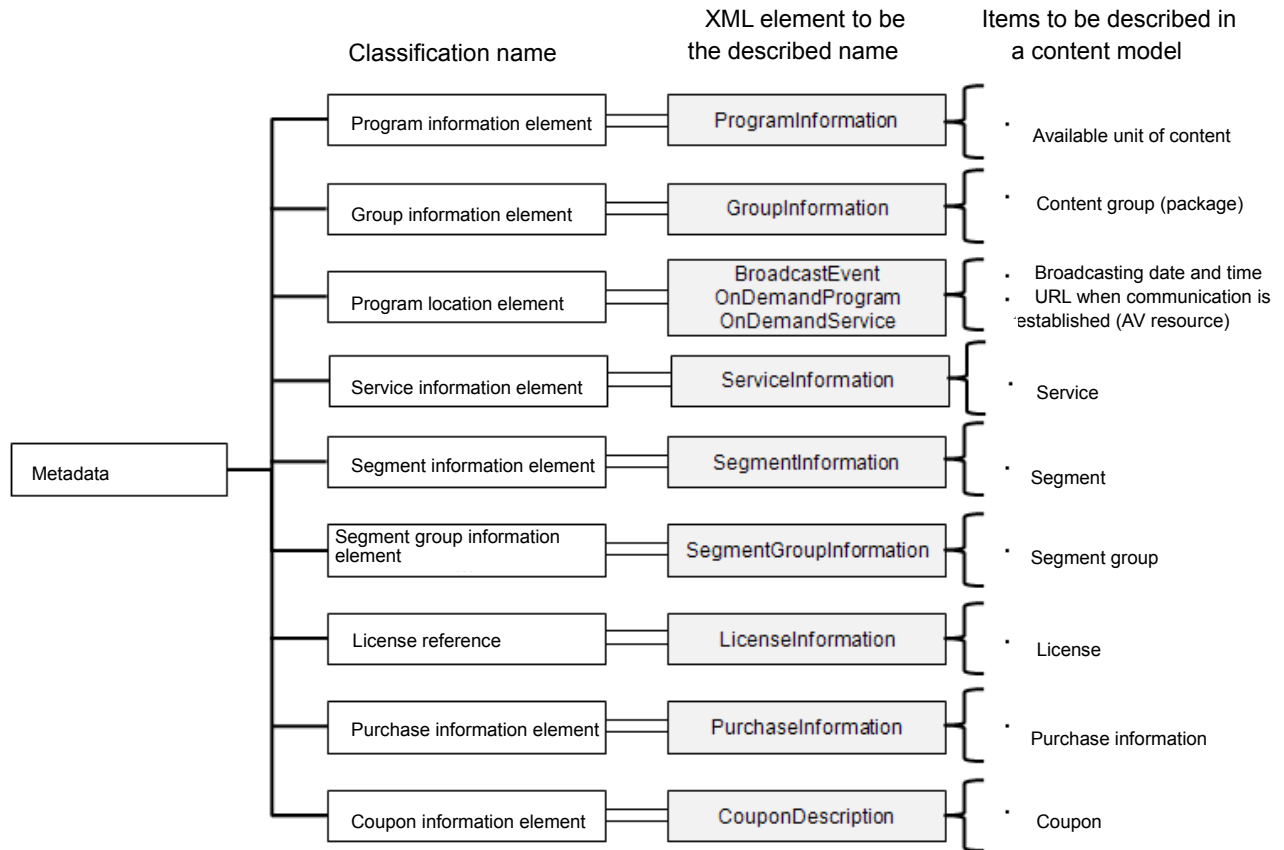


Fig. 2-1: Classification of description language-type metadata information elements

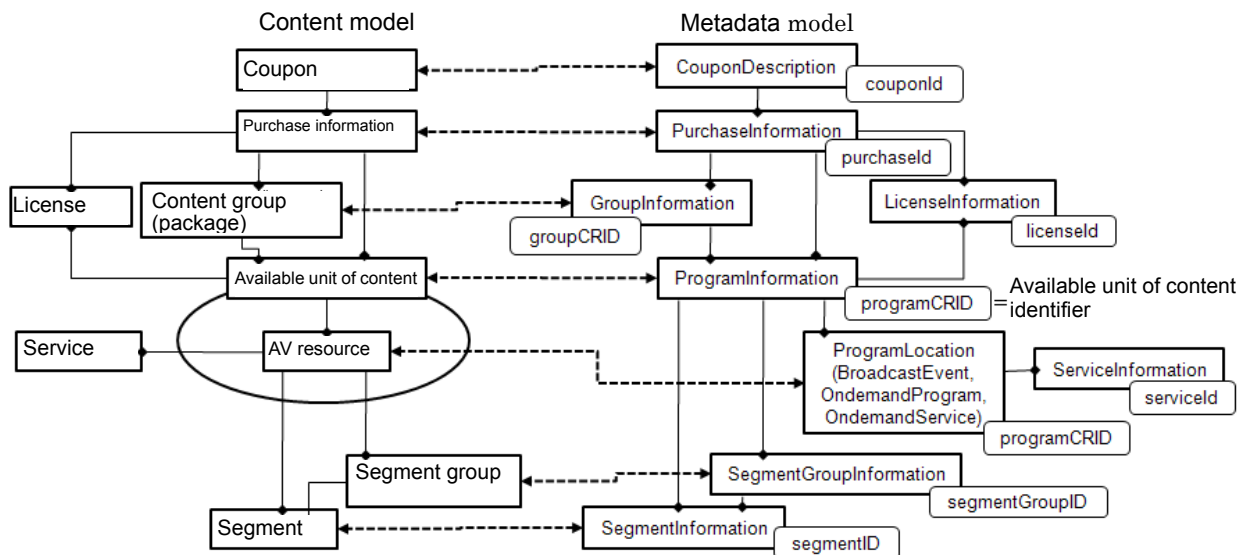


Fig. 2-2: Correspondence relationship between a content model and a metadata model

Figure 2-2 shows the correspondence relationship between each element of a content model and each element of a metadata model. To handle the content in a receiver, the metadata that corresponds to the content is obtained using the applicable identifier to use it for the operations

related to the billing of usage fees and the operations related to the resources to be provided.

The following table describes the detailed correspondence relationship between the content model and metadata model.

Table 2-1: Correspondence relationship between a content model and a metadata model

Content model	Metadata model	Identification method
Available unit of content	ProgramInformation	Indicates a logical unit used in a receiver application for presenting information to users, viewing, managing, and copyright protection; it is identified with the programCRID that is described in the programId attribute of the ProgramInformation element.
Location of available unit of content (AV resource, etc.)	ProgramLocation	Specifies the broadcasting transmission of an available unit of content, the location of the content within a communication site or in a terminal, etc.; this also indicates the physical substance of the available unit of the content (AV resource, etc.). This is identified by the programCRID of the available unit of content described in the Program of the ProgramLocation (BroadcastEvent, OnDemandProgram) element (or in combination with InstanceMetadataId).
Service	ServiceInformation	Holds the information related to the service (service name, service type, channels, etc.) and indicates the entire stream content; this service is identified by an identifier that is described in the serviceId attribute of the ServiceInformation element.
License	LicenseInformation	Represents the right to use the content and holds the license expiration date, license type, and license acquisition information; a license is identified by the LicenseID of the LicenseInformation element (or in combination with MainLicenseID). The LicenseInformation (license reference information element) does not have license privilege and is used to reference the license overview. One package can hold multiple content under a license. In addition, multiple licenses for one content can belong to a different package, respectively. In the CAS/DRM server, a license is issued appropriately to a receiver when viewing/using. In this case, the license does not refer to the meaning defined earlier in this section but refers to “the right to use” each piece of content that is obtained by a viewer when purchasing/contracting the service.

Purchase information	PurchaseInformation	<p>Indicates the billing unit of an available unit of content (group) that is defined in “2.4.1 Merchandise in Volume 5” of this operational standard; purchase information is identified by the identifier that is described in the purchaseId attribute of the PurchaseInformation element.</p> <p>If a single item of merchandise is purchased, the package information is described in the purchase information element (PurchaseInformation) and program information element (ProgramInformation). Otherwise, the package information is described in the purchase information element (PurchaseInformation) and group information element (GroupInformation). When describing the package information, write in information specific to the package type in the purchase information element and write in the information that is irrelevant to the package type in the program information element or group information element.</p>
Group	GroupInformation	<p>Indicates a set of multiple available units of content (series and packages); a group can be composed of two or more layers.</p> <p>A group is identified by the groupCRID described in the groupID attribute of the GroupInformation element.</p>
Segment	SegmentInformation	<p>Indicates one section (scene, etc.) divided by time in an AV resource; a segment is identified by the identifier that is described in the segmentId attribute of the SegmentInformation element.</p>
Segment group	SegmentGroupInformation	<p>A group of multiple segments that are gathered together; a group can be composed of only one layer. A segment group includes only segments. A segment group is identified by the identifier that is described in the segmentGroupId attribute of the SegmentGroupInformation element.</p>
Coupon	CouponDescription	<p>Indicates coupon information (discount, etc.); a coupon is identified by the identifier that is described in the couponId attribute of the CouponDescription element.</p>

2.4 Reference Model Related to Metadata

Figure 2-3 shows the reference model related to metadata.

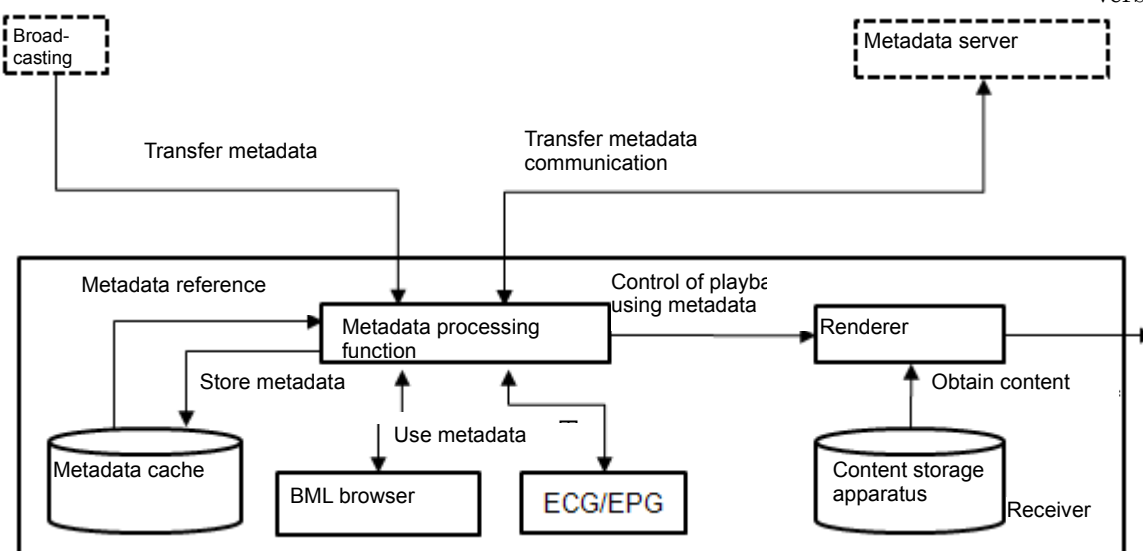


Fig. 2-3: Metadata system reference model

This reference model is created to match the standard defined in this operational specification based on the system reference model provided in “Appendix A1.3 in ARIB STD-B38.” Unless otherwise specifically specified in this specification, the configuration elements that are not defined in the following conform to “ARIB STD-B38.”

Table 2-2 shows the configuration elements of the reference model.

Table 2-2: Configuration elements of the metadata system reference model

Configuration element	Function
Metadata processing function	A receiver function that receives metadata, obtains distributed data, and manages locations
Metadata cache	Storage medium inside a receiver that (temporarily) stores metadata
EPG/ECG	A receiver function that provides user interfacing, such as content search and content display, using metadata (It can be assumed that this function is realized by a browser using metadata.)
Renderer	A receiver function to play back content based on a metadata description
Content storage apparatus	Storage medium inside a receiver that stores content
BML browser	Browser function that controls the entire BML-related operations

■ **Metadata transmission:**

Metadata is transmitted as a data event for metadata in the ES, which is transmitted in the partial reception hierarchy allocated to the central part of the 13 segments. Otherwise, metadata is transmitted as one storable piece of broadcasting content in the ES that transmits storable broadcasting content. The metadata description format is defined in “Chapter 3 Use of the Description Format of Description Language-type Metadata,” encoding for transmission is defined in “Chapter 4 Use of Metadata Encoding for Transmission,” and the transmission/storage control method during broadcasting is defined in “Chapter 5 Use of Metadata Transmission/Storage Control Method in Broadcasting.”

■ **Metadata communication transmission:**

Metadata is distributed via communication based on the standard defined in “Chapter 6 Use of Metadata Distribution Method via Communication.”

■ **Metadata storage:**

Metadata is stored/controlled based on the standard defined in “Chapter 7 Metadata Storage Control.”

■ **Metadata reference:**

Stored metadata is referenced using the metadata processing function of a receiver as needed. For details on how to manage metadata in a receiver, see “Chapter 4 User Interface Requirements” in Volume 2 of this operational standard.

■ **Using metadata:**

The applications that use metadata are the BML browser and the EPG/ECG of a receiver. For details on the extension function to use metadata from the BML content, see “3.4.10 Operation Concerning Procedure Description” in Volume 3 of this operational standard. For details on how to use metadata from the EPG/ECG, see Volume 2 of this operational standard.

■ **Control of playback using metadata:**

Content is played back/presented when an instruction to play back/use content is given from the metadata processing function of the EPG/ECG to a renderer. For the operations related to the activation of content, see Volume 2 of this operational standard. In order to control playback using the segment information element, a method to provide a segment ID is defined in “3.7.4 Use of segmentID and segment groupID” in this volume. For the playback control of content using the segment information element, see Volume 2 of this operational standard.

2.4.1 About metadata

Metadata described in this standard is transmitted separately from content based on the description in “Chapter 4 Use of Metadata Encoding for Transmission” and “Chapter 5 Use of Metadata Transmission and Storage Control Method in Broadcasting.” Metadata is also stored in a receiver in a certain format and used by the EPG/ECG of the receiver.

2.4.2 Identification information

The identifier that uniquely identifies an available unit of content in multimedia broadcasting is defined in “8.1 Use of a Content Reference Identifier (CRID).” An identifier is used for referencing content, and the content reference identifier (CRID) described in metadata is used as the base.

Metadata is used to associate an identifier with a namespace for transmission or an identifier with a namespace for storage. The use of location resolution is defined in “8.2 Use of Location Resolution.”

2.4.3 Authority and publisher

A publisher is a producer of metadata (TVAMain/@publisher in Table 3-2 in “3.3 Use of a Metadata Document”) and an authority (“8.1.2 Use of authority (<authority>”) and is used to mean both.

<Blank Page>

Chapter 3 Use of the Description Format of Description Language-type Metadata

3.1 Use of Metadata Character Coding

3.1.1 Character coding for metadata

The coded character sets for metadata is UCS (Unicode 2.0), and the character coding scheme is UTF-8 without BOM (Byte Order Mark). The coding scheme conforms to the basic character sets defined in "7.2 Universal Multi-octet Coded Character Set " in Part 2, Volume 1 of ARIB STD-B24. The code point conforms to the ISO/IEC 10646:2003 (including Revision 5 and 6).

As a receiver does not always support all of the UCS character sets, the subset defined in this section can be used to describe metadata for presentation.

Table 7-4 Kanji Set (1) to (8) in "Monomedia Coding" in Part 2, Volume 1 of ARIB STD-B24 is applied as the character range that the receiver is expected to have. In this case, among the special characters (additional symbols) defined in rows 90 to 94 in Table 7-4 (7) and (8), the special characters that are included in the shaded area in Table 3-1 are excluded. In addition, in Table 7-19 "Code Values for the Added Symbols Set," the characters that are not defined in ISO/IEC 10646:2003 are excluded (row 85, cells 39, 46, 47, 52, 67, 91, 93, and 94, and row 86, cells 1, 11, 13, 15, 18, 22, 25, 29, 37, and 40). Combining characters are not used.

If a character is applied that is not included in the range above, it is not guaranteed that a receiver can present it. If a receiver has received a character code other than one in the Basic Multilingual Plane, whether or not the receiver can present the character is determined based on the implementation. However, even if the character is not presented, it must be ignored properly.

All the byte numbers written in this chapter are based on UTF-8. The font is based on the sample fonts provided in Table 7-4 in Volume 1, Part 2 of ARIB STD-B24 (same as Table 1 in Attachment 3 in JIS X 0208:1997 = ISO/IEC 10646:2003 J0) and the fonts provided in Table 7-11 (Table 1 in JIS X 0213:2000 = ISO/IEC 10646: 2003 J3, J4).

Table 3-1: Additional symbols

To take into account compatibility with external systems, the character code specified in the ISO/IEC 10646: 2003 rather than the character code specified in “Table 7-19 Code Values for the Added Symbols Set” in “Monomedia Coding” in Volume 1, Part 2 of ARIB STD-B24 is used for the following characters.

Row/Cell	Character	B24	ISO/IEC 10646:2003
Row 85 cell 1	(崑)	E080	3402
Row 85 cell 10	(勢)	E082	351F
Row 85 cell 14	(詹)	E083	8A79
Row 85 cell 32	(崎)	E085	FA11
Row 85 cell 33	(寄)	E086	37E2
Row 85 cell 84	(珊)	E08F	3EDA
Row 85 cell 88	(規)	E090	4093
Row 86 cell 6	(籓)	E094	4264

In addition, in order to handle the following characters as the characters that are different from the characters in JIS X 0208, the character codes defined in the ISO/IEC 10646:2003 rather than the character codes specified in “Table 7-19 Code Values for the Added Symbols Set” in “Monomedia Coding” in Volume 1, Part 2 of ARIB STD-B24 are used for the following characters.

Row/Cell	Character	B24	ISO/IEC 10646:2003	Variant character of JIS X 0208
Row 85 cell 26	(塚)	585A	FA10	塚 (585A)
Row 85 cell 59	(海)	6D77	FA45	海 (6D77)
Row 85 cell 63	(渚)	6E1A	FA46	渚 (6E1A)
Row 85 cell 78	(琢)	7422	FA4A	琢 (7422)

The following character codes are assigned to the characters other than those in Basic Multilingual Plane based on Table 7-11 “Additional Kanji Characters” and Table 7-19 “Code Values for the Added Symbols Set” in “Monomedia Coding” in Volume 1, Part 2 of ARIB STD-B24—not on the character codes specified in the ISO/IEC 10646:2003.

Row/cell	Character	B24	ISO/IEC 10646:2003
Row 85 cell 2	(豊)	E081	20518
Row 85 cell 15	(吉)	E084	20BB7
Row 85 cell 53	(杷)	E08A	233CC

Row 85 cell 54	(棧)	E08B	233FE
Row 85 cell 55	(梳)	E08C	235C4
Row 85 cell 68	(熙)	E08E	242EE

3.1.2 Pictographic code elements

It can be assumed that an image is used as pictographs (animation GIF, etc.), and a document, which includes the IMG tag that references the image, is described in the CDATA section (see “2.1 Embedding Links/Images in an Article” in Annex 4). However, as this operation is dependent on the service, the receiver operation is based on the product design.

3.1.3 Use of tab, line feed, carriage return, and space

Tab (CHARACTER TABULATION: U+0009), line feed (LINE FEED (LF): U+000A), carriage return (CARRIAGE RETURN (CR): U+000D), and space (SPACE (SP): U+0020) are blank characters, among which tab cannot be used. A method for processing when a tab is used in character string data is determined based on the receiver. Whether the other three blank characters can be used or not is specified in the standard for each element in this chapter. A receiver processes line feed, carriage return, and line feed + carriage return as line feed code, with blanks as a space. Line feed codes must be processed properly in conformity with XML1.0. In this chapter, a byte string that expresses characters is used on the premise that it is encoded in UTF-8. Therefore, line feed, carriage return, and space are counted as one character and one byte. The end of a line in the character string data in metadata is a character string, which is composed of the two successive characters of carriage return and line feed that are presented in this order. They are counted as two characters and two bytes.

3.2 Use of Metadata Namespace

The namespace defined in “3.2.1 in ARIB STD-B38” is used for the namespace of the metadata description scheme. The schema defined in “Annex B.2 in ARIB STD-B38” is used as the metadata schema.

3.3 Use of Metadata Documents

Among the elements/attributes of metadata documents as defined in “3.2.10.1 in ARIB STD-B38,” the elements/attributes provided in Table 3-2 are used. The table elements of the ClassificationSchemeTable, ProgramInformationTable, GroupInformationTable, ProgramLocationTable, ServiceInformationTable, ProgramReviewTable, PurchaseInformationTable, LicenseInformationTable, and CouponDescriptionTable are stored in the metadata documents defined in this regulation.

The following categories are used for the description of the use field in tables in this chapter.

○: Mandatory or optional item

When a table element with this element/attribute is described, a mandatory or optional item always or may exist.

—: Item that is not to be used

This element/attribute is not used.

The number of appearances (use) indicates the number of appearances with conditions when the parent element of an applicable element/attribute appear.

Table 3-2: Use of metadata documents

Element/Attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
TVAMain					
@xml:lang	○	0..1	1	Fixed to “ja”	
@publisher	○	0..1	1	Maximum 127 bytes	Note 1
@publicationTime	—				
@rightsOwner	—				
@originID	—				
@version	—				
CopyrightNotice	○	0..1	0..1	Maximum 300 two-byte characters, maximum 900 bytes; SP can be used. CR and LF cannot be used.	Note 2 See Section 3.1.3.
MetadataOriginationInformationTable	—				
ClassificationSchemeTable	○	0..1	0..1		Note 3
@xml:lang	—				
ClassificationScheme	○	1	1		See Table 3-3.
ProgramDescription	○	0..1	1		
ProgramInformationTable	○	0..1	0..1		
@metadataOriginIDRef	—				
@xml:lang	—				
ProgramInformation	○	0-∞	1		See Table 3-4.
GroupInformationTable	○	0..1	0..1		
@metadataOriginIDRef	—				
@xml:lang	—				
GroupInformation	○	0-∞	1		See Table 3-26.
ProgramLocationTable	○	0..1	0..1		
@metadataOriginIDRef	—				
@xml:lang	—				
Schedule	—				
BroadcastEvent	○	0-∞	0..1		See Table 3-48.
OnDemandProgram	○	0-∞	0..1		See Table 3-50.
OnDemandService	○	0-∞	0..1		See Table 3-52.
ServiceInformationTable	○	○	0..1		
ServiceInformation	○	0-∞	1		See Table 3-53.

	CreditsInformationTable	—		Note 4	
	ProgramReviewTable	○	0-∞	0..1	
	@metadataOriginIDRef	—			
	+xml:lang	—			
	Review	○	0-∞	1..10	See Table 3-46.
	SegmentInformationTable	○	0..1	0..1	
	SegmentList	○	0..1	0..1	
	SegmentInformation	○	0-∞	1..100	See Table 3-56.
	SegmentGroupList	○	0..1	0..1	
	SegmentGroupInformation	○	0-∞	1..100	See Table 3-65.
	@metadataOriginIDRef	—			
	+xml:lang	—			
	PurchaseInformationTable	○	0..1	0..1	
	@metadataOriginIDRef	—			
	+xml:lang	—			
	PurchaseInformation	○	0-∞	1 Note 5	See Table 3-47.
	LicenseInformationTable	○	0..1	0..1	
	@metadataOriginIDRef	—			
	+xml:lang	—			
	LicenseInformation	○	1-∞	1 Note 5	See Table 3-55.
	CouponDescriptionTable	○	0..1	0..1	
	@metadataOriginIDRef	—			
	+xml:lang	—			
	CouponDescription	○	1-∞	1..N Note 6	See Table 3-72.
	UserDescription	—			

Note: A fragmented metadata document contains only one set of either “a set of ClassificationTable + ClassificationScheme,” “a set of ProgramInformationTable + ProgramInformation,” “a set of GroupInformationTable + GroupInformation,” “a set of ProgramLocationTable + BroadcastEvent,” “a set of ProgramLocationTable + OnDemandProgram,” “a set of ProgramLocationTable + OnDemandService,” “a set of ServiceInformationTable + ServiceInformation,” “a set of ProgramReviewTable + Review,” “a set of SegmentInformationTable + SegmentList/SegmentInformation,” “a set of SegmentInformationTable + SegmentGroupList/SegmentGroupInformation,” “a set of PurchaseInformationTable + PurchaseInformation,” “a set of LicenseInformationTable + LicenseInformation,” or “a set of CouponDescriptionTable + CouponDescription.” The main elements in the metadata document (CRID unit), which includes all fragments related to a program that is expressed using a specific CRID, must exist as the following set.

- 1 ProgramInformation element
- + 1 BroadcastEvent or OnDemandProgram or OnDemandService element
- + 0 or more ServiceInformation elements
- + 0 or more PurchaseInformation elements
- + 1 or more LicenseInformation elements
- + 0 or more SegmentList/SegmentInformation elements
- + 0 or more SegmentGroupList/SegmentGroupInformation elements
- + 0 or more Review elements
- + 0 or more CouponDescription elements

- 1 GroupInformation element
- + 0 or more PurchaseInformation elements
- + 0 or more CouponDescription elements

Note 1: A publisher and an authority (see “2.4.3 Authority and publisher”) mean the same thing.

Note 2: Copyright display for metadata itself; not used in a receiver

Note 3: The ClassificationSchemeTable is provided for CS definition. It is not returned as a metadata search result.

Note 4: The CreditsInformationTable is used only as an actor search result (getactor). It is not described in metadata. For details, see “6.4 Reception of Search Results.”

Note 5: If the transmission unit is a CRID unit (fragments that are connected with the same CRID are stored in a single TVAMain in PGSET format), then a maximum of 10 items can be transmitted.

Note 6: N is a value that makes one metadata document size remain within 1 Mb (1,048,576 bytes).

Figure 3-1 shows the reference relationship between fragmented metadata documents.

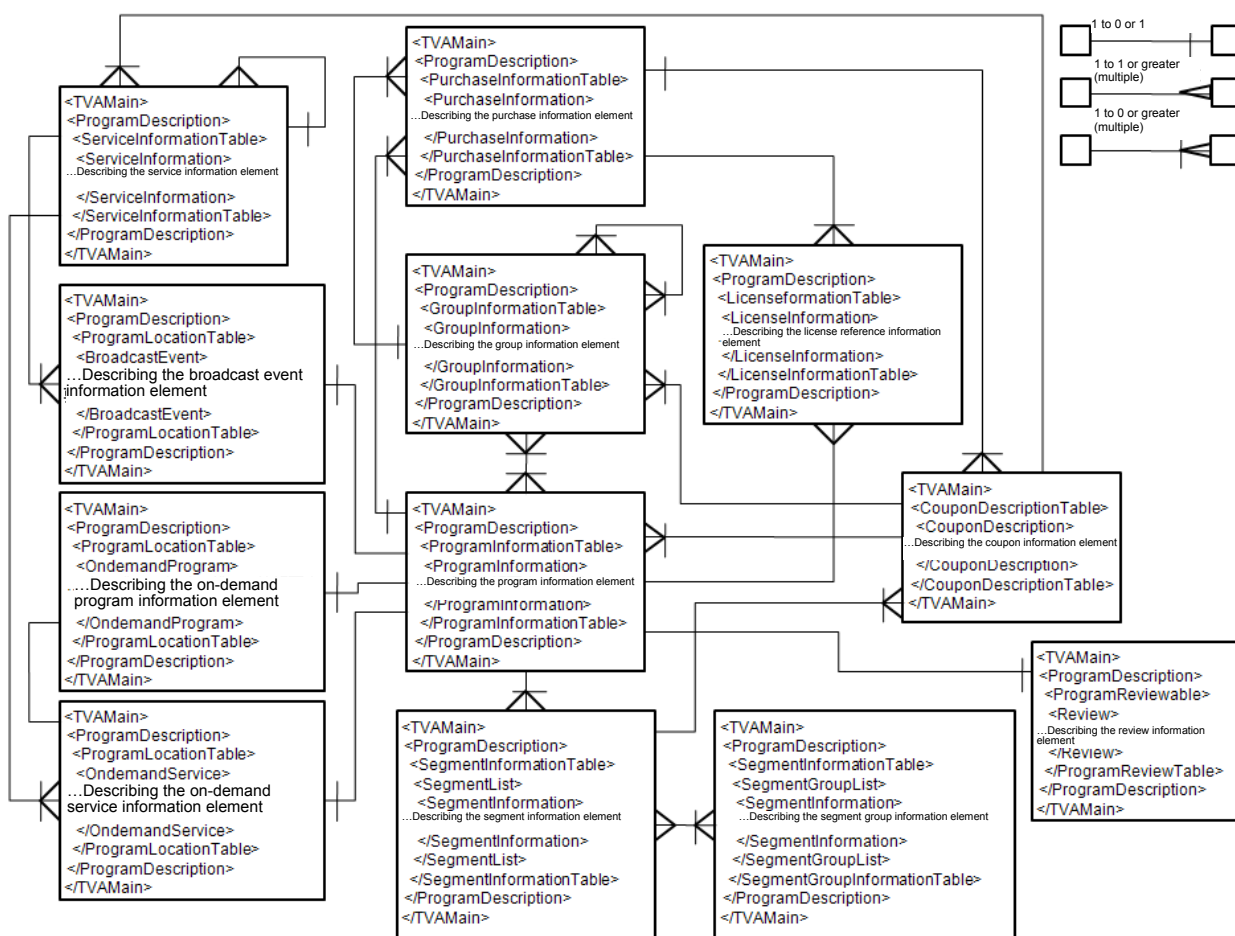


Fig. 3-1: Reference relationship between fragmented metadata documents

3.4 Use of Content Description Metadata

This section defines the use of the classification scheme element (ClassificationScheme), program information element (ProgramInformation), group information element (GroupInformation), review information element (Review), and purchase information element (PurchaseInformation), from

among the content description metadata defined in 3.2.3 in ARIB STD-B38.

3.4.1 Use of classification scheme elements

Among the elements/attributes defined in 3.2.3.1 in ARIB STD-B38, the elements/attributes shown in Table 3-3 are used.

Table 3-3: Use of ClassificationScheme

Element/attribute name		Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
ClassificationScheme						
	@fragmentId	○	0..1	1	10-digit hexadecimal integer value (10 characters)	Note 1 See 4.4.1.
	@fragmentVersion	○	0..1	1	14-digit decimal numerical value (14 characters)	Note 1 See 4.4.2.
	@fragmentExpirationDate	○	0..1	0..1	Specify in the YYYY-MM-DDThh:mm:ss+09:00 format (25 bytes).	See 7.3.1.
	@uri	○	1	1	Maximum 128 bytes	
	@domain	○	0..1	0..1	Maximum 128 bytes	
	mpeg7:Term	○	0-∞	1-1024		
	@termID	○	1	1	Maximum 16 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.
	mpeg7:Name	○	0-∞	0..3	Maximum 32 two-byte characters Maximum 96 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.
	@xml:lang	○	0..1	0..1	Note 2	
	mpeg7:Definition	○	0-∞	0..3	Maximum 128 two-byte characters Maximum 384 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.
	@xml:lang	○	0..1	0..1	Note 2	

Note 1: A ClassificationScheme that has the same URI and a different fragmentID does not exist. A URI remains the same if the fragmentVersion is updated when the fragmentId is the same.

Note 2: Select from “ja” (Japanese), “en” (English), “de” (German), “fr” (French) “it” (Italian), “ru” (Russian), “zh” (Chinese), “ko” (Korean), “es” (Spanish), and “xx” (a language other than those above, multiple languages, unknown language).

3.4.2 Use of program information elements

Among the elements/attributes defined in 3.2.3.6 in ARIB STD-B38, the elements/attributes shown in Table 3-4 are used.

Table 3-4: Use of ProgramInformation

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
ProgramInformation					
@programId	○	1	1	Maximum 128 bytes	Note 1 See 8.1.
@fragmentId	○	0..1	1	10-digit hexadecimal integer value (10 characters)	Note 1 See 4.4.1.
@fragmentVersion	○	0..1	1	14-digit decimal numerical value (14 characters)	Note 1 See 4.4.2.
@fragmentExpirationDate	○	0..1	0..1	Specify in the YYYY-MM-DDThh:mm:ss+09:00 format (25 bytes).	See 7.3.1.
@metadataOriginIDRef	—				
@xml:lang	—				
BasicDescription	○	1	1		See Table 3-5.
OtherIdentifier	○	0-∞	0..1		See Table 3-24.
AVAttributes	○	0..1	0..1		See Table 3-25.
MemberOf	○	0-∞	0-20		
@crid	○	1	1	Maximum 128 bytes	See 8.1.
@index	○	0..1	0..1	5-digit decimal numerical value (0 to 65535)	
@xsi:type	○	1	1	“MemberOfType” (Attach a namespace prefix if needed.)	
DerivedFrom	—				

	EpisodeOf	—			
	PartOfAggregatedProgram	—			
	AggregationOf	—			
	Period	○	0-∞	1..3	Note 3
	@type	○	0..1	1	Maximum 12 bytes Note 2
	Start	○	1	1	Specify in the YYYY-MM-DDThh:mm :ss+09:00 format (25 bytes). Note 4
	End	○	0..1	0..1	Specify in the YYYY-MM-DDThh:mm :ss+09:00 format (25 bytes). Note 4

Note 1: A ProgramInformation that has the same programId and a different fragmentID does not exist. A programId remains the same if the fragmentVersion is updated when the fragmentId is the same.

Note 2: Select from “display” (display duration), “availability” (distribution period), and “new_arrival” (new arrival period).

Note 3: Period[@type=“availability”]/Start (disclosure date and time) is mandatory.

Note 4: YYYY: 4-digit decimal western calendar notation, MM: 2-digit decimal month notation, DD: 2-digit decimal date notation, hh: 2-digit decimal time notation, mm: 2-digit decimal minute notation, ss: 2-digit decimal second notation

Table 3-5: Use of ProgramInformation/BasicDescription

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
BasicDescription					
Title	○	0-∞	2..7		See Table 3-6.
MediaTitle	○	0-∞	0..1		
TitleImage	○	0..1	0..1		See Table 3-7.
TitleVideo	○	0..1	0..1		See Table 3-8.
TitleAudio	—				
ShortTitle	—				
Synopsis	○	0-∞	0..3 Note 1)		See Table 3-9.
PromotionalInformation	○	0-∞	0..1		See Table 3-10.
Keyword	○	0-∞	0..15		See Table 3-11.
Genre	○	0-∞	0..13		See Table 3-12.
ParentalGuidance	○	0..1	1		See Table 3-13.
Language	○	0-∞	0..4		See Table 3-14.
CaptionLanguage	○	0-∞	0..2		See Table 3-15.
SignLanguage	○	0-∞	0..1		See Table 3-16.
CreditsList	○	0..1	0..1		See Table 3-17.

AwardsList	—				
RelatedMaterial	○	0-∞	0..50		See Table 3-18.
ProductionDate	○	0..1	0..1		See Table 3-19.
ProductionLocation	—				
CreationCoordinates	—				
DepictedCoordinates	—				
ReleaseInformation	○	0-∞	0..1		See Table 3-20.
Duration	○	0..1	0..1	Specified in the PThhHmmMssS (xsd: duration) format Note 2	
PurchaseList	○	0..1	0..1		See Table 3-21.
ContentProperties	○	0..1	0..1		See Table 3-22.

Note 1: 0 to 3 (three types: short, medium, and long) when the content type is storable broadcasting content; 0 to 18 (short, medium, long*16) when the content type is real-time broadcasting content; in this case, short-format EIT (80 characters) is mapped to medium, and extension format (item name [8 characters]: item description [200 characters]) is mapped to long.

Note 2: P: A character that indicates a lapse of time, T: a character that separates date and time, hhH: elapsed 2-digit 10 decimal hour, mmM: elapsed 2-digit decimal minute, ssS: elapsed 2-digit decimal second

Table 3-6: Use of ProgramInformation/BasicDescription/Title

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
Title				Maximum 128 two-byte characters Maximum 384 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.
@xml:lang	—				
@phoneticTranscription	—				
@phoneticAlphabet	—				
@type	○	0..1	1	Select from “main,” “secondary,” “alternative,” “popular,” “seriesTitle,” and “episodeTitle.” Maximum 12 bytes	

Note: The title element that has the @type specified as “main” is mandatory, and only one such title element exists. The title element that has the @type specified as “alternative” is also mandatory, and only one such title element exists. A maximum of one title element can have the @type element that is specified as “secondary.” A maximum of three title elements can have the @type that is specified as “popular.” A title character string for presentation is described in the @type specified as “main.” A subtitle character string is described in the @type specified as “secondary.” The ruby of the title character string is described in the @type specified as

“alternative,” which is used as a search key and a sort key for sorting. The @type specified as “popular” is the attribute that is specified when there are two or more rubies, which is used not as a sort key but as a search key. Whether or not ruby as a ruby of a title character string is described in the @type specified as “main” and the @type specified as “secondary” is determined based on the operation. The processing of a title when the @type is specified as “seriesTitle” and the @type is specified as “episodeTitle” is determined based on the implemented receiver. For details on ruby, see “1 Use of *Furigana*” in Annex 4.

Table 3-7: Use of ProgramInformation/BasicDescription/MediaTitle/TitleImage

Element/attribute name		Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
TitleImage						
	Select ① or ②.	○	0..1	1		
①	mpeg7:MediaUri	○	1	1	Maximum 128 bytes	
②	mpeg7:InlineMedia	—				
	mpeg7:StreamID	—				
	CopyrightNotice	○	0-∞	0..1	Maximum 50 two-byte characters Maximum 150 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.

Note 1: The TitleImage specifies a still picture that is used for displaying a thumbnail. This still picture is not encrypted. When displaying a thumbnail picture, it can be displayed by scaling within the extent that the impression of the original picture is not largely altered. However, the aspect ratio shall be maintained, and the picture shall not be trimmed.

Note 2: See “8.2.3.3 Location resolution in a receiver” for the correspondence relationship between the reference information of each resource described in metadata (e.g., <http://example.jp/a/v/c.png>) and the storage locations of an applicable resource on a broadcasting stream and in a receiver.

Table 3-8: Use of ProgramInformation/BasicDescription/MediaTitle/TitleVideo

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
TitleVideo					
Select ① or ②.	○	0..1	1		
① mpeg7:MediaUri	○	1	1	Maximum 128 bytes	
② mpeg7:InlineMedia	—				
mpeg7:StreamID	—				
CopyrightNotice	○	0-∞	0..1	Maximum 50 two-byte characters Maximum 150 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.

Note: The TitleVideo specifies a piece of content that is used for displaying a preview, etc. The locator of video content (e.g., <http://example.jp/a/b/c.mp4>) and the CRID of a preview content (e.g. <http://example.jp/trailer/content1>) are described in MediaUri.

Table 3-9: Use of ProgramInformation/BasicDescription/Synopsis

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
Synopsis				Maximum 408 two-byte characters Maximum 1224 bytes SP, CR, and LF can be used.	Note 1 See Section 3.1.3.
@xml:lang	—				
@phoneticTranscription	—				
@phoneticAlphabet	—				
@length	○	0..1	0..1	Select from “short,” “medium,” and “long.” Maximum 6 bytes	

Note 1: Maximum 25 two-byte characters and maximum 75 bytes when “short” is selected for @length; maximum 100 two-byte characters and maximum 300 bytes when “medium” is selected for @length; maximum 408 two-byte characters and maximum 1224 bytes when “long” is selected for @length

Table 3-10: Use of ProgramInformation/BasicDescription/PromotionalInformation

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
PromotionalInformation				Maximum 200 two-byte characters Maximum 600 bytes SP, CR, and LF can be used.	See Section 3.1.3.
@href	○	0..1	0..1	Select from the ARIBPromotionalType CS. Maximum 128 bytes	See 1 in Annex 1.
@xml:lang	—				
@phoneticTranscription	—				
@phoneticAlphabet	—				

Table 3-11: Use of ProgramInformation/BasicDescription/Keyword

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
Keyword				Maximum 40 two-byte characters Maximum 120 bytes SP can be used CR and LF cannot be used	See Section 3.1.3.
@xml:lang	—				
@phoneticTranscription	—				
@phoneticAlphabet	—				
@type	○	0..1	0..1	Select from “main,” “secondary,” and “other (supplemental).” Only one keyword can be selected as “main.”	Note 1

Note 1: In the ECG, a keyword to which “main” or “secondary” is specified can be presented; however, a keyword to which “other” is specified should not be used for presentation. If a type is not specified, then the attribute is regarded as “main.” A type can be omitted when one keyword is used. However, when multiple keywords are used, a type must be described.

Table 3-12: Use of ProgramInformation/BasicDescription/Genre

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
Genre					
@href	○	1	1	Select from the ARIBGenreCS, ARIBGenreCS2 (when “main” or “secondary” is selected for the @type), and another CS (when “other” is selected for the @type, see 2.2 in Annex 4). Maximum 128 bytes	See 1 in Annex 1.
@type	○	0..1	0..1	Select from “main,” “secondary,” and “other.” Only one genre can be selected as “main.”	Note 1
Name	—				
Definition	—				

Note 1: If only one genre is used/presented using a content list presenting method, such as ECG and EPG, priority is given to a genre that is specified as “main.” If there is only one Genre element, the @type can be omitted. However, if there are multiple Genre elements, the @type must be described. If the ARIBGenreCS or ARIBGenreCS2 is specified, the first one is specified as main and the second and subsequent are specified as secondary. “Other” can be specified for any CS other than the CSs stated earlier (see 2.2 in Annex 4). In this case, the processing is dependent on the implemented receiver.

Table 3-13: Use of ProgramInformation/BasicDescription/ParentalGuidance

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
ParentalGuidance					
Select ① or ②.	○	1	1		
① mpeg7:ParentalRating	○	1	1		
@href	○	1	1	Select from the ARIBParentalRatingCS. Maximum 128 bytes	See 2 in Annex 1.
mpeg7:Name	—				
mpeg7:Definition	—				
② mpeg7:MinimumAge	—				
mpeg7:Region	—				

Table 3-14: Use of ProgramInformation/BasicDescription/Language

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
Language				Select from “ja” (Japanese), “en” (English), “de” (German), “fr” (French), “it” (Italian), “ru” (Russian), “zh” (Chinese), “ko” (Korean), “es” (Spanish), and “xx” (a foreign language other than above, multiple languages, unknown language).	Corresponds to the language names described in ISO_639_language_code and ISO_639_language_code_2 in “Table 31-62 Audio component descriptor” in ARIB TR-B14
@type	—				
@supplemental	—				

Table 3-15: Use of ProgramInformation/BasicDescription/CaptionLanguage

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
CaptionLanguage				Select from “ja” (Japanese), “en” (English), “de” (German), “fr” (French), “it” (Italian), “ru” (Russian), “zh” (Chinese), “ko” (Korean), “es” (Spanish), and “xx” (a foreign language other than above, multiple languages, unknown language).	Corresponds to the language names described in ISO_639_language_code and ISO_639_language_code_2 in “Table 31-62 Audio component descriptor” in ARIB TR-B14
@closed	○	0..1	1		
@supplemental	—				
@description	○	0..1	0..1	Maximum 33 bytes SP can be used. CR and LF cannot be used.	Mapping the data_content_descriptor/text in ARIB-SI (See 3.9.2) See Section 3.1.3.

Table 3-16: Use of ProgramInformation/BasicDescription/SignLanguage

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
SignLanguage				Select from “ja” (Japanese), “en” (English), “de” (German), “fr” (French), “it” (Italian), “ru” (Russian), “zh” (Chinese), “ko” (Korean), “es” (Spanish), and “xx” (a foreign language other than above, multiple languages, unknown language).	Describes sign language
@primary	○	0..1	0..1		
@translation	○	0..1	0..1		
@type	—				

Table 3-17: Use of ProgramInformation/BasicDescription/CreditsList

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
CreditsList					
CreditsItem	○	0~∞	1..20		
@role	○	1	1	Select from the ARIBRoleCS and the business operator's original RoleCS. Maximum 128 bytes	See 1 in Annex 1.
Select from ①, ⑥, ⑦, and ⑧.	○	0~∞	1..5		
① PersonName	○	1	1		
@dateFrom	—				
@dateTo	—				
@type	○	0..1	1	Select from “main,”	Only one element

									“variant,” and “former.” Maximum 7 bytes	can have the @type specified as “main.” Note 1
				@xml:lang	—					
				Select from ② to ⑤.	○	1-∞	1			
			②	mpeg7:Given Name	○	1	1	Maximum 60 two-byte characters Maximum 180 bytes SP can be used. CR and LF cannot be used.	Describe the full name. See Section 3.1.3.	
				@xml:lang	—					
				@initial	—					
				@abbrev	○	0..1	0..1	Maximum 16 two-byte characters Maximum 48 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.	
			③	mpeg7:Family Name	—					
			④	mpeg7:Title	—					
			⑤	mpeg7:Numer ation	—					
		⑥	PersonNameIDRef		—					
		⑦	OrganizationName		—					
		⑧	OrganizationNameI DRef		—					
			Character		○	0-∞	0..1			
				@dateFrom	—					
				@dateTo	—					
				@type	—					
				@xml:lang	—					
				Select from ⑨ to ⑫.	○	1-∞	1			
			⑨	mpeg7:GivenNa me	○	1	1	Maximum 60 two-byte characters Maximum 180 bytes SP can be used. CR and LF cannot be used.	Describe the full name. See Section 3.1.3.	
				@xml:lang	—					
				@initial	—					

			@abbrev	○	0..1	0..1	Maximum 16 two-byte characters Maximum 48 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.
		⑩	mpeg7:FamilyName	—				
		⑪	mpeg7:Title	—				
		⑫	mpeg7:Numeration	—				

Note 1: Only one PersonName element can have the @type that is specified as “main” with a credit character string for presentation described. Only one PersonName element can have the @type that is specified as “variant” with the ruby of the credit character string described. This attribute is used as a search key and a sort key for sorting. A maximum of three PersonName elements can have the @type that is specified as “former.” The @type specified as “former” is the attribute that is specified when there are two or more rubies, which is used not as a sort key but as a search key. For details on ruby, see “1 Use of Furigana” in Annex 4.

Table 3-18: Use of ProgramInformation/BasicDescription/RelatedMaterial

Element/attribute name		Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
RelatedMaterial						
	HowRelated	○	0..1	1		
	@href	○	1	1	Select from the ARIBHowRelatedCS, ISDBTMMHowRelatedCS, and the business operator's original HowRelatedCS. Note 1 Maximum 128 bytes	See 3 in Annex 1.
	Name	○	0..1	0..1	Maximum 32 two-byte characters Maximum 96 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.
	Definition	—				
	Format	○	0..1	0..1		

	@href	○	1	1	Select from the ARIBFileFormatCS, ISDBTMMFileFo rmatCS, and the business operator's original FileFormatCS. Note 1 Maximum 128 bytes	See 4 in Annex 1.
	Name	—				
	Definition	—				
	Select ① or ④.	○	1	1		
①	MediaLocator	○	1	1		
	Select ② or ③.	○	0..1	1		
	② mpeg7:MediaUri	○	1	1	Maximum 128 bytes	Note 2
	③ mpeg7:InlineMedia	—				
	mpeg7:StreamID	—				
④	SegmentReference	—				
	PromotionalText	○	0-∞	0..1	Maximum 3000 bytes SP, CR, and LF can be used.	Note 3 See Section 3.1.3.
	@xml:lang	—				
	@phoneticTranscription	—				
	@phoneticAlphabet	—				
	PromotionalMedia	○	0-∞	0..1		Note 4
	TitleImage	○	0..1	1		See Table 3-7.
	TitleVideo	—				
	TitleAudio	—				
	SourceMediaLocator	—				

Note 1: If the business operator's original CS is specified, the processing in the receiver is dependent on the receiver.

Note 2: This attribute specifies the reference URI. The URL of a content or browser script file is specified. The relationship with a reference is specified using a dictionary item of the HowRelatedCS in the HowRelated/@href. The same dictionary item (relationship) can be specified for multiple different references.

Note 3: Describe the catch copy, copyright, etc., for the substance specified in the MediaLocator/mpeg7: MediaUri for sales promotion.

Note 4: Specify a thumbnail still picture file for the substance specified in the MediaLocator/mpeg7: MediaUri.

Table 3-19: Use of ProgramInformation/BasicDescription/ProductionDate

Element/attribute name		Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
ProductionDate						
	TimePoint	○	1	1	Describe in the YYYY-MM-DD or YYYY format. Maximum 10 bytes	
	Duration	—				
	YearRange	○	0..1	0..1	Describe in the YYYY-YY or YYYY-YYYY format. Maximum 9 bytes	Note 1

Note 1: When the YearRange is specified, YearRange rather than TimePoint is used for displaying the production year.

Table 3-20: Use of ProgramInformation/BasicDescription/ReleaseInformation

Element/attribute name		Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
ReleaseInformation						
	ReleaseDate	○	0..1	0..1		
	Select ① or ②.	○	1	1		
①	DayAndYear	○	1	1	Specify in the YYYY-MM-DD format.	
②	Year	○	1	1	Specify in the YYYY format.	
	ReleaseLocation	○	0..1	0..1	Specify with a 2-character country code as defined in “ISO3166-2.”	

Table 3-21: Use of ProgramInformation/BasicDescription/PurchaseList

Element/attribute name	Use	Number of appearances (B38)	Number of appearances	Character number limit Value limit	Remarks
PurchaseList					
Select ① or ②.	○	0-∞	0..10		
①PurchaseItem	—				
②PurchaseIdRef	○	1	1	Describe a purchase identifier. Maximum 24 bytes	

Table 3-22: Use of ProgramInformation/BasicDescription/ContentProperties

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
ContentProperties					
ContentType	○	0..1	0..1		
@href	○	1	1	Select from the ISDBTMMContentTypes and the business operator's original ContentTypeCS. Maximum 128 bytes	See 5 in Annex 1.
FileProperties	○	0..1	0..1		
FileFormat	○	0..1	0..1		
@href	○	1	1	Select arbitrarily from the ARIBFileFormatCS, ISDBTMMFileFormatCS, and the business operator's original FileFormatCS. Maximum 128 bytes	See 4 in Annex 1.
FileSize	○	0..1	0..1	10-digit decimal integer value, 10 characters	Describe in a byte unit.
FileChecksum	—				
ContentAttributes	○	0-∞	0..10		See Table 3-23.
ContextAttributes	—				

Table 3-23: Use of ProgramInformation/BasicDescription/ContentProperties/ContentAttributes

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
ContentAttributes					
@xsi:type	○	1	1	“StillImageContentAttributesType”	Note 1
Width	○	0..1	0..1	5-digit decimal integer value, 5 characters	Describe in a pixel unit.
Height	○	0..1	0..1	5-digit decimal integer value, 5 characters	Describe in a pixel unit.
Orientation	○	0..1	0..1	0 to 360	Express the rotation component of the original image in a clockwise rotation angle.

Note 1: The content information of audio and video content is defined in ProgramInformation/AVAttributes. Therefore, “tva2: AudioAttributes” and “tva2: VideoAttributes” cannot be specified for the @xsi: type.

Table 3-24: Use of ProgramInformation/OtherIdentifier

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
OtherIdentifier				Maximum 9 bytes SP, CR, and LF cannot be used.	See Section 3.1.3.
@type	○	0..1	1	“VideoID” Maximum 10 bytes	Note 1
@organization	—				
@authority	—				
@encoding	—				

Note 1: If the @type is specified as “VideoID,” it indicates the VideoID of the information described for an applicable ProgramInformation. A 9-digit or less decimal integer value that is unique for a business operator is specified.

Table 3-25: Use of ProgramInformation/AVAttributes

Element/attribute name		Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
AVAttributes						
	FileFormat	○	0..1	0..1		
	@href	○	1	1	Select arbitrarily from the ARIBFileFormatCS, ISDBTMMFileFormatCS, and the business operator's original FileFormatCS. Maximum 128 bytes	See 4 in Annex 1.
	FileSize	○	0..1	0..1	Maximum 10 bytes	Specify in a byte unit.
	System	—				
	BitRate	○	0..1	0..1	Decimal notation Maximum 20 digits Without prefix 0	When CBR is used, a constant bit rate is specified in a bit unit/seconds. When VBR is used, an average bit rate is specified in a bit unit/seconds.
	@variable	—				
	@minimum	—				
	@average	—				
	@maximum	—				
	AudioAttributes	○	0-∞	0..2		Note 1
	Coding	○	0..1	0..1		
	@href	○	1	1	Select from the ARIBAudioCodingFormatCS. Maximum 128 bytes	See 1 in Annex 1.
	Name	—				
	Definition	—				
	NumOfChannels	○	0..1	0..1	2-digit decimal numerical value (2 characters) without prefix 0	
	MixType	○	0..1	0..1		

	@href	○	1	1	Select arbitrarily from the ARIBAudioComponentCS. Maximum 128 bytes	See 1 in Annex 1.
	Name	—				
	Definition	—				
	AudioLanguage	○	0..1	0..2 Note 2)	Select from “ja” (Japanese), “en” (English), “de” (German), “fr” (French), “it” (Italian), “ru” (Russian), “zh” (Chinese), “ko” (Korean), “es” (Spanish), and “xx” (a foreign language other than above, multiple languages, unknown language).	Corresponds to the language names described in ISO_639_language_code and ISO_639_language_code_2 in “Table 31-62 Audio component descriptor” in ARIB TR-B14
	@type	○	0..1	0..1	Select “original,” “dubbed,” or “background.” Maximum 10 bytes	If the @type is not specified, it is regarded as “original.”
	@channel	○	0..1	0..1	Select from “main,” “sub,” “alternate,” and “other.” Maximum 9 bytes	When dual mono stereo or dual stereo is used, the first audio (main) and the second audio (sub) are distinguished from each other. If a type is not specified, then the attribute is regarded as “main.” “Alternate” and “other” are not used.
	@description	○	0..1	0..1	Maximum 33 bytes (character string) SP can be used. CR and LF cannot be used.	Mapping the audio_component_descriptor/text in ARIB-SI See Section 3.1.3.
	StreamDescription	○	0..1	0..1	Maximum 15 two-byte characters Maximum 45 bytes SP can be used.	Content for presentation, such as “main audio,” “sub audio” and “audio

					CR and LF cannot be used.	multiplex,” each of which represents an ES unit, is described. See Section 3.1.3.	
		@type	○	0..1	0..1	Select from “main,” “sub,” “alternate,” and “other.”	For distinguishing an ES stream; if a type is not specified, then the attribute is regarded as “main.” “Alternate” and “other” are not used.
		@id	○	0..1	1	4-digit hexadecimal without a prefix	Mapping the audio_component_descriptor/component_tag in ARIB-SI
		BitRate	○	0..1	0..1	Specify in a bit unit/seconds. Maximum 20-character decimal, without prefix 0	Mapping the audio_component_descriptor/quality_indicator in ARIB-SI
		SamplingRate	○	0..1	0..1	Maximum 8 bytes. Describe in Hz units.	The type is changed in this operational standard. Mapping the audio_component_descriptor/sample_rate in ARIB-SI
		VideoAttributes	○	0..1	0..1		Note 3
		Coding	○	0..1	0..1		
		@href	○	1	1	Select from the ARIBVisualCodingFormatCS. Maximum 128 bytes	See 1 in Annex 1.
		Name	—				
		Definition	—				
		Scan	○	0..1	0..1	Specify either “interlaced” or “progressive.”	Note 4
		HorizontalSize	○	0..1	0..1	Specify in a pixel unit. Decimal without prefix 0	Note 4
		VerticalSize	○	0..1	0..1	Specify in a pixel unit.	Note 4

					Decimal without prefix 0	
	AspectRatio	○	0..2	0..1	Select either “4:3” or “16:9.”	Note 4
	@type	—				
	Color	○	0..1	0..1		
	@type	○	1	1	Specify either “color” or “blackAndWhite.”	
	StreamDescription	○	0..1	0..1	Maximum 15 two-byte characters Maximum 45 bytes SP can be used. CR and LF cannot be used.	Describe a video type name, such as “video,” in ES units. See Section 3.1.3
	@type	○	0..1	0..1	Select from “main,” “sub,” “alternate,” and “other.”	For distinguishing an ES stream; if a type is not specified, then the attribute is regarded as “main.” “Alternate” and “other” are not used.
	@id	○	0..1	1	4-digit hexadecimal without a prefix	Mapping the component_descriptor/ component_tag in ARIB-SI
	StereoVideoFormat	○	0..1	0..1		Describe when the content is a 3D video.
	@href	○	1	1	Select from the ISDBTMMStereoVideoFormatC. Maximum 128 bytes	See 9 in Annex 1.
	Name	—				
	Definition	—				

Note 1: See 3.9.2 for the correspondence relationship with the audio_component_descriptor described in ARIB-SI.

Note 2: As for AudioLanguage, describe the language information of each audio track in the AudioAttributes in which technical attributes, such as coding scheme for audio, are described. The number of repetitions is two, in order to support 1ES in which the left and right channels use different languages (dual mono stereo, bilingual, and multiplex broadcasting). When dual mono stereo is used, describe a language for the left channel first.

Note 3: See 3.9.2 for the correspondence relationship with the component_descriptor defined in ARIB-SI.

Note 4: Select from the parameters defined in “2.1 Coding Scheme of Real-time Broadcasting Content” and “2.2 Coding Scheme of Storable Broadcasting Content” in Volume 7 of this operational standard.

3.4.3 Use of group information elements

Among the elements/attributes defined in “3.2.3.7 Group information” in ARIB STD-B38, the elements/attributes provided in Table 3-26 are used.

Table 3-26: Use of GroupInformation

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
GroupInformation					
@groupId	○	1	1	Maximum 128 bytes	Note 1 See 8.1.
@ordered	○	0..1	0..1	Select “true” or “false.”	
@numOfItems	○	0..1	0..1	5-digit decimal numerical value (0 to 65535)	
@fragmentId	○	0..1	1	10-digit hexadecimal integer value (10 characters)	Note 1 See 4.4.1.
@fragmentVersion	○	0..1	1	14-digit decimal numerical value (14 characters)	Note 1 See 4.4.2.
@fragmentExpirationDate	○	0..1	0..1	Specify in the YYYY-MM-DDThh:mm:ss+09:00 format (25 bytes).	See 7.3.1.
@metadataOriginIDRef	—				
@xml:lang	—				
GroupType					
@xsi:type	○	1	1	“ProgramGroupTypeType” (Attach a namespace prefix if needed.)	
@value	○	1	1	Select from “series,” “show,” “otherCollection,” “package,” “programConcept,” “programCompilation,” and “otherChoice.” Maximum 20 bytes	Note 2
BasicDescription	○	1	1		See Table 3-27.
MemberOf	○	0-∞	0-20		
@crId	○	1	1	Maximum 128 bytes	See 8.1.
@index	○	0..1	0..1	5-digit decimal numerical value (0 to 65535)	
@xsi:type	○	1	1	“MemberOfType”	

					(Attach a namespace prefix if needed.)	
	OtherIdentifier	○	0-∞	0..1		See Table 3-45.
	Period	○	0-∞	0..3		
	@type	○	0..1	1	Maximum 12 bytes Note 3	
	Start	○	1	1	Specify in the YYYY-MM-DDThh:mm:ss+09:00 format (25 bytes). Note 4	
	End	○	0..1	0..1	Specify in the YYYY-MM-DDThh:mm:ss+09:00 format (25 bytes). Note 4	

Note 1: A GroupInformation that has the same groupId and a different fragmentId does not exist. A groupId remains the same if the fragmentVersion is updated when the fragmentId is the same.

Note 2: If “series,” “show,” “programConcept,” “programCompilation,” “otherCollection,” or “otherChoice” is selected, a receiver interprets that the attribute indicates a grouping that is different from a “package.”

Note 3: Select from “display” (display duration), “availability” (distribution period), and “new_arrival” (new arrival period).

Note 4: YYYY: 4-digit decimal western calendar notation, MM: 2-digit decimal month notation, DD: 2-digit decimal date notation, hh: 2-digit decimal time notation, mm: 2-digit decimal minute notation, ss: 2-digit decimal second notation

Table 3-27: Use of GroupInformation/BasicDescription

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
BasicDescription					
Title	○	0-∞	1..7		See Table 3-28.
MediaTitle	○	0-∞	0..1		
TitleImage	○	0..1	0..1		See Table 3-29.
TitleVideo	○	0..1	0..1		See Table 3-30.
TitleAudio	—				
ShortTitle	—				
Synopsis	○	0-∞	0..3		See Table 3-31.
PromotionalInformation	○	0-∞	0..1		See Table 3-32.
Keyword	○	0-∞	0..15		See Table 3-33.
Genre	○	0-∞	0..13		See Table 3-34.
ParentalGuidance	○	0..1	0..1		See Table 3-35.
Language	○	0-∞	0..4		See Table 3-36.
CaptionLanguage	○	0-∞	0..2		See Table 3-37.

SignLanguage	○	0-∞	0..1		See Table 3-38.
CreditsList	○	0..1	0..1		See Table 3-39.
AwardsList	—				
RelatedMaterial	○	0-∞	0..65535		See Table 3-40.
ProductionDate	○	0..1	0..1		See Table 3-41.
ProductionLocation	—				
CreationCoordinates	—				
DepictedCoordinates	—				
ReleaseInformation	○	0-∞	0..1		See Table 3-42.
Duration	—				
PurchaseList	○	0..1	0..1		See Table 3-43.
ContentProperties	○	0..1	0..1		See Table 3-44.

Table 3-28: Use of GroupInformation/BasicDescription/Title

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
Title				Maximum 128 two-byte characters Maximum 384 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.
@xml:lang	—				
@phoneticTranscription	—				
@phoneticAlphabet	—				
@type	○	0..1	1	Select from “main,” “secondary,” “alternative,” “popular,” “seriesTitle,” and “episodeTitle.” Maximum 12 bytes	

Note: The title element that has the @type specified as “main” is mandatory, and only one such title element exists.

A maximum of one title element can have the @type element that is specified as “alternative.” A maximum of one title element can have the @type element that is specified as “secondary.” A maximum of three title elements can have the @type that is specified as “popular.” A title character string for presentation is described in the @type specified as “main.” A subtitle character string for presentation is described in the @type specified as “secondary.” The ruby of the title character string is described in the @type specified as “alternative,” which is used as a search key and a sort key for sorting. The @type specified as “popular” is the attribute that is specified when there are two or more rubies, which is used not as a sort key but as a search key. Whether or not ruby as the ruby of a title character string is described in the @type specified as “main” and the @type specified as “secondary” is determined based on the operation. The processing of a title when the @type is specified as “seriesTitle” and the @type is specified as “episodeTitle” is determined based on the implemented receiver. For details on ruby, see “1 Use of *Furigana*” in Annex 4.

Table 3-29: Use of GroupInformation/BasicDescription/MediaTitle/TitleImage

Element/attribute name		Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
TitleImage						
Select ① or ②.		○	0..1	1		
①	mpeg7:MediaUri	○	1	1	Maximum 128 bytes	
②	mpeg7:InlineMedia	—				
mpeg7:StreamID		—				
CopyrightNotice		○	0-∞	0..1	Maximum 50 two-byte characters Maximum 150 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.

Note 1: The TitleImage specifies a still picture that is used for displaying a thumbnail. This still picture is not encrypted. When displaying a thumbnail picture, it can be displayed by scaling within the extent that the impression of the original picture is not largely altered. However, the aspect ratio shall be maintained, and the picture shall not be trimmed.

Note 2: For the correspondence relationship among the reference information of each resource described in metadata (e.g. <http://example.jp/a/v/c.png>), the applicable resource on a broadcasting stream, and the storage location in the receiver, see “8.2.3.3 Location resolution in a receiver.”

Table 3-30: Use of GroupInformation/BasicDescription/MediaTitle/TitleVideo

Element/attribute name		Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
TitleVideo						
Select ① or ②.		○	0..1	1		
①	mpeg7:MediaUri	○	1	1	Maximum 128 bytes	
②	mpeg7:InlineMedia	—				
mpeg7:StreamID		—				
CopyrightNotice		○	0-∞	0..1	Maximum 50 two-byte characters Maximum 150 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.

Note: The TitleVideo specifies the content used for preview display, etc. The locator of video content (e.g., <http://example.jp/a/b/c.mp4>) and the CRID of preview content (e.g., [crid: //example.jp/trailer/content1](http://example.jp/trailer/content1)) are described in MediaUri.

Table 3-31: Use of GroupInformation/BasicDescription/Synopsis

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
Synopsis				Maximum 408 two-byte characters Maximum 1224 bytes SP, CR, and LF can be used.	Note 1 See Section 3.1.3.
@xml:lang	—				
@phoneticTranscription	—				
@phoneticAlphabet	—				
@length	○	0..1	0..1	Select from “short,” “medium,” and “long.” Maximum 6 bytes	

Note 1: Maximum 25 characters and maximum 75 bytes when “short” is selected for @length; maximum 100 characters and maximum 300 bytes when “medium” is selected for @length; maximum 408 characters and maximum 1224 bytes when “long” is selected for @length

Table 3-32: Use of GroupInformation/BasicDescription/PromotionalInformation

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
PromotionalInformation				Maximum 200 two-byte characters Maximum 600 bytes SP, CR, and LF can be used.	See Section 3.1.3.
@href	○	0..1	0..1	Select from the ARIBPromotionalType CS. Maximum 128 byte	See 1 in Annex 1.
@xml:lang	—				
@phoneticTranscription	—				
@phoneticAlphabet	—				

Table 3-33: Use of GroupInformation/BasicDescription/Keyword

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
Keyword				Maximum 40 two-byte characters Maximum 120 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.
@xml:lang	—				
@phoneticTranscription	—				
@phoneticAlphabet	—				
@type	○	0..1	0..1	Select from “main,” “secondary,” and “other (supplemental).” Only one keyword can be selected as “main.”	Note 1

Note 1: In the ECG, a keyword to which “main” or “secondary” is specified can be presented; however, a keyword to which “other” is specified should not be used for presentation. If a type is not specified, then the attribute is regarded as “main.” A type can be omitted when one keyword is used. However, when multiple keywords are used, a type must be described.

Table 3-34: Use of GroupInformation/BasicDescription/Genre

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
Genre					
@href	○	1	1	Select from the ARIBGenreCS, ARIBGenreCS2 (when main or secondary is selected for the @type), and another CS (when “other” is selected for the @type, see 2.2 in Annex 4). Maximum 128 bytes	See 1 in Annex 1.
@type	○	0..1	0..1	Select from “main,” “secondary,” and “other.” Only one genre can be selected as “main.”	Note 1
Name	—				
Definition	—				

Note 1: If only one genre is used/presented using a content list presenting method, such as ECG and EPG, priority is given to a genre that is specified as “main.” If there is only one Genre element, the @type can be omitted. However, if there are multiple Genre elements, the @type must be described. If the ARIBGenreCS or ARIBGenreCS2 is specified, the first one is specified as main, and the second and subsequent are specified as secondary. “Other” can be specified for any CS other than the CSs stated earlier (see 2.2 in Annex 4). In this case, the processing is dependent on the implemented receiver.

Table 3-35: Use of GroupInformation/BasicDescription/ParentalGuidance

Element/attribute name		Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
ParentalGuidance						
	Select ① or ②.	○	1	1		
①	mpeg7:ParentalRating	○	1	1		Note 1
	@href	○	1	1	Select from the ARIBParentalRatingCS. Maximum 128 bytes	See 2 in Annex 1.
	mpeg7:Name	—				
	mpeg7:Definition	—				
②	mpeg7:MinimumAge	—				
	mpeg7:Region	—				

Note 1: If a value specified for the @href is different from the value specified for ProgramInformation/BasicDescription/ParentalGuidance/mpeg7: ParentalRating/@href, priority is given to the value specified for ProgramInformation/BasicDescription/ParentalGuidance/mpeg7: ParentalRating/@href.

Table 3-36: Use of GroupInformation/BasicDescription/Language

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
Language				Select from “ja” (Japanese), “en” (English), “de” (German), “fr” (French) “it” (Italian), “ru” (Russian), “zh” (Chinese), “ko” (Korean), “es” (Spanish), and “xx” (a foreign language other than above, multiple languages, unknown language).	Corresponds to the language names described in the ISO_639_language_code and ISO_639_language_code_2 in “Table 31-62 Audio component descriptor” in ARIB TR-B14
@type	—				
@supplemental	—				

Table 3-37: Use of GroupInformation/BasicDescription/CaptionLanguage

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
CaptionLanguage				Select from “ja” (Japanese), “en” (English), “de” (German), “fr” (French) “it” (Italian), “ru” (Russian), “zh” (Chinese), “ko” (Korean), “es” (Spanish), and “xx” (a foreign language other than above, multiple languages, unknown language).	Corresponds to the language names described in the ISO_639_language_code and ISO_639_language_code_2 in “Table 31-62 Audio component descriptor” in ARIB TR-B14
@closed	○	0..1	1		
@supplemental	—				
@description	—	—	—		

Table 3-38: Use of GroupInformation/BasicDescription/SignLanguage

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
SignLanguage				Select from “ja” (Japanese), “en” (English), “de” (German), “fr” (French) “it” (Italian), “ru” (Russian), “zh” (Chinese), “ko” (Korean), “es” (Spanish), and “xx” (a foreign language other than above, multiple languages, unknown language).	Describes the sign language
@primary	○	0..1	0..1		
@translation	○	0..1	0..1		
@type	—				

Table 3-39: Use of GroupInformation/BasicDescription/CreditsList

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
CreditsList					
CreditsItem	○	0-∞	1..20		
@role	○	1	1	Select from the ARIBRoleCS and the business operator's original RoleCS. Maximum 128 bytes	See 1 in Annex 1.
Select from ①, ⑥, ⑦, and ⑧.	○	0-∞	1..5		
① PersonName	○	1	1		
@dateFrom	—				
@dateTo	—				
@type	○	0..1	1	Select from “main,”	Only one element can have the @type

							“variant,” and “former.” Maximum 7 bytes.	specified as “main.” Note 1
			@xml:lang	—				
			Select from ② to ⑤.	○	1-∞	1		
		②	mpeg7:GivenName	○	1	1	Maximum 60 two-byte characters Maximum 180 bytes SP can be used. CR and LF cannot be used.	Describe the full name. See Section 3.1.3.
			@xml:lang	—				
			@initial	—				
			@abbrev	○	0..1	0..1	Maximum 16 two-byte characters Maximum 48 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.
		③	mpeg7:FamilyName	—				
		④	mpeg7:Title	—				
		⑤	mpeg7:Numeration	—				
	⑥	PersonNameIDRef		—				
	⑦	OrganizationName		—				
	⑧	OrganizationNameIDRef		—				
		Character		○	0-∞	0..1		
			@dateFrom	—				
			@dateTo	—				
			@type	—				
			@xml:lang	—				
			Select from ⑨ to ⑫.	○	1-∞	1		
		⑨	mpeg7:GivenName	○	1	1	Maximum 60 two-byte characters Maximum 180 bytes	Describe the full name. See Section 3.1.3.

								SP can be used. CR and LF cannot be used.	
				@xml:lang	—				
				@initial	—				
				@abbrev	○	0..1	0..1	Maximum 16 two-byte characters Maximum 48 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.
			⑩	mpeg7:FamilyName	—				
			⑪	mpeg7:Title	—				
			⑫	mpeg7:Numeration	—				

Note 1: The PersonName element that has the @type specified as “main” is mandatory, and only one such element exists, in which a credit character string for presentation is described. Only one PersonName element can have the @type that is specified as “variant,” in which the ruby of the credit character string is described. This attribute is used as a search key and a sort key for sorting. A maximum of three PersonName elements can have the @type that is specified as “former.” The @type specified as “former” is the attribute that is specified when there are two or more rubies, which are used not as a sort key but as a search key. For details on ruby, see “1 Use of *Furigana*” in Annex 4.

Table 3-40: Use of GroupInformation/BasicDescription/RelatedMaterial

Element/attribute name		Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
RelatedMaterial						
	HowRelated	○	0..1	1		
	@href	○	1	1	Select from the ARIBHowRelatedCS, ISDBTMMHowRelatedCS, and the business operator's original HowRelatedCS. Note 1 Maximum 128 bytes	See 3 in Annex 1.
	Name	○	0..1	0..1	Maximum 32 two-byte characters Maximum 96 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.
	Definition	—				
	Format	○	0..1	0..1		
	@href	○	1	1	Select from the ARIBFileFormatCS, ISDBTMMFileFormatCS, and the business operator's original FileFormatCS. Note 1 Maximum 128 bytes	See 4 in Annex 1.
	Name	—				
	Definition	—				
	Select ① or ④.	○	1	1		
①	MediaLocator	○	1	1		
	Select ② or ③.	○	0..1	1		
	② mpeg7:MediaUri	○	1	1	Maximum 128 bytes	Note 2
	③ mpeg7:InlineMedia	—				
	mpeg7:StreamID	—				

④	SegmentReference	—				
	PromotionalText	○	0-∞	0..1	Maximum 3000 bytes SP, CR, and LF can be used.	Note 3 See Section 3.1.3.
	@xml:lang	—				
	@phoneticTranscription	—				
	@phoneticAlphabet	—				
	PromotionalMedia	○	0-∞	0..1		Note 4
	TitleImage	○	0..1	1		See Table 3-7.
	TitleVideo	—				
	TitleAudio	—				
	SourceMediaLocator	—				

Note 1: If the business operator's original CS is specified, the processing in the receiver is dependent on the receiver.

Note 2: This attribute specifies the reference URI. The URL of a content or browser script file is specified. The relationship with a reference is specified using a dictionary item of the HowRelatedCS in the HowRelated/@href. The same dictionary item (relationship) can be specified for multiple different references.

Note 3: Describe the catch copy, copyright, etc., for the substance specified in the MediaLocator/mpeg7: MediaUri for sales promotion.

Note 4: Specify a thumbnail still picture file for the substance specified in MediaLocator/mpeg7: MediaUri.

Table 3-41: Use of GroupInformation/BasicDescription/ProductionDate

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
ProductionDate					
TimePoint	○	1	1	Describe in the YYYY-MM-DD or YYYY format. Maximum 10 bytes	
Duration	—				
YearRange	○	0..1	0..1	Describe in the YYYY-YY or YYYY-YYYY format. Maximum 9 bytes	Note 1

Note 1: When the YearRange is specified, YearRange rather than TimePoint is used for displaying the production year.

Table 3-42: Use of GroupInformation/BasicDescription/ReleaseInformation

Element/attribute name		Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
ReleaseInformation						
	ReleaseDate	○	0..1	0..1		
	Select ① or ②.	○	1	1		
	① DayAndYear	○	1	1	Specify in the YYYY-MM-DD format.	
	② Year	○	1	1	Specify in the YYYY format.	
	ReleaseLocation	○	0..1	0..1	Specify with a 2-character country code as defined in “ISO3166-2.”	

Table 3-43: Use of GroupInformation/BasicDescription/PurchaseList

Element/attribute name		Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
PurchaseList						
	Select ① or ②.	○	0-∞	0..10		
	①PurchaseItem	—				
	②PurchaseIdRef	○	1	1	Describe a purchase identifier. Maximum 24 bytes	

Table 3-44: Use of GroupInformation/BasicDescription/ContentProperties

Element/attribute name		Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
ContentProperties						
	ContentType	○	0..1	0..1		
	@href	○	1	1	Select from the ISDBTMMContentTypeCS and the business operator's original ContentTypeCS. Maximum 128 bytes	See 5 in Annex 1.
FileProperties		○	0..1	0..1		
	FileFormat	○	0..1	0..1		
	@href	○	1	1	Select arbitrarily from the ARIBFileFormatCS, ISDBTMMFileFormatCS, and the business operator's original FileFormatCS. Maximum 128 bytes	See 4 in Annex 1.
	FileSize	○	0..1	0..1	10-digit decimal integer value, 10 characters	Describe in a byte unit.
	FileChecksum	—				
	ContentAttributes	—				
	ContextAttributes	—				

Table 3-45: Use of GroupInformation/OtherIdentifier

Element/attribute name		Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
OtherIdentifier					Maximum 9 bytes SP, CR and LF cannot be used.	See Section 3.1.3.
@type		○	0..1	1	“VideoID” Maximum 10 bytes	Note 1
@organization		—				
@authority		—				
@encoding		—				

Note 1: if the @type is specified as “VideoID,” it indicates the VideoID of the information described for an applicable ProgramInformation. A 9-digit or less decimal integer value that is unique for a business operator is specified.

3.4.4 Use of review information elements

Among the elements/attributes defined in “3.2.3.5 Media review description scheme” in ARIB STD-B38, the elements/attributes provided in Table 3-46 are used.

Table 3-46: Use of Review

Element/attribute name		Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
Review						
@programId		○	1	1	Maximum 128 bytes	Note 1 See 8.1.
@fragmentId		○	0..1	1	10-digit hexadecimal integer value (10 characters)	Note 1 See 4.4.1.
@fragmentVersion		○	0..1	1	14-digit decimal numerical value (14 characters)	Note 1 See 4.4.2.
@fragmentExpirationDate		○	0..1	0..1	Specify in the YYYY-MM-DDThh:mm:ss+09:00 format (25 bytes).	See 7.3.1.
@metadataOriginIDRef		—				
@xml:lang		—				

	Rating	○	0-∞	0..1		
	mpeg7:RatingValue	○	1	1	1 to 10 integer value	
	mpeg7:RatingScheme	○	1	1		
	@best	—				
	@worst	—				
	@style	○	1	1	Select “higherBetter” or “lowerBetter.”	
	FreeTextReview	○	0-∞	0..1	Maximum 255 two-byte characters Maximum 765 bytes SP, CR, and LF can be used.	See Section 3.1.3.
	Reviewer	○	0-∞	0..1		
	Select ① or ②.	○	1	1		
	①PersonName	○	1	1	Maximum 60 two-byte characters Maximum 180 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.
	@dateFrom	—				
	@dateTo	—				
	@type	○	0..1	1	Select from “main,” “variant,” and “former.” Maximum 7 bytes.	Only one element can have “main” as the @type. Note 3
	@xml:lang	—				
	Select from ③ to ⑥.	○	1-∞	1		
	③ mpeg7:GivenName	○	1	1	Maximum 60 two-byte characters Maximum 180 bytes SP can be used. CR and LF cannot be used.	Describe the full name. See Section 3.1.3.
	@xml:lang	—				
	@initial	—				
	@abbrev	○	0..1	0..1	Maximum 16 two-byte characters Maximum 48 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.
	④ mpeg7:FamilyName	—				

		⑤	mpeg7:Title	—				
		⑥	mpeg7:Numeration	—				
		②	OrganizationName	○	1	1	Maximum 100 two-byte characters Maximum 300 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.
			Publication	○	0..1	0..1	Maximum 100 two-byte characters Maximum 300 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.
			ReviewReference	○	0..1	0..1	Maximum 128 bytes	Note 2

Note 1: A programId remains the same if the fragmentVersion is updated when the fragmentId is the same.

Note 2: Specify the URI of a material in which the review is extracted or cited.

Note 3: The PersonName element that has the @type specified as “main” is mandatory, and only one such element exists, in which a credit character string for presentation is described. Only one PersonName element can have the @type that is specified as “variant,” in which the ruby of the credit character string is described. This attribute is used as a search key and a sort key for sorting. A maximum of three PersonName elements can have the @type that is specified as “former.” The @type specified as “former” is the attribute that is specified when there are two or more rubies, which are used not as a sort key but as a search key. For details on ruby, see “1 Use of Furigana” in Annex 4.

3.4.5 Use of purchase information elements

The purchase information element is used to describe the information in order to reference the price of a billing package, sales pattern, and corresponding license. The details on the conditions of use applied after purchasing each content included in a package are described in the license reference information element defined in “3.6 Use of license reference information elements.”

(1) If the PurchaseType of a purchase information element is “single” (single item), the purchase information element is associated with the program information element. The reference from a purchase information element to a program information element is described in the CRIDRef element of the purchase information element. The reference from a program information element to a purchase information element is described in the PurchaseIdRef element of the program information element (1 to 10 elements). The reference from a license reference information element to a program information element is described in the ProgramCRID element of the license reference information element. One program information element can be referenced from 1 to 10 license reference information elements. The reference from a license reference information element to a purchase information element is described in

the PurchaseIdRef element of the license reference information element. One purchase information element can be referenced from one license reference information element.

In addition to the reference relationship among a purchase information element, license reference information element, and program information element, there is a reference relationship between the program information element and the group information element in which a series is described, as well as a reference relationship between the group information elements in which a series is described. The reference from a program information element to a group information element in which a series is described is described in the crid attribute of the MemberOf element in the program information element (0 to 20 elements). One group information element, in which one series is described, can be referenced from 0 to 500 program information elements. The reference from a group information element, in which a lower series is described, to a group information element, in which an upper series is described, is described in the crid attribute of the MemberOf element in the group information element in which the lower series is described (0 to 20 items). One group information element, in which an upper series is described, can be referenced from 0 to 100 group information elements in which lower series are described.

(2) If the PurchaseType of a purchase information element is “pack,” the purchase information element is associated with the program information element via the group information element in which the package is described. The reference from a purchase information element to the group information element, in which a package is described, is described in the CRIDRef element of the purchase information element. The reference from a group information element, in which a package is described, to a purchase information element is described in the PurchaseIdRef element of the group information element, in which a package is described (1 to 10 items). The reference from a license reference information element to a purchase information element is described in the PurchaseIdRef element of the license reference information element. One purchase information element can be referenced from 1 to 500 license reference information elements. The reference from a license reference information element to a program information element is described in the ProgramCRID element of the license reference information element. One program information element can be referenced from 1 to 10 license reference information elements.

The reference from a program information element to a group information element, in which a package is described, is described in the crid attribute of the MemberOf element in the program information element (1 to 20 elements). A group information element, in which one

package is described, can be referenced from 1 to 500 program information elements. The reference from a group information element, in which a package is described, to a program information element is described in the RelatedMaterial element of the group information element in which the package is described (ARIBHowRelatedCS/21.2) (0 to 500 elements). One program information element can be referenced from 1 to 20 group information elements in which a package is described.

If a group information element, in which a package is described, is associated with a group information element, in which a series is described, the RelatedMaterial element is used to cross-reference them. For the reference from a group information element, in which a package is described, to a group information element, in which a series is described, is described in the RelatedMaterial element of the group information element in which the package is described (ARIBHowRelatedCS/21.1) (0 to 20 elements). For the reference from a group information element, in which a series is described, to a group information element, in which a package is described, is described in the RelatedMaterial element of the group information element in which the series is described (ARIBHowRelatedCS/21) (0 to 20 elements).

(3) If the PurchaseType of a purchase information element is “subscription” (unlimited), the purchase information element is associated with a program information element via a group information element in which the package is described. The reference relationship between the purchase information element and the group information element in which a package is described are the same as (2) above. The reference from a license reference information element to a purchase information element is described in the PurchaseIdRef element of the license reference information element. One purchase information element can be referenced from 0 to 65,535 license reference information elements. The reference from a license reference information element to a program information element is described in the ProgramCRID element of the license reference information element. One program information element can be referenced from 1 to 10 license reference information elements.

The reference from a program information element to a group information element, in which a package is described, is described in the crid attribute of the MemberOf element in the program information element (1 to 20 elements). One group information element, in which a package is described, can be referenced from 0 to 65,535 program information elements. The reference from a group information element, in which a package is described, to a program information element is described in the RelatedMaterial element of the group information element in which the package is described (ARIBHowRelatedCS/21.2) (0 to 65,535 elements).

One program information element can be referenced from 1 to 20 group information elements in which a package is described.

If a group information element in which a package is described is associated with a group information element in which a series is described, the reference relationship between them is the same as (2) above.

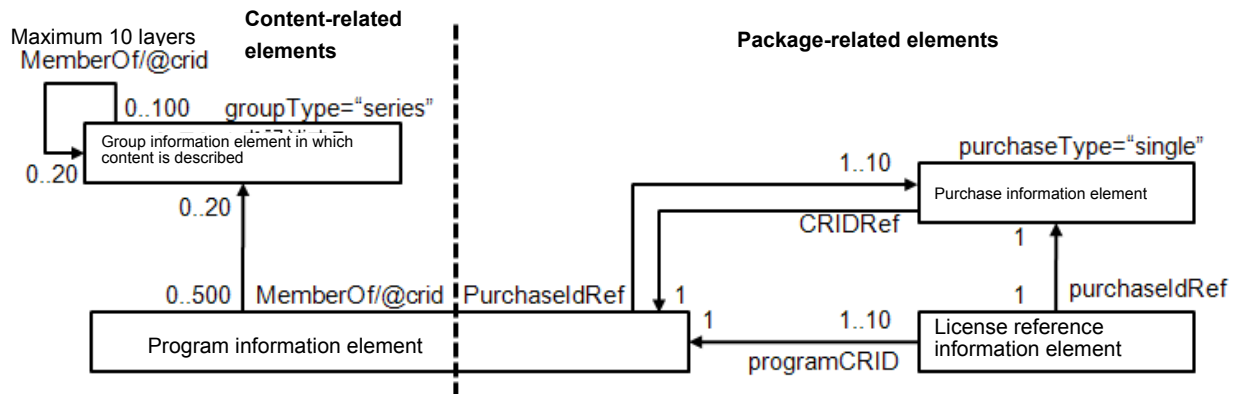


Fig. 3-2: Relationship among purchase information elements, program information elements, and license reference information elements (single item: contract type is PPC or PPV)

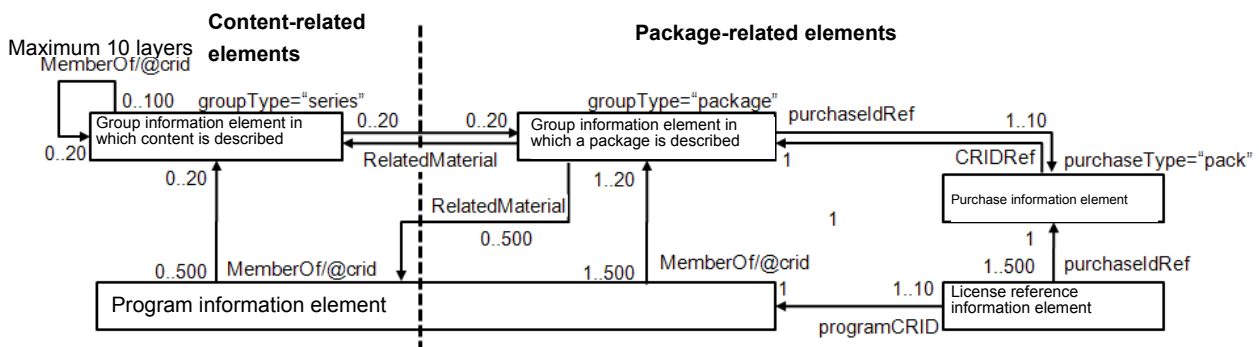


Fig. 3-3: Relationship among purchase information elements, program information elements, and license reference information elements (pack: contract type is PPP)

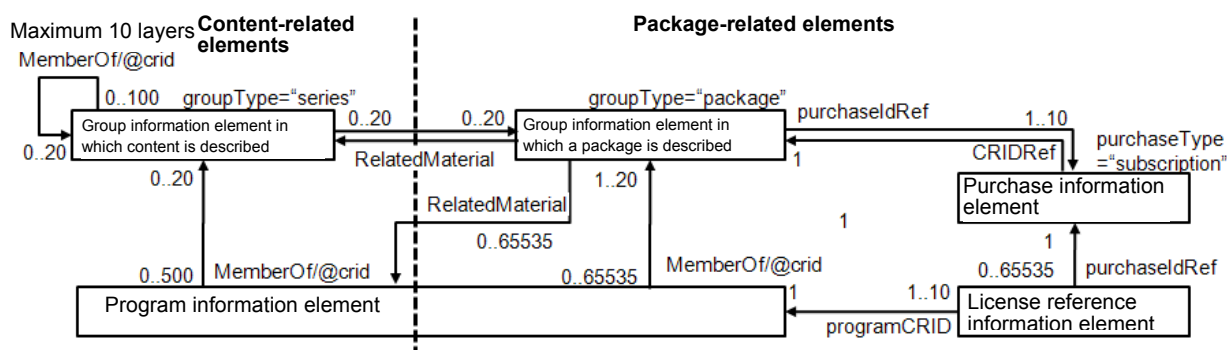


Fig. 3-4: Relationship among purchase information elements, program information elements, and license reference information elements (unlimited: contract type is PPM)

Among the elements/attributes defined in “3.2.7 Purchase information metadata” in ARIB STD-B38, the elements/attributes provided in Table 3-47 are used.

Table 3-47: Use of PurchaseInformation

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
PurchaseInformation					
@start	○	0..1	0..1	Specify in the YYYY-MM-DDTh h:mm:ss+09:00 format (25 bytes).	Date and time when a billing package can be purchased
@end	○	0..1	0..1	Specify in the YYYY-MM-DDTh h:mm:ss+09:00 format (25 bytes).	Expiration date and time when a billing package cannot be purchased
@purchaseId	○	1	1	Maximum 24 bytes	Note 1 Purchase identifier
@fragmentId	○	0..1	1	10-digit hexadecimal integer value (10 characters)	Note 1 See 4.4.1.
@fragmentVersion	○	0..1	1	14-digit decimal numerical value (14 characters)	Note 1 See 4.4.2.
@fragmentExpiration Date	○	0..1	0..1	Specify in the YYYY-MM-DDTh h:mm:ss+09:00 format (25 bytes).	See 7.3.1.

	@metadataOriginIDRef	—				
	@xml:lang	—				
	Price	○	1..∞	1	10-digit decimal integer value (10 characters) without prefix 0	Purchase price of a billing package in yen
	@currency	○	1	1	Specify “JPY” (3 characters).	Currency unit of the price paid for a billing package
	Purchase	○	0..∞	1		
	PurchaseType	○	0..1	1		
	@href	○	1	1	Select either “single,” “pack,” or “subscription” from ARIBPurchaseTypeCS. Note 2 Maximum 128 bytes	See 1 in Annex 1.
	Name	○	0..1	0..1	Maximum 64 two-byte characters Maximum 192 bytes Note 3	
	QuantityUnit	○	0..1	0..1 Note 4		
	@href	○	1	1	Select either “minute, hour, date, week, month, or year” from the ARIBUnitTypeCS (maximum 128 bytes)	See 1 in Annex 1.
	QuantityRange	○	0..1	0..1 Note 4		
	@min	—				
	@max	○	0..1	1	5-digit decimal integer value (5 bytes) without prefix 0	
	Description	○	0-∞	0..1	Maximum 200 two-byte characters	See Section 3.1.3.

					Maximum 600 bytes SP, CR, and LF can be used.	
	PricingServerURL	○	0-∞	0..1	Maximum 128 bytes	Note 5
	NameTag	○	0..1	0..1	Maximum 128 two-byte characters Maximum 384 bytes SP can be used. CR and LF cannot be used.	Describe an item name (price tag name). Note 6
	PurchaseOption	○	0..1	0..1	Maximum 200 two-byte characters Maximum 600 bytes SP, CR, and LF can be used.	Describe additional information when purchasing. Note 7 See Section 3.1.3.
	ParentPurchase	○	0..1	0..1	Maximum 24 bytes	Purchase identifier Note 8
	RequiredCoupon	○	0-∞	0..10		
	@couponId	○	0..1	1	Maximum 24 bytes	Specify a coupon descriptor.
	@authorityName	○	0..1	0..1	Maximum 64 bytes	Authority that provides the coupon
	@requiredNumber	○	0..1	0..1	2-digit decimal integer value (2 characters)	Indicate the number of coupons that a client must be prepared to use. The default value is 1. Note 9
	RewardCoupon	○	0-∞	0..1		
	@userConfirmationRequired	○	0..1	0..1	Select “true” or “false.”	
	@number	○	0..1	0..1	2-digit decimal integer value (2 characters)	Describe the number of coupons to be provided. The default is 1.
	Select ① or ②.	○	1	1		
	①CouponRef	○	1	1		
	CouponIdRef	○	1	1	Maximum 24 bytes	
	AuthorityName	○	0..1	0..1	Maximum 64 bytes	Authority that provides the coupon

	② CouponDescription	—				
	NonTVARightsInformationRef	—				
	CRIDRef	○	1	1		
	@crid	○	1	1	Maximum 128 bytes	Describe the ProgramInformation/programId of the ProgramInformation that references the applicable PurchaseInformation/@purchaseId. Or, describe the GroupInformation/groupId of the GroupInformation. See 8.1.

Note 1: A purchaseId remains the same if the fragmentVersion is updated when the fragmentId is the same.

Note 2: A type can be selected from the business operator's original PurchaseTypeCS. In this case, the operation is based on the service business operator.

Note 3: Describe the “free (free item)/basic (basic contract item)/premium (premium contract item)” as an item type.

Note 4: As the conditions of use are various in multimedia broadcasting, such as “absolute duration, relative duration, and number of times,” a unit of items that can be used or a usage period of an item may not match the conditions of use of the license (for instance, if the purchase information element is specified as a pack, a license type varies depending on each content). Therefore, 0 can be set to the QuantityUnit and QuantityRange in the item information.

Note 5: The service business operator can decide the detailed setting of PricingServerURL. For instance, the service business operator can decide a format and other settings if the site ID of the service business operator for billing is described in the URI format.

Note 6: NameTag is used when it is necessary to describe an item name (price tag name) separately from the program information element that corresponds to the purchase information element or the title of a group (package) information element.

Note 7: Describe the information that is used as additional information when processing a purchase. The service business operator can decide the settings of the items to be described. For instance, if limitations are applied on the number of times and the location when selling an applicable item, they can be described in the following format.

Format: Describe them as the set of <Control information name>:<control value>. If multiple <control values> exist, the values can be described by separating them with a slash (/). If there are multiple additional information, describe the information as the set of <control information name>:<control value> by separating each information with a comma (,).

Example: Limits of the number of people who can purchase, purchase area (latitude and longitude), purchase area code, and whether or not an item can be purchased from abroad (if it is not specified, “false,” which means an item cannot be purchased from abroad, is described) are described as follows:

```
<PurchaseOption>limit:100,Area:N48 51 29.538/E2 17 39.692/radius 100m,
AreaCode:123,Abroad:true</PurchaseOption>
```

Note 8: If the purchase information to be described has a parent-child relationship with other purchase information, the purchase information descriptor of the parent purchase information is described (for instance, a certain basic contract must be signed to have a contract for a premium contract item).

Note 9: For instance, as for coupon A, which is provided when purchasing a movie ticket, a movie ticket discount is offered for three coupons.

3.5 Use of Instance Description Metadata

This section defines the use of a broadcast event information element (BroadcastEvent), on-demand program information element (OnDemandProgram), and on-demand service information element (OnDemandService) among the instance description metadata defined in “3.2.4 Instance description metadata” in ARIB STD-B38.

3.5.1 Use of broadcast event information elements

Among the broadcast event information elements (BroadcastEvent) defined in “3.2.4.2 Program location” in ARIB STD-B38, the elements/attributes provided in Table 3-48 are used.

Table 3-48: Use of BroadcastEvent

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
BroadcastEvent					
@serviceIDRef	○	0..1	1	Maximum 4 bytes	Describe the service identifier that identifies the service that broadcasts this event.
@fragmentId	○	0..1	1	10-digit hexadecimal integer value (10 characters)	Note 1 See 4.4.1.
@fragmentVersion	○	0..1	1	14-digit decimal numerical value (14 characters)	Note 1 See 4.4.2.
@fragmentExpirationDate	○	0..1	0..1	Specify in the YYYY-MM-DDThh:mm:ss+09:00 format (25 bytes).	See 7.3.1.
@metadataOriginIDRef	—				
@xml:lang	—				
Program	○	1	1		
@crId	○	1	1	Maximum 128 bytes	Note 1 See 8.1.
ProgramURL	○	0..1	0..1	Maximum 128 bytes	See 8.2.
InstanceMetadataId	○	0..1	1	Maximum 16 bytes	Note 1 See 3.5.5.
InstanceDescription	○	0..1	0..1		See Table 3-49.

PublishedStartTime	○	0..1	1	Specify in the YYYY-MM-DDThh:mm:ss+09:00 format (25 bytes).	
PublishedEndTime	○	0..1	0..1	Specify in the YYYY-MM-DDThh:mm:ss+09:00 format (25 bytes).	
PublishedDuration	○	0..1	0..1	Specify in the PThhHmmMssS (xsd:duration) format. Note 2	
Live	○	0..1	0..1		
@value	○	1	1	Select “true” or “false.”	
Repeat	○	0..1	0..1		Note 3
@value	○	1	1	Select “true” or “false.”	
FirstShowing	○	0..1	0..1		
@value	○	1	1	Select “true” or “false.”	
LastShowing	○	0..1	0..1		
@value	○	1	1	Select “true” or “false.”	
Free	○	0..1	0..1		Note 4
@value	○	1	1	Select “true” or “false.”	
RepeatLabel	○	0..1	0..1	1 or more 2-digit decimal integer	Note 3, Note 4

Note 1: A BroadcastEvent that has the same set of Program/@crid and InstanceMetadataId and that has a different fragmentID does not exist. The Program/@crid and InstanceMetadataId remain the same if the fragmentVersion is updated when the fragmentId is the same.

Note 2: P: A character that indicates a lapse of time, T: a character that separates date and time, hhH: elapsed 2-digit 10 decimal time, mmM: elapsed 2-digit decimal minute, ssS: elapsed 2-digit decimal second

Note 3: Basically, the “RepeatLabel” is used. However, taking into account compatibility, it is recommended that the Repeat/@value is set to “true” if the “RepeatLabel” is not specified and if the Repeat/@value is set to “false” and 1 or more values are set to the “RepeatLabel.”

Note 4: See 3.9.2 for the correspondence relationship with ARIB-SI.

Table 3-49: Use of BroadcastEvent/InstanceDescription

Element/Attribute		Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Definition
InstanceDescription						
	Title	○	0..1	0..1		See Table 3-6.
	Synopsis	○	0..1	0..1		See Table 3-9.
	Genre	○	0-∞	0..40		See Table 3-12.
	PurchaseList	—				
	AVAttributes	○	0..1	0..1		See Table 3-25.
	MemberOf	—				
	OutputRestriction	○	0-∞	0..20		Note 1
	Port	○	1	1		
	@href	○	1	1	Select from the ARIBOutputPortCS. Maximum 128 bytes	See 1 in Annex 1.
	Name	—				
	Definition	—				
	Mode	○	1	1		
	@href	○	1	1	Select from the ARIBCopyControlMethodCS. Maximum 128 bytes	See 1 in Annex 1.
	Name	—				
	Definition	—				
	ContentProperties	—				

Note: If the same information as ProgramInformation is described in InstanceDescription, priority is given to the information in InstanceDescription.

Note 1: As metadata in a channel does not have an LIT, the copy control information of the channel is described using this information. This attribute is mapped from the APS_control_data, copy_control_type, and digital_recording_control_data of ARIB-SI. This attribute is used only in the BroadcastEvent.

3.5.2 Use of on-demand program information elements

Among the on-demand program information elements (OnDemandProgram) defined in “3.2.4.2 Program location” in ARIB STD-B38, the elements/attributes provided in Table 3-50 are used.

Table 3-50: Use of OnDemandProgram

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
OnDemandProgram					
@fragmentId	○	0..1	1	10-digit hexadecimal integer value (10 characters)	Note 1 See 4.4.1.
@fragmentVersion	○	0..1	1	14-digit decimal numerical value (14 characters)	Note 1 See 4.4.2.
@fragmentExpirationDate	○	0..1	0..1	Specify in the YYYY-MM-DDThh:mm:ss+09:00 format (25 bytes).	See 7.3.1.
@metadataOriginIDRef	—				
@xml:lang	—				
Program	○	1	1		
@crid	○	1	1	Maximum 128 bytes	Note 1 See 8.1.
ProgramURL	○	0..1	0..1	Maximum 128 bytes	See 8.2.
InstanceMetadataId	○	0..1	1	Maximum 16 bytes	Note 1 See 3.5.5.
InstanceDescription	○	0..1	0..1		See Table 3-51.
PublishedDuration	○	0..1	0..1	Specify in the PThhHmmMssS (xsd:duration) format. Note 2	
StartOfAvailability	○	0..1	1	Specify in the YYYY-MM-DDThh:mm:ss+09:00 format (25 bytes).	
EndOfAvailability	○	0..1	0..1	Specify in the YYYY-MM-DDThh:mm:ss+09:00 format (25 bytes).	
FirstAvailability	—				
LastAvailability	—				

ImmediateViewing	—				
------------------	---	--	--	--	--

Note 1: An OnDemandProgram that has the same set of the Program/@crid and InstanceMetadataId and that has a different fragmentID does not exist. The Program/@crid and InstanceMetadataId remains the same if the fragmentVersion is updated when the fragmentId is the same.

Note 2: P: A character that indicates a lapse of time, T: a character that separates date and time, hhH: elapsed 2-digit 10 decimal time, mmM: elapsed 2-digit decimal minute, ssS: elapsed 2-digit decimal second

Table 3-51: Use of OnDemandProgram/InstanceDescription

Element/Attribute	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Definition
InstanceDescription					
Title	○	0..1	0..1		See Table 3-6.
Synopsis	○	0..1	0..1		See Table 3-9.
Genre	○	0-∞	0..40		See Table 3-12.
PurchaseList	—				
AVAttributes	○	0..1	0..1		See Table 3-25.
MemberOf	—				
OutputRestriction	—				
ContentProperties	—				

Note: If the same information as ProgramInformation is described in InstanceDescription, priority is given to the information in InstanceDescription.

3.5.3 Use of on-demand service information elements

Among the on-demand service information elements (OnDemandService) defined in “3.2.4.2 Program location” in ARIB STD-B38, the elements/attributes provided in Table 3-50 are used.

Table 3-52: Use of OnDemandService

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
OnDemandService					
@serviceIDRef	○	1	1	Maximum 4 bytes	Identifier that identifies the on-demand service information element related to an on-demand program

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
					list
@fragmentId	○	0..1	1	10-digit hexadecimal integer value (10 characters)	See 4.4.1.
@fragmentVersion	○	0..1	1	14-digit decimal numerical value (14 characters)	See 4.4.2.
@fragmentExpirationDate	○	0..1	0..1	Specify in the YYYY-MM-DDThh:mm:ss+09:00 format (25 bytes).	See 7.3.1.
@metadataOriginIDRef	—				
@xml:lang	—				
OnDemandProgram	○	1-∞	1		See Table 3-50.

3.5.4 Reference relationship between program information elements and instance description metadata

Figure 3-5 shows the reference relationship among the program information elements, broadcast event information elements, and on-demand program information elements.

- One program information element can be associated with zero or one broadcast event information element. One broadcast event information element must be associated with one program information element. The reference from a broadcast event information element to a program information element is described in the Program/@crid attribute of the broadcast event information element.
- One program information element can be associated with zero or one on-demand program information element. One on-demand program information element must be associated with one program information element. The reference from an on-demand program information element to a program information element is described in the Program/@crid attribute of the on-demand program information element.

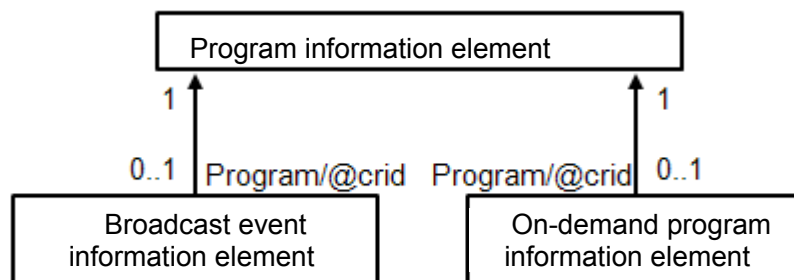


Fig. 3-5: Relationship between program information elements and instance description metadata

3.5.5 Use of instance metadata IDs

An instance metadata ID is used to identify multiple broadcast event information elements and multiple on-demand program information elements that correspond to the same content reference identifier (CRID). The instance metadata ID is expressed as a unique character string within a CRID scope. In multimedia broadcasting, as the program information element has zero or one BroadcastEvent and OnDemandProgram, an arbitrary character string can be shared with multiple instance metadata IDs.

The syntax of an instance metadata ID is expressed as `imi:<data>` based on the standards defined in “3.2.4.2 Program location” in ARIB STD-B38.

The following shows the characters that can be used as `<data>`.

```

<data>      =      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”
  
```

3.5.6 Use of service information elements

Among the elements/attributes defined in “3.2.4.3 Service information” in ARIB STD-B38, the elements/attributes provided in Table 3-53 are used.

Table 3-53: Use of ServiceInformation

Element/attribute name		Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
ServiceInformation						
	@serviceId	○	1	1	Maximum 4 bytes	An identifier that identifies a service Note 1
	@fragmentId	○	0..1	1	10-digit hexadecimal integer value (10 characters)	Note 1 See 4.4.1.
	@fragmentVersion	○	0..1	1	14-digit decimal numerical value (14 characters)	Note 1 See 4.4.2.
	@fragmentExpirationDate	○	0..1	0..1	Specify in the YYYY-MM-DDThh:m m:ss+09:00 format (25 bytes).	See 7.3.1.
	@metadataOriginIDRef	—				
	@xml:lang	—				
	Name	○	1-∞	1	Maximum 128 two-byte characters Maximum 384 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.
	@length	○	0..1	0..1	Select “short,” “medium,” or “long.” Maximum 6 bytes	
	Owner	○	0..1	0..1	Maximum 128 two-byte characters Maximum 384 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.
	ServiceURL	○	0..1	0..1	Maximum 128 bytes	URL of an option for service
	Logo	○	0-∞	0..3		
	Select ① or ②.	○	0..1	1		
	①mpeg7:MediaUri	○	1	1	Maximum 128 bytes	Specify the reference URI.

	②mpeg7:InlineMedia	—				
	mpeg7:StreamID	○	0..1	0..1	An integer value within 2-digit decimal 2 bytes	
	ServiceDescription	○	0-∞	0..1	Maximum 128 two-byte characters Maximum 384 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.
	@length	○	0..1	0..1	Select “short,” “medium,” or “long.” Maximum 6 bytes	
	ServiceGenre	○	0-∞	0..20		Genre that characterizes a program organization in a service
	@href	○	1	1	Select from the ARIBGenreCS, ARIBGenreCS2 (when main or secondary is selected for the @type), and another CS (when “other” is selected for the @type, see 2.2 in Annex 4). Maximum 128 bytes	See 1 in Annex 1.
	@type	○	0..1	0..1	Select from “main,” “secondary,” and “other.” Only one genre can be selected as “main.”	Note 2
	ServiceLanguage	○	0-∞	0..1	Select from “ja” (Japanese), “en” (English), “de” (German), “fr” (French) “it” (Italian), “ru” (Russian), “zh” (Chinese), “ko” (Korean), “es” (Spanish), and “xx” (a foreign language other than above,	Corresponds to the language names described in the ISO_639_language_code and ISO_639_language_code_2 in “Table 31-62 Audio component

					multiple languages, unknown language).	descriptor” in ARIB TR-B14
	ParentService	○	0-∞	0..1		
	@serviceIDRef	○	1	1	Maximum 4 bytes	Specify the ServiceId of the service to be referenced (parent).
	ValidPeriod	○	0-∞	0..1		
	ValidFrom	○	0..1	1	Specify in the YYYY-MM-DDThh:m m:ss+09:00 format (25 bytes).	Start date and time when the viewing of the service is active
	ValidTo	○	0..1	0..1	Specify in the YYYY-MM-DDThh:m m:ss+09:00 format (25 bytes).	Final date and time when the viewing of the service is active (includes that date and time)
	CRIDRef	○	0..1	0..1		
	@crid	○	1	1	Maximum 128 bytes	See 8.1. Note 3

Note 1: A ServiceInformation that has the same serviceId and a different fragmentID does not exist. A serviceId remains the same if the fragmentVersion is updated when the fragmentId is the same.

Note 2: If only one genre is used/presented using a content list presenting method, such as ECG and EPG, priority is given to a genre that is specified as “main.” If there is only one Genre element, the @type can be omitted. However, if there are multiple Genre elements, the @type must be described. If the ARIBGenreCS or ARIBGenreCS2 is specified, the first one is specified as main, and the second and subsequent are specified as secondary. “Other” can be specified for any CS other than the CSs stated earlier (see 2.2 in Annex 4). In this case, the processing is dependent on the implemented receiver.

Note 3: Describe the serviceId in the PLT/BroadcastEvent/@serviceIdRef. Or, describe the ProgramInformation/programId of a channel that is specified in the PLT/OnDemandService/@serviceIdRef.

Figure 3-6 shows the reference relationship among the service information elements, broadcast event information elements, and on-demand service information elements (on-demand program information element).

- One service information element can be associated with zero or one broadcast event information element. One broadcast event information element must be associated with one service information element. The reference from a broadcast event information element to a service information element is described in the @serviceIDRef attribute of the broadcast event information element.
- One service information element can be associated with zero or one on-demand service information element. One on-demand service information element must be associated with

one service information element. The reference from an on-demand service information element to a service information element is described in the @serviceIDRef attribute of the on-demand service information element.

- One on-demand service information element must include one on-demand program information element.

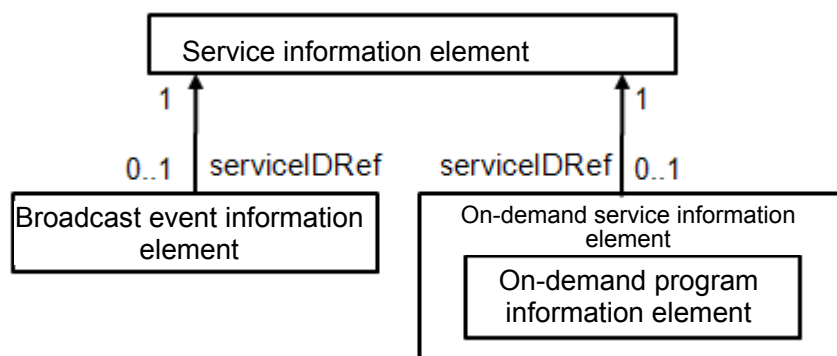


Fig. 3-6: Relationship between service information elements and instance description metadata

3.6 Use of License Reference Information Elements

3.6.1 License reference information elements

The metadata structure, in which the information related to a license is stored, is referred to as a “license reference information element.” The license reference information element is implemented to summarize a license type and the conditions for use (RMPI) after purchasing content and to reference the reference relationship between the license and content. The detailed conditions of use for each piece of content included in a package are described in the license reference information element. When content is selected by a viewer from the EPG/ECG, the purchase information element is referenced for the package content and the license reference information element is referenced for the conditions of use for each piece content that is included in the package.

The license reference information element itself is not protected (encryption, etc.) by digital rights management (DRM). It can be handled with the same security level as other metadata. The license reference information element is the universal information of a license. It doesn't include information on the conditions of use and status of use, which are different for each CAS/DRM client. For instance, if a viewer starts viewing content using a license with conditions for use, in which the viewing period is specified as 8 days and 7 nights, the actual viewing period is set to the date after 8 days and 7 nights counted from the date when the viewer purchased or selected the content. Therefore, in a receiver, the license reference information element is used to refer to the information, such as conditions of use for each content, before obtaining a license. After obtaining a license, the conditions for use and status of use regarding the license are obtained from the license status information stored in the receiver.

3.6.2 Information elements of license reference information elements

Table 3-54 shows the main information elements included in a license reference information element.

Table 3-54: Main information elements of license reference information elements

Information	Mandatory/optional	Description
License ID	Mandatory	The identifier (LicenseID) of a license for the license reference information element; if a license type is a one-layer license, the license ID of a storable broadcasting license functions as the license ID. If a license type is a two-layer license, the program ID of the ECM functions as the license ID.
Main license ID	Mandatory only for a second-layer license	The identifier (MainLicenseID) of a main license, which distinguishes it from the sub-license for the license reference information; the tier bit of a real-time broadcasting license functions the same way as the main license ID.
License type	Mandatory	A type to categorize the layer type of a license; a one-layer license is categorized as “Simple,” while a two-layer license is categorized as “Layered.”
Content identifier	Mandatory	Identifier (CRID) of a piece of content that is a subject of licensing
Purchase identifier	Mandatory	Identifier of a package with a license (PurchaseIdRef)
RMPI description	Mandatory	Summarized description of the conditions for using a license (RMPIDescription). <ul style="list-style-type: none"> • Number of viewable times • Viewable period (start/end) • Viewable period from viewing start time • With or without copy restrictions on outputting to the outside at playback <p>* If the license type is a second-layer license, the summary of the conditions for using a sub-license (ECM) is described.</p>

3.6.3 Use of license reference information elements

Among the elements/attributes defined in “3.2.8 License metadata” in ARIB STD-B38, the elements/attributes provided in Table 3-55 are used in a metadata document in which license reference information elements are stored.

Table 3-55: Use of LicenseInformation

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
LicenseInformation					
@fragmentId	○	0..1	1	10-digit hexadecimal integer value (10 characters)	Note 1 See 4.4.1.
@fragmentVersion	○	0..1	1	14-digit decimal numerical value (14 characters)	Note 1 See 4.4.2.
@fragmentExpirationDate	○	0..1	0..1	Specify in the YYYY-MM-DDThh:mm:ss+09:00 format (25 bytes).	See 7.3.1.
@metadataOriginIDRef	—				
@xml:lang	—				
LicenseID	○	1	1	16-digit hexadecimal integer value (16 characters)	Note 1, Note 2, Note 8
MainLicenseID	○	0..1	0..1	Maximum 32 characters	Note 1, Note 7
LicenseType	○	1	1	Select “Simple” or “Layered.”	
LicenseTypeExtension	—				
AcquisitionTiming	○	0..1	0..1	Select “immediately” or “at use.”	Timing to obtain a licenseImmediate (after purchase)/at use
ProgramCRID	○	1	1		
@crid	○	1	1	Maximum 128 bytes	See 8.1.
PurchaseIdRef	○	1	1	Maximum 24 bytes	Specify a purchase identifier.
RMPIDescription	○	1	1		
TextualDescription	○	0-∞	0..1	Maximum 127 bytes SP, CR, and LF can	Describe a character string that is summarized in the RMPIDescription for presentation in a receiver. Note 3, Note 4

					be used.	See Section 3.1.3.	
		PlayCountsLimit	○	0..1	0..1	Integer value (0 to 100, 3 characters)	Note 4 Number of viewable times; if the setting value of the RMPI is set to “00..00h” (this value indicates that the number of viewable times is not set), this element/attribute is not described.
		PlayCountsThreshold	○	0..1	0..1	5-digit decimal integer value (1 to 65535) Note 6	Threshold when counting the number of viewable times (number of seconds)Note 4 * Mapping from the storable broadcasting license
		ValidityIntervalStart	○	0..1	0..1	Specify in the YYYY-MM-DDThh:mm:ss+09:00 format (25 bytes).	Viewable period start date and time Note 4 If the setting value of the RMPI is set to “FF..FFh” (this value indicates that the viewable period start date and time is not set), this element/attribute is not described.
		ValidityIntervalEnd	○	0..1	0..1	Specify in the YYYY-MM-DDThh:mm:ss+09:00 format (25 bytes).	Viewable period end date and time Note 4 If the setting value of the RMPI is set to “FF..FFh” (this value indicates that the viewable period end date and time is not set), this element/attribute is not described.
		ValidityIntervalAfterFirstUse	○	0..1	0..1	Specify in the PThhHmmMssS (xsd:duration) format. Note 5	Viewable period from viewing start time Note 4 If the setting value of the RMPI is set to “00..00h” (this value indicates that the viewable period is not set), this

						element/attribute is not described.
C		ValidityInterval	—			Validity period from the start date and time when a download playback license is used
		TrickPlayEnabled	○	0..1	0..1	Select “true” or “false.” Whether or not trick play can be performed
		TrickPlayDetail	○	0..1	0..1	2-digit hexadecimal integer value (1 to FF) Whether or not content can be fast forwarded/rewind * Mapping from a storable broadcasting license See “2.3.6.3 Content usage conditions (RMPI)” in Chapter 5 of this operational standard.
P		PreviewDescription	○	0..1	0..1	* Mapping from the ECM's preview control identifier
		CountsLimit	○	0..1	0..1	Integer value (1 to 100, 3 characters) Note 6 Number of previewable times
		TotalTime	○	0..1	0..1	Specify in the PThhHmmMssS (xsd:duration) format. Note 5 Total previewable time * In seconds for ECM
		IntervalEnd	○	0..1	0..1	Specify in the YYYY-MM-DDThh:mm:ss+09:00 format (25 bytes). Preview end time
		OutputRestriction	○	0-∞	0..20	
		Port	○	1	1	
		@href	○	1	1	Select from the ARIBOutputPortCS Maximum 128 bytes See 1 in Annex 1.
		Name	—			
		Definition	—			
		Mode	○	1	1	
		@href	○	1	1	Select from the ARIBCopyControl See 1 in Annex 1.

						MethodCS Maximum 128 bytes	
			Name	—			
			Definition	—			
			ImageConstraintTo ken	○	0..1	0..1	Select “true” or “false.” Resolution control bit * Mapping from a storable broadcasting license
			IssueOption	○	0..1	0..1	License issuance option information * Mapping from a storable broadcasting license
			IssueStartTime	○	0..1	0..1	Specify in the YYYY-MM-DDThh: mm:ss+09:00 format (25 bytes). Date and time when a license is activated
			IssueEndTime	○	0..1	0..1	Specify in the YYYY-MM-DDThh: mm:ss+09:00 format (25 bytes). Date and time when a license is deactivated.
			IssueCountsLimi t	○	0..1	0..1	5-digit decimal integer value (1 to 65535) Note 6 Limit of the number of people to whom licenses can be issued
			IssueAbroadEna bled	○	0..1	0..1	Select “true” or “false.” Whether or not a license can be obtained from abroad; if it is not specified, then “false” (this means that a license cannot be obtained) is specified for this attribute.
			ExportRestriction	—			Export license information
			ExportMedia	—			Media type of an export destination
			@href	—			
			ExportCountsLi mit	—			Export counts
			ChargeOption	—			Billing-related information before use
			@Timing	—			Billing timing
			ChargeMessage	—			Message for presentation

Note 1: LicenseInformation that has the same LicenseID (or set of LicenseID and MainLicenseID) and a different fragmentId does not exist. The LicenseID (or set of LicenseID and MainLicenseID) remains the same if the fragmentVersion is updated when the fragmentId is the same.

Note 2: Describe a LicenseID using a 16-digit hexadecimal integer value. Do not add “0x” or another value that represents a hexadecimal code at the beginning of the ID. If a LicenseID is less than 16 digits, then “0” is added to the beginning of the LicenseID to bring it to 16 digits.

Note 3: Specify a viewing period. For instance, describe as “3 days and 2 nights.”

Note 4: Any of the ValidityIntervalEnd elements, PlayCountsLimit elements, ValidityIntervalAfterFirstUse elements, and TextualDescription elements must be described. If the ValidityIntervalEnd element is described, then the ValidityIntervalStart element can also be described.

Note 5: P: A character that indicates a lapse of time, T: a character that separates date and time, hhH: elapsed 2-digit decimal time, mmM: elapsed 2-digit decimal minute, ssS: elapsed 2-digit decimal second

Note 6: If this element is not described in the PlayCountsThreshold, CountsLimit, and IssueCountsLimit, it is regarded that 0 is described in this element.

Note 7: A business entity identifier is described in the license reference information element of the storable broadcasting content.

Note 8: If the contract type is PPV, the program ID of the ECM is described as the last 4 digits of the license reference information element.

3.7 Use of Segmentation Metadata

This section defines the use of the segment information element and segment group information element of the segmentation metadata defined in “3.2.5 Segmentation metadata” in ARIB STD-B38.

3.7.1 Use of segment information elements

Among the elements/attributes of the segment information element (SegmentInformation) defined in “3.2.5.5 Segment information” in ARIB STD-B38, the elements/attributes provided in Table 3-56 are used.

Table 3-56: Use of SegmentInformation

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
SegmentInformation					
@segmentId	○	1	1	Maximum 128 bytes	Note 1 See 3.7.4.
@fragmentId	○	0..1	1	10-digit hexadecimal integer value (10 characters)	Note 1 See 4.4.1.
@fragmentVersion	○	0..1	1	14-digit decimal numerical value (14 characters)	Note 1 See 4.4.2.

@fragmentExpirationDate	○	0..1	0..1	Specify in the YYYY-MM-DDT hh:mm:ss+09:00 format (25 bytes).	See 7.3.1.
@metadataOriginIDRef	—				
@xml:lang	—				
ProgramRef	○	0..1	1		Specify a program to which a segment belongs.
@crid	○	1	1	Maximum 128 bytes	See 8.1.
TimeBaseReference	—				
Description	○	0..1	0..1		See Table 3-57.
SegmentLocator	○	1	1		See Table 3-64.
KeyFrameLocator	—				

Note 1: A SegmentInformation that has the same SegmentId and a different fragmentId does not exist. The SegmentId remains the same if the fragmentVersion is updated when the fragmentId is the same.

Table 3-57: Use of SegmentInformation/Description

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
Description					
Title	○	0-∞	2..6		See Table 3-58.
Synopsis	○	0-∞	0..3		See Table 3-59.
Genre	○	0-∞	0..30		See Table 3-60.
Keyword	○	0-∞	0..20		See Table 3-61.
RelatedMaterial	○	0-∞	0..20		See Table 3-62.
CreditsList	○	0..1	0..1		See Table 3-63.
RewardCoupon	○	0-∞	0..1		
@userConfirmationRequired	○	0..1	0..1	Select “true” or “false.”	
@number	○	0..1	0..1	2-digit decimal integer value (2 characters)	Describe the number of coupons to be provided. The default is 1.
Select ① or ②.	○	1	1		
①CouponRef	○	1	1		
CouponIDRef	○	1	1	Maximum 24 bytes	
AuthorityName	○	0..1	0..1	Maximum 64 bytes	Authority that provides the coupon.

	②CouponDescription	—			
--	--------------------	---	--	--	--

Table 3-58: Use of SegmentInformation/Description/Title

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
Title				Maximum 128 two-byte characters Maximum 384 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.
@xml:lang	—				
@phoneticTranscription	—				
@phoneticAlphabet	—				
@type	○	0..1	1	Select from “main,” “secondary,” “alternative,” and “popular.” Maximum 12 bytes	

Note: The title element that has the @type specified as “main” is mandatory, and only one such title element exists. The title element that has the @type specified as “alternative” is mandatory, and only one such title element exists. A maximum of one title element can have the @type element that is specified as “secondary.” A maximum of three elements can have the @type that is specified as “popular.” A title character string for presentation is described in the @type specified as “main.” A subtitle character string is described in the @type specified as “secondary.” The ruby of the title character string is described in the @type specified as “alternative,” which is used as a search key and a sort key for sorting. The @type specified as “popular” is the attribute that is specified when there are two or more rubies, which are used not as a sort key but as a search key. Whether or not a ruby as the ruby of a title character string is described in the @type specified as “main” and the @type specified as “secondary” is determined based on the operation. For details on ruby, see “1 Use of *Furigana*” in Annex 4.

Table 3-59: Use of SegmentInformation/Description/Synopsis

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
Synopsis				Maximum 408 two-byte characters Maximum 1224 bytes SP, CR, and LF can be used.	Note 1 See Section 3.1.3.

@xml:lang	—				
@phoneticTranscription	—				
@phoneticAlphabet	—				
@length	○	0..1	0..1	Select from “short,” “medium,” and “long.” Maximum 6 bytes	

Note 1: Maximum 25 characters and maximum 75 bytes when “short” is selected for @length; maximum 100 characters and maximum 300 bytes when “medium” is selected for @length; maximum 408 characters and maximum 1224 bytes when “long” is selected for @length.

Table 3-60: Use of SegmentInformation/Description/Genre

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
Genre					
@href	○	1	1	Select from the ARIBGenreCS, ARIBGenreCS2 (when main or secondary is selected for the @type), and another CS (when “other” is selected for the @type, see 2.2 in Annex 4). Maximum 128 bytes	See 1 in Annex 1.
@type	○	0..1	0..1	Select from “main,” “secondary,” and “other.” Only one genre can be selected as “main.”	Note 1
Name	—				
Definition	—				

Note 1: If only one genre is used/presented using a content list presenting method, such as ECG and EPG, priority is given to the genre that is specified as “main.” If there is only one Genre element, the @type can be omitted. However, if there are multiple Genre elements, the @type must be described. If the ARIBGenreCS or ARIBGenreCS2 is specified, the first one is specified as main, and the second and subsequent are specified as secondary. “Other” can be specified for any CS other than the CSs stated earlier (see 2.2 in Annex 4). In this case, the processing is dependent on the implemented receiver.

Table 3-61: Use of SegmentInformation/Description/Keyword

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
Keyword				Maximum 40 two-byte characters Maximum 120 bytes SP can be used. CR and	See Section 3.1.3.

				LF cannot be used.	
@xml:lang	—				
@phoneticTranscription	—				
@phoneticAlphabet	—				
@type	○	0..1	0..1	Select from “main,” “secondary,” and “other (supplemental).” Only one @type can be selected as “main.”	Note 1

Note 1: In the ECG, a keyword to which “main” or “secondary” is specified can be used for presentation; however, a keyword to which “other” is specified should not be used for presentation. If a type is not specified, then the attribute is regarded as “main.” A type can be omitted when one keyword is used. However, when multiple keywords are used, a type must be described.

Table 3-62: Use of SegmentInformation/Description/RelatedMaterial

Element/attribute name		Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
RelatedMaterial						
	HowRelated	○	0..1	1		
	@href	○	1	1	Select from the ARIBHowRelatedCS, ISDBTMMHowRelatedCS, and the business operator's original HowRelatedCS. Note 1 Maximum 128 bytes	See 3 in Annex 1.
	Name	○	0..1	0..1	Maximum 32 two-byte characters Maximum 96 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.
	Definition	—				
	Format	○	0..1	0..1		
	@href	○	1	1	Select from the ARIBFileFormatCS, ISDBTMMFileFor	See 4 in Annex 1.

					matCS, and the business operator's original FileFormatCS. Note 1 Maximum 128 bytes	
	Name	—				
	Definition	—				
	Select ① or ④.	○	1	1		
①	MediaLocator	○	1	1		
	Select ② or ③.	○	0..1	1		
	② mpeg7:MediaUri	○	1	1	Maximum 128 bytes	Note 2
	③ mpeg7:InlineMedia	—				
	mpeg7:StreamID	—				
④	SegmentReference	○	1	1		Note 3
	@segmentType	○	0..1	1	Select “segment” or “segmentgroup.”	
	@ref	○	1	1	Maximum 128 bytes	Specify a segment ID and a segment group ID.
	PromotionalText	○	0-∞	0..1	Maximum 3000 bytes SP, CR, and LF can be used.	Note 4 See Section 3.1.3.
	@xml:lang	—				
	@phoneticTranscription	—				
	@phoneticAlphabet	—				
	PromotionalMedia	○	0-∞	0..1		Note 5
	TitleImage	○	0..1	1		See Table 3-7.
	TitleVideo	—				
	TitleAudio	—				
	SourceMediaLocator	—				

Note 1: If the business operator's original CS is specified, the processing in a receiver is dependent on the receiver.

Note 2: This attribute specifies the reference URI. The URL of a content or browser script file is specified. The relationship with a reference is specified using a dictionary item of the HowRelatedCS in the HowRelated/@href. The same dictionary item (relationship) can be specified for multiple different references.

Note 3: SegmentReference is used only when referencing a segment from the BML content.

Note 4: Describe the catch copy, copyright, etc., for the substance specified in the MediaLocator/mpeg7:MediaUri for sales promotion.

Note 5: Specify a thumbnail still picture file for the substance specified in the MediaLocator/mpeg7:MediaUri.

Table 3-63: Use of SegmentInformation/Description/CreditsList

Element/attribute name		Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
CreditsList						
	CreditsItem	○	0-∞	1..20		
	@role	○	1	1	Select from the ARIBRoleCS and the business operator's original RoleCS. Maximum 128 bytes	See 1 in Annex 1.
	Select from ①, ⑥, ⑦, and ⑧.	○	0-∞	1..5		
	① PersonName	○	1	1		
	@dateFrom	—				
	@dateTo	—				
	@type	○	0..1	0..1	Select from “main,” “variant,” and “former.” Maximum 7 bytes	Only one element can have the @type specified as “main.” Note 1
	@xml:lang	—				
	Select from ② to ⑤.	○	1-∞	1		
	② mpeg7:GivenName	○	1	1	Maximum 60 two-byte characters Maximum 180 bytes SP can be used. CR and LF cannot be used.	Describe the full name. See Section 3.1.3.
	@xml:lang	—				
	@initial	—				
	@abbrev	○	0..1	0..1	Maximum 16 two-byte characters Maximum 48 bytes SP can be used.	See Section 3.1.3.

									CR and LF cannot be used.	
			③	mpeg7:Family Name	—					
			④	mpeg7:Title	—					
			⑤	mpeg7:Numeration	—					
		⑥		PersonNameIDRef	—					
		⑦		OrganizationName	—					
		⑧		OrganizationNameIDRef	—					
				Character	○	0-∞	0..1			
				@dateFrom	—					
				@dateTo	—					
				@type	—					
				@xml:lang	—					
				Select from ⑨ to ⑫.	○	1-∞	1			
			⑨	mpeg7:GivenName	○	1	1	Maximum 60 two-byte characters Maximum 180 bytes SP can be used. CR and LF cannot be used.	Describe the full name. See Section 3.1.3.	
				@xml:lang	—					
				@initial	—					
				@abbrev	○	0..1	0..1	Maximum 16 two-byte characters Maximum 48 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.	
			⑩	mpeg7:FamilyName	—					
			⑪	mpeg7:Title	—					
			⑫	mpeg7:Numeration	—					

Note 1: The PersonName element that has the @type specified as “main” is mandatory, and only one such element exists, in which a credit character string for presentation is described. Only one PersonName element that has the @type specified as “variant” exists, in which the ruby of the credit character string is described. This attribute is used as a search key and a sort key for sorting. A maximum of three PersonName elements can have the @type that is specified as “former.” The @type specified as “former” is the attribute that is specified when there are two or more rubies. This attribute is used not as a sort key but as a search key. For details

on ruby, see “1 Use of Furigana” in Annex 4.

Table 3-64: Use of SegmentInformation/SegmentLocator

Element/attribute name		Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
SegmentLocator						
Select ① or ②.		○	1	1		
①	MediaRelTimePoint	○	1	1	Maximum 20 bytes	Note 1
②	MediaRelIncrTimePoint	—				
Select ③ or ④.		○	0..1	1		
③	MediaDuration	○	1	1	Maximum 20 bytes	Note 1
④	MediaIncrDuration	—				

Note 1: Describe the MediaRelTimePoint and MediaDuration in the “PThhHmmMssSnnnN1000F” format by providing it accuracy to 1 ms.

3.7.2 Use of segment group information elements

Among the elements/attributes of the segment group information element (SegmentGroupInformation) defined in “3.2.5.6 Segment group information” in ARIB STD-B38, the elements/attributes provided in Table 3-65 are used.

Table 3-65: Use of SegmentGroupInformation

Element/attribute name		Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
SegmentGroupInformation						
@groupId		○	1	1	Maximum 128 bytes	Note 1 See 3.7.4.
@ordered		○	0..1	0..1	Select “true” or “false.”	
@numberOfSegments		○	0..1	0..1	Integer value (0 to 100)	Note 2
@numberOfKeyFrames		—				
@duration		○	0..1	0..1	Specify in the PThhHmmMssS (xsd: duration) format. Note 3	Total playback time for a segment group that is specified in the Segments/@refList

						(in seconds)
	@topLevel	—				
	@fragmentId	○	0..1	1	10-digit hexadecimal integer value (10 characters)	Note 1 See 4.4.1.
	@fragmentVersion	○	0..1	1	14-digit decimal numerical value (14 characters)	Note 1 See 4.4.2.
	@fragmentExpirationDate	○	0..1	0..1	Specify in the YYYY-MM-DDThh:mm:ss+09:00 format (25 bytes).	See 7.3.1.
	@metadataOriginIDRef	—				
	@xml:lang	—				
	ProgramRef	○	1	1		
	@crid	○	1	1		Specify a program to which the segment group belongs. Note 2
	TimeBaseReference	—				
	GroupType	○	1-∞	1		
	@xsi:type	○	1	1	“SegmentGroupType” (Attach a namespace prefix if needed.)	
	@value	○	1	1	Note 4	
	Description	○	0..1	0..1		See Table 3-66.
	GroupInterval	—				
	Select ① or ②.	○	0..1	1		
	① Segments	○	1	1		
	@refList	○	1	1	Maximum 25600 bytes	Note 5
	② Groups	—				
	@refList	—				
	KeyFrameLocator	—				

Note 1: A SegmentGroupInformation that has the same groupId and a different fragmentId does not exist. The groupId remains the same if the fragmentVersion is updated when the fragmentId is the same.

Note 2: Specify “crid://.x/x” if a segment group is not associated to a program or is associated with multiple programs.

Note 3: P: A character that indicates a lapse of time, T: a character that separates date and time, hhH: elapsed 2-digit decimal time, mmM: elapsed 2-digit decimal minute, ssS: elapsed 2-digit decimal second

Note 4: Select from “highlights,” “highlights/objects,” “highlights/events,” “bookmarks,” “bookmarks/objects,” “bookmarks/events,” “themeGroup,” “preview,” “preview/title,” “preview/slideshow,” “tableOfContents,” “synopsis,” “shots,” “insertionPoints,” “alternativeGroups,” and “other.” For details on each type, see “3.2.5.6 Segment group” in ARIB STD-B38.

Note 5: Describe segmentIDs for the number of @numberOfSegments (maximum 100 segments) continuously by using a delimiter as a space.

Table 3-66: Use of SegmentGroupInformation/Description

Element/attribute name		Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
Description						
	Title	○	0-∞	2..6		See Table 3-67.
	Synopsis	○	0-∞	0..3		See Table 3-68.
	Genre	○	0-∞	0..15		See Table 3-69.
	Keyword	○	0-∞	0..13		See Table 3-70.
	RelatedMaterial	—				
	CreditsList	○	0..1	0..1		See Table 3-71.
	RewardCoupon	—				
	@userConfirmationRequired	—				
	@number	—				
	Select ① or ②.	—				
	①CouponRef	—				
	CouponIdRef	—				
	AuthorityName	—				
	②CouponDescription	—				

Table 3-67: Use of SegmentGroupInformation/Description/Title

Element/attribute name		Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
	Title				Maximum 128 two-byte characters Maximum 384 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.
	@xml:lang	—				
	@phoneticTranscription	—				
	@phoneticAlphabet	—				
	@type	○	0..1	1	Select from “main,” “secondary,” “alternative,” and “popular.” Maximum 12 bytes	

Note: The title element that has the @type specified as “main” is mandatory, and only one such title element exists. The title element that has the @type specified as “alternative” is mandatory, and only one such title element exists. A maximum of three elements can have the @type that is specified as “popular.” A maximum of one title element can have the @type element that is specified as “secondary.” A title character string for presentation is described in the @type specified as “main.” A subtitle character string for presentation is described in the @type specified as “secondary.” The ruby of the title character string is described in the @type specified as “alternative,” which is used as a search key and a sort key for sorting. The @type specified as “popular” is the attribute that is specified when there are two or more rubies, which are used not as a sort key but as a search key. Whether or not a ruby as the ruby of a title character string is described in the @type specified as “main” and the @type specified as “secondary” is determined based on the operation. For details on ruby, see “1 Use of *Furigana*” in Annex 4.

Table 3-68: Use of SegmentGroupInformation/Description/Synopsis

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
Synopsis				Maximum 408 two-byte characters Maximum 1224 bytes SP, CR, and LF can be used.	Note 1 See Section 3.1.3.
@xml:lang	—				
@phoneticTranscription	—				
@phoneticAlphabet	—				
@length	○	0..1	0..1	Select from “short,” “medium,” and “long.” Maximum 6 bytes	

Note 1: Maximum 25 characters and maximum 75 bytes when “short” is selected for @length; maximum 100 characters and maximum 300 bytes when “medium” is selected for @length; maximum 408 characters and maximum 1224 bytes when “long” is selected for @length

Table 3-69: Use of SegmentGroupInformation/Description/Genre

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
Genre					
@href	○	1	1	Select from the ARIBGenreCS, ARIBGenreCS2 (when main or secondary is selected for the @type), and another CS (when “other” is selected for the @type, see 2.2 in Annex 4).	See 1 in Annex 1.

				Maximum 128 bytes	
@type	○	0..1	0..1	Select from “main,” “secondary,” and “other.” Only one genre can be selected as “main.”	Note 1
Name	—				
Definition	—				

Note 1: If only one genre is used/presented using a content list presenting method, such as ECG and EPG, priority is given to the genre that is specified as “main.” If there is only one Genre element, the @type can be omitted. However, if there are multiple Genre elements, the @type must be described. If the ARIBGenreCS or ARIBGenreCS2 is specified, the first one is specified as main, and the second and subsequent are specified as secondary. “Other” can be specified for any CS other than the CSs stated earlier (see 2.2 in Annex 4). In this case, the processing is dependent on the implemented receiver.

Table 3-70: Use of SegmentGroupInformation/Description/Keyword

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
Keyword				Maximum 40 two-byte characters Maximum 120 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.
@xml:lang	—				
@phoneticTranscription	—				
@phoneticAlphabet	—				
@type	○	0..1	0..1	Select from “main,” “secondary,” and “other (supplemental).” Only one keyword can be selected as “main.”	Note 1

Note 1: In the ECG, a keyword to which “main” or “secondary” is specified can be used for presentation; however, a keyword to which “other” is specified should not be used for presentation. If a type is not specified, then the attribute is regarded as “main.” A type can be omitted when one keyword is used. However, when multiple keywords are used, a type must be described.

Table 3-71: Use of SegmentGroupInformation/Description/CreditsList

Element/attribute name		Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
CreditsList						
	CreditsItem	○	0-∞	1..20		
	@role	○	1	1	Select from the ARIBRoleCS and the business operator's original RoleCS. Maximum 128 bytes	See 1 in Annex 1.
	Select from ①,⑥, ⑦, and ⑧.	○	0-∞	1..5		
	① PersonName	○	1	1		
	@dateFrom	—				
	@dateTo	—				
	@type	○	0..1	0..1	Select from “main,” “variant,” and “former.” Maximum 7 bytes	Only one element can have the @type specified as “main.” Note 1
	@xml:lang	—				
	Select from ② to ⑤.	○	1-∞	1		
	② mpeg7:GivenName	○	1	1	Maximum 60 two-byte characters Maximum 180 bytes SP can be used. CR and LF cannot be used.	Describe the full name. See Section 3.1.3.
	@xml:lang	—				
	@initial	—				
	@abbrev	○	0..1	0..1	Maximum 16 two-byte characters Maximum 48 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.

		③	mpeg7:Family Name	—				
		④	mpeg7>Title	—				
		⑤	mpeg7:Numeration	—				
	⑥		PersonNameIDRef	—				
	⑦		OrganizationName	—				
	⑧		OrganizationNameIDRef	—				
			Character	○	0-∞	0..1		
			@dateFrom	—				
			@dateTo	—				
			@type	—				
			@xml:lang	—				
			Select from ⑨ to ⑫.	○	1-∞	1		
		⑨	mpeg7:GivenName	○	1	1	Maximum 60 two-byte characters Maximum 180 bytes SP can be used. CR and LF cannot be used.	Describe the full name. See Section 3.1.3.
			@xml:lang	—				
			@initial	—				
			@abbrev	○	0..1	0..1	Maximum 16 two-byte characters Maximum 48 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.
		⑩	mpeg7:FamilyName	—				
		⑪	mpeg7>Title	—				
		⑫	mpeg7:Numeration	—				

Note 1: The PersonName element that has the @type specified as “main” is mandatory, and only one such element exists, in which a credit character string for presentation is described. Only one PersonName element can have the @type specified as “variant,” in which the ruby of the credit character string is described. This attribute is used as a search key and a sort key for sorting. A maximum of three PersonName elements can have the @type that is specified as “former.” The @type specified as “former” is the attribute that is specified when there are two or more rubies. This attribute is used not as a sort key but as a search key. For details on ruby, see “1 Use of *Furigana*” in Annex 4.

3.7.3 Reference relationship between program information elements and segmentation metadata

Figure 3-7 shows the reference relationship among the program information elements, segment information elements, and segment group information elements.

- One program information element can be associated with 0 to 100 segment information elements. One segment information element must be associated with one program information element. The reference from a segment information element to a program information element is described in the ProgramRef/@crid attribute of the segment information element.
- One segment information element can be associated with 0 to 10 segment group information elements. One segment group information element must be associated with 1 to 100 segment information elements. The reference from a segment group information element to a segment information element is described in the Segments/@refList attribute of the segment group information element.

Note: The reference to one program information element (ProgramRef/@crid attribute) must be described in one segment group information element (see Table 3-65). However, if the element is not associated with the program or is associated with multiple programs, “crid://.x/x” is specified.

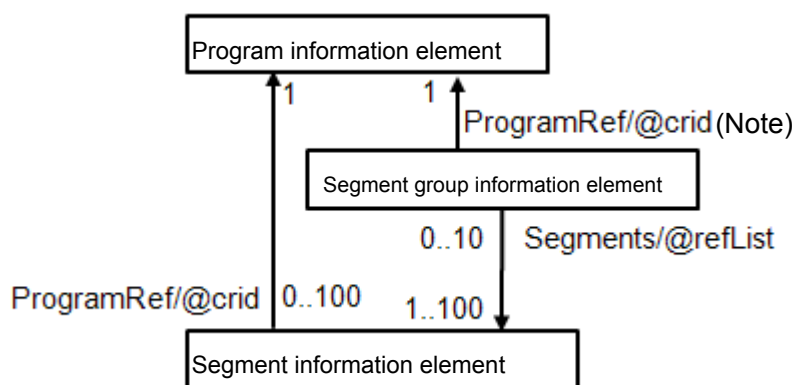


Fig. 3-7: Relationship among the program information elements, segment information elements, and segment group information elements

3.7.4 Use of segmentID and segment groupID

A segmentID that uniquely identifies a segment is described in the SegmentInformation/@segmentId attribute. A segment groupID that uniquely identifies a segment group is described in the SegmentGroupInformation/@groupId attribute. A segmentID and segment groupID are expressed in a unique character string that shares a domain within the range of <authority> (see 8.1.2 in this volume), and they are not reused. The following shows the characters that can be used for segmentID and segment groupID.

When comparing a segment ID with a segment group ID, uppercase and lowercase letters are not distinguished.

<data>	=	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”

3.8 Use of Coupon Description Information

3.8.1 Coupon description information

A metadata structure, in which information related to a coupon is stored, is referred to as “coupon description information.” Coupon description information is used to refer to coupon details (discount rate, valid period, number of valid coupons), reference relationships between the coupon and purchase information element, and the reference relationship among the coupon, program information element, and group information element.

The coupon description information is used to describe universal information about coupons. The status of coupons offered per user is separately managed. A client management server or billing server is used to offer a coupon to a user and to offer a discount with a coupon. Therefore, in a receiver, the coupon description information is used to reference the coupon information that is related to each piece of content and item and the coupon information that can be used when purchasing an item or viewing/using content, etc.

3.8.2 Use of coupon description information

Among the elements/attributes defined in “3.2.9 Coupon description metadata” in ARIB STD-B38, the elements/attributes provided in Table 3-72 are used for metadata documents in which coupon description information is stored.

Table 3-72: Use of CouponDescription

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
CouponDescription					
@couponId	○	1	1	Maximum 24 bytes	Note 1
@fragmentId	○	0..1	1	10-digit hexadecimal integer value (10 characters)	Note 1 See 4.4.1.
@fragmentVersion	○	0..1	1	14-digit decimal numerical value (14 characters)	Note 1 See 4.4.2.
@fragmentExpirationDate	○	0..1	0..1	Specify in the YYYY-MM-DDThh:mm:ss+09:00 format (25 bytes).	See 7.3.1.
@authorityName	○	0..1	0..1	Maximum 64 bytes	Describe an authority that has responsibility for coupons (the same as the authority for a CRID).
@requiredNumber	—				Note 2
@start	○	0..1	1	Specify in the YYYY-MM-DDThh:mm:ss+09:00 format (25 bytes).	Start date and time of the coupon valid period
@end	○	0..1	0..1	Specify in the YYYY-MM-DDThh:mm:ss+09:00 format (25 bytes).	End date and time of the coupon valid period
@acquisitionMode	○	0..1	0..1	Select “immediate” or “linkedToContent.”	A method to obtain a coupon The default is linkedToContent (at purchase/use/viewing).
@metadataOriginIDRef	—				
@xml:lang	—				
RequiredCoupon	—				
@couponId	—				
@authorityName	—				
@requiredNumber	—				

	Select from ①, ②, and ③. Note7	○	0..1	0..1		
	NonTVACouponRef	○	1	1	Maximum 128 bytes	Describe a URI to refer to a coupon that is offered with an item other than TV-Anytime or with a service.
	@couponAuthority	○	1	1	128 two-byte characters SP can be used. CR and LF can be used.	Describe who provided the coupon. See Section 3.1.3.
	②CouponValue	○	1	1		
	@purchaseIdRef	○	0..1	1	Maximum 24 bytes	Specify the purchaseId to which a coupon is applied.
	Select from ④, ⑤ and ⑥.	○	1	1		
	④ ReductionAmount	○	0~∞	1	An integer value within 10-digit decimal (10 characters)	Describe a discount price.
	@currency	○	1	1	“JPY”	
	⑤NewPrice	○	0~∞	1	An integer value within 10-digit decimal (10 characters)	Describe a new price.
	@currency	○	1	1	“JPY”	
	⑥ ReductionPercentage	○	0..1	1	An integer value within 3-digit decimal (3 characters)	Describe a discount rate in a numerical value.
	③ContentTarget	○	0~∞	0..1		Describe the attribute information of the content that can use the coupon.
	Genre	○	0~∞	0..40		
	@href	○	1	1	Select from the ARIBGenreCS, ARIBGenreCS2 (when main or secondary is specified for the @type), and	See 1 in Annex 1

					another CS (when “other” is specified for @type, see 2.2 in Annex 4). Maximum 128 bytes	
		@type	○	0..1	0..1	Select from “main,” “secondary,” and “other.” Note 3
		Program	○	0-∞	0..100	
		@crid	○	1	1	Maximum 128 bytes See 8.1.
		ServiceIdRef	○	0-∞	0..1	Maximum 100 bytes Note 4
		CouponURL	○	0..1	0..1	Maximum 128 bytes URI that is used to obtain coupon information or to update the coupon status in a receiver
		CouponText	○	0-∞	0..1	Maximum 200 two-byte characters Maximum 600 bytes SP, CR, and LF can be used. See Section 3.1.3.
		CRIDRef	○	0..1	0..1	
		@crid	○	1	1	Maximum 128 bytes See 8.1. Note 5
		CouponName	○	0..1	0..1	Maximum 128 two-byte characters Maximum 384 bytes SP can be used. CR and LF cannot be used. Describe a coupon name. See Section 3.1.3.
		CouponImage	○	0..1	0..1	Note 6
		mpeg7:MediaUri	○	1	1	Maximum 128 bytes
		CopyrightNotice	○	0-∞	0..1	Maximum 50 two-byte characters Maximum 150 bytes See Section 3.1.3.

					SP can be used. CR and LF cannot be used.	
	CouponThumbnail	○	0..1	0..1		Note 6
	mpeg7:MediaUri	○	1	1	Maximum 128 bytes	
	CopyrightNotice	○	0-∞	0..1	Maximum 50 two-byte characters Maximum 150 bytes SP can be used. CR and LF cannot be used.	See Section 3.1.3.

Note 1: A CouponDescription that has the same couponId and a different fragmentId does not exist. The couponId remains the same if the fragmentVersion is updated when the fragmentId is the same.

Note 2: The requiredNumber is a number at the PuI/Requiredcoupon/@requiredNumber side.

Note 3: Unlike the description rule in content, “main” and “secondary” can be specified multiple times. If the @type is omitted, then “main” is specified as the @type.

Note 4: Describe serviceIds continuously by using a delimiter as space.

Note 5: Describe the CRID (the CRID described in PuI/CRIDRef or SegI/ProgramRef) of a purchase information element with a couponID, which is described in PuI/RewardCoupon/CouponRef/CouponIdRef or SegI/RewardCoupon/CouponRef/CouponIdRef, the CRID of the program information element that corresponds to the segment information element, or the CRID of the group information element.

Note 6: The CouponImage specifies still images used on a screen that display coupon details, and the Coupon Thumbnail specifies thumbnail images used on the list display screen. When specifying these images, they are not encrypted. When displaying the images, it can be displayed by scaling within the extent that the impression of the original picture is not largely altered. However, the aspect ratio shall be maintained, and the picture shall not be trimmed. See “8.2.3.3 Location resolution in a receiver” for the correspondence between the reference information of each resource described in metadata (e.g., http://example.jp/a/v/c.png) and the storage locations of an applicable resource on a broadcasting stream and in a receiver.

Note 7: CouponValue and ContentTarget are used as a pair. They can be used by describing only the CouponValue. However, if the ContentTarget is described, the CouponValue also needs to be described.

Note: As for the CRID unit metadata document (fragments associated with the same CRID are stored in a single TVAMain in PGSET format), the CouponDescription is described in a metadata document with an applicable CouponId that is described in the PuI/RewardCoupon or SegI/RewardCoupon. In addition to the FragmentExpirationDate of a piece of content or group described in the metadata document, the expiration date of the coupon (such as the expiration date of the item information to which a coupon is applied) is set to the FragmentExpirationDate of the CouponDescription.

3.8.3 Reference relationship among the program information elements, item information elements, segment information elements, and coupon metadata

Figure 3-8 shows the reference relationship among the program information elements, group information elements, item information elements, segment information elements, and coupon description elements.

- One item information element (RewardCoupon) can be associated with 0 to 1 coupon description element. The reference from an item information element (RewardCoupon) to a

coupon description element is described in the CouponRef/CouponIdRef of the item information element (RewardCoupon).

- One item information element (RequiredCoupon) can be associated with 0 to 1 coupon description element. The reference from an item information element (RequiredCoupon) to a coupon description element is described in the @couponId of the item information element (RequiredCoupon).
- One segment information element can be associated with 0 to 1 coupon description element. The reference from a segment information element to a coupon description element is described in the RewardCoupon/CouponRef/CouponIdRef of the segment information element.
- One coupon description element can be associated with 0 to 1 item information element. The reference from a coupon description element to an item information element is described in the CouponValue/@purchaseIdRef of the coupon description element.

One coupon description element can be associated with 0 to X program information elements, group information elements, and service information elements. The reference from a coupon information element to a program information element, group information element, and service information element is described in the ContentTarget/Genre/@href, ContentTarget/Program/@crid, and ContentTarget/ServiceIdRef of the coupon description element. Zero to 10 ContentTargets can be described in one coupon description element, and 0 to 40 Genres, 0 to 100 Programs, and 100 bytes of ServiceIdRef can be described in one ContentTarget (by separating them with a comma [,]) (see Table 3-72).

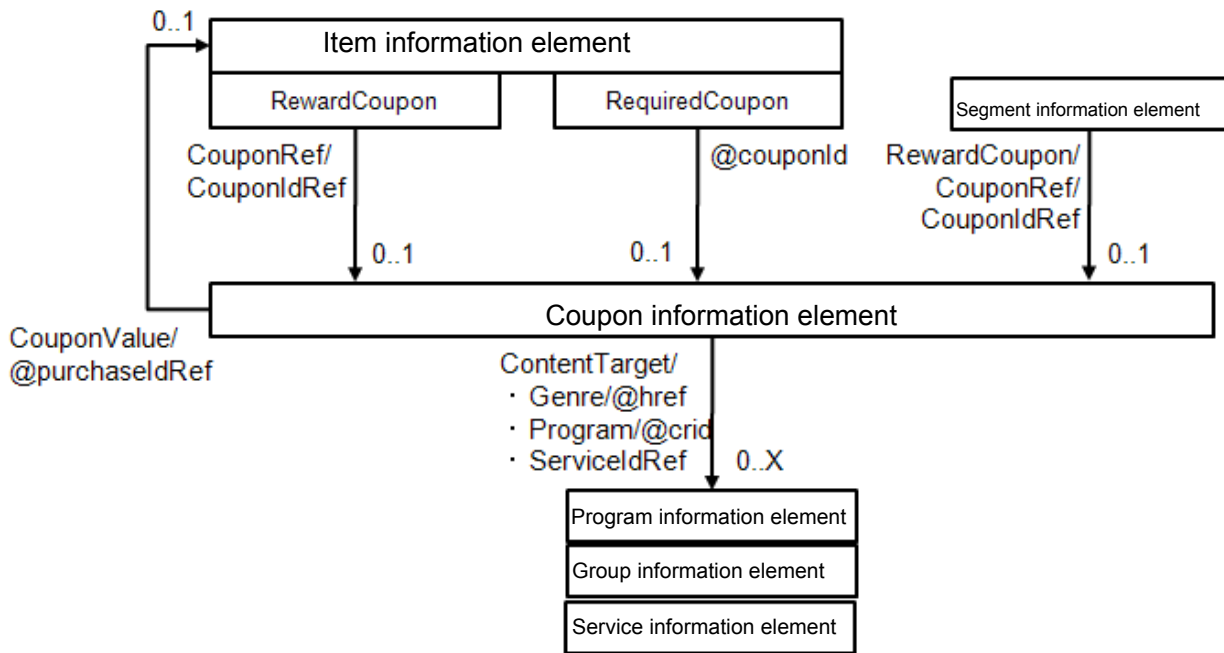


Fig. 3-8: Relationship among the program information elements, item information elements, segment information elements, and coupon description elements

3.9 Others

3.9.1 Use of credit information elements

Among the elements/attributes defined in the credit information (CreditsInformationTableType) in “3.2.10.1 Information table” in ARIB STD-B38, the elements/attributes provided in Table 3-73 are used. The credit information is only used as an actor search (getactor) result and is not described in the content metadata. For details, see 6.4 “Reception of Search Results.”

Table 3-73: Use of CreditsInformationTable

Element/attribute name	Use	Number of appearances (B38)	Number of appearances (use)	Character number limit Value limit	Remarks
CreditsInformationTable					
@copyrightNotice	—				
@metadataOriginIDRef	—				
@xml:lang	—				
Select ① or ②.	—				
PersonName ①	—				
OrganizationName ②	—				

	CreditsItem	○	—	0..X Note 1		
	@role	○	1	1	Select from the ARIBRoleCS and ISDBTMMRoleCS. Maximum 128 bytes	See 1 in Annex 1.
	Select from ①, ⑥, ⑦, and ⑧.	○	0-∞	1..5		
	①PersonName	○	1	1		
	@dateFrom	—				
	@dateTo	—				
	@type	○	0..1	0..1	Select from “main,” “variant,” and “former.” Maximum 7 bytes	Only one element can have “main” as @type. Note 2
	@xml:lang	—				
	Select from ② to ⑤.	○	1-∞	1		
	②mpeg7:GivenName	○	1	1	Maximum 60 two-byte characters Maximum 180 bytes SP can be used. CR and LF cannot be used.	Describe the full name. See Section 3.1.3.
	@xml:lang	—				
	@initial	—				
	@abbrev	—				
	③mpeg7:FamilyName	—				
	④mpeg7:Title	—				
	⑤mpeg7:Numeration	—				
	⑥PersonNameIDRef	—				
	⑦OrganizationName	—				
	⑧OrganizationNameIDRef	—				
	Character	—				

Note 1: X corresponds to the number of search results returned per getactor search. It is set separately by the service operator.

Note 2: The PersonName element that has the @type specified as “main” is mandatory, and only one such element exists, in which a credit character string for presentation is described. Only one PersonName element can have the @type specified as “variant,” in which the ruby of the credit character string is described. This attribute is used as a search key and a sort key for sorting. A maximum of three PersonName elements can

have the @type specified as “former.” The @type specified as “former” is the attribute that is specified when there are two or more rubies. This attribute is used not as a sort key but as a search key. For details on ruby, see “1 Use of Furigana” in Annex 4.

3.9.2 Correspondence between EIT and metadata

In this operational standard, schemas are extended, taking into account not only the creation of an electronic program guide (EPG) using the existing program arrangement information (SI), but also the creation of an EPG using metadata. The following table shows the correspondence between each field of the EIT (Event Information Table) [schedule], which is described in digital terrestrial television broadcasting (TR-B14) and BS/wide band CS digital broadcasting (TR-B15), along with each element/attribute of the metadata.

In the table, the correspondence between metadata and the event group descriptor, component group descriptor, time shift event descriptor, or carousel-compatible composite descriptor is not described, as those descriptors are not used in multimedia broadcasting.

Abbreviation In the following table, each XPATH is abbreviated as follows:

- /TVAMain/ProgramDescription/ProgramLocationTable/BroadcastEvent→”PLT/BE”
- /TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation→”PIT/PI”
- /TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/
BasicDescription→”PIT/PI/BD”
- /TVAMain/ProgramDescription/GroupInformationTable/GroupInformation→”GIT/GI”
- /TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/
BasicDescription→”GIT/GI/BD”
- /TVAMain/ProgramDescription/LicenseInformationTable/LicenseInformation→”LIT/LI”

Table 3-74: Correspondence between EIT and metadata (EIT body)

EIT field	Element/attribute of metadata	Supplemental note
Table_id	None	No need to map this field because it is used to identify an EIT in SI
section_syntax_indicator	None	Fixed to “1” in TR-B14 and B15
reserved_future_use	—	
Reserved	—	
section_length	None	No mapping
service_id	PLT/BE/@serviceIdRef PIT/PI/@programId	Note 1
reserved	—	

version_number	None	PI/@fragmentVersion indicates the date and time when metadata is created.
current_next_indicator	None	Fixed to “1” in TR-B14 and B15
section_number	None	Not used
last_section_number	None	Not used
transport_stream_id	PLT/BE/ProgramURL	Note 2
original_network_id	PLT/BE/ProgramURL	Note 2
segment_last_section_number	None	Not used
last_table_id	None	Not used
event_id	PIT/PI/@programId	Note 1
start_time	PLT/BE/PublishedStartTime	
duration	PLT/BE/PublishedDuration	
running_status	None	Always describe “0” (not defined) in TR-B14 and B15.
free_CA_mode	PLT/BE/Free/@value	
descriptors_loop_length	None	Not used
CRC_32	None	Not used

Note 1: The CRID of real-time broadcasting content is used as a character string including <service_id> and <event_id> (e.g., crid: //<authority>/<data>/<service_id>/<event_id>).

Note 2: A URL to view an applicable program is described in the ProgramURL. The ProgramURL is used by including <transport_stream_id> and <original_network_id> in the URL (e.g., arib://<original_network_id>.<transport_stream_id>/<service_id>/<event_id>).

Table 3-75: Correspondence between EIT and metadata (short-format event descriptor)

EIT field	Element/attribute of metadata	Supplemental note
descriptor_tag	None	Fixed to “0x40” in TR-B14 and B15
descriptor_length	None	No mapping
ISO_639_language_code	None	
event_name_length	None	No mapping
event_name_char	PIT/PI/BD/Title[@type=“main”]	
text_length	None	No mapping
text_char	PIT/PI/BD/Synopsis[@length=“medium”]	

Table 3-76: Correspondence between EIT and metadata (component descriptor)

EIT field	Element/attribute of metadata	Supplemental note
descriptor_tag	None	Fixed to “0x50” in TR-B14 and B15
descriptor_length	None	No mapping
stream_content	None	Fixed to “0x01” (video) in TR-B14 and B15

component_type	PIT/PI/AVAttributes/videoAttributes/ ./HorizontalSize, ./VerticalSize, ./AspectRatio	
component_tag	PIT/PI/AVAttributes/videoAttributes/ StreamDescription/@id	
ISO_639_language_code	None	Fixed to “jpn” in TR-B14 and B15
text_char	PIT/PI/AVAttributes/videoAttributes/ StreamDescription	

Table 3-77: Correspondence between EIT and metadata (audio component descriptor)

EIT field	Element/attribute of metadata	Supplemental note
descriptor_tag	None	Fixed to “0xC4” in TR-B14 and B15
descriptor_length	None	No mapping
stream_content	None	Fixed to “0x2” (audio) in TR-B14 and B15
component_type	PIT/PI/AVAttributes/AudioAttributes/ MixType/@href PIT/PI/AVAttributes/AudioAttributes/ NumOfChannles	
component_tag	PIT/PI/AVAttributes/AudioAttributes/ StreamDescription/@id	
stream_type	PIT/PI/AVAttributes/AudioAttributes/ ./Coding/@href	
simulcast_group_tag	None	Not used
ES_multilingual_flag	None Note 1	
main_component_flag	PIT/PI/AVAttributes/AudioAttributes/ StreamDescription/@type	
quality_indicator	PIT/PI/AVAttributes/AudioAttributes/ BitRate	
sampling_rate	PIT/PI/AVAttributes/AudioAttributes/ SamplingRate	
ISO_639_language_code	PIT/PI/AVAttributes/AudioAttributes/ AudioLanguage	
ISO_639_language_code2	PIT/PI/AVAttributes/AudioAttributes/ AudioLanguage	Second AudioLanguage
text_char	PIT/PI/AVAttributes/AudioAttributes/ AudioLanguage/@description	

Note 1: If two AudioLanguages are described after PIT/PI/AudioAttributes/, it is regarded that “dual mono stereo” and “bilingual and multiplex broadcasting” are used.

Table 3-78: Correspondence between EIT and metadata (content descriptor)

EIT field	Element/attribute of metadata	Supplemental note
descriptor_tag	None	Fixed to “0x54” in TR-B14 and B15
descriptor_length	None	No mapping
content_nibble_level_1	PIT/PI/BD/Genre[@type=“main”]	Specify the ARIBGenreCS or ARIBGenreCS2.
content_nibble_level_2	PIT/PI/BD/Genre[@type=“secondary”]	
user_nibble	PIT/PI/BD/Keyword[@type=“other”]	Describe in the EIT content_un:x (x is arbitrary numerical value) format. For details, see 2.3 in Annex 4.

Table 3-79: Correspondence between EIT and metadata (parental rate descriptor)

EIT field	Element/attribute of metadata	Supplemental note
descriptor_tag	None	Fixed to “0x55” in TR-B15
descriptor_length	None	No mapping
country_code	None	Fixed to “jpn” in TR-B15
rating	PIT/PI/BD/ParentalGuidance/mpeg7:ParentalRating/@href	Specify ARIBParentalRatingCS.

Table 3-80: Correspondence between EIT and metadata (digital copy control descriptor)

EIT field	Element/attribute of metadata	Supplemental note
descriptor_tag	None	Fixed to “0xC1” in TR- B15
descriptor_length	None	No mapping
digital_recording_control_data	Mode/@href of LIT/LI/RMPIDescription/OutputRestriction/Port[@href=“ARIBOutputPortCS/digital_serial”]	Specify ARIBCopyControlMethodCS.
maximum_bit_rate_flag	None	Not used
component_control_flag	None	Not used * The PMT is used for the entire program.
copy_control_type	None	Note 1
APS_control_data	/Mode/@href of LIT/LI/RMPIDescription/OutputRestriction/Port[@href=“ARIBOutputPortCS/component”]	Specify ARIBCopyControlMethodCS.
maximum_bit_rate	PIT/PI/AVAttributes/BitRate	
component_control_length	None	No mapping

Note 1: Reference the digital_recording_control_data and APS_control_data when mapping.

Table 3-81: Correspondence between EIT and metadata (CA contract information descriptor)

EIT field	Element/attribute of metadata	Supplemental note
descriptor_tag	None	Fixed to “0xCB” in TR- B15
descriptor_length	None	No mapping
CA_system_id	None	Fixed value that indicates BS in TR-B15
CA_unit_id	None	Fixed value
num_of_component	None	Determine whether this field is valid or invalid in an entire applicable descriptor based on the following attribute setting.
component_tag	None	No distinguishing between components
contract_verification_info_length	None	No mapping
contract_verification_info	Note 1	
fee_name_length	None	Fixed to “0” in TR-B15
fee_name	None	Not used in TR-B15

Note 1: The contract verification information managed in a terminal is described. For instance, the information can be managed using a character string created by combining a LIT/LI/MainLicenseID and LI/LicenseID. The service business operator can decide how to manage the information.

Table 3-82: Correspondence between EIT and metadata (series descriptor)

EIT field	Element/attribute of metadata	Supplemental note
descriptor_tag	None	Fixed to “0xD5” in TR-B14 and B15
descriptor_length	None	No mapping
series_id	GIT/GI/@groupId Note 1 MediaLocator/mpeg7:MediaUri of GIT/GI/RelatedMaterial/HowRelated/[@ href=“ISDBTMMHowRelatedCS/22”]	
repeat_label	PLT/BE/RepeatLabel	
program_pattern	PIT/PI/Keyword[@type=“other”]	Describe in the EIT series_pp:x (x is an arbitrary numerical value) format. For details, see 2.3 in Annex 4.
expire_date_valid_flag	None	No mapping
expire_date	GIT/GI/Period[@type=“display”]/End	
episode_number	PIT/PI/MemberOf/@index Note 2	
last_episode_number	GIT/GI/@numOfItems	
series_name_char	GIT/GI/BD/Title[@type=“main”]	

Note 1: Series metadata is metadata in which information related to series is described. The GIT/GI/@groupType is specified as a “series.” A CRID is created as a character string including <series_id> (e.g., crid://<authority>/<data>/<series_id>).

Note 2: MemberOf/@index that has a CRID (CRID of series GI), which is described in the MediaLocator/mpeg7:MediaUri of the RelatedMaterial/HowRelated[@href=“ISDBTMMHowRelatedCS/23”], in the MemberOf/@crid

Table 3-83: Correspondence between EIT and metadata (extension format event descriptor)

EIT field	Element/attribute of metadata	Supplemental note
descriptor_tag	None	Fixed to “0x4E” in TR-B14 and B15
descriptor_length	None	No mapping
descriptor_number	None	No mapping
last_descriptor_number	None	No mapping
ISO_639_language_code	None	Same as the ISO_639_language_code of a short-format event descriptor
length_of_items	None	No mapping
item_description_length	None	No mapping
item_description_char	PI/BD/Synopsis[@length=“long”] Note 1	
item_length	None	No mapping
item_char	PI/BD/Synopsis[@length=“long”] Note 1	
text_length	None	No mapping
text_char	None	Not used in TR-B14 and 15

Note 1: Mapping a character string created as “<item_description_char>:<item_char>” by combining them with a period (.); the service business operator can decide whether or not to map to a CreditsItem with reserved words in <item_char>.

Table 3-84: Correspondence between EIT and metadata (hyperlink descriptor)

EIT field	Element/attribute of metadata	Supplemental note
descriptor_tag	None	Fixed to “0xC5” in TR- B14 and B15
descriptor_length	None	No mapping
hyper_linkage_type	PIT/PI/BD/RelatedMaterial/ HowRelated/@href	Specify ISDBTMMHowRelatedCS/24.X. * X is selected from a defined value (see 3 in Annex 1).
link_destination_type	PIT/PI/BD/RelatedMaterila/ Format/@href	Specify ISDBTMMFileFormatCS/eit.Y. * Y is selected from a defined value (see 4 in Annex 1).
selector_length	None	No mapping
selector_byte	PIT/PI/BD/RelatedMaterila/ MediaLocator/mpeg7:MediaUri	

private_data	PIT/PI/BD/RelatedMaterial/ HowRelated/Name	Mapping when the hyper_linkage_type is specified as “0x0A” (Authority)
--------------	---	--

Table 3-85: Correspondence between EIT and metadata (data content descriptor)

EIT field	Element/attribute of metadata	Supplemental note
descriptor_tag	None	Fixed to “0xC7” in TR-B14 and B15
descriptor_length	None	No mapping
data_component_id	None	See PMT.
entry_component	None	See PMT.
selector_length	None	No mapping
selector_byte	None	Not used
num_of_component_ref	None	Not used
component_ref	None	Not used
ISO_639_language_code	PIT/PI/BD/CaptionLanguage	
text_length	None	No mapping
text_char	PIT/PI/BD/CaptionLanguage/@description	
arib_bxml_info()		
transmission_format	As this element is not used, mapping to metadata is not necessary.	Fixed to “00” in TR-B14 and B15
auto_start_flag		Fixed to “0” in TR-B14 and B15
document_resolution		
use_xml		Fixed to “0” in TR-B14 and B15
default_version_flag		No mapping
independent_flag		
content_id_flag		No mapping
associated_content_flag		
style_for_tv_flag		
update_flag		Fixed to “0” in TR-B14 and B15
ISO_639_language_code		
content_id		
content_version		
bml_major_version bml_minor_version		Connect with a period (.)
ondemand_retrieval_flag file_storable_flag		
arib_carousel_info()		
num_of_carousels	As this element is not used, mapping	

component_tag	to metadata is not necessary.	
event_section_flag		Fixed to “1” in TR-B15
component_size_flag		No mapping
default_transaction_id_flag		No mapping
default_timeoutDII_flag		No mapping
default_leak_rate_flag		No mapping
component_size		
transaction_id		Not used in TR-B15
timeout_value_DII		
leak_rate		Not used in TR-B15

<Blank Page>

Chapter 4 Use of Metadata Encoding for Transmission

4.1 Coding Scheme of Metadata

The metadata transmission coding scheme conforms to “3.3.1 Binary format coding” and “3.3.2 Text format coding” in ARIB STD-B38. However, for character coding, see “3.1 Use of Metadata Character Coding.”

4.2 Metadata Description Unit

The metadata coded in the text format is described as an XML document that uses TVAMain as the route element.

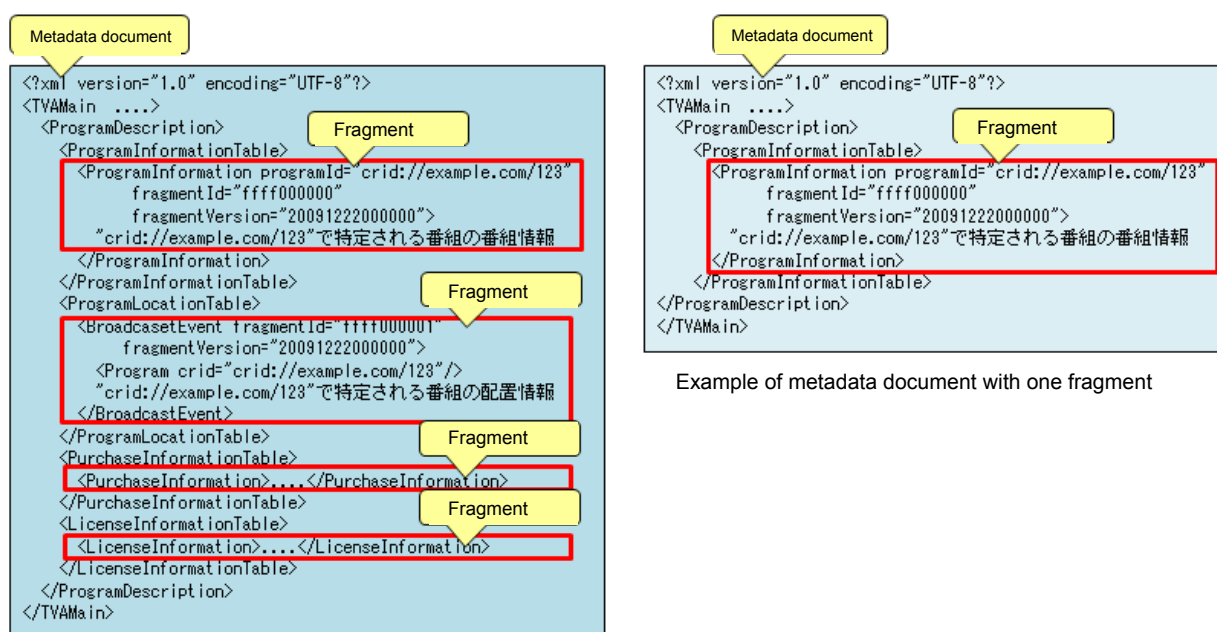
There are two types of metadata documents: metadata documents in which only one fragment is described and metadata documents in which all the fragments identified by a single CRID are described. fragmentId and fragmentVersion are used to identify, update, or manage each fragment.

Description:

An explanation of a metadata document and fragment is as follows.

- A minimum unit of metadata being distributed is referred to as a metadata fragment.
- A file in which metadata fragments are stored is an independent XML document that has a TVAMain element as a route element.

Figure 4-1 shows an example of metadata documents and fragments.



Example of metadata document with one fragment

Example of a metadata document with all fragments (CRID unit, PGSET format)

* For the fragment combination, see Note 1 in Table 3-4.

Fig. 4-1: Example of metadata documents and fragments

4.3 File Format of Metadata

The metadata received by a receiver is provided as one file (resource) in which one metadata document (an XML document in which the XML declaration is described in the first line and in which a range starting from <TVAMain ...> and ending at </TVAMain> is described in the second and subsequent line) is stored. This file (resource) is called a “metadata resource.” A receiver can also receive the file in which all fragments with the same CRIDs are included in a single TVAMain (CRID unit, see Fig. 4-1).

The description content of an XML declaration is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
```

4.4 Use of Fragment Identification

4.4.1 Use of fragmentId

A fragmentID is uniquely used throughout multimedia broadcasting by selecting from the value range of “0” to “a value obtained by subtracting 1 from 2 raised to the 40th power.” A fragmentID used in a metadata document is described in a 10-digit hexadecimal integer value. “0x” or another value that represents a hexadecimal code is not added to the beginning of the ID. If the ID is less than 10 digits, then “0” is added to the beginning of a fragmentID to bring it to 10 digits. In this operational standard, the authority and the publisher are used as the same thing, as described in “2.4.3 Authority and publisher.”

Fragments to which the same fragmentID is applied are not described in a different description target. Multiple fragments are not described in the same description target by applying a different fragmentId.

The same description target is any of the following fragments.

- ProgramInformationfragment with the same programId
- GroupInformationfragment with the same groupId
- ProgramLocationfragment with the same Program (crId) and ProgramURL
- LicenseInformationfragment with the same LicenseId
- PurchaseInformationfragment with the same purchaseId
- SegmentInformationfragment with the same segmentId
- SegmentGroupInformationfragment with the same groupId
- CouponDescriptionfragment with the same couponId

Reference information:

An entrusted broadcaster can freely decide on how to assign fragmentIds. However, it should be

noted that a duplicated `fragmentId` cannot be used and that a `fragmentId` cannot be basically reused in an entire multimedia broadcasting instance. For reference, the following shows an example that provides a standard when creating metadata in order to uniquely apply a `fragmentId` in `<authority>`.

4.4.2 Use of `fragmentVersion`

The date and time when an applicable fragment is updated is described in the `fragmentVersion`. A `fragmentVersion` in a metadata document is described in a 14-digit decimal numerical value (in the `YYYYMMDDhhmmss` format).

4.4.3 Reuse of `fragmentId`

A `fragmentId` can be reused if it has been issued two months after the latest date and time among the `fragmentExpirationDates` of all fragments issued in the past. For details on the expiration date of metadata in a receiver, see “7.3.2 Notes on the expiration date of metadata.”

<Blank Page>

Chapter 5 Use of Metadata Transmission and Storage Control Method in Broadcasting

The following cases are expected when transmitting description language-type metadata.

- Transmitting description language-type metadata at the partial reception layer (Layer A)
- Transmitting description language-type metadata at a layer other than the partial reception layer (Layer B) as one storable piece of broadcasting content

For details on partial reception, see “3.2.10 Use of partial reception” and “2.4 Service Pattern” in Volume 7 of this operational standard. For details on multiplexing, see “Chapter 3 Multiplexing” in Volume 7 of this operational standard. For details on metadata transmission, see “2.2 Transmission Method of EPG/ECG Metadata” in Volume 11 of this operational standard and 6.3 in Part 2 of ARIB STD-B45.

5.1 Use of Modules to Transmit Metadata

When transmitting metadata, CRID unit metadata (in PGSET format, see Section 3.3), in which all the existing fragments among the fragments with the same CRID are stored in a single TVMain, is transmitted. This means that all information related to one piece of content/program is described in a metadata file. To transmit multiple metadata in bulk, each CRID unit metadata file is transmitted together. Figure 5-1 shows the conceptual diagram of a metadata transmission unit.

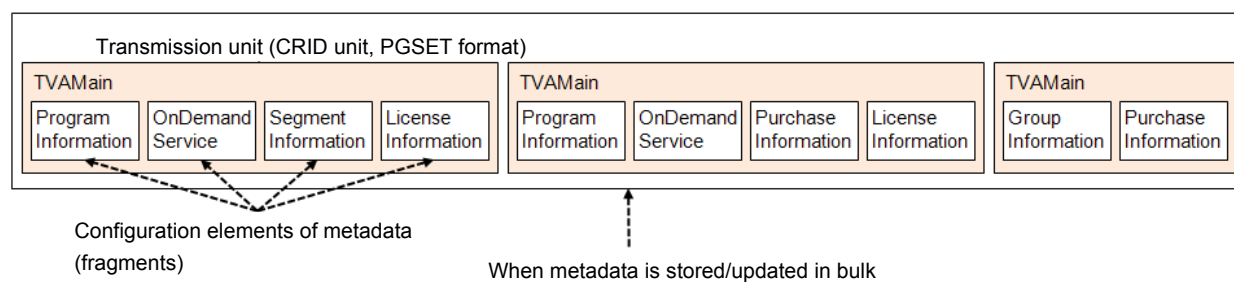


Fig. 5-1: Conceptual diagram of a metadata transmission unit

5.1.1 Metadata file format

When transmitting metadata, the metadata file format defined in 3.4.3 in ARIB STD-B38 is used.

When transmitting transmission control metadata (see “2.1.5.4 Transmission control metadata” in Volume 11 of this operational standard), a metadata file format defined in 6.3.1.4 in Part 2 of ARIB STD-B45 is used.

5.2 Transmission at the Partial Reception Layer (Layer A)

A metadata file in a CRID unit, which is a metadata transmission unit, is transmitted using the partial reception layer in 13-segment broadcasting. At this time, one metadata transmission unit must be set to the partial reception layer (Layer A) and transmitted. If multiple metadata files are transmitted in bulk, a service provider can decide whether to transmit each metadata file separately or to transmit an arbitrary file in which multiple metadata files are stored.

5.3 Transmission Using a Layer Other Than a Partial Reception Layer (Layer B)

A metadata file in a CRID unit, which is a metadata transmission unit, or an arbitrary file, in which multiple metadata files are stored, is transmitted using the same method as ordinary storable broadcasting content. This means that a metadata file or an arbitrary file with metadata is separated in a block size defined in the FLUTE and AL-FEC. Then, the AL-FEC is applied to the file and the file is transmitted based on the UDP/IP or IOverMPEG-2. If part of the data has been lost due to the condition of a transmission route, the data can be restored using the compensation of the stored content function.

For details on FLUTE, AL-FEC, UDP/IP, and IOverMPEG-2, etc., see “Chapter 5 Use of Storable Broadcasting Service” in Volume 3 of this operational standard. For details on the compensation of stored content, see “Chapter 6 Compensation of Stored Content” in Volume 6 of this operational standard.

5.4 Guidelines for Organizing Metadata

This section provides the guidelines to be followed when organizing metadata.

5.4.1 Guidelines for organizing metadata at the partial reception layer (Layer A)

- Metadata is transmitted at the partial reception layer (Layer A) in a metadata transmission FLUTE session in 13-segment broadcasting. In addition to metadata, storable broadcasting content (content distributed separately from an ordinary storable broadcasting schedule, such as extras and content for promotion) and PSI are transmitted at the partial reception layer.
- The metadata of a piece of content/program, which will be distributed via real-time broadcasting or storable broadcasting, is distributed within a period of time determined based on the service requirement (i.e., the time for immediately applying a metadata update, such as 48 hours after the current time).
- The content, which will be distributed via real-time broadcasting or storable broadcasting within the period of time specified above, and the metadata for a package/group that has such content (excluding contents without metadata) are transmitted.
- The metadata, which is scheduled to be distributed within the period of time specified above, is distributed regularly, and the metadata to be transmitted is changed as appropriate based on the lapse of time.

- The metadata transmitted at the partial reception layer (Layer A) is obtained as appropriate (such as before obtaining storable broadcasting content or before starting to view real-time content). Whether to update the EPG/ECG as required can be determined based on a business operator or an implemented terminal.

Figure 5-2 shows an example of organizing metadata based on the guidelines above.

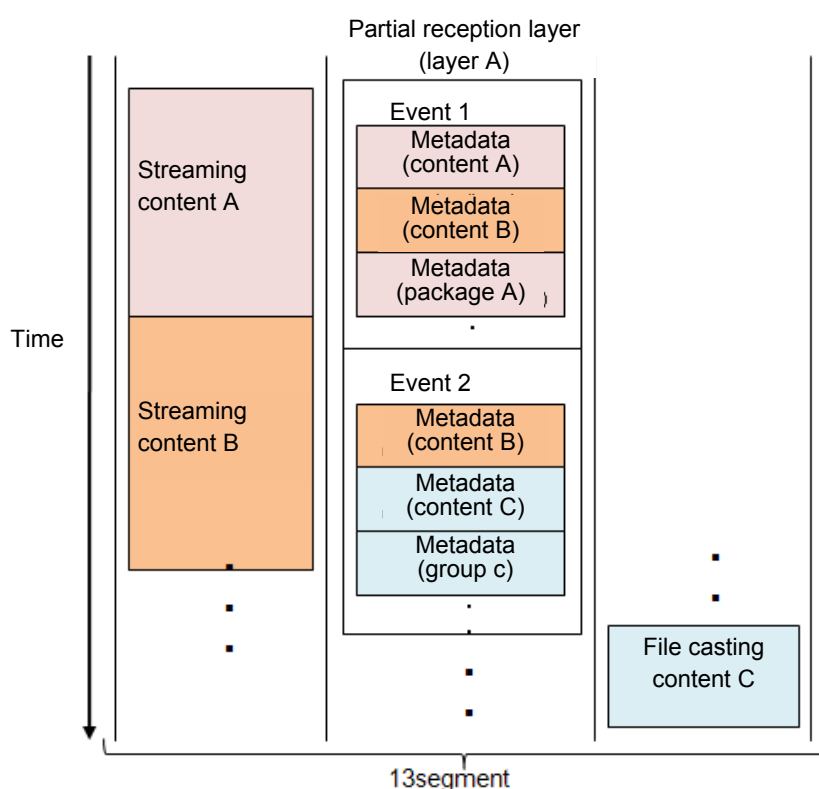


Fig. 5-2: Example of organizing metadata at the partial reception layer (Layer A)

5.4.2 Guidelines for organizing metadata at a layer other than the partial reception layer (Layer B)

- The metadata of a piece of content/program, which will be distributed via real-time broadcasting or storable broadcasting, by combining it in one storable piece of broadcasting content (such as program guide content), is transmitted in a content transmission FLUTE session within a period of time determined based on the service requirement (a period that is the required minimum for displaying the EPG/ECG in a receiver, such as eight days including the current day).
- The storable broadcasting content with the metadata mentioned above is updated at a cycle determined based on the service requirement (such as one day) and distributed regularly.

- The storable broadcasting content with the metadata mentioned above has the content that will be distributed within the period of time specified above via real-time broadcasting or storable broadcasting, along with the metadata for a package/group that has such content (excluding content without metadata).
- If metadata with the same CRID or FragmentId is received with the metadata transmitted at the partial reception layer (Layer A) (see 5.4.1) and the metadata transmitted at a layer other than the partial reception layer (Layer B), the FragmentVersion are compared and the metadata with the latest version is managed in a terminal.
- If the information related to storable broadcasting content, in which the metadata mentioned above is stored, is described in the metadata, then ISDBTMMContentypeCS/1.7 (metadata set) or ISDBTMMContentypeCS/1.11 (metadata set 2) must be described in :ContentProperties/Contentype/@href.

Figure 5-3 shows an example of organizing metadata based on the guidelines above.

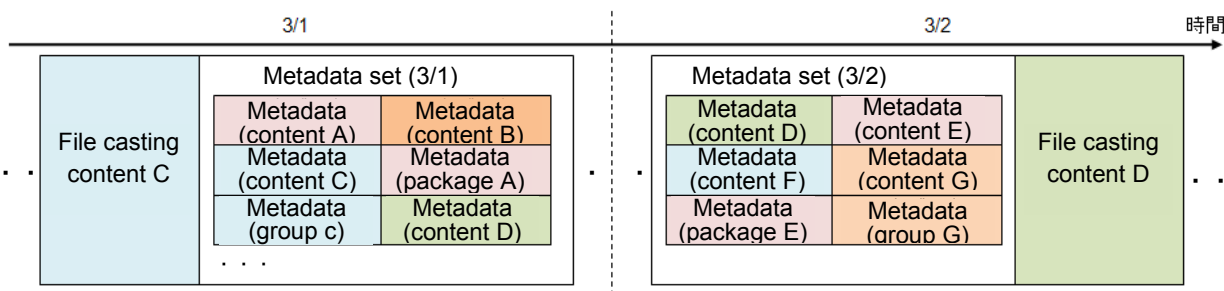


Fig. 5-3: Example of organizing metadata at a layer other than the partial reception layer (Layer B)

Chapter 6 Use of Metadata Distribution Methods via Communication

6.1 Overview of Metadata Distribution Methods

This section defines the distribution of metadata via communication.

A receiver can request the acquisition of metadata by entering the search condition for specifying the metadata. Then, it can obtain a list of appropriate metadata and the CRIDs. As a precondition, this distribution method uses bidirectional communication over TCP/IP. Figure 6-1 shows an overview of metadata distribution.

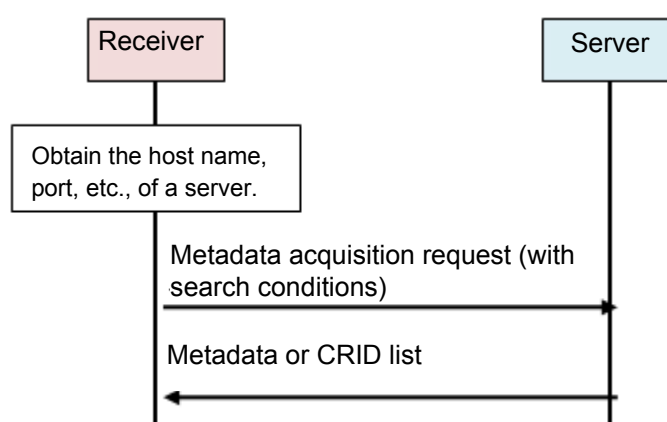


Figure 6-1: Overview of metadata distribution

6.2 Protocols Used

HTTP 1.1 [RFC-2616] or HTTP/TLS (HTTPS) [RFC-2818] is used when sending an acquisition request (search request transmission) and receiving a search result. The details are defined in the following sections.

The time-out value of the receiver when obtaining metadata is based on an implemented receiver. It does not take over 15 seconds to obtain metadata.

6.3 Search Request

The following section describes how to send a search request over HTTP.

6.3.1 Overview of a search request

A search request is composed of a set of multiple parameters' names and the parameter values and is sent using the HTTP Method POST and GET Method. For details on format, see [HTML4.01] 17.13 Form submission. The following describes the POST search request format. In the notation used hereinafter, <~> indicates parameter, and [~] indicates that it can be omitted.

```
POST /get_Data
Host:<server>
Content-Type: application/x-www-form-urlencoded
<parameters>
```

- ◆ <server> Server URL
- ◆ <parameters> <parameter>[&<parameters>]
- ◆ <parameter> <paraname>=<paravalue>
- ◆ <paraname> Parameter name
- ◆ <paravalue> Parameter value

When requesting a POST, the body of a search request is sent only in x-www-form-urlencoded format. Just as when requesting a GET, the parameter name and parameter value must be appropriately encoded in the URL. For the specification of the URL encode, see [RFC-1738] and [RFC-2396]. The character codes used for URL encoding are pursuant to the standard defined in “3.1.1 Character coding for metadata.”

6.3.2 Search request details

6.3.2.1 Common format

This section describes a common format that is independent from a search pattern. The following shows the format. The order of appearance of an item that can be omitted is fixed.

【Format】 <predicate-bag>[&<range>][&<fragment>][&<format>][&<type>]

- ◆ <predicate-bag> Specifies the search condition (described in (a) below)
- ◆ <range> Specifies the range of a search result to be returned
(described in (b) below)
- ◆ <fragment> Specifies the fragment to be returned (described in (c) below)
- ◆ <format> Specifies the coding scheme of data to be returned (described in
(d) below)
- ◆ <type> Specifies the format of the data to be returned (described in (e)
below)

(a) <predicate-bag>

【Format】 predicate=<binary-predicates>[&<sort>]

- ◆ <binary-predicates> A collection of <binary-predicates>; multiple listed <binary-predicates> are expressed by applying an AND operation.

【Format】 <binary-predicate>[,<binary-predicates>]

The symbol"! " indicates a NOT operation, which selects a complementary set of fragments that is selected immediately after <binary-predicate> .

- ◆ <binary-predicate> Specifies the detailed condition for each search pattern
Details are defined in "6.3.2.2 Format for each search pattern."

- ◆ <sort> Specifies a sort order

【Format】 sort = crid | fragmentId | title | epg | none | index |

index-asc | index-desc | duration-asc |

duration-desc | date-asc | date-desc

crid: Ascending sort by CRID

fragmentid: Ascending sort by FragmentId

title: Ascending sort by title ruby*¹

epg: First sort by service ID in ascending order, second sort by program start date in ascending order

none: Sort not performed

index: Ascending sort by order in parent groups*² when searching parent groups (MemberOf)

index-asc/desc: Ascending/descending sort by order in parent groups*² when searching parent groups (MemberOf)

duration-asc/desc: Ascending/descending sort by duration when searching duration

date-asc/desc: Ascending/descending sort by date and time (disclosure start date and time [metadata]); if specified date and time information does not exist,

results are not sorted in ascending/descending order

but are sorted by the ending's title ruby.*¹

*¹ The ProgramInformation or GroupInformation

for which a BasicDescription/Title[@type is specified as
“alternative” is used.

*2 The MemberOf/@index of the ProgramInformation
or GroupInformation is used.

If this parameter is not specified, it is regarded that sort is specified as none.

Note: If data to be sorted is NULL, the data is not sorted in ascending/descending order but is put at the end of the list. If the order of the first sorted data is the same as the order before sorting, the data is sorted again by title ruby*1 in ascending order. This processing is applied to all sort types excluding “title” and “epg.” If the “epg” is applied, the third sort is performed by sorting data by title ruby *1 in ascending sort.

* For details on the overview of how the evaluation result of each predicate is created, see “6.3.2.3 Overview of search processing.”

(b) <range>

【Format】 range=<from>[,<count>] | unlimit

- | | |
|-----------|--|
| ◆ <from> | Specifies the top order of the returned search results (1–) |
| ◆ <count> | Specifies the number of search results returned starting from the top |
| ◆ unlimit | All search results are returned starting from the top of the search results. |

If this parameter is not specified, it is regarded that the range is specified as unlimit.

* Search results may be returned based on a search range that is different from the specified range due to processing on a server. Even if this happens, search results based on the specified range are included. The actual range of the returned search results is notified using the response HTTP header “X-metaserver-range:”.

(c) <fragment>

【Format】 fragment =<singlefragmentspecs> | ALL | PGSET

singlefragmentspecs=<singlefragmentspec>[,<singlefragmentspecs>]

singlefragmentspec=PIT | GIT | PLT | PRT | PuIT | SIT | SvIT | CPT | LIT

- | | |
|-------|---------------------------------------|
| ◆ PIT | Specifies the ProgramInformationTable |
| ◆ GIT | Specifies the GroupInformationTable |
| ◆ PLT | Specifies the ProgramLocationTable |

- | | |
|---------|---|
| ◆ PRT | Specifies the ProgramReviewTable |
| ◆ PuIT | Specifies the PurchaseInformationTable |
| ◆ SIT | Specifies the SegmentInformationTable |
| ◆ SvIT | Specifies the ServiceInformationTable |
| ◆ CPT | Specifies the CouponDescriptionTable |
| ◆ LIT | Specifies the LicenseInformationTable |
| ◆ ALL | All existing fragments among the fragments with the same CRID |
| ◆ PGSET | Returns the same fragments as ALL above by storing them in a single TVAMain (CRID unit) |

If this parameter is not specified, it is regarded that the fragment is specified as ALL.

* If multiple fragments are specified, they can be specified in an arbitrary order; however, the same fragments cannot be specified more than once. If data is sorted by CRID or another condition by specifying a sort type, including ALL, the returned result of the fragments with the same CRID appears in an arbitrary order.

(d) <format>

【Format】 format=textual | binary

- | | |
|-----------|---------------------------------|
| ◆ textual | Specifies the xml format (text) |
| ◆ binary | Specifies the binary form (BiM) |

If this parameter is not specified, it is regarded that the format is specified as textual.

(e) <type>

【Format】 type=cridlist | body | countonly | segidlist

- | | |
|-------------|---|
| ◆ cridlist | Specifies a CRID list that corresponds to the search conditions |
| ◆ body | Specifies the xml body of the metadata that corresponds to the search conditions |
| ◆ countonly | Only the counts are returned. |
| ◆ segidlist | Corresponds to the search conditions when searching a scene
The SegmentId list of the SegmentInformation is specified. |

If this parameter is not specified, it is regarded that the type is specified as a body.

6.3.2.2 Format for each search pattern

The format of <binary-predicate> explained in (a) in 6.3.2.1 varies depending on the search pattern. More specifically, the format varies depending on which element/attribute in the metadata is used as the key for the search.

This standard defines <binary-predicate> formats for the following search patterns and fieldIds. A fieldId functions as a label that represents a search target in the metadata. See Table 6-2 for the correspondence between each fieldId and the information element/attribute of the metadata that is actually searched.

- Title search (fieldId: Title)
- Keyword search (fieldId: Keyword)
- Actor search (fieldId: Role, GivenName)
- Broadcasting period/time range search (fieldId: PublishedStart, PublishedEnd)
- Broadcasting period/time search (fieldId: PublishedStart, PublishedEnd)
- CRID search (fieldId: Crid)
- FragmentId/FragmentVersion search (fieldId: FragmentId, FragmentVersion)
- Genre search (fieldId: Genre)
- Parent group (MemberOf) search (fieldId: MemberOf)
- Group type search (fieldId: GroupType)
- License ID search (fieldId: LicenseId)
- PromotionalStatus search (fieldId: PromotionalStatus)
- Parental rating search (fieldId: ParentalRating)
- FragmentExpirationDate search (fieldId: FragmentExpirationDate)
- Identifier search (fieldId: OtherIdentifier)
- Period search (fieldId: PeriodType, PeriodStart, PeriodEnd)
- Period exclusion search (fieldId: PeriodType, PeriodStart, PeriodEnd)
- LI target sales ID search (fieldId: PurchaseID_LI)
- PuI target sales ID search (fieldId: PurchaseID_PuI)
- Scene search (fieldId: Crid, StartTime, EndTime)
- Coupon search (fieldId: CouponID_PuI)
- Duration search (fieldId: Duration)
- Search by a fieldId other than those provided above (see Table 6-2 for a fieldId that can be specified).

Table 6-1 shows the <binary-predicate> of each search pattern.

Table 6-1: <binary-predicate> format

Search pattern	<binary-predicate> format	fieldId	Remarks
Title search	title(<value>,...)	Title	Multiple values can be searched using the OR condition.
Keyword search	keyword(<value>,...)	Keyword	Multiple values can be searched using the OR condition.
Actor search	credit((<value_1[,value_2>]) ,...) ※ value_1: GivenName, value_2: role	Role	Multiple values can be searched using the OR condition. <value_1> and <value_2> are specified using the AND condition.
		GivenName	
Broadcasting period/time range search	timerange([<start-time>],[<end-time>])	PublishedStart	
		PublishedEnd	
Broadcasting period/time search	time(<start>[,<time-range>])	PublishedStart	
		PublishedEnd	
CRID search	crid(<value>,...)	Crid	Multiple values can be searched using the OR condition.
FragmentId search	fragmentid(<value>,...)	FragmentId	Multiple values can be searched using the OR condition.
FragmentVersion search	fragmentversion([<minimum-version>],[<maximum-version>])	FragmentVersion	
Genre search	genre(<value>,...)	Genre	Multiple values can be searched using the OR condition.
Parent group search	memberof(<value>,...)	MemberOf	Multiple values can be searched using the OR condition.
Group type search	grouptype(<value>,...)	GroupType	Multiple values can be searched using the OR condition.
License ID search	licenseid(<value>,...)	LicenseId	Multiple values can be searched using the OR condition.
PromotionalStatus search	promotionalstatus(<value>,..)	PromotionalStatus	Multiple values can be searched using the OR condition.
Parental rating search	parentalrating(<value>,...)	ParentalRating	Multiple values can be searched using the OR condition.

FragmentExpirationDate search	expirationdate(<not-before>)	FragmentExpirationDate	
Identifier search	identifier(<id>,[<id-type>])	OtherIdentifier	
Period search	period(<type>,[<start>],[<end>])	PeriodType	
		PeriodStart	
		PeriodEnd	
Period exclusion search	periodex(<type>,[<start>],[<end>])	PeriodType	
		PeriodStart	
		PeriodEnd	
LI target sales ID search	purchaseid_li(<value>,...)	PurchaseID Note 1	Multiple values can be searched using the OR condition.
PuI target sales ID search	purchaseid_pui(<value>,...)	PurchaseID Note 2	Multiple values can be searched using the OR condition.
Scene search	scene(<value>,[<start>],[<end>])	Scene	
		StartTime	
		MediaDuration	
Coupon search	coupon(<value>,...)	CouponId_PuI	Multiple values can be searched using the OR condition.
Duration search	duration([<minimum-duration>],[<maximum-duration>])	Duration	

Note 1: Special expression used only for LI/PurchaseIdRef/text() in a PurchaseID

Note 2: Special expression used for specifying a condition to {PI|GI}/BD/PurchaseList/PurchaseIdRef/text() or PuI/@purchaseID in a PurchaseID

A fieldId that is not described in Table 6-1 is searched using the following general format. Multiple <values> can be described, to which the evaluation result of each condition is added using the OR condition. The fieldIds described in Table 6-1 cannot be used in this format. The formats provided in Table 6-2 are used.

`field(<fieldId>,<value>[,...])`

If the following characters are included in the <value>, the escape processing needs to be performed using '\':

'(', ')', ',', '*', '&', '\'

A search condition can be specified for a <value> in the following format.

`[*]<string>[*]` (<string> is a search character string.)

- Add “*” to the beginning only → Backward match
- Add “*” to the end only → Forward match
- Add “*” to the both ends → Partial match
- Without “*” → Perfect match

The following provides a supplementary explanation about search formats in Table 6-1, which are not used for a simple character string retrieval.

(1) credit

The name (GivenName) and role (Role) can be specified as a pair using the AND condition. When such multiple pairs are specified, each pair is connected using the OR condition.

The following shows an example.

- credit ((name)): Selects the data with the GivenName specified as “name”
- credit ((name, http://www.arib.or.jp/cs/2011/03/ARIBRoleCS/3.1), (name, http://www.arib.or.jp/cs/2011/03/ARIBRoleCS/1)): Selects the data for which the GivenName is specified as “name” and the role is specified as “director” or “actor”

(2) timerange

An applicable metadata fragment is selected if a period of time from a specified start time to an end time overlaps the broadcasting period/time described in the metadata. Whether such an overlap exists can be checked as follows.

(Broadcasting distribution start time < <end-time>) and (Broadcasting distribution end time > <start-time>)

The <start-time> and <end-time> are described in the “YYYYMMDDHH” or “YYYY-MM-DDTHH[+zz: zz]” format. The place of a minute or second is processed as 0. If the latter format is used, a timezone can be added. In either case, if a timezone is not specified, the timezone is processed as the default timezone of a server. The information of a default timezone is notified as the header information of a returned result.

Either <start-time> or <end-time> can be omitted. In this case, it is regarded that any condition matches the omitted constraint. For instance, if a timerange (2006010100,) is specified as a condition and if the <end-time> is omitted, then the metadata fragment of a content that ends after 2006/01/01 0:00:00’ is selected.

The following shows an example when the search condition of a broadcasting period is specified as StartOfAvailability=2005/12/24 0:00:00 and EndOfAvailability=2006/1/1 0:00:00.

- Example of timeranges when the metadata fragment of the content is selected

Search condition	Description
timerange(2005122423,2005122504)	The start of a broadcasting period overlaps a search range.
timerange(2005122500,2005122501)	A search range is included in a

	broadcasting period.
timerange(2005123123,2006010104)	The end of a broadcasting period overlaps a search range.
timerange(2005122000,2006020100)	A search range is included in a broadcasting period.

- Example of timeranges when the metadata fragment of the content is not selected

Search condition	Description
timerange(2005122320,2005122323)	A broadcasting period does not overlap a search range.
timerange(2005122323,2005122400)	The end of a search range and the start of a broadcasting period match.
timerange(2006010100,2006010110)	The start of a search range and the end of a broadcasting period match.

(3) time

An applicable metadata fragment is selected when the period of time between the start time specified in <start> and the end time, which is calculated by adding <time-range> to the <start>, and a broadcasting period described in metadata overlap. Whether such an overlap exists is evaluated using the same period condition as timerange.

(4) fragmentversion

A metadata fragment, which has a FragmentVersion included in a specified <minimum-version> and <maximum-version>, is selected. Whether such a metadata fragment is included is checked as follows.

$$\boxed{(\text{FragmentVersion} \geq \text{<minimum-version>}) \text{ and } (\text{FragmentVersion} \leq \text{<maximum-version>})}$$

In the < minimum-version > and < maximum-version >, a time expressed in the “YYYYMMDDhhmmss” format is described.

Either <minimum-version> or <maximum-version> can be omitted. In this case, it is regarded that any condition matches the omitted constraint. For instance, if a FragmentVersion (20060101000000,) is specified as a condition, a metadata fragment that corresponds to '20060101000000" or a metadata fragment that has a greater FragmentVersion is selected.

(5) expirationdate

A metadata fragment that matches the time specified in <not-before> or a metadata fragment for which the FragmentExpirationDate is later than the specified time is selected.

A time is described in the <not-before> in “YYYY-MM-DDThh:mm:ss[+zz:zz]” format. A timezone is processed in the same way as timerage in (2).

(6) period

A Period field with the @type specified in <type> is evaluated using the same period condition as

timerange.

(7) periodex

A Period field with the @type specified in <type> is evaluated using the same period condition as ! timerange (denial of timerange). An overlap between periods is checked as follows.

$$\boxed{(\text{Broadcasting distribution start time} \geq \text{<end-time>}) \text{ or } (\text{Broadcasting distribution end time} \leq \text{<start-time>})}$$

(8) scene

An applicable metadata fragment is selected if there is an overlap among the start time specified in <start> in the SegmentInformation, in which the ProgramRef/@crid has a CRID specified in <value>, the end time specified in <end>, and the scene period described in the metadata. Such an overlap among periods is evaluated using the same period condition as timerange. When evaluating, the <start> and <end> time is described in “hhmmss.” The end time of a scene period is calculated by StartTime + MediaDuration.

(9) duration

A metadata fragment that has a duration included in the specified <minimum-duration> and <maximum-duration> is selected. Whether such a duration is included is evaluated using the same boundary condition as FragmentVersion. When evaluating, the <minimum-duration> and <maximum-duration> are described in “seconds.”

6.3.2.3 Overview of the search process

This section describes the basic idea of how a search request is processed based on the syntax mentioned earlier. The processing method described in this section does not define a processing method used in an actual server, but only defines a group of results to be returned for clarification.

Creating the evaluation result of each <binary-predicate>

Each <binary-predicate> individually evaluates the format of every path included in a fieldId that is the subject of the predicate. Then, a list is created by fragment type, showing the correspondence between the fragment IDs and CRIDs of all fragments in which the results are true.

a) Extracting a result list of each binary-predicate

The following processing is performed for each binary-predicate, and a candidate result list of fragment IDs is created.

(1) When one fragment is returned

Check the fragment type of each fragment in a result list. If the fragment type is different from the returned fragment, obtain a CRID of the fragment ID. Extract a fragment ID of the fragment that has the same CRID as the obtained CRID and for which the fragment type is the same as the

returned fragment. If the fragment type is the same as the returned fragment, obtain the fragment ID and create a list by adding the obtained fragment ID with the earlier extracted fragment ID.

(2) Cases other than above

Create a list of the fragment IDs of all fragments that have the CRIDs in a result list and for which the fragment types match the returned fragments.

b) Creating a last result

As a final result, create a collection of the fragment IDs by gathering the fragment IDs in all the candidate list of each binary-predicate's result.

6.3.2.4 Acquisition of an actor name list

In addition to the predicate search explained in the previous section, there is also an “actor name list acquisition” function that returns the list of actor names in the actor information (CreditItem), in which a specified word is specified to an actor ruby (PersonName[@type=“variant/former”]/GivenName) and in which a specified ID is applied to a role ID (CreditItem/@role). The following provides the format for the actor name list acquisition function.

【Format】

getactor=<actorsearchspecs>[&<range>]

<actorsearchspecs> Collection of <actorsearchspec>; multiple listed
<actorsearchspecs> are expressed by applying an
AND operation.

<actorsearchspec> **【Format】** <actorsearchspec> [<actorsearchspecs>]
[name(<value>)]: Actor search
(fieldId: GivenName*¹)
[role(<value>)] : Role ID search (fieldId: Role)

¹A special expression used for specifying a condition
when the PersonName/@type is specified as
“variant” or “former.”

The <value> is handled the same way as the
<value> of the predicate search.

6.3.3 Supplementary notes on search request description

6.3.3.1 Rounding of each specified value

When a search is performed on a server, some search condition values may be corrected

(rounded) to fit the interval boundaries of the data managed in the server or for other reasons. Such a correction is performed so as to include the range specified in the search condition. Therefore, a search response may include data outside the range supported by a receiver. In such a case, necessary data needs to be selected in a receiver. The following describes a method to select a data range for each search condition in which the value may be rounded.

(1) range specification

When a range is specified, more data, including the range from “from” to “from+count,” may be returned. In a search response from a metadata server, the range of data to be returned is always provided in the HTTP header, X-metaserver-range:. Therefore, the range to be selected can be specified by referencing this value.

6.3.3.2 Definition and use of fieldId

Table 6-2 shows the relationship between a fieldId definition and corresponding search target Xpath, and a format type described in the <binary-predicate>.

Table 6-2: Relationship between a fieldId and search target Xpath and the condition format

- The “condition format” item in this table indicates a format that is used to describe the conditions of a fieldId.

fieldId	Corresponding XPath	Condition format A: Format in Table 6-1 B: fieldId ()
Crid	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/@programId	A
	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/@groupId	
	/TVAMain/ProgramDescription/ProgramLocationTable/BroadcastEvent/Program/@crid	
	/TVAMain/ProgramDescription/ProgramLocationTable/OnDemandProgram/Program/@crid	
	/TVAMain/ProgramDescription/ProgramLocationTable/OnDemandService/OnDemandProgram/Program/@crid	
	/TVAMain/ProgramDescription/LicenseInformationTable/LicenseInformation/ProgramCRID/@crid	
	/TVAMain/ProgramDescription/ServiceInformationTable/ServiceInformation/CRIDRef/@crid	
	/TVAMain/ProgramDescription/SegmentInformationTable/SegmentList/SegmentInformation/ProgramRef/@crid	

	/TVAMain/ProgramDescription/SegmentInformationTable/SegmentGroupList/SegmentGroupInformation/ProgramRef/@crid /TVAMain/ProgramDescription/ProgramReviewTable/Review/@programId /TVAMain/ProgramDescription/PurchaseInformationTable/PurchaseInformation/CRIDRef/@crid	
MemberOf	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/MemberOf/@crid /TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/MemberOf/@crid	A
GroupType	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/GroupType/@value	A
FragmentId	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/@fragmentId /TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/@fragmentId /TVAMain/ProgramDescription/ProgramLocationTable/BroadcastEvent/@fragmentId /TVAMain/ProgramDescription/ProgramLocationTable/OnDemandProgram/@fragmentId /TVAMain/ProgramDescription/ProgramLocationTable/OnDemandService/@fragmentId /TVAMain/ProgramDescription/ProgramReviewTable/Review/@fragmentId /TVAMain/ProgramDescription/ServiceInformationTable/ServiceInformation/@fragmentId /TVAMain/ProgramDescription/SegmentInformationTable/SegmentList/SegmentInformation/@fragmentId /TVAMain/ProgramDescription/SegmentInformationTable/SegmentGroupList/SegmentGroupInformation/@fragmentId /TVAMain/ProgramDescription/LicenseInformationTable/LicenseInformation/@fragmentId /TVAMain/ProgramDescription/CouponDescriptionTable/CouponDescription/@fragmentId /TVAMain/ProgramDescription/PurchaseInformationTable/PurchaseInformation/@fragmentId	A
FragmentVersion	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/@fragmentVersion /TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/@fragmentVersion /TVAMain/ProgramDescription/ProgramLocationTable/BroadcastEvent/@fragmentVersion /TVAMain/ProgramDescription/ProgramLocationTable/OnDemandProgram/@fragmentVersion /TVAMain/ProgramDescription/ProgramLocationTable/OnDemandService/@fragmentVersion	A

	<p>/TVAMain/ProgramDescription/ProgramReviewTable/Review/@fragmentVersion</p> <p>/TVAMain/ProgramDescription/ServiceInformationTable/ServiceInformation/@fragmentVersion</p> <p>/TVAMain/ProgramDescription/SegmentInformationTable/SegmentList/SegmentInformation/@fragmentVersion</p> <p>/TVAMain/ProgramDescription/SegmentInformationTable/SegmentGroupList/SegmentGroupInformation/@fragmentVersion</p> <p>/TVAMain/ProgramDescription/LicenseInformationTable/LicenseInformation/@fragmentVersion</p> <p>/TVAMain/ProgramDescription/CouponDescriptionTable/CouponDescription/@fragmentVersion</p> <p>/TVAMain/ProgramDescription/PurchaseInformationTable/PurchaseInformation/@fragmentVersion</p>	
FragmentExpire	<p>/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/@fragmentExpirationDate</p> <p>/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/@fragmentExpirationDate</p> <p>/TVAMain/ProgramDescription/ProgramLocationTable/BroadcastEvent/@fragmentExpirationDate</p> <p>/TVAMain/ProgramDescription/ProgramLocationTable/OnDemandProgram/@fragmentExpirationDate</p> <p>/TVAMain/ProgramDescription/ProgramLocationTable/OnDemandService/@fragmentExpirationDate</p> <p>/TVAMain/ProgramDescription/ProgramReviewTable/Review/@fragmentExpirationDate</p> <p>/TVAMain/ProgramDescription/ServiceInformationTable/ServiceInformation/@fragmentExpirationDate</p> <p>/TVAMain/ProgramDescription/SegmentInformationTable/SegmentList/SegmentInformation/@fragmentExpirationDate</p> <p>/TVAMain/ProgramDescription/SegmentInformationTable/SegmentGroupList/SegmentGroupInformation/@fragmentExpirationDate</p> <p>/TVAMain/ProgramDescription/LicenseInformationTable/LicenseInformation/@fragmentExpirationDate</p> <p>/TVAMain/ProgramDescription/CouponDescriptionTable/CouponDescription/@fragmentExpirationDate</p> <p>/TVAMain/ProgramDescription/PurchaseInformationTable/PurchaseInformation/@fragmentExpirationDate</p>	A
OtherIdentifier	<p>/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/OtherIdentifier/text()</p> <p>/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/OtherIdentifier/text()</p>	A
OtherIdType	<p>/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/OtherIdentifier/@type</p>	A

	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/OtherIdentifier/@type	
Title	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/BasicDescription/Title/text() (Note 1)	A
	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/BasicDescription/Title/text() (Note 1)	
Duration	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/BasicDescription/Duration/text()	A
Synopsis	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/BasicDescription/Synopsis/text()	B
	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/BasicDescription/Synopsis/text()	
Keyword	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/BasicDescription/Keyword/text()	A
	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/BasicDescription/Keyword/text()	
Genre	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/BasicDescription/Genre/@href	A
	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/BasicDescription/Genre/@href	
Role	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/BasicDescription/CreditsList/CreditsItem/@role	A
	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/BasicDescription/CreditsList/CreditsItem/@role	
GivenName	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/BasicDescription/CreditsList/CreditsItem/PersonName/mpeg7:GivenName/text()	A
	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/BasicDescription/CreditsList/CreditsItem/PersonName/mpeg7:GivenName/text()	
Character	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/BasicDescription/CreditsList/CreditsItem/Character/mpeg7:GivenName/text()	B
	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/BasicDescription/CreditsList/CreditsItem/Character/mpeg7:GivenName/text()	
Language	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/BasicDescription/Language/text()	B
	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/BasicDescription/Language/text()	
CaptionLanguage	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/BasicDescription/CaptionLanguage/text()	B
	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/BasicDescription/CaptionLanguage/text()	

SignLanguage	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/BasicDescription/SignLanguage/text()	B
	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/BasicDescription/SignLanguage/text()	
PromotionInfo	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/BasicDescription/PromotionalInformation/text()	B
	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/BasicDescription/PromotionalInformation/text()	
PromotionStatus	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/BasicDescription/PromotionalInformation/@href	A
	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/BasicDescription/PromotionalInformation/@href	
ProductDate	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/BasicDescription/ProductionDate/TimePoint/text()	B
	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/BasicDescription/ProductionDate/TimePoint/text()	
ReleaseDate	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/BasicDescription/ReleaseInformation/ReleaseDate/DayAndYear/text()	B
	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/BasicDescription/ReleaseInformation/ReleaseDate/DayAndYear/text()	
ReleaseYear	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/BasicDescription/ReleaseInformation/ReleaseDate/Year/text()	B
	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/BasicDescription/ReleaseInformation/ReleaseDate/Year/text()	
ReleaseLocation	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/BasicDescription/ReleaseInformation/ReleaseLocation/text()	B
	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/BasicDescription/ReleaseInformation/ReleaseLocation/text()	
PeriodType	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/Period/@type	A
	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/Period/@type	
PeriodStart	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/Period/Start/text()	A
	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/Period/Start/text()	
PeriodEnd	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/Period/End/text()	A
	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/Period/End/text()	
ProgramURL	/TVAMain/ProgramDescription/ProgramLocationTable/OnDemandProgram/ProgramURL/text()	B
	/TVAMain/ProgramDescription/ProgramLocationTable/BroadcastEvent/ProgramURL/text()	

	/TVAMain/ProgramDescription/ProgramLocationTable/OnDemandService/OnDemandProgram/ProgramURL/text()	
PublishedStart	/TVAMain/ProgramDescription/ProgramLocationTable/OnDemandProgram/StartOfAvailability/text()	A
	/TVAMain/ProgramDescription/ProgramLocationTable/BroadcastEvent/PublishedStartTime/text()	
	/TVAMain/ProgramDescription/ProgramLocationTable/OnDemandService/OnDemandProgram/StartOfAvailability/text()	
PublishedEnd	/TVAMain/ProgramDescription/ProgramLocationTable/OnDemandProgram/EndOfAvailability/text()	A
	/TVAMain/ProgramDescription/ProgramLocationTable/BroadcastEvent/PublishedEndTime/text()	
	/TVAMain/ProgramDescription/ProgramLocationTable/OnDemandService/OnDemandProgram/EndOfAvailability/text()	
Price	/TVAMain/ProgramDescription/PurchaseInformationTable/PurchaseInformation/Price/text()	B
PurchaseStart	/TVAMain/ProgramDescription/PurchaseInformationTable/PurchaseInformation/@start	B
PurchaseEnd	/TVAMain/ProgramDescription/PurchaseInformationTable/PurchaseInformation/@end	B
NameTag	/TVAMain/ProgramDescription/PurchaseInformationTable/PurchaseInformation/NameTag	B
LicenseID	/TVAMain/ProgramDescription/LicenseInformationTable/LicenseInformation/LicenseID/text()	A
MainLicenseID	/TVAMain/ProgramDescription/LicenseInformationTable/LicenseInformation/MainLicenseID/text()	B
LicenseType	/TVAMain/ProgramDescription/LicenseInformationTable/LicenseInformation/LicenseType/text()	B
ParentalRating	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/BasicDescription/ParentalGuidance/mpeg7:ParentalRating/@href	A
	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/BasicDescription/ParentalGuidance/mpeg7:ParentalRating/@href	
ServiceID	/TVAMain/ProgramDescription/ServiceInformationTable/ServiceInformation/@serviceId	A
	/TVAMain/ProgramDescription/ProgramLocationTable/BroadcastEvent/@serviceIdRef	
	/TVAMain/ProgramDescription/ProgramLocationTable/OnDemandService/@serviceIdRef	

PurchaseId	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/BasicDescription/PurchaseList/PurchaseIdRef/text()	B
	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/BasicDescription/PurchaseList/PurchaseIdRef/text()	
	/TVAMain/ProgramDescription/LicenseInformationTable/LicenseInformation/PurchaseIdRef/text()	
	/TVAMain/ProgramDescription/PurchaseInformationTable/PurchaseInformation/@purchaseId	
SegmentId	/TVAMain/ProgramDescription/SegmentInformationTable/SegmentList/SegmentInformation/@segmentId	B
SegmentGroupId	/TVAMain/ProgramDescription/SegmentInformationTable/SegmentGroupList/SegmentGroupInformation/@groupId	B
Scene	/TVAMain/ProgramDescription/SegmentInformationTable/SegmentList/SegmentInformation/ProgramRef/@crid	A
StartTime	/TVAMain/ProgramDescription/SegmentInformationTable/SegmentList/SegmentInformation/SegmentLocator/MediaRelTimePoint	A
StartTime + MediaDuration	/TVAMain/ProgramDescription/SegmentInformationTable/SegmentList/SegmentInformation/SegmentLocator/MediaRelTimePoint + /TVAMain/ProgramDescription/SegmentInformationTable/SegmentList/SegmentInformation/SegmentLocator/MediaDuration	A
CouponId	/TVAMain/ProgramDescription/CouponDescriptionTable/CouponDescription/@couponId	B
CouponId_PuI	/TVAMain/ProgramDescription/PurchaseInformationTable/PurchaseInformation/RequiredCoupon/@couponId	A
CouponId_Rew	/TVAMain/ProgramDescription/PurchaseInformationTable/PurchaseInformation/RewardCoupon/CouponRef/CouponIdRef	B
	/TVAMain/ProgramDescription/SegmentInformationTable/SegmentList/SegmentInformation/Description/RewardCoupon/CouponRef/CouponIdRef	
ContentType	/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/BasicDescription/ContentProperties/ContentType/ @href	B
	/TVAMain/ProgramDescription/GroupInformationTable/GroupInformation/BasicDescription/ContentProperties/ContentType/@href	

Note 1: Only for Title [@type="main, secondary, alternative, popular"] (not for "seriesTitle and episodeTitle")

Note 2: The prefix "tva:" is omitted.

6.4 Reception of the Search Result

This section describes how to receive search results over HTTP.

6.4.1 Normal response

Search results are returned by storing them in the body of HTTP. Information may be described in an original header field in the HTTP header when search results are returned.

6.4.1.1 Syntax

Search results obtained by specifying search conditions other than the condition, in which “countonly” is specified as the type, is stored in the Multipart/Mixed (RFC2046) format.

If “countonly” is specified as a type when requesting a search, the body of a search result response will be empty. The following provides the multi-part syntax definition specified in RFC2046 again.

```
boundary := 0*69<bchars> bcharsnospace
bchars := bcharsnospace / “ “
bcharsnospace := DIGIT / ALPHA / ““ / “(“ / “)” / “+” / “_” / “,” / “-” / “.” / “/” / “:” / “=” / “?”
dash-boundary := “-” boundary
; Obtain a boundary from the boundary parameter of the Content-Type field.
multipart-body := [preamble CRLF]
dash-boundary transport-padding CRLF
body-part *encapsulation
close-delimiter transport-padding
[CRLF epilogue]
transport-padding := *LWSP-char
; Composers must not generate non-zero length padding.
; But, receivers must be able to handle padding
; added by message transports.
encapsulation := delimiter transport-padding
CRLF body-part
delimiter := CRLF dash-boundary
close-delimiter := delimiter “-”
preamble := discard-text
epilogue := discard-text
discard-text := *(*text CRLF) *text
; Is ignored
body-part := MIME-part-headers [CRLF *OCTET]
; Lines in a body-part must not start with the specified dash-boundary, and
; the delimiter must not appear anywhere
; in the body part. Note that the semantics of a body-part differ from
; the semantics of a message, as described in the text.
OCTET := <any 0-255 octet value>
```

6.4.1.2 Additional information described in the entity header

The following information is added to the entity header based on specified search conditions and search results.

(1) count

Count indicates the total number (<totalcount>) of the metadata that matches search conditions. This header field is added only when “countonly” is specified to the type when requesting search.

```
X-metaserver-count: <totalcount>
```

(2) range

The range indicates the total amount of metadata that matches the range (<from>, <to>) of the metadata that was returned as a search result and that matches the search conditions. This header field is always added if a type other than “countonly” is specified as the type when requesting search.

```
X-metaserver-range: <from>·<to>/<totalcount>
```

(3) latest version

A maximum value (maximum-version) of the FragmentVersion of the metadata that was returned as a search result is shown in YYYYMMDDhhmmss format. This header field is always added if a type other than “countonly” is specified as the type when requesting search.

```
X-metaserver-latestversion: <maximum-version>
```

(4) timezone

Timezone presents which timezone is used for a specified time, which is included in a search request and to which a timezone is not set, in the +zzzz format. This information is always returned when a specified time without timezone is included in a search request.

```
X-metaserver-default-timezone: +0900
```

(5) warning

Warning indicates warning information (<warnings>) when processing a search request. A receiver does not need to do any processing based on this information.

```
X-metaserver-warning: <warnings>
```

(6) error

Error information is returned using the following header in a situation where results cannot be correctly returned when processing a search request. A receiver does not need to do any processing based on this value. For details, see “6.4.2 Error response.”

```
X-metaserver-error: <error code>
```

6.4.1.3 Additional information described in the header of a part

When the type is specified as the body, the following information is added to the header of each part when responding.

(1) Fragment information

When a format other than PGSET is specified to a fragment, the FragmentId and FragmentVersion of the fragment stored in the part is added to the header of the part in the following format.

```
Content-Description: <fragmentId> <FragmentVersion>
```

(2) CRID

If a format other than PGSET is specified to a fragment, the CRID of the metadata stored in the part is added to the header of the part in the following format.

```
Content-Description: <CRID>
```

6.4.1.4 Data type included in each part of a search result

There are only two kinds of data that are included in each part when a search result is multi-part. The Content-Length field is used for all headers of a part. The Content-Location field is not used.

(1) Fragment/metadata

A metadata fragment, which corresponds to one search result, is stored in text XML format. The Content-Type field in the header of a part including a fragment is as follows.

```
Content-Type: application/X-arib-meta+xml; charset="UTF-8"
```

(2) Fragment/metadata in binary format (option)

If binary is specified as the format in a search request, the fragment or metadata of each part is returned in binary format (BiM). In such a case, the Content-Type header of the part is as follows.

```
Content-Type: application/X-arib-meta+bim
```

6.4.1.5 Structure of a body part

As a search result, a list of multiple XML bodies and search results that correspond to search conditions is returned. The configuration of the information included in the body part is determined based on the type specified when requesting a search.

The following shows the configuration of a part per each specified type.

(1) type=countonly

A result is described in the X-metaserver-count field of the header without the body

(2) type=body

A result is returned in multi-part format; if one or more search results exist, all parts after the

first part are fragments. If the search result is zero, the first part is empty, and the second and subsequent parts do not exist.

(3) type=cridlist

A result is returned in text format. One CRID per line; the line feed is CR-LF.

(4) type=segidlist

A result is returned in text format. One SegmentId per line; the line feed is CR-LF.

6.4.1.6 Example of a search result

The following shows an example of a response when the type is specified as the body as explained in “6.3.2 Search request details” (predicate).

```
(Example)
-----
Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08j34c0p
Content-Length: 80192
X-metaserver-range: 1-10/1000
X-metaserver-latestversion: 20060101000000
X-metaserver-warning: Some parameters are ignored
X-metaserver-default-timezone: +0900
--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/X-arib-meta+xml;charset="UTF-8"
Content-Description: 0cd321101e,20080101000000
Content-Length: 6452
<?xml version="1.0" encoding="UTF-8"?>
<TVAMain>
....
</TVAMain>
--gc0p4Jq0M2Yt08j34c0p
...
--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/X-arib-meta+xml;charset="UTF-8"
Content-Description: 20d33313fb,20080101000000
Content-Length: 3446
<?xml version="1.0" encoding="UTF-8"?>
<TVAMain>
...
</TVAMain>
--gc0p4Jq0M2Yt08j34c0p--
-----
```

First fragment

Last fragment

The following shows an example of a response when searching using the function described in “6.3.2.4 Acquisition of an actor name list” (getactor).

(Example)

Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08j34c0p

Content-Length: 80192

X-metaserver-range: 1-10/1000

X-metaserver-latestversion: 20060101000000

X-metaserver-warning: Some parameters are ignored

X-metaserver-default-timezone: +0900

--gc0p4Jq0M2Yt08j34c0p

Content-Type: application/X-arib-meta+xml;charset="UTF-8"

Content-Description: 0cd321101e,20080101000000

Content-Length: 6452

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<TVAMain xml:lang="ja" xmlns="...">
```

```
<ProgramDescription>
```

```
<CreditsInformationTable>
```

```
<CreditsItem role="http://www.arib.or.jp/cs/2011/03/ARIBRoleCS/1">
```

```
<PersonName type="main">
```

```
<mpeg7:GivenName xml:lang="ja">田中太郎</mpeg7:GivenName>
```

```
</PersonName>
```

```
<PersonName type="variant">
```

```
<mpeg7:GivenName xml:lang="ja">タナカタロウ</mpeg7:GivenName>
```

```
</PersonName>
```

```
</CreditsItem>
```

```
...
```

```
</CreditsInformationTable>
```

```
</ProgramDescription>
```

```
</TVAMain>
```

6.4.2 Error response

If an error occurs after a search request has successfully arrived at a server, an error response is

returned. The following three types of errors are returned.

(1) Server process error

A server program is running, but if a search process cannot be done due to a format error in a search request or other reasons, then an error code is returned in the X-metaserver-error: header with HTTP status 200. A business operator can then decide details on error codes.

(2) Error code 500 Internal Error

This code may be returned when a server program does not operate.

(3) Error code 503 Service Unavailable

This error code is returned when a request cannot be temporarily processed. If this error occurs, an appropriate time to retry with the Retry-After header is specified. If the Retry-After is not specified, an operation is determined based on a receiver.

The following shows the example of an error response for a server processing error.

(Example)

```
-----  
HTTP/1.1 200 OK  
X-metaserver-error: ERR00004  
Content-Type: Text/Plain; charset='UTF-8'  
Content-Length: xx  
. . . . .  
errorno=ERR00004  
errorstr=Unexpected predicate parameter format  
-----
```


Chapter 7 Metadata Storage Control

7.1 Metadata Storage Control Method

7.1.1 Metadata storage guidelines

Realizing metadata storage (storage method, storage directory, etc.) varies based on an implemented receiver. However, the expiration date of stored metadata is managed according to the instructions described in “7.3.1 Expiration date of metadata.”

If content that corresponds to metadata is stored, the metadata is stored in the storage directory of the content (for details, see “8.2.3.3 Location resolution in receivers”). A storage method (such as how to change the storage location of the metadata and how to arrange copies) and the file name of the stored metadata vary depending on the implemented receiver.

7.2 Adding/updating Metadata

A receiver can identify the logical addition/update of metadata in a fragment unit by identifying the fragment of the metadata that belongs to the same authority. A fragment is identified using the FragmentVersion defined in “4.4.2 Use of fragmentVersion.”

If the FragmentId of the obtained metadata is new, a receiver can add the metadata by identifying that the process to be performed is to add the metadata. If the FragmentID of the obtained metadata already exists in a receiver and if the FragmentVersion is newer than that of the stored metadata's fragment, then the receiver can update the metadata.

In a metadata document in a CRID unit (fragments that are associated with the same CRID are stored in a single TVAMain in PGSET format), the common date and time are added to the FragmentVersion of each fragment. Therefore, when metadata document in a CRID unit is stored in a receiver, it can be added/updated using the ProgramInformation or the GroupInformation of the FragmentVersion included in the applicable metadata document.

7.3 Deleting Metadata

Whether or not metadata is deleted is determined based on the expiration date and the storage status of the content that corresponds to the applicable metadata. Metadata is not deleted if the expiration date is valid or if the content of the metadata is stored even though the expiration date is invalid. A method to delete metadata from a terminal varies depending on the implemented terminal.

As described in “3.8.2 Use of coupon description information,” the expiration date of ProgramInformation/GroupInformation and the expiration date of the CouponDescription, which are described in the metadata document, are set separately for the metadata document in a CRID unit. Therefore, it should be noted that coupon information should be deleted separately from

program/group information by taking into account whether or not there are any coupons.

7.3.1 Expiration date of metadata

The expiration date of metadata is described as the last date and time of the valid date (Japanese standard time) in the `FragmentExpirationDate` attribute in `DateTime` format (YYYY-MM-DDThh:mm:ss+09:00 format). A timezone is fixed to +09:00 in Japan. If the `FragmentExpirationDate` attribute is not specified, the description method of the expiration date varies depending on the implemented receiver.

7.3.2 Notes on the expiration date of metadata

It is important to note that expired metadata may be retained for a long period of time in the obtained metadata. As shown in Fig. 7-1, when the expiration date of the metadata presented by a server is shortened during the period, a receiver may store the metadata presented first and then add/update the metadata after the shortened expiration date. If a receiver cannot obtain the metadata with the shortened expiration date, it continues to present the metadata in the receiver until the old expiration date. As a result, it is expected that a server also presents the expired metadata until the old expiration date. Thus, the receiver needs to properly update the metadata.

If the expired metadata is not necessary, it can be searched using a metadata search string to delete. More specifically, the metadata to be deleted can be searched by specifying the current date and time to the `ExpirationDate` (<not-before>).

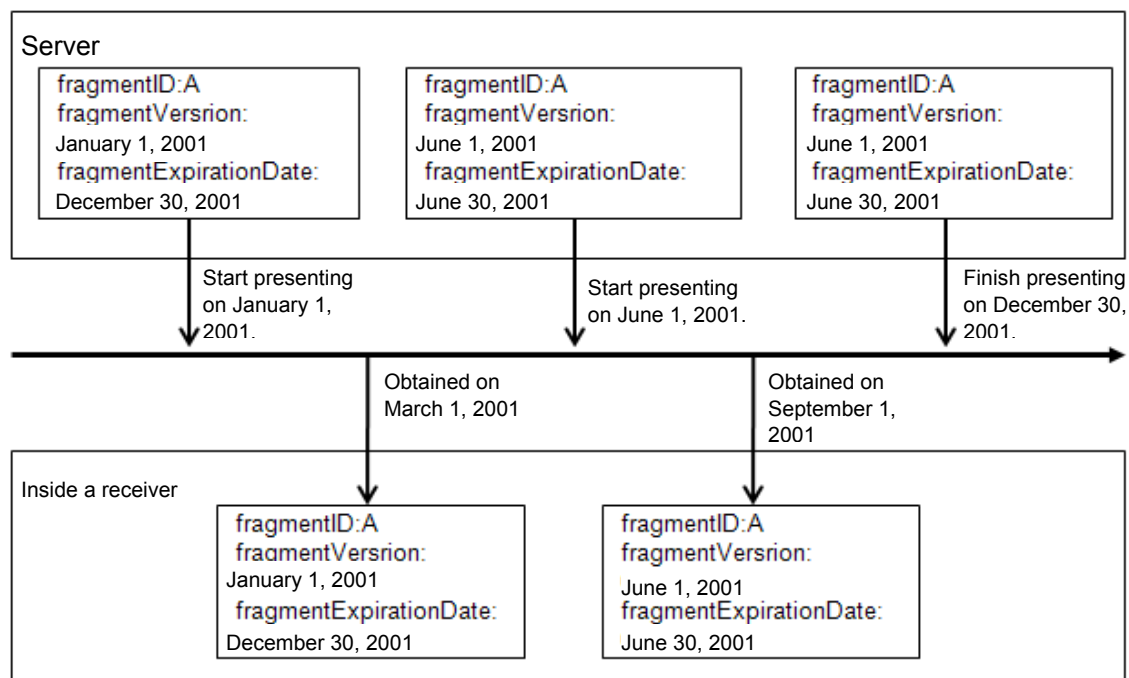


Fig. 7-1: Example when the metadata expiration date is shortened

Chapter 8 Use of Content and Metadata Identification Information

8.1 Use of a Content Reference Identifier (CRID)

The content reference identifier is used to identify an available unit of content or group in multimedia broadcasting.

The content reference identifier enables identifying an available unit of content and resource, establishing an association between metadata and an available unit of content (resource), and establishing an association between each control resource, such as license and transmission control metadata, and an available unit of content.

8.1.1 Description format of the content reference identifier (CRID)

An available unit of content and group is identified using the content reference identifier (CRID). The CRID is described according to “4.1.1 Content reference identifier (CRID)” in ARIB STD-B38 as follows:

`crid://<authority>/<data >`

The number of characters that can be described in the whole length of a content reference identifier is 255 bytes, including “`crid://<authority>/.`”

Note: As described later, uppercase and lowercase characters can be used for a CRID. However, when comparing CRIDs, they are not case-sensitive.

8.1.2 Use of authority (<authority>)

<authority> is used to uniquely identify a business operator. <authority> is described as follows according to the instruction written in “4.1.1.1 Authority” in ARIB STD-B38.

`<authority> = <DNS name>`

In <DNS name>, the domain name of the DNS (Domain Name System) defined in “IETF-RFC1034” and “IETF-RFC1035” is described to ensure the uniqueness among business operators.

The following shows the characters that can be used as <DNS name>.

```

<DNS name>      =      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”

8.1.3 Use of data (<data>)

A character string that is unique in <authority> is used in the <data> part. The following shows the characters that can be used as <data>.

<data>	=	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”

As described in “8.2.3.3 Location resolution in a receiver,” the content reference identifier is used as a logical directory in a receiver, and “#” is used as a delimiter. Therefore, the content reference identifier must not be used for the identifier (ProgramId) of an available unit of content and the identifier of a group (GroupId).

As for real-time broadcasting content, the <data> part is segmented and defined as <data>/<service ID>/<event ID>. By doing so, an event ID can be retrieved in a receiver. This enables the receiver to support changes added to the schedule of real-time broadcasting content to which a video recording or viewing reservation is made.

A CRID in real-time broadcasting content is described as follows:

crid://<authority>/<data>/<service ID>/<event ID>

The following shows the description example of a CRID.

- Example of storable broadcasting content: `crid://example.jp/20100401/fd/0010/cid001`
In the <data> part, the initial broadcast distribution date (2010/04/01), content category (fd), service ID (0010), content ID (cid001), and other information are described.
- Example of real-time broadcasting content: `crid://example.jp/20100401/rt/0001/12345`
In the <data> part, the broadcast date (2010/04/01), content category (rt) service ID (0001), event ID (12345), and other information are described.
- Example of group (package): `crid://example.jp/package/p0001`
In the <data> part, the group type (package), group ID (p0001), and other information are described.
- Example of group (series): `crid://example.jp/series/s0001`
In the <data> part, the group type (series), series ID (s0001), and other information are described.
- Example of metadata set (see “5.3 Transmission Using a Layer Other Than the Partial Reception Layer (Layer B)”): `crid://exampleA.jp/metadataset/20100401/cidA01`
In the <data> part, the content classification (metadataset), broadcasting date (2010/04/01), content ID (cidA01), and other information are described.
- Example of a thumbnail image set (see “2.3 Identifier of a Thumbnail Image Set” in Annex 4): `crid://example.jp/thumbnailset/thumb00A/cidB01`
In the <data> part, the content classification (thumbnailset), thumbnail set ID (thumb00A), content ID (cidB01), and other information are described.

8.2 Use of Location Resolution

Location resolution means to obtain time and space information that can be acquired by a piece of content, using a locator. Metadata is used as a resolution method (location resolution) to associate a content reference identifier and a locator, which indicates the time and space information that can be acquired by said content.

8.2.1 Description format of locator

A locator indicates the time and space information that can be acquired by a piece of content and is the information to obtain applicable content. The locator is the location information of the content that can be obtained by resolving a content reference identifier (location resolution) and that enables a receiver to obtain content.

<transport mechanism>:<transport system specific> is used as a locator format according to the format defined in “4.1.2.1 Format” in ARIB STD-B38.

- The namespace defined in “Chapter 9, Volume 2 of ARIB STD-B24” is used as a locator value of the content transmitted in broadcasting.
- The namespace defined in “Chapter 9, Volume 2 of ARIB STD-B24” is used as a locator value that indicates the content in a communication site.

The following shows a description example of a locator.

A description example when referencing an event in which an available unit of content is transmitted

```
arib://<original_network_id>.<transport_stream_id>.<service_id><event_id>
```

8.2.2 Transferring a locator

Metadata is used for transferring locator information.

The following provides the metadata location in which each locator is placed.

- Locator of real-time broadcasting content
Place in
/TVAMain/ProgramDescription/ProgramLocationTable/BroadcastEvent/ProgramURL.
- Locator of storable broadcasting content
Place in
/TVAMain/ProgramDescription/ProgramLocationTable/OnDemandService/OnDemandProgram/ProgramURL.
- Locator of an available unit of content in a receiver
It is not necessary to separately place the locator in metadata because a logical storage location in a receiver is described in a content reference identifier (see “8.2.3.3 Location resolution in a receiver”).

8.2.3 Guidelines for using location resolution

8.2.3.1 Location resolution of real-time broadcasting content

Location resolution can be performed using the following method if an available unit of content is transferred in broadcasting.

The location of the available unit of content that is transferred in broadcasting is specified in the namespace of the event to be transferred. The location resolution of the namespace for transmission from the content reference identifier can be performed by obtaining a content reference identifier, which is described in /TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/@programId of the metadata (program information element) that indicates an available unit of content, and by obtaining the URI described in /TVAMain/ProgramDescription/ProgramLocationTable/BroadcastEvent/ProgramURL of the

metadata (program location information element) in which the same content reference identifier as the content reference identifier obtained earlier is described.

Example: Metadata description when an available unit of content identifier is “crd://example.jp/20100401/rt/0001/12345” and when the broadcasting event's URI of an applicable available unit of content is “arib://0001.00111.0001.12345?time=2010-04-01T13:55/PT01H55M”

```
<ProgramInformationTable>
<ProgramInformation programId="crd://example.jp/20100401/rt/0001/12345" />
</ProgramInformationTable>

<ProgramLocationTable>
  <BroadcastEvent>
    <Program crd="crd://example.jp/20100401/rt/0001/12345"/>
    <ProgramURL>
arib://0001.00111.0001.12345?time=2010-04-01T13:55/PT01H55M</ProgramURL>
    </BroadcastEvent>
  </ProgramLocationTable>
```

8.2.3.2 Location resolution of storable broadcasting content

Location resolution can be performed using the following method if an available unit of content is transferred in storable broadcasting.

The location of an available unit of content that is distributed in storable broadcasting is specified using transmission control metadata (see “2.1.5.4 Transmission control metadata in Volume 11 of this operational standard). The location resolution of transmission control metadata from a content reference identifier can be performed by obtaining the content reference identifier that is described in /TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/@programId of the metadata (program information element) that indicates an available unit of content and by obtaining the URI that is described in the /TVAMain/ProgramDescription/ProgramLocationTable/OnDemandService/OnDemandProgram/ProgramURL of the metadata (program location information element) in which the same content reference identifier as the content reference identifier obtained earlier is described.

Example: Metadata description when an available unit of content identifier is “crd://example.jp/20100401/fd/0010/cid001” and when the transmission control metadata's URI of an applicable available unit of content is “http://example.jp/isdb-t/densouseigyocid001.fcj”

```
<ProgramInformationTable>
```

```
<ProgramInformation programId="crid://example.jp/20100401/fd/0010/cid001" />
</ProgramInformationTable>

<ProgramLocationTable>
  <OnDemandService>
    <OnDemandProgram>
      <Program crid="crid://example.jp/20100401/fd/0010/cid001"/>
      <ProgramURL>http://example.jp/isdb-t/densouseigyo/cid001.fci</ProgramURL>
    </OnDemandProgram>
  </OnDemandService>
</ProgramLocationTable>
```

Next, the information of the FLUTE session, in which an applicable available unit of content is transmitted, is obtained by acquiring the transmission object information (see “2.1.5.4.2 Session description” in Volume 11 of this operational standard) that is described in SessionDescription of the transmission control metadata. Then, the location of the available unit of content can be identified by obtaining the URI that is described in the Content-Location attribute of the FDT instance (see “2.1.5.3 FDT instance” in Volume 11 of this operational standard) in the FLUTE session.

8.2.3.3 Location resolution in a receiver

The location of each available unit of content, resource, and image, etc., in a receiver can be determined from the logical storage location allocated to a content reference identifier. In other words, a content reference identifier (CRID) is handled as the logical storage location (directory) of an available unit of content, and the available unit of content, transmission control metadata, and related files, etc., are placed in the directory. Each data file can be uniquely specified using a character string that is created by combining a content reference identifier and the logical location of each data file (including directory name) with a delimiter of “#.” It is only necessary that a receiver has access to each corresponding available unit of content using the obtained logical storage location. The physical storage location and storage method are determined based on an implemented receiver, and the storage location and method must conform to the guidelines for applicable storage media. For instance, when data is stored in a SD memory card, SD standard Part H2 is referenced. However, when data is stored in a SD memory card, it is handled as a special case and a route directory of “/private/tmm/” is used.

Example: When the content reference identifier is `crid://example.jp/20100401/fd/0010/cid001`, the video file is `fd1/movie.mpg`, and the transmission control metadata is located at `http://example.jp/isdb-t/densouseigyo/cid001.fci`.

- Logical location of an available unit of content (directory): `example.jp/20100401/fd/0010/cid001/`

- Storage location of a video file: example.jp/20100401/fd/0010/cid001/fd1/movie.mpg
- URI of a video file: crid://example.jp/20100401/fd/0010/cid001#fd1/movie.mpg
- Storage location of transmission control metadata:
example.jp/20100401/fd/0010/cid001/cid001.fci
- URI of transmission control metadata (in a receiver):
crid://example.jp/20100401/fd/0010/cid001#cid001.fci

* As described in “7.1.1 Metadata storage guidelines,” when metadata file (metadata.becg) that corresponds to stored content is stored in a directory in which the available unit of content is stored, the storage location is example.jp/20100401/fd/0010/cid001/metadata.becg and the URI (metadata file) is crid://example.jp/20100401/fd/0010/cid001#metadata.becg.

Note: As for a thumbnail image set (see “Identifier of a thumbnail image set” in 2.3 in Annex 4), given that multiple sets of thumbnail images for which the sizes are different are provided as described in “Grouping content with different versions” in 2.3 in Annex 4, a character string, which is created by deleting the last part of a content reference identifier as shown below, is used.

For instance, when the content reference identifier is crid://example.jp/thumbnailset/thumb00A/cidB01 and the thumbnail image file is a01.jpg to a99.jpg, a character string is created by deleting the last part of the content reference identifier as follows:

- Logical location of an available unit of content (directory): example.jp/thumbnailset/thumb00A/
- Storage location of a thumbnail image file: example.jp/thumbnailset/thumb00A/a01.jpg–a99.jpg
- URL of each thumbnail image file: crid://example.jp/thumbnailset/thumb00A#a01.jpg–a99.jpg

When a thumbnail image file that is included in an applicable thumbnail image set is used from certain metadata, the URI of each thumbnail image file as that above is described.

8.2.4 Layer resolution

8.2.4.1 Layer resolution of content and groups

The hierarchical relationship between a content reference identifier, which groups multiple content (package, group), and each piece of content, which configures a group, is expressed using the group information element in metadata. The relationship is mutually resolved based on the information.

The information that should be described in the group information element of metadata is as follows.

- The amount of content included in a group is described in GroupInformation/@numOfItems.

Example: Example of a group information element that expresses a package with two pieces of content

```
<GroupInformation groupId="crd://example.jp/pack/p0001" numOfItems="2">  
<GroupType value="package"/>  
</GroupInformation>
```

The following information should be described in the program information element of the metadata for each piece of content that configures a group.

- A group ID that organizes groups based on the crid attribute of a MemberOf element is described.

Example: Example of the program information element that expresses the content included in a package

```
<ProgramInformation programId="crd://example.jp/20100401/fd/0010/cid001">  
<MemberOf crid="crd://example.jp/pack/p0001" />  
</ProgramInformation>
```

Annex 1 Use of Classification Schemes

1 Common Items

The following table provides a list of the classification schemes referenced in this specification. If “this specification” is described in the reference standard field, then a standard is uniquely defined in this specification or added to the specification defined in “ARIB STD-B38.” If classification using a layered structure is applied, the top of the structure is specified. If a middle layer is specified, it is regarded that all the layers above the layer are specified.

It must be guaranteed that a classification scheme in the same category has upper compatibility. If a new dictionary item is added to an old classification scheme and if the classification scheme version is changed, the category part that is the last part of a newly provided URI is not changed and only the year-and-month part is changed. For instance, if a classification scheme version in “http://www.isdb-t.jp/cs/2011/03/ISDBTMMContentTypesCS” is changed, the last part of the old URI is used in a new URI as follows: “http://www.isdb-t.jp/cs/2012/03/ISDBTMMContentTypesCS.”

As for “http://www.isdb-t.jp/cs/2011/03/ISDBTMMContentTypesCS/1” and “http://www.isdb-t.jp/cs/2012/03/ISDBTMMContentTypesCS/1,” the same dictionary item is referenced if a classification scheme version is changed.

Table S1-1: Use of classification schemes

URI of classification scheme	Reference standard	Use	Place to be shown	Remarks
http://www.arib.or.jp/cs/2006/03/ARIBGenreCS (Note 1)	ARIB STD-B38	○	:Genre/@href	Usage range is “3. GENERAL AUDIENCE” and “4.2 TARGET GROUP.”
http://www.arib.or.jp/cs/2010/04/ARIBGenreCS2 (Note 1)	ARIB STD-B38	○	:Genre/@href	All items are used.
http://www.arib.or.jp/cs/2006/03/ARIBParentalRatingCS	ARIB STD-B38	○	:ParentalGuidance/mpeg7:ParentalRating/@href	For details on the usage range and for the usage per item, see 2 in Annex 1.
http://www.arib.or.jp/cs/2011/03/ARIBHowRelatedCS	ARIB STD-B38	○	:RelatedMaterial/HowRelated/@href	For details on the dictionary definition and for the usage per item, see 3 in Annex 1.
http://www.isdb-t.jp/cs/2011/03/ISDBTMMHowRelatedCS	This specification	○		

http://www.arib.or.jp/cs/2006/03/ARIBAudioPurposeCS	ARIB STD-B38	—		
http://www.arib.or.jp/cs/2010/04/ARIBPurchaseTypeCS	ARIB STD-B38	○	:PurchaseItem/Purchase/PurchaseType/@href	Usage range is “1 Single-unit sale,” “2 Pack sale,” and “4 Unlimited sale.” “3, 3.1, 3.2 Select sale” is not included.
http://www.arib.or.jp/cs/2010/04/ARIBUnitTypeCS	ARIB STD-B38	○	:PurchaseItem/Purchase/QuantityUnit/@href	Usage range is any items other than the “number of counts.”
http://www.arib.or.jp/cs/2010/04/ARIBPromotionalTypeCS	ARIB STD-B38	○	:PromotionalInformation/@href	All items are used.
http://www.arib.or.jp/cs/2011/03/ARIBRoleCS	ARIB STD-B38	○	:CreditsList/CreditsItem/@role	All items are used.
http://www.arib.or.jp/cs/2010/04/ARIBAudioComponentCS	ARIB STD-B38	○	:AVAttributes/AudioAttributes/MixType/@href	All items are used.
http://www.arib.or.jp/cs/2011/03/ARIBAudioCodingFormatCS	ARIB STD-B38	○	:AVAttributes/AudioAttributes/Coding/@href	All items are used.
http://www.arib.or.jp/cs/2011/03/ARIBVisualCodingFormatCS	ARIB STD-B38	○	:AVAttributes/VideoAttributes/Coding/@href	All items are used.
http://www.arib.or.jp/cs/2011/03/ARIBFileFormatCS	ARIB STD-B38	○	:RelatedMaterial/FileFormat/@href	All items are used.
http://www.isdb-t.jp/cs/2011/03/ISDBTMMFileFormatCS	This specification	○	:AVAttributes/FileFormat :ContentProperties/FileProperties/@href	All items are used. See 4 in Annex 1 for the dictionary definition.
http://www.arib.or.jp/cs/2013/07/ARIBOutputPortCS	ARIB STD-B38	○	:OutputRestriction/Port/@href	All items are used.
http://www.arib.or.jp/cs/2013/07/ARIBCopyControlMethodCS	ARIB STD-B38	○	:OutputRestriction/Mode/@href	All items are used.
http://www.arib.or.jp/cs/2010/04/ARIBExportMediaCS	ARIB STD-B38	—		
http://www.isdb-t.jp/cs/2011/03/ISDBTMMContentTypeCS	This specification	○	:ContentProperties/ContentType/@href	See 5 in Annex 1 for details on the dictionary definition and for the usage of each item.
http://www.isdb-t.jp/cs/2011/03/ISDBTMMDeliveryStatusCS	This specification	○	:Genre/@href	All items are used. See 6 in Annex 1 for the dictionary definition.

http://www.isdb-t.jp/cs/2011/03/ISDBTMMItemCategoryCS	This specification	○	:Genre/@href	See 7 in Annex 1 for details on the dictionary definition and for the usage of each item.
http://www.isdb-t.jp/cs/2011/03/ISDBTMMIntendedAudienceCS	This specification	○	:Genre/@href	All items are used. See 8 in Annex 1 for the dictionary definition.
http://www.isdb-t.jp/cs/2011/03/ISDBTMMStereoVideoFormatCS	This specification	○	:AVAttributes/VideoAttributes/StereoVideoFormat/@href	See 9 in Annex 1 for details on the dictionary definition and for the usage of each item.

Note 1: A genre that is individually used by a service provider can be used by separately defining one's own GenreCS.

2 Use of the ARIBParentalRatingCS Dictionary

As for the PG rating provided as the parental rating specification defined in “Table Annex A.2-1 Name of parental rating” in “Annex A.2 Parental Guidance Dictionary” in ARIB STD-B38, only PG-12 can handle R ratings from R-4 to R-20. All other PG ratings are specified as a G rating (if there is no age limit). The following table describes a recommended minimum viewing age for each specified item and precautions implementing a receiver. For details on parental control in multimedia broadcasting, see “9. Parental Control Guidelines” in Appendix 1, Volume 2 of this operational standard. As for each rating, it is defined as follows: a PG rating indicates that a program includes improper content for viewers under a specified age; an R rating indicates that a program should have a restricted viewing so that it is not viewed by those under a specified age; and an N rating indicates that caution needs to be taken when viewing a program because it is difficult to set an age limit.

Table S1-2: Use of the ARIBParentalRatingCS dictionary

Name	Recommended minimum viewing age	Precautions on the implementation of a receiver
R-4	4	Display and viewing limitations in the ECG are not applied regardless of the result obtained when comparing a parental rating specification (parental route) and a parental level (minimum viewing age) set to a receiver.
R-5	5	
R-6	6	
R-7	7	
R-8	8	
R-9	9	
R-10	10	
R-11	11	Viewing limitation is applied by comparing a parental rating specification (parental route) and a parental level (minimum viewing age) set to a receiver. Viewing can be limited without restricting display in the ECG. However, if thumbnail images are displayed, they can be displayed only when the parental control is removed.
PG-12, R-12	12	
R-13	13	
R-14	14	
R-15	15	
R-16	16	
R-17	17	
R-18	18	
R-19	19	Applicable content cannot be displayed and viewed in the ECG. However, such content can be displayed and viewed in the ECG only when the parental control is removed by entering a password or by another method.
R-20	20	
G	Without age limit	Display and viewing limitations in the ECG are not applied regardless of the result obtained when comparing a parental rating specification (parental route) and a parental level (minimum viewing age) set to a receiver.

```

<ClassificationScheme uri="http://www.arib.or.jp/cs/2006/03/ARIBParentalRatingCS">
  <Term termID="PG-4">
    <Name xml:lang="en">PG-4</Name>
    <Definition xml:lang="ja">A program contained improper content for viewers under 4 years
of age.</Definition>
  </Term>
  <Term termID="PG-5">
    <Name xml:lang="en">PG-5</Name>
    <Definition xml:lang="ja">A program contained improper content for viewers under 5 years
of age .</Definition>
  </Term>
  <Term termID="PG-6">
    <Name xml:lang="en">PG-6</Name>
    <Definition xml:lang="ja">A program contained improper content for viewers under 6 years
of age.</Definition>
  </Term>
  <Term termID="PG-7">
    <Name xml:lang="en">PG-7</Name>
    <Definition xml:lang="ja">A program contained improper content for viewers under 7 years
of age.</Definition>
  </Term>
  <Term termID="PG-8">
    <Name xml:lang="en">PG-8</Name>
    <Definition xml:lang="ja">A program contained improper content for viewers under 8 years
of age.</Definition>
  </Term>
  <Term termID="PG-9">
    <Name xml:lang="en">PG-9</Name>
    <Definition xml:lang="ja">A program contained improper content for viewers under 9 years
of age.</Definition>
  </Term>
  <Term termID="PG-10">
    <Name xml:lang="en">PG-10</Name>
    <Definition xml:lang="ja">A program contained improper content for viewers under 10
years of age.</Definition>
  </Term>
  <Term termID="PG-11">
    <Name xml:lang="en">PG-11</Name>
    <Definition xml:lang="ja">A program contained improper content for viewers under 11
years of age.</Definition>
  </Term>
  <Term termID="PG-12">
    <Name xml:lang="en">PG-12</Name>
    <Definition xml:lang="ja">A program contained improper content for viewers under 12
years of age.</Definition>
  </Term>
  <Term termID="PG-13">
    <Name xml:lang="en">PG-13</Name>

```

```
<Definition xml:lang="ja">A program contained improper content for viewers under 13
years of age.</Definition>
</Term>
<Term termID="PG-14">
  <Name xml:lang="en">PG-14</Name>
  <Definition xml:lang="ja">A program contained improper content for viewers under 14
years of age.</Definition>
</Term>
<Term termID="PG-15">
  <Name xml:lang="en">PG-15</Name>
  <Definition xml:lang="ja">A program contained improper content for viewers under 15
years of age.</Definition>
</Term>
<Term termID="PG-16">
  <Name xml:lang="en">PG-16</Name>
  <Definition xml:lang="ja">A program contained improper content for viewers under 16
years of age.</Definition>
</Term>
<Term termID="PG-17">
  <Name xml:lang="en">PG-17</Name>
  <Definition xml:lang="ja">A program contained improper content for viewers under 17
years of age.</Definition>
</Term>
<Term termID="PG-18">
  <Name xml:lang="en">PG-18</Name>
  <Definition xml:lang="ja">A program contained improper content for viewers under 18
years of age.</Definition>
</Term>
<Term termID="PG-19">
  <Name xml:lang="en">PG-19</Name>
  <Definition xml:lang="ja">A program contained improper content for viewers under 19
years of age.</Definition>
</Term>
<Term termID="PG-20">
  <Name xml:lang="en">PG-20</Name>
  <Definition xml:lang="ja">A program contained improper content for viewers under 20
years of age.</Definition>
</Term>
<Term termID="R-4">
  <Name xml:lang="en">R-4</Name>
  <Definition xml:lang="ja"> A program that should be restricted for any viewer under 4 years
of age.</Definition>
</Term>
<Term termID="R-5">
  <Name xml:lang="en">R-5</Name>
  <Definition xml:lang="ja"> A program that should be restricted for any viewer under 5 years
of age.</Definition>
</Term>
<Term termID="R-6">
```



```

    <Name xml:lang="en">R-6</Name>
    <Definition xml:lang="ja"> A program that should be restricted for any viewer under 6 years
of age.</Definition>
  </Term>
  <Term termID="R-7">
    <Name xml:lang="en">R-7</Name>
    <Definition xml:lang="ja"> A program that should be restricted for any viewer under 7 years
of age.</Definition>
  </Term>
  <Term termID="R-8">
    <Name xml:lang="en">R-8</Name>
    <Definition xml:lang="ja"> A program that should be restricted for any viewer under 8 years
of age.</Definition>
  </Term>
  <Term termID="R-9">
    <Name xml:lang="en">R-9</Name>
    <Definition xml:lang="ja"> A program that should be restricted for any viewer under 9 years
of age.</Definition>
  </Term>
  <Term termID="R-10">
    <Name xml:lang="en">R-10</Name>
    <Definition xml:lang="ja"> A program that should be restricted for any viewer under 10
years of age.</Definition>
  </Term>
  <Term termID="R-11">
    <Name xml:lang="en">R-11</Name>
    <Definition xml:lang="ja"> A program that should be restricted for any viewer under 11
years of age.</Definition>
  </Term>
  <Term termID="R-12">
    <Name xml:lang="en">R-12</Name>
    <Definition xml:lang="ja"> A program that should be restricted for any viewer under 12
years of age.</Definition>
  </Term>
  <Term termID="R-13">
    <Name xml:lang="en">R-13</Name>
    <Definition xml:lang="ja"> A program that should be restricted for any viewer under 13
years of age.</Definition>
  </Term>
  <Term termID="R-14">
    <Name xml:lang="en">R-14</Name>
    <Definition xml:lang="ja"> A program that should be restricted for any viewer under 14
years of age.</Definition>
  </Term>
  <Term termID="R-15">
    <Name xml:lang="en">R-15</Name>
    <Definition xml:lang="ja"> A program that should be restricted for any viewer under 15
years of age.</Definition>
  </Term>

```

```
<Term termID="R-16">
  <Name xml:lang="en">R-16</Name>
  <Definition xml:lang="ja"> A program that should be restricted for any viewer under 16
years of age.</Definition>
</Term>
<Term termID="R-17">
  <Name xml:lang="en">R-17</Name>
  <Definition xml:lang="ja"> A program that should be restricted for any viewer under 17
years of age.</Definition>
</Term>
<Term termID="R-18">
  <Name xml:lang="en">R-18</Name>
  <Definition xml:lang="ja"> A program that should be restricted for any viewer under 18
years of age.</Definition>
</Term>
<Term termID="R-19">
  <Name xml:lang="en">R-19</Name>
  <Definition xml:lang="ja"> A program that should be restricted for any viewer under 19
years of age.</Definition>
</Term>
<Term termID="R-20">
  <Name xml:lang="en">R-20</Name>
  <Definition xml:lang="ja"> A program that should be restricted for any viewer under 20
years of age.</Definition>
</Term>
<Term termID="N">
  <Name xml:lang="en">N</Name>
  <Definition xml:lang="ja">A program to which caution should be applied when viewing
because it is difficult to specify an age limit.</Definition>
</Term>
<Term termID="G">
  <Name xml:lang="en">G</Name>
  <Definition xml:lang="ja">Program without an age limit</Definition>
</Term>
</ClassificationScheme>
</ClassificationSchemeTable>
```

3 Use of the HowRelatedCS Dictionary

The following provides a definition for the ARIBHowRelatedCS dictionary and ISDBTMMHowRelatedCS dictionary referenced in this operational standard.

```
<ClassificationScheme uri="http://www.arib.or.jp/cs/2011/03/ARIBHowRelatedCS">
  <Term termID="1">
    <Name xml:lang="ja">Preview</Name>
    <Definition xml:lang="ja">Relationship: The current A/V content is a preview of a program that
is associated with a CRID. Example: Record a movie preview.</Definition>
  </Term>
  <Term termID="2">
```

```

<Name xml:lang="ja">Group preview</Name>
<Definition xml:lang="ja">Relationship: The current A/V content is a preview of a program
group that is associated with a CRID. Example: Record the preview of a series program that will
start in the near future.</Definition>
</Term>
<Term termID="3">
  <Name xml:lang="ja">Brother</Name>
  <Definition xml:lang="ja">Relationship: A program that is associated with a CRID and the
current A/V content have a brotherly relationship. Example: Watch the previous episode while
recording the next episode of the series.</Definition>
</Term>
<Term termID="4">
  <Name xml:lang="ja">Substitution</Name>
  <Definition xml:lang="ja">Relationship: The identifier of a CRID references another version of a
program that has the same content and that is edited the same way as the current A/V stream.
Example: While watching a program or part of a program, a user finds that the high-definition
version of the program can be obtained from another location.</Definition>
</Term>
<Term termID="5">
  <Name xml:lang="ja">Parent</Name>
  <Definition xml:lang="ja">Relationship: Pieces of content in a program group are associated
with each other using a CRID, with which the current A/V content is also associated. Example: While
watching an episode, all series are recorded.</Definition>
</Term>
<Term termID="6">
  <Name xml:lang="ja">Recommendation</Name>
  <Definition xml:lang="ja">Relationship: Broadcasters think that there is a relationship between
the current A/V content and a program that is associated with a CRID. Example: Record a program
recommended by a broadcaster because a user watches it.</Definition>
</Term>
<Term termID="7">
  <Name xml:lang="ja">Group recommendation</Name>
  <Definition xml:lang="ja">Relationship: Broadcasters think that there is a relationship between
the current A/V content and a program group that is associated with a CRID. Example: Record a
program series recommended by broadcasters because many users watch it.</Definition>
</Term>
<Term termID="8">
  <Name xml:lang="ja">Commercial advertisement</Name>
  <Definition xml:lang="ja">Relationship: Items or services that appeared in the current A/V
content are advertised in another place. A CRID is used to create an association with the A/V content
of the advertisement. Example: A user watches a video in which an item appears that they want. If
the user shows an interest in the item, an advertisement that provides the detailed information is
recorded.</Definition>
</Term>
<Term termID="9">
  <Name xml:lang="ja">Direct item purchase</Name>
  <Definition xml:lang="ja">Relationship: Items or services are directly linked to a program being
viewed. Users can directly purchase from the linked resource. Example: A user is watching a video in
which an item or service is advertised that the user wants (for instance, a recipe book from a cooking

```

series). If the user shows an interest in the item, then they are guided to a website where a purchase can be made (or an interactive application).</Definition>

</Term>

<Term termID="10">

<Name xml:lang="ja">Additional information</Name>

<Definition xml:lang="ja">Relationship: A program has additional information in a format such as "audio/video/character/image/interactive application/web content." Example: A user is watching a program in which additional information is prepared by the content provider for the program. When the user shows an interest, they are then guided by the additional content and returned to the original content after viewing the additional content.</Definition>

</Term>

<Term termID="11">

<Name xml:lang="ja">Program review information</Name>

<Definition xml:lang="ja">Relationship: A program has a review or criticism that may influence a user's decision of whether to continue watching the program. Example: A user can view additional information and use the information to decide whether to continue watching the program.</Definition>

</Term>

<Term termID="12">

<Name xml:lang="ja">Summary</Name>

<Definition xml:lang="ja">Relationship: A program in a series shows a text summary or AV summary. Example: A user can read or view a summary when the user misses the previous episode or forgets the story line of the series.</Definition>

</Term>

<Term termID="13">

<Name xml:lang="ja">Making</Name>

<Definition xml:lang="ja">Relationship: Broadcasters provide the "making" of a program or information. Example: A user can view the background of how a program is produced if the user has an interest.</Definition>

</Term>

<Term termID="14">

<Name xml:lang="ja">Support</Name>

<Definition xml:lang="ja">Relationship: A program that contains a problem on which users may want to ask questions. Example: As the intention of a program, a user can know whether support through telephone consultation or via mail, e-mail, or a website is provided to get advice.</Definition>

</Term>

<Term termID="15">

<Name xml:lang="ja">Derivation</Name>

<Definition xml:lang="ja">Relationship: The current A/V content is developed from a program that is associated with a CRID. Example: A user can know that an item (such as a novel based on a scenario and a program based on a side story related to a character) is developed from a program that the user is watching.</Definition>

</Term>

<Term termID="16">

<Name xml:lang="ja">Segmentation</Name>

<Definition xml:lang="ja">Relationship: Pointer to a segmentation group</Definition>

</Term>

<Term termID="17">

```

<Name xml:lang="ja">Reference</Name>
<Definition xml:lang="ja">Relationship: Content referenced from the current content. Example:
Real-time broadcasting content, which is used by referencing storable broadcasting content, can be
known. If terrestrial multimedia broadcasting with a segment connection transmission method is
provided as a download, then real-time broadcasting content, which is used by referencing
downloaded-based broadcasting content, can be known.</Definition>
</Term>
<Term termID="18">
  <Name xml:lang="ja">License information</Name>
  <Definition xml:lang="ja">Relationship: Pointer to license information. Example: The license
information of the content (available period, etc.) can be known.</Definition>
</Term>
<Term termID="19">
  <Name xml:lang="ja">Main part</Name>
  <Definition xml:lang="ja">Relationship: This indicates the main part of the
content.</Definition>
</Term>
<Term termID="20">
  <Name xml:lang="ja">High image quality version</Name>
  <Definition xml:lang="ja">Relationship: This indicates high-definition content.</Definition>
</Term>
<Term termID="21">
<Name xml:lang="ja">Package reference</Name>
<Definition xml:lang="ja">Relationship: Reference to a group information element in which a
package is described</Definition>
</Term>
  <Term termID="21.1">
    <Name xml:lang="ja">GI under a pack</Name>
    <Definition xml:lang="ja">Relationship: A link to a group information element in which a series
is described from a group information element in which a package is described</Definition>
  </Term>
  <Term termID="21.2">
    <Name xml:lang="ja">PI under a pack</Name>
    <Definition xml:lang="ja">Relationship: A link to PI from a group information element in which
a package is described.</Definition>
  </Term>
</ClassificationScheme>

```

The table below provides the detailed usage on each item described above. In the “Use” field in the table, “○” indicates that an item is used and “—” indicates that an item is not used. “Target” represents a target indicated by the RelatedMaterial/MediaLocator/MediaUri of an applicable program information element, group information element, segment information element, or segment group information element of the applicable content. A “text displayed in a receiver” is a character string that represents the relationship between applicable content and is recommended for use when presenting information about a target on a receiver. “Usage type” indicates how to process a target in a receiver (expected receiver operation). ① is applied when a target is a piece of

content. ② is applied when a target specifies the URL of a browser script file. When ① is applied, the metadata details of applicable content is displayed or applicable content is played back. When ② is applied, an applicable script is downloaded to be processed in a browser. The definition of each item above (<Definition> element details) is used by redefining it based on the “Relationship definition” details in the table below.

Table S1-3: Use of the ARIBHowRelataCS dictionary

TermID	Use	Text displayed in a receiver	Target (type)	Usage type	Relationship definition	Usage example
1	○	Preview	Content	①	As described above	
2	○	Group preview	Content	①	As described above	
3	○	Brother content	Content	①	As described above	
4	○	Another version	Content	①	As described above	
5	○	Parent content	Content	①	As described above	
6	○	Recommended content	Content	①	Recommended work related to applicable content	The target is content recommended by a business operator.
7	○	Related content	Content	①	Work related to applicable content	
8	○	Relevant advertisement information	Content URL	①, ②	Advertisement information related to applicable content	The target is a relevant advertisement or sponsor site.
9	○	Relevant item purchase	Content URL	①, ②	Item purchase information related to applicable content	The target is relevant item purchase information or the item purchase site.
10	○	Related information	Content URL	①, ②	Relative information of applicable content	The target is relevant information of applicable content.
11	—	Program review				
12	—	Summary				
13	○	Making	Content	①	“Making of”	

					video of applicable content	
14	—	Support				
15	○	Derivation content	Content	①	As described above	
16	○	Segmentation group	Content	①	As described above	
17	○	Reference	Content	①	As described above	
18	○	License information	License reference information element (LicenseID)	Note 1	As described above	
19	○	Main part	Content	①	As described above	
20	○	High-image-quality version	Content	①	As described above	
21	○	Pack reference	Group information element in which a package is described.	①	A link to a group information element, in which a package is described, from a group information element, in which a series is described	
21.1	○	GI under a pack	Group information element in which a series is described	①	A link to a group information element, in which a series is described, from a group information element, in which a package is described	Cross-layer transition is realized from a group information element in which a package is described.
21.2	○	PI under a pack	Program information element	①	A link to a program information	Cross-layer transition is realized from a group information

					element from a group information element in which a package is described	element in which a package is described.
--	--	--	--	--	--	--

Note 1: A license ID is specified.

In this operational standard, in addition to the ARIBHowRelatedCS dictionary above, the ISDBTMMHowRelatedCS dictionary that defines the ISDB-tmm specific items is referenced. The following items are used.

```

<ClassificationScheme uri="http://www.isdb-t.jp/cs/2011/03/ISDBTMMHowRelatedCS">
  <Term termID="22">
    <Name xml:lang="en">Series Info</Name>
    <Name xml:lang="ja">Series information</Name>
    <Definition xml:lang="ja">A pointer to the series information to be referenced. Example: The series identification (series_id) of the program arrangement information (SI) to be referenced can be known.</Definition>
  </Term>
  <Term termID="23">
    <Name xml:lang="en">Series GI</Name>
    <Name xml:lang="ja">Series GI</Name>
    <Definition xml:lang="ja">A pointer to a series GI to be referred. Example: A link from a program information element (PI) to a series GI (CRID contains series_id)</Definition>
  </Term>
  <Term termID="24">
    <Name xml:lang="en">hyper linkage type</Name>
    <Name xml:lang="ja">Hyperlink type</Name>
    <Definition xml:lang="ja">Mapping from EIT/hyperlink_descriptor/hyper_linkage_type</Definition>
  </Term>
  <Term termID="24.1">
    <Name xml:lang="en">combined data</Name>
    <Definition xml:lang="ja">When relative information is broadcasted in a timeframe that is different from a TV program or in a different service, combined data is used to specify a data broadcasting program in which the relative information is broadcasted in the TV program.</Definition>
  </Term>
  <Term termID="24.2">
    <Name xml:lang="en">combined stream</Name>
    <Definition xml:lang="ja">When relative information is broadcasted in a timeframe that is different from a TV program or in a different service, a combined stream is used to specify a TV program from a data broadcasting program in which relative information is broadcasted.</Definition>
  </Term>
  <Term termID="24.3">

```



```

<Name xml:lang="en">content to index</Name>
<Definition xml:lang="ja">When the index information in a program is broadcasted in a
timeframe that is different from a TV program or in a different service, the content to index is used to
specify a program, in which the index information in the program is broadcasted, from a TV
program.</Definition>
</Term>
<Term termID="24.4">
  <Name xml:lang="en">index to content</Name>
  <Definition xml:lang="ja">When the index information in a program is broadcasted in a
timeframe that is different from a TV program or in a different service, the index to content is used to
specify a TV program from a program in which the index information in the program is
broadcasted.</Definition>
</Term>
<Term termID="24.5">
  <Name xml:lang="en">guide data</Name>
  <Definition xml:lang="ja">When the program guide information of an applicable program is
broadcasted in a different data broadcasting program, the program guide data broadcasting program
is specified.</Definition>
</Term>
<Term termID="24.7">
  <Name xml:lang="en">content to metadata</Name>
  <Definition xml:lang="ja">When metadata is broadcasted in a timeframe that is different
from a TV program/data broadcasting program or in a different service, the content to metadata is
used to specify a program or service in which the metadata is broadcasted from the TV program/data
broadcasting program.</Definition>
</Term>
<Term termID="24.8">
  <Name xml:lang="en">metadata to content</Name>
  <Definition xml:lang="ja">When metadata is broadcasted in a timeframe that is different
from a TV program/data broadcasting program or in a different service, the metadata to content is
used to specify a TV program/data broadcasting program from a program or service in which the
metadata is broadcasted.</Definition>
</Term>
<Term termID="24.9">
  <Name xml:lang="en">portal URI</Name>
  <Definition xml:lang="ja">This is used to indicate a portal link destination URI.</Definition>
</Term>
<Term termID="24.10">
  <Name xml:lang="en">authority URI</Name>
  <Definition xml:lang="ja">This is used to indicate an authority URI.</Definition>
</Term>
<Term termID="24.40">
  <Name xml:lang="en">index module</Name>
  <Definition xml:lang="ja">An index module is used only in the LIT that is used as an index in
a data broadcasting program, and also used to indicate a correspondence between a local event
identification module and a data broadcasting program module.</Definition>
</Term>
<Term termID="25">
  <Name xml:lang="en">image</Name>

```

```
<Name xml:lang="ja">Image</Name>
<Definition xml:lang="ja">This is used to specify an image in an element other than the
MediaTitle/TitleImage.</Definition>
</Term>
<Term termID="25.1">
  <Name xml:lang="en">image large</Name>
  <Name xml:lang="ja">Image (large)</Name>
</Term>
<Term termID="25.2">
  <Name xml:lang="en">image small</Name>
  <Name xml:lang="ja">Image (small)</Name>
</Term>
<Term termID="25.3">
  <Name xml:lang="en">channel logo</Name>
  <Name xml:lang="ja">Channel logo</Name>
<Definition xml:lang="ja">Channel logo image that is used for EPG, etc.</Definition>
</Term>
<Term termID="26">
  <Name xml:lang="en">external application</Name>
  <Name xml:lang="ja">External application</Name>
  <Definition xml:lang="ja">This is used when starting an external application.</Definition>
</Term>
<Term termID="26.1">
  <Name xml:lang="en">browser</Name>
  <Name xml:lang="ja">Browser</Name>
  <Definition xml:lang="ja">Starting a browser (providing a described URL)</Definition>
</Term>
<Term termID="26.2">
  <Name xml:lang="en">mail</Name>
  <Name xml:lang="ja">Mail</Name>
  <Definition xml:lang="ja">Starting an e-mail (providing a described e-mail
address)</Definition>
</Term>
<Term termID="26.3">
  <Name xml:lang="en">native</Name>
  <Name xml:lang="ja">Native application</Name>
  <Definition xml:lang="ja">Starting a native application</Definition>
</Term>
<Term termID="27">
  <Name xml:lang="en">comment</Name>
  <Name xml:lang="ja">Comment</Name>
  <Definition xml:lang="ja">This indicates a URL for posting comments on applicable
content.</Definition>
</Term>
<Term termID="28">
  <Name xml:lang="en">simultaneous store</Name>
  <Name xml:lang="ja">Simultaneous storage</Name>
  <Definition xml:lang="ja">A CRID of the content that is stored (reserved) is described when
storing (reserving) the content.</Definition>
```

```
</Term>
</ClassificationScheme>
```

4 Use of the FileFormatCS Dictionary

This section describes the definition of the ISDBTMMFileFormatCS dictionary that is referenced in this operational standard. A file format for the contractor's original content or receiver business operator's original content can be used by separately defining their original FileFormatCS. It is allowed to uniquely define the same file format (the same termID) in each business operator's original FileFormatCS. If the same file format is defined, a receiver distinguishes between them with a URI (http://***/XXFileFormatCS/<termID>).

“html, mp4, jpg, gif” in ARIBFileFormatCS and “container” in ISDBTMMFileFormatCS are the file formats for the ISDBTMM common content (any terminals can be used).

```
<ClassificationScheme uri="http://www.isdb-t.jp/cs/2011/03/ISDBTMMFileFormatCS">
  <!-- ISDBTMM common (all terminals can be used) -->
    <!-- "html, mp4, jpg, gif" in the ARIBFileFormatCS -->
    <!-- For content that includes multiple resources (multiple moving image files) * See Section
2.4 in Annex 4.-->
    <Term termID="container">
      <Name xml:lang="en">container</Name>
      <Definition xml:lang="ja">Content with multiple resources</Definition>
    </Term>

    <!-- Contractor's original (availability dependent on a terminal) -->
    <!-- "pdf, flv, swf, wmv, wma, mp3, mpo, png, mpg, ts, txt, shtml, xhtml, xml, bml, cgi, saac,
etc." in the ARIBFileFormatCS -->
    <!--The contractor's original content can be defined not only in ISDBTMMFileFormatCS but also in
the contractor's original FileFormatCS. -->

    <!-- Receiver business operator's original (only a specific receiver business operator's terminal can
be used) -->
    <!-- Defined in the original FileFormatCS of each receiver business operator -->

    <!-- Mapping EIT/hyperlink_descriptor/link_destination_type -->
    <Term termID="eit.1 ">
      <Name xml:lang="en">link to service</Name>
      <Definition xml:lang="ja">A link target is a service.</Definition>
    </Term>
    <Term termID="eit.2">
      <Name xml:lang="en">link to event</Name>
      <Definition xml:lang="ja">A link target is an event.</Definition>
    </Term>
    <Term termID="eit.3">
      <Name xml:lang="en">link to module</Name>
      <Definition xml:lang="ja">A link target is a specific module of an event.</Definition>
```

```
</Term>
<Term termID="eit.4">
  <Name xml:lang="en">link to content</Name>
  <Definition xml:lang="ja">A link target is a piece of content.</Definition>
</Term>
<Term termID="eit.5">
<Name xml:lang="en">link_to_content_module</Name>
  <Definition xml:lang="ja">A link target is the specific module of a content.</Definition>
</Term>
<Term termID="eit.6">
  <Name xml:lang="en">link to ert node</Name>
  <Definition xml:lang="ja">A link target is the node of an event-related table.</Definition>
</Term>
<Term termID="eit.7">
  <Name xml:lang="en">link to stored content</Name>
  <Definition xml:lang="ja">A link target is a storage content.</Definition>
</Term>
</ClassificationScheme>
```

5 Use of the ISDBTMMContentTypesCS Dictionary

This section describes the definition of ISDBTMMContentTypesCS dictionary that is referenced in this operational standard. ISDBTMMContentTypesCS is described in ContentProperties/ContentType/@href to define the type of applicable content. The type of a contractor's original content or receiver business operator's original content can be used by being separately defined in each business operator's original ContentTypesCS. It is allowed to uniquely define the same content classification (the same termID) in each business operator's original ContentTypesCS. If the same content classification is defined, a receiver distinguishes each other with a URI (http://***/XXContentTypesCS/<termID>).

The metadata that is specified as ISDBTMMContentTypesCS/1.7 is processed soon after reception is done in all receivers regardless of whether or not the content version, autoplay, synchplay, force_receive, or auto_receive is specified.

The metadata that is specified as ISDBTMMContentTypesCS/1.11 is processed based on the conditions of content version, autoplay, synchplay, force_receive, and auto_receive specifications. If this classification is specified, Base: 3.0 or later must be specified as the content version.

```
<ClassificationScheme uri="http://www.isdb-t.jp/cs/2011/03/ISDBTMMContentTypesCS">
<!--
Content classification is defined and described in :ContentProperties/ContentType/@href.
// -->
<!-- Content classification of storable broadcasting content -->
<!-- Common to ISDBTMM (all terminals can be used) -->
<Term termID="1">
  <Name xml:lang="ja">Common to ISDBTMM</Name>
```

```

    <Definition>This ID is used for classification and not actually applied.</Definition>
  </Term>
<Term termID="1.1">
<Name xml:lang="ja">Audio</Name>
</Term>
<Term termID="1.2">
<Name xml:lang="ja">Video</Name>
</Term>
<Term termID="1.3">
<Name xml:lang="ja">Image</Name>
</Term>
<Term termID="1.4">
<Name xml:lang="ja">Document</Name>
</Term>
<Term termID="1.5">
<Name xml:lang="ja">HTML document</Name>
</Term>
<Term termID="1.6">
<Name xml:lang="ja">Package</Name>
</Term>
<Term termID="1.7">
<Name xml:lang="ja">Metadata set</Name>
</Term>
<Term termID="1.9">
<Name xml:lang="ja">Thumbnail set</Name>
</Term>
<Term termID="1.10">
<Name xml:lang="ja">CS dictionary set</Name>
</Term>
<Term termID="1.11">
<Name xml:lang="ja">Metadata set 2</Name>
</Term>

  <!-- Contractor's original (availability dependent on a terminal) -->
  <Term termID="2">
    <Name xml:lang="ja">Contractor's original</Name>
    <Definition>This ID is used for classification and not actually applied.</Definition>
  </Term>
<Term termID="2.1">
<Name xml:lang="ja">Banner advertisement (individual)</Name>
<Definition>This is specified when a certain file format, such as JPG, GIF, MP4, SWF, or FLV, is
used.</Definition>
</Term>
<Term termID="2.2">
<Name xml:lang="ja">PDF document</Name>
</Term>

<!-- The contractor's original content can be defined not only in ISDBTMMContentypeCS but also
in the contractor's original ContentypeCS. -->

```

```
<!-- Receiver business operator's original (only a terminal of a specific receiver business operator
can be used) -->
<Term termID="3">
  <Name xml:lang="ja">Receiver business operator's original</Name>
  <Definition>A receiver business operator's original content is defined in each receiver business
operator's original ContentTypeCS.</Definition>
</Term>

<!-- Content classification of real-time broadcasting content -->
<Term termID="4">
  <Name xml:lang="ja">Real-time broadcasting content</Name>
  <Definition>This ID is used for classification and not actually applied.</Definition>
</Term>
  <Term termID="4.1">
    <Name xml:lang="ja">Audio streaming broadcasting</Name>
  </Term>
  <Term termID="4.2">
    <Name xml:lang="ja">Video streaming broadcasting</Name>
  </Term>
  <Term termID="4.3">
    <Name xml:lang="ja">Program linked data broadcasting</Name>
  </Term>
  <Term termID="4.4">
    <Name xml:lang="ja">Independent data broadcasting</Name>
  </Term>
</ClassificationScheme>
```

6 Use of the ISDBTMMDeliveryStatusCS Dictionary

This section defines the definition of the ISDBTMMDeliveryStatusCS dictionary that is referenced in this operational standard. ISDBTMMDeliveryStatusCS is described in :Genre/@href and defines the distribution status of applicable content. For details, see “2.2 Description of various control information using Genre [@type=“other”]” in Annex 4.

```
<ClassificationScheme uri="http://www.isdb-t.jp/cs/2011/03/ISDBTMMDeliveryStatusCS">
<!--
  The distribution status of the content is defined and described in :Genre/@href.
// -->
  <Term termID="1">
    <Name xml:lang="ja">Delivery status</Name>
    <Name xml:lang="en">delivery status</Name>
  </Term>
  <Term termID="1.1">
    <Name xml:lang="ja">Scheduled for distribution</Name>
    <Name xml:lang="en">coming soon</Name>
    <Definition xml:lang="ja">The distribution start date and time have not been decided, but the
content will be distributed in the near future. Metadata can be displayed.</Definition>
```

```

</Term>
<Term termID="1.2">
  <Name xml:lang="ja">Distributing</Name>
  <Name xml:lang="en">available</Name>
  <Definition xml:lang="ja">Distributing</Definition>
</Term>
<Term termID="1.3">
  <Name xml:lang="ja">Distribution suspended</Name>
  <Name xml:lang="en">suspension</Name>
  <Definition xml:lang="ja">Distribution has been temporarily suspended. It has not been
decided yet whether distribution will be restarted. Metadata can be displayed. </Definition>
</Term>
<Term termID="1.4">
  <Name xml:lang="ja">Distribution ended</Name>
  <Name xml:lang="en">end</Name>
  <Definition xml:lang="ja">Distribution has been ended. At present, there is no plan to restart
distribution. The display of metadata is optional.</Definition>
</Term>
<Term termID="1.5">
  <Name xml:lang="ja">Distribution cancelled</Name>
  <Name xml:lang="en">cancel</Name>
  <Definition xml:lang="ja">Distribution has been cancelled. At present, there is no plan to
restart distribution. Metadata cannot be displayed. It is recommended to display a message
indicating that distribution has been cancelled.</Definition>
</Term>
<Term termID="1.6">
  <Name xml:lang="ja">Postponed</Name>
  <Name xml:lang="en">postponement</Name>
  <Definition xml:lang="ja">Distribution is postponed. At present, a new distribution date and
time are not being considered. Metadata can be displayed. </Definition>
</Term>
<Term termID="1.7">
  <Name xml:lang="ja">Delete</Name>
  <Name xml:lang="en">delete</Name>
  <Definition xml:lang="ja">Delete is specified when the content, licenses, or keys stored in a
receiver need to be deleted. Metadata cannot be displayed. It is recommended to separately display a
message indicating that such data will be deleted.</Definition>
</ClassificationScheme>

```

7 Use of the ISDBTMMItemCategoryCS Dictionary

This section describes the definition of the ISDBTMMItemCategoryCS dictionary that is referenced in this operational standard.

ISDBTMMItemCategoryCS is described in :Genre/@href and defines the distribution method of applicable content and the attributes of metadata. For details, see “2.2 Description of various control information using Genre [@type=“other”]” in Annex 4.

```
<ClassificationScheme uri="http://www.isdb-t.jp/cs/2011/03/ISDBTMMItemCategoryCS">
```

```
<!-- The category of content and attributes of metadata are defined and described in :Genre/@href.
// -->
<!-- Category of content; one category must be described. -->
  <Term termID="RT">
    <Name xml:lang="ja">Real-time broadcasting</Name>
    <Name xml:lang="en">real-time broadcasting</Name>
    <Definition xml:lang="ja">Real-time broadcasting content</Definition>
  </Term>
  <Term termID="FD">
    <Name xml:lang="ja">Storable broadcasting</Name>
    <Name xml:lang="en">file download type broadcasting</Name>
    <Definition xml:lang="ja">Storable broadcasting content</Definition>
  </Term>
  <Term termID="CH">
    <Name xml:lang="ja">Channel (service)</Name>
    <Name xml:lang="en">channel(service)</Name>
    <Definition xml:lang="ja">Channel service information</Definition>
  </Term>

<!-- Extra flag -->
  <Term termID="gougai">
    <Name xml:lang="ja">Extra content</Name>
    <Name xml:lang="en">extra content</Name>
    <Definition xml:lang="ja">An extra flag is added to the metadata of extra content. The flag
must be removed when the content is not classified as "extra."</Definition>
  </Term>

  <!--trailer flag -->
  <Term termID="trailer">
    <Name xml:lang="ja">Trailer (preview) content</Name>
    <Name xml:lang="en">trailer content</Name>
    <Definition xml:lang="ja">A trailer flag is added to the metadata of trailer content. Trailer
content (the CRID) is specified in the MediaTitle/TitleVideo or RelatedMaterial of the main
content.</Definition>
  </Term>

  <!-- Whether or not a search target can be searched -->
  <Term termID="search.ok">
    <Name xml:lang="ja">search target</Name>
    <Name xml:lang="en">Searchable</Name>
    <Definition xml:lang="ja">This is determined based on the metadata displayed in a search
result. Example: A group is displayed in a search result by adding "search.ok" to the group but by not
adding "search.ok" to the children of the group.</Definition>
  </Term>

  <!-- A flag that is not displayed on a UI -->
  <Term termID="display.ng">
    <Name xml:lang="ja">Non-display</Name>
    <Name xml:lang="en">nondisplay</Name>
```



```

    <Definition xml:lang="ja">This flag is added to content (metadata) that is not to be displayed
in the EPG/ECG. This flag can be used with the search target (search.ok).</Definition>
  </Term>
</ClassificationScheme>

```

8 Use of the ISDBTMMIntendedAudienceCS Dictionary

This section describes the definition of the ISDBTMMIntendedAudienceCS dictionary that is referenced in this operational standard.

ISDBTMMIntendedAudienceCS is described in :Genre/@href and defines the content type of applicable content. The details on the items are T.B.D. (the details are described after entrusted broadcasters have decided them). For the use of ISDBTMMIntendedAudienceCS, see “2.2 Description of various control information using Genre [@type=“other”]” in Annex 4.

```

<ClassificationScheme uri="http://www.isdb-t.jp/cs/2011/03/ISDBTMMIntendedAudienceCS">
<!-- A content (metadata) type is defined and described in :Genre/@href.
// -->
  <Term termID="1">
    <Name xml:lang="ja">Common to ISDB-Tmm</Name>
    <Definition>Content that can be used in any terminal; the content type is determined by
referencing the ContentTypeCS.</Definition>
  </Term>
  <Term termID="2">
    <Name xml:lang="ja">Contractor's original</Name>
    <Definition>Contractor's original content; availability is dependent on a terminal (not
dependent on a receiver business operator). A terminal determines availability by referencing a
content type (ContentTypeCS) or file format (FileFormatCS).</Definition>
  </Term>
  <Term termID="3">
    <Name xml:lang="ja">Receiver business operator's original</Name>
    <Definition>This ID is used for classification and is not actually applied.</Definition>
  </Term>
    <Term termID="3.1">
      <Name xml:lang="ja">Receiver business operator A</Name>
      <Definition>Content available only using the terminal of receiver business operator
A</Definition>
    </Term>
      <Term termID="3.1.1">
        <Name xml:lang="ja">Standard terminal</Name>
        <Definition>Content available only using the standard terminal of receiver business
operator A</Definition>
      </Term>
      <Term termID="3.1.2">
        <Name xml:lang="ja">Multi-function terminal</Name>
        <Definition>Content available only using the multi-function terminal of receiver business
operator A</Definition>
      </Term>
    <Term termID="3.2">

```

```
<Name xml:lang="ja">Receiver business operator B</Name>
<Definition>Content available only using the terminal of receiver business operator
B</Definition>
</Term>
<Term termID="3.2.1">
  <Name xml:lang="ja">Standard terminal</Name>
  <Definition>Content available only using the standard terminal of receiver business
operator B</Definition>
</Term>
...
...
</ClassificationScheme>
```

9 Use of the ISDBTMMStereoVideoFormatCS Dictionary

This section describes the definition of the ISDBTMMStereoVideoFormatCS dictionary that is referenced in this operational standard. ISDBTMMStereoVideoFormatCS is described in :AVAttributes/VideoAttributes/StereoVideoFormat/@href and defines the 3D video format. Among the following items, only “3.1 Side-by-side” is used.

```
<ClassificationScheme uri="http://www.isdb-t.jp/cs/2011/03/ISDBTMMStereoVideoFormatCS">
<!--
  A 3D video format is defined.
// -->
<!-- With lenses -->
  <Term termID="1">
    <Name xml:lang="ja">Anaglyph method</Name>
    <Name xml:lang="en">anaglyph</Name>
  </Term>
  <Term termID="2">
    <Name xml:lang="ja">Polarization method</Name>
    <Name xml:lang="en">polarization</Name>
  </Term>
  <Term termID="3">
    <Name xml:lang="ja">Frame sequential method</Name>
    <Name xml:lang="en">frame sequential</Name>
  </Term>
  <Term termID="3.1">
    <Name xml:lang="ja">Side-by-side method</Name>
    <Name xml:lang="en">side by side</Name>
  </Term>
  <Term termID="3.2">
    <Name xml:lang="ja">Top and bottom</Name>
    <Name xml:lang="en">top and bottom</Name>
  </Term>
<!-- Without lenses -->
  <Term termID="4">
```

```
<Name xml:lang="ja">Parallax barrier method</Name>
<Name xml:lang="en">parallax barrier</Name>
</Term>
<Term termID="5">
  <Name xml:lang="ja">Lenticular lens method</Name>
  <Name xml:lang="en">lenticular lens</Name>
</Term>
</ClassificationScheme>
```

10 Acquisition of a Classification Scheme and Management Guidelines

The ISDBTMM common dictionaries in the classification schemes (hereinafter referred to as “CS dictionary files”) in the previous sections are basically implemented when manufacturing a receiver. However, if a receiver has a communication function, a CS dictionary file stored in a portal server can be obtained.

If an addition or update is made to a CS dictionary file, the addition or update is sent as storable broadcasting content (ISDBTMMContentTypeCS/1.10 describes it to ContentProperties/ContentType/@href; see Section 5 in Annex 1.). A receiver obtains the CS dictionary file from the storable broadcasting content and identifies the content of the CS dictionary file from ClassificationScheme/@uri of the CS dictionary file. At this time, if the obtained CS dictionary file is new, the file is stored in a receiver. If the obtained CS dictionary file already exists, then ClassificationScheme/@FragmentVersion instances of the obtained file and existing file are compared. If the FragmentVersion of the obtained file is new, the applicable CS dictionary is updated.

The expiration date of a CS dictionary file is described in ClassificationScheme/@fragmentExpirationDate. Then, the receiver checks the expiration date of each CS dictionary file and deletes the CS dictionary file for which the expiration date has lapsed.

Whether or not a business operator's original CS dictionary file is implemented when manufacturing a receiver is determined based on the receiver. A business operator's original CS dictionary file can also be obtained by transferring the CS dictionary file in storable broadcasting as described above or via communication to a portal server.

<Blank Page>

Annex 2 Metadata Usage Guidelines

1 Summary of Metadata Related Identifiers

Table S2-1 illustrates the reference information related to the use of various identifiers described in metadata documents. For details on usage, see the applicable part of the operational rules in the table. Uppercase and lowercase letters can be used for the identifiers, including Authority and CRID, described in Table S2-1. However, when comparing them, they are not case-sensitive (see Section 8.1).

Table S2-1: Summary of metadata related identifiers

Identification name	Unit to add an identifier	Description format	Limit on the number of characters	Range of uniqueness	Life cycle	Applicable part of the operational standard
Authority	Business operator unit	DNS name Example: www.isdb-t.jp	Maximum 57 bytes	Unique among the service providers	Static use	8.1.2 in Volume 10
programID (CRID)	Available unit of content unit	crid://<authority>/<data> Example: crid://example.jp/content1	Maximum 255 bytes	Unique among business operators (<authority>). This range is applied to all identifiers in CRID format.	Not reusable	8.1 in Volume 10
groupID (CRID)	Group unit	crid://<authority>/<data>			Not reusable	8.1 in Volume 10
segmentId	Segment (scene) unit	Character string Note 1	Maximum 255 bytes	Unique among business operators (<authority>)	Not reusable	3.7.4 in Volume 10
segmentGroupID	Segment group unit	Character string	Maximum 255 bytes	Unique among business operators (<authority>)	Not reusable	3.7.4 in Volume 10
PurchaseId	Item unit	Character string	Maximum 24 bytes	Unique among service providers	Not reusable	3.4.5 in Volume 10
LicenseID	License unit	16-digit hexadecimal integer	16 bytes	Unique among service providers	Not reusable	3.6 in Volume 10
MainLicenseID	Main license unit	Character string	Maximum 32 bytes	Unique among service providers	Not reusable	3.6 in Volume 10

				Note 2		
serviceId	Service unit	Character string	Maximum 4 bytes	Unique among business operators (<authority>)	Not reusable	3.5.6 in Volume 10
couponId	Coupon unit	Character string	Maximum 24 bytes	Unique among business operators (<authority>)	Not reusable	3.8.1 in Volume 10
fragmentID	Metadata's fragment ID unit	10-digit hexadecimal integer Example: ffff00001	10 bytes	Unique among service providers	Not reusable	4.4.1 in Volume 10
fragmentVersion	Metadata's fragment ID unit	14-digit decimal (YYYYMMDDhhmmss format: Update date and time of metadata)	14 bytes	—	—	4.4.2 in Volume 10
Imi	Multiple broadcast event information elements and multiple on-demand program information elements that correspond to the same CRID are identified.	imi:<data> Character string	Maximum 255 bytes	Unique among CRIDs	—	3.5.5 in Volume 10

Note 1: If multiple resources (multiple moving image files) are included in storable broadcasting content (see 2.4 in Annex 4), it is recommended to use a SegmentId as follows in order to uniquely identify the segment (scene) of each resource file. The scene information described in a manifest file (see “5.3.2 Manifest files” in Volume 3 of this operational standard) is associated with the file name of a video to be referenced and the start/end time and is not associated with an ID.

• Format of segmentId: <scene_id>_<content_id>#<file_name>

Example: 1_cid001#movie1

If multiple resources are included in a piece of storable broadcasting content, multiple SegmentInformation may be returned as a search result of a scene search (see section 6.3). At this time, a segment (scene) can be uniquely identified using the <file_name> part of the SegmentId.

Note 2: A business entity identifier is described in the MainLicenseID of storable broadcasting content (see 3.6.3).

Annex 3 Guidelines for Creating the EPG/ECG Using Metadata

1 Displaying EPG/ECG

This chapter summarizes correspondence between metadata and the information displayed in the EPG/ECG for storable broadcasting content and real-time broadcasting content. For details on the information to be displayed, see “4.5 Content Navigation” in Volume 2 of this operational standard.

1.1 EPG

The following table provides correspondence between metadata and the information displayed in the EPG.

Table S3-1: Content Information

Item no.	Information to be displayed	Acquisition source of the information (XPATH)
1	Program name	ProgramInformation/BasicDescription/Title[@type=“main”]
2	Still picture thumbnail	ProgramInformation/BasicDescription/MediaTitle/TitleImage
3	Broadcasting date and time	BroadcastEvent/PublishedStartTime BroadcastEvent/PublishedEndTime
4	Program description	ProgramInformation/BasicDescription/Synopsis (Note 1)
5	Program genre	ProgramInformation/BasicDescription/Genre (Note 2)
6	Program language	ProgramInformation/BasicDescription/Language
7	Program video, audio mode	ProgramInformation/AVAttributes
8	Playback time of a program	ProgramInformation/BasicDescription/Duration
9	Price (charged/free) * When the contract type is PPV (Note 3)	PurchaseInformation/Price of the PurchaseInformation that corresponds to the ProgramInformation/BasicDescription/PurchaseList/PurchaseIdRef
10	Package that corresponds to a program * When the contract type is PPM (Note 4)	GroupInformation with a GroupInformation/BasicDescription/PurchaseList/PurchaseIdRef that has a PurchaseId specified in LicenseInformation/PurchaseIdRef.
11	Purchase status	See “5 License Related Information” in Appendix 1, Volume 2 of this operational standard.
12	View limit License status (conditions of use)	See “5 License Related Information” in Appendix 1, Volume 2 of this operational standard.
13	Viewing reservation status	Information created by a receiver

14	Video reservation status	Information created by a receiver
15	Storage status	Information created by a receiver

Note 1: It is expected that Synopsis [*@length*="medium"] is used for a content list and that Synopsis [*@length*="long"] is used for content details according to the display area for the content explanation. However, a service provider can decide the actual length.

Note 2: If ARIBGenreCS or ARIBGenreCS2 is used as genre, Genre [*@type*="main, secondary"] is used. For a CS other than the ARIBGenreCS and ARIBGenreCS2 (see 2.2 in Annex 4), Genre [*@type*="other"] is used. However, a service provider can decide the CS to be used.

Note 3: When the contract type is PPM, a price is described in the package (GroupInformation).

Note 4: When the contract type is PPV, ProgramInformation/BasicDescription/PurchaseList/PurchaseIdref of the ProgramInformation, which corresponds to an applicable program, has a PurchaseId that is indicated by the LicenseInformation/PurchaseIdRef.

Table S3-2: Group information (series, package)

Item no.	Information to be displayed	Acquisition source of the information (XPATH)
1	Group name	GroupInformation/BasicDescription/Title[<i>@type</i> ="main"]
2	Still picture thumbnail	GroupInformation/BasicDescription/MediaTitle/TitleImage
3	Explanation of a group	GroupInformation/BasicDescription/Synopsis (Note 1)
4	Genre of a group	GroupInformation/BasicDescription/Genre (Note 2)
5	Program included in a group	ProgramInformation with ProgramInformation/MemberOf/ <i>@crid</i> in which the CRID of an applicable group is described
6	Package price (charged/free)	The PurchaseInformation/Price of the PurchaseInformation that corresponds to GroupInformation/BasicDescription/PurchaseList/PurchaseIdRef
7	Purchase status of a package	See "5 License Related Information" in Appendix 1, Volume 2 of this operational standard.
8	Reservation status	Information created by a receiver
9	Storage status	Information created by a receiver

Note 1: It is expected that Synopsis [*@length*="medium"] is used for a group list and that Synopsis [*@length*="long"] is used for group details according to the display area for the group explanation. However, a service provider can decide the actual length.

Note 2: If the ARIBGenreCS or ARIBGenreCS2 is used as the genre, Genre [*@type*="main, secondary"] is used. For a CS other than the ARIBGenreCS and ARIBGenreCS2 (see 2.2 in Annex 4) is used, Genre [*@type*="other"] is used. However, a service provider can decide the CS to be used.

Note 3: A group type is specified in GroupType. If the group type is a series, "series" is described. If a group type is a package, "package" is described.

1.2 ECG

The following table provides the correspondence between metadata and the information displayed in the ECG.

Table S3-3: Content information

Item no.	Information to be displayed	Acquisition source of the information (XPATH)
1	Content name	ProgramInformation/BasicDescription/Title[@type="main"]
2	Still picture thumbnail	ProgramInformation/BasicDescription/MediaTitle/TitleImage
3	Video preview	ProgramInformation/BasicDescription/MediaTitle/TitleVideo
4	Broadcasting period (Note 1)	OnDemandService/OnDemandProgram/StartOfAvailability OnDemandService/OnDemandProgram/EndOfAvailability
5	Explanation of content	ProgramInformation/BasicDescription/Synopsis (Note 2)
6	Genre of content	ProgramInformation/BasicDescription/Genre (Note 3)
7	Language of content	ProgramInformation/BasicDescription/Language
8	Content video, audio mode	ProgramInformation/AVAttributes
9	Playback time of content	ProgramInformation/BasicDescription/Duration
10	Price (charged/free) * When the contract type is PPV (Note 4)	PurchaseInformation/Price of the PurchaseInformation that corresponds to ProgramInformation/BasicDescription/PurchaseList/PurchaseIdRef
11	Package that corresponds to content * When the contract type is PPM (Note 5)	GroupInformation with the GroupInformation/BasicDescription/PurchaseList/PurchaseIdRef that has a PurchaseId specified in the LicenseInformation/PurchaseIdRef * The CRID of an applicable package is described in the ProgramInformation/MemberOf/@crid.
12	Purchase status	See "5 License Related Information" in Appendix 1, Volume 2 of this operational standard.
13	Whether or not playback can be performed, view limit License status (conditions of use)	See "5 License Related Information" in Appendix 1, Volume 2 of this operational standard.
14	Reservation status	Information created by a receiver
15	Storage status	Information created by a receiver
16	Storage capacity	See "4.4.5 Ensuring storage capacity when scheduling storage" in Appendix 1 in Volume 2 of this operational standard.

Note 1: The broadcasting period of applicable content is described in the metadata and each schedule period when content is actually broadcasted is described in the SessionDescription of the transmission control metadata (for details, see "2.1.5.4 Transmission control metadata" in Volume 11 of this operational standard).

Note 2: It is expected that Synopsis [@length="medium"] is used for a content list and that Synopsis [@length="long"] is used for content details according to the display area for the content explanation. However, a service provider can decide the actual length.

Note 3: If the ARIBGenreCS or ARIBGenreCS2 is used as genre, Genre [@type=“main, secondary”] is used. For a CS other than the ARIBGenreCS and ARIBGenreCS2 (see 2.2 in Annex 4) is used, Genre [@type=“other”] is used. However, a service provider can decide the CS to be used.

Note 4: When the contract type is PPM, the price is described in the package (GroupInformation).

Note 5: When the contract type is PPV, ProgramInformation/BasicDescription/PurchaseList/PurchaseIdref of the ProgramInformation, which corresponds to an applicable program, has a PurchaseId that is indicated by the LicenseInformation/PurchaseIdRef.

Table S3-4: Group information (series, package)

Item no.	Information to be displayed	Acquisition source of the information (XPATH)
1	Group name	GroupInformation/BasicDescription/Title[@type=“main”]
2	Still picture thumbnail	GroupInformation/BasicDescription/MediaTitle/TitleImage
3	Explanation of group	GroupInformation/BasicDescription/Synopsis (Note 1)
4	Genre of group	GroupInformation/BasicDescription/Genre (Note 2)
5	Content included in group	ProgramInformation with ProgramInformation/MemberOf/ @crid in which the CRID of an applicable group is described (Note 3)
6	Package price (charged/free)	PurchaseInformation/Price of the PurchaseInformation that corresponds to the GroupInformation/BasicDescription/PurchaseList/PurchaseIdRef.
7	Purchase status of package	See “5 License Related Information” in Appendix 1, Volume 2 of this operational standard.
8	Reservation status	Information created by a receiver
9	Storage status	Information created by a receiver

Note 1: It is expected that Synopsis [@length=“medium”] is used for a group list and that Synopsis [@length=“long”] is used for group details according to the display area for the group explanation. However, a service provider can decide the actual length.

Note 2: If the ARIBGenreCS or ARIBGenreCS2 is used as genre, Genre [@type=“main, secondary”] is used. For a CS other than the ARIBGenreCS and ARIBGenreCS2 (see 2.2 in Annex 4), Genre [@type=“other”] is used. However, a service provider can decide the CS to be used.

Note 3: For details on the reference relationship between the package and content, see “3.4.5 Use of purchase information elements.”

Note 4: A group type is specified in GroupType. If a group type is a series, “series” is described. If a group type is a package, “package” is described. If a group is created by collecting series (groups), “show” is described. If a group is created by collecting content or groups (series) for editorial purposes, such as ranking and special editions, “otherCollection” is described. In addition, “otherChoice” is described if a group is created by collecting content that has different version information (see “2.3 Description of various control information using Keyword [@type=“other”]” in Annex 4) (for instance, program A group that collects the SD [Ver.1] version and HD [Ver.2] version of program A; see “Section 2.3 Grouping content with different versions” in Annex 4).

2 Example of a Metadata Search

While “Chapter 6 Use of a Metadata Distribution Method via Communication” provides the protocols used for searching metadata, this chapter explains how to use a metadata search using an example of a simple screen transition.

2.1 Example of retrieving metadata in a receiver without a large-capacity cache

Figure S3-1 illustrates a case in which a receiver that does not have a large-capacity cache requests obtaining necessary metadata for a screen transition by taking a new item list display and title search from a top screen as examples.

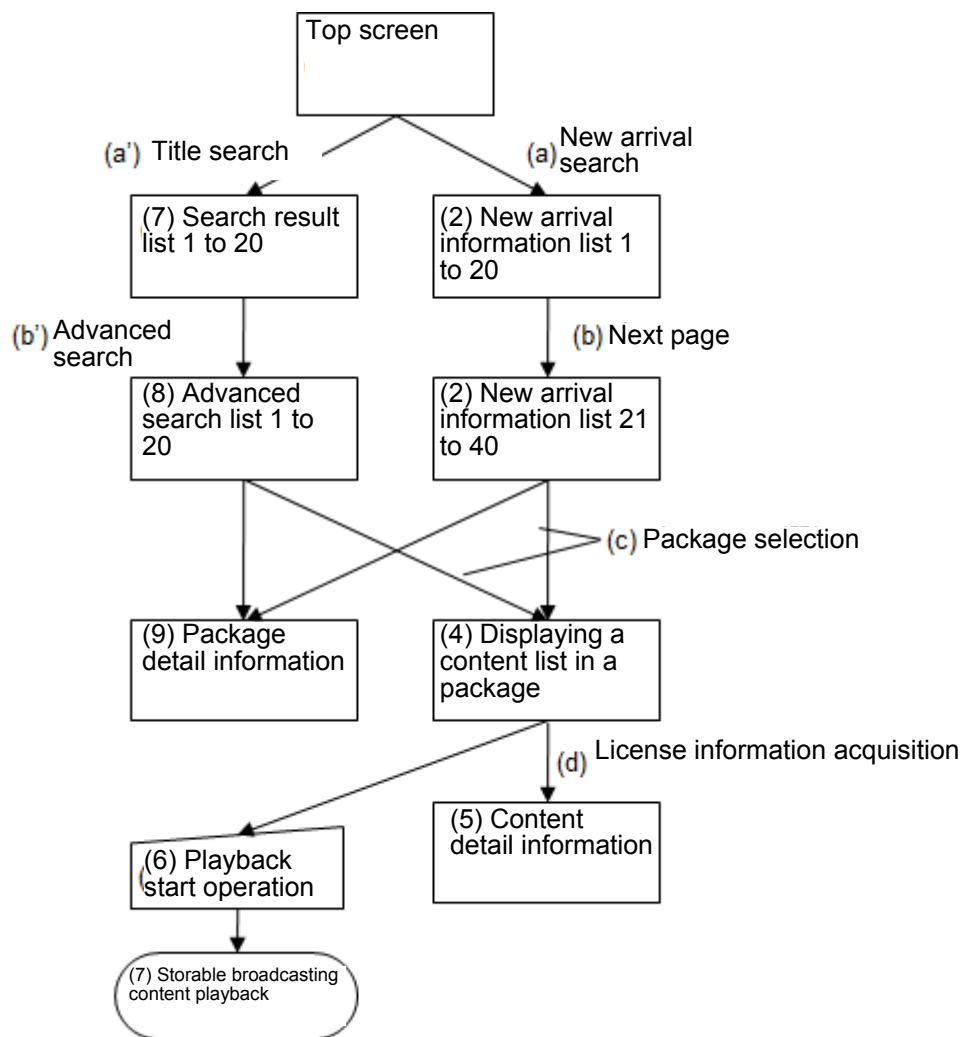


Fig. S3-1: Search example in content navigation

(a) Obtain the top 20 items by searching a package information list that is marked as new, excluding adult content. An example of the search string is as follows.

* In the search string example below, line feeds are added as necessary for easy viewing.

* An actual search string is encoded in x-www-form-urlencoded format.

```

Predicate=
period(new_arrival,2008040100,2008040101),
!parentalrating(http://www.arib.or.jp/cs/2006/03/ARIBParentalRatingCS/R-20),
  
```

```
grouptype(package),  
expirationdate(2008-04-01T00:00:00)&range=1,20&fragment=PGSET
```

(b) Obtain the 20 items that follow the items obtained in (a). An example of the search string is as follows. The range specification is different.

```
predicate=  
period(new_arrival,2008040100,2008040101),  
!parentalrating("http://www.arib.or.jp/cs/2006/03/ARIBParentalRatingCS/R-20"),  
grouptype(package),  
expirationdate(2008-04-01T00:00:00)&range=21,20&fragment=PGSET
```

(a') Obtain the top 20 items by searching packages starting with titles “スターボーズ.” An example of the search string is as follows.

```
predicate=title(スターボーズ*),  
!parentalrating(http://www.arib.or.jp/cs/2006/03/ARIBParentalRatingCS/R-20),  
grouptype(package),  
expirationdate(20080401000000)&range=1,20&fragment=PGSET
```

(b') Obtain the top 20 items by performing an advanced search after adding the condition that a title includes “3.” An example of the search string is as follows.

```
predicate=title(スターボーズ*),title(*3*),  
!parentalrating(http://www.arib.or.jp/cs/2006/03/ARIBParentalRatingCS/R-20),  
grouptype(package),  
expirationdate(20080401000000)&range=1,20&fragment=PGSET
```

(c) Obtain the information of content included in the packages that are selected by performing a MemberOf search. An example of the search string is as follows.

```
predicate=memberof(crid://example.jp/group/1245124_1369457_12834756),  
expirationdate(2008-04-01T00:00:00)&fragment=PGSET
```

(d) Obtain a license reference information element to display the detailed information of the specified content. An example of the search string is as follows. This search string also specifies a PurchaseId that is described in the :PurchaseList/PurchaseIdRef of a package (GIT), in order to uniquely specify license information.

```
predicate=crid(crid://example.jp/video/0654381_5792573_04158457)  
purchaseid_li(pack000001),
```

expirationdate(2008-04-01T00:00:00) &fragment=LIT

Figure S3-2 illustrates a detailed example of a sequence diagram representing a search that is performed to display each screen for “(2) and (3) New information list” and “(4) Displaying a content list in a package” in Figure S3-1. The sequence diagram that shows the route for “(7) and (8) Search result list” is basically the same except for the inquiry statements.

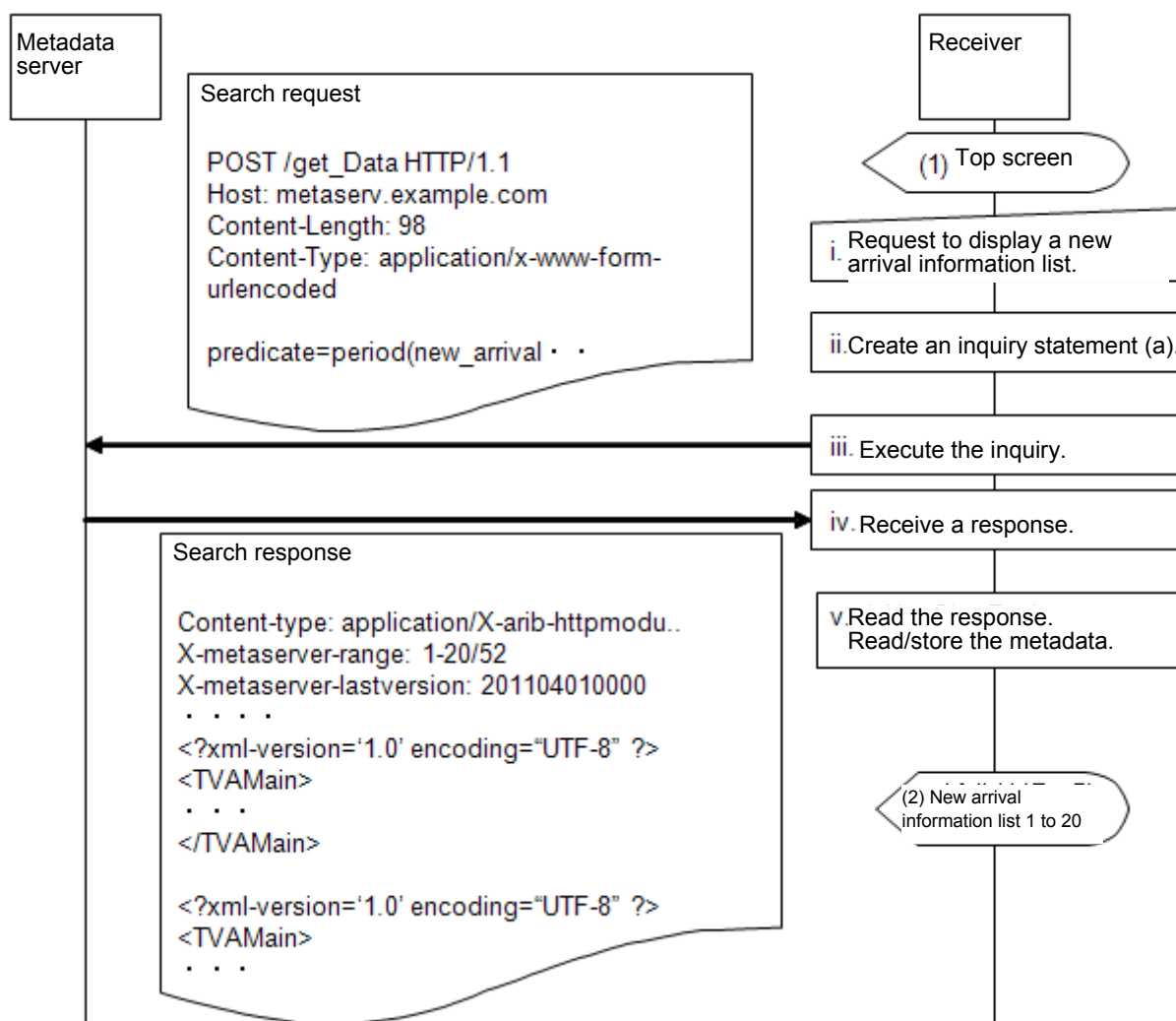


Fig. S3-2: Metadata search sequence for content navigation

- i. A user requests a new content information list.
- ii. An inquiry statement (Fig. S3-1 (a)) is created to request the metadata for the new content information list to a metadata server.
- iii. An inquiry is made to the metadata server.
- iv. A response returned from the metadata server is received.
- v. The received data is parsed to retrieve data.

2.2 Example of retrieving metadata in a receiver with a large-capacity cache

As for a receiver with a large-capacity cache, the sequence provided in 2.1 in Annex 3 is basically performed in a receiver in order to hold all metadata in the receiver. A request is made to a metadata server so as not to obtain the data for an individual screen but to obtain all data. The sequence for this case is explained in the next section. For details on how to handle obtained metadata in a receiver, see “Chapter 7 Metadata Storage Control.”

2.2.1 Acquisition of initial data

If metadata is not stored in a receiver, the receiver requests all currently valid data in a server to a metadata server. Figure S3-3 illustrates the example of the sequence.

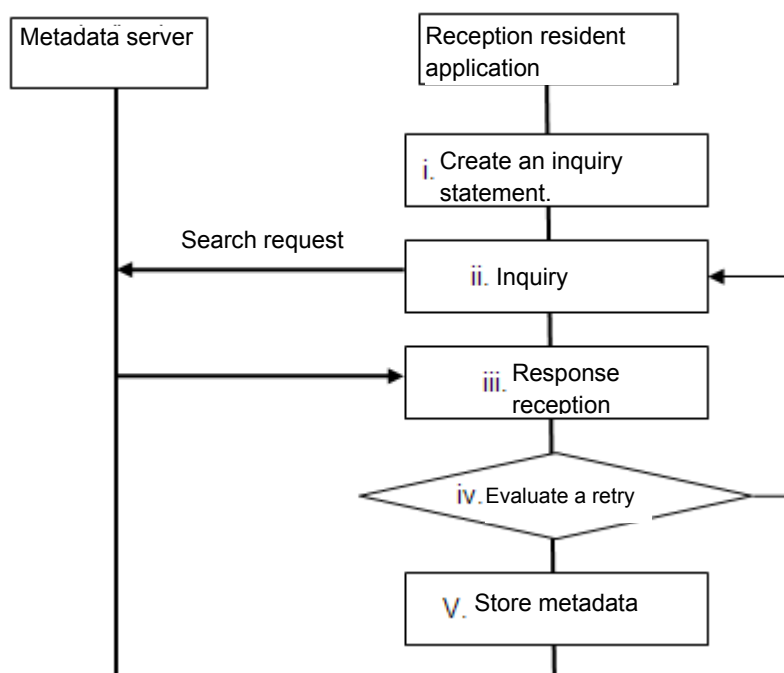


Fig. S3-3: Sequence diagram of obtaining initial metadata

i. The following shows the example of an inquiry statement to obtain all the metadata that is currently valid.

```
predicate=expirationdate(2008-04-01T00:00:00)&fragment=PGSET
```

ii. A search request is performed by sending the inquiry statement created in i to the metadata server.

iii. A response from the metadata server is received.

iv. If data cannot be returned promptly due to overload on a metadata server or other reason, then

a 503 (Service Unavailable) response with a Retry-After header may be returned as a response. In this case, a search request is resent after a specified time has passed. In some cases, the retry wait time may be specified in few-hour units.

- v. The received metadata is stored in a cache area.

2.2.2 Updating differences

After obtaining initial data according to the description in 2.1.1 in Annex 3, a receiver updates the internal data at appropriate intervals. Figure S3-4 shows the sequence.

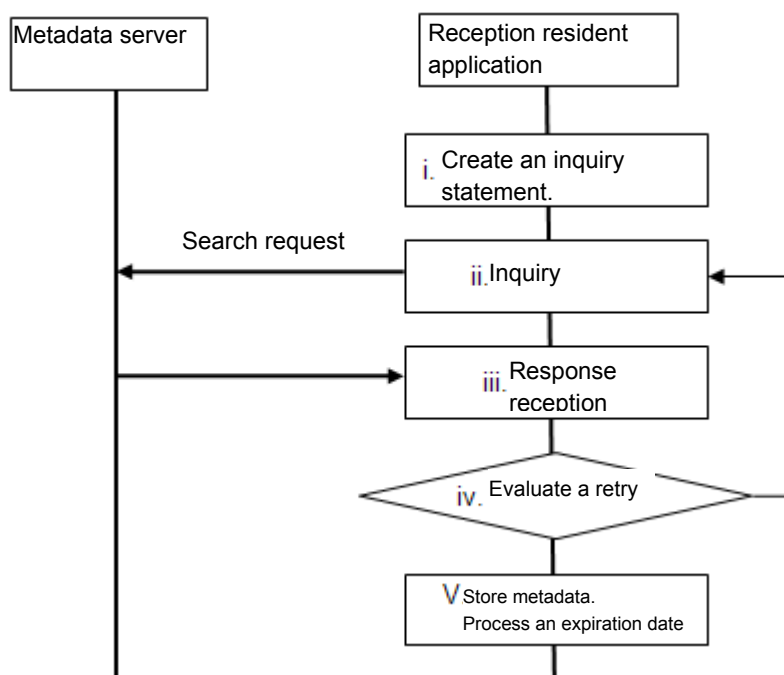


Fig. S3-4: Updating differences when deploying a large-capacity cache of metadata

i. The following shows the example of an inquiry statement for a FragmentVersion search using a maximum value (LatestVersion) of the FragmentVersion of a fragment, which is currently held in a receiver, as a parameter.

```
predicate=fragmentversion(LatestVersion+1)&fragment=PGSET
```

In this case, an expiration date is not specified because a cached fragment may be disabled by being overwritten with a past FragmentExpirationDate. Therefore, a receiver needs to obtain a fragment with a past FragmentExpirationDate. For details, see “Chapter 7 Metadata Storage Control.”

ii. Make a search request on the metadata server using the inquiry statement created in i.

iii, iv. Receive a response from the metadata server. The processing, including retry processing, are the same as iv and v in Fig. S3-4.

v. Update the cache based on the data received from the metadata server. For details, see “Chapter 7 Metadata Storage Control.”

<Blank Page>

Annex 4 Metadata Operation Guidelines in Multimedia Broadcasting

1 Use of Furigana

A business operator can describe the title ruby of applicable content using the Title element for which the @type attribute is specified as “alternative.” *Furigana* is configured with name and episode number and is used as follows.

A. Name is described in “full-width *katakana*.”

(Example: CAB NY→キャブニューヨーク, 23→トゥエンティースリー)

B. The notation method of an episode number is optional. If multiple episodes exist, it is recommended to write in a two-digit one-byte number by entering a one-byte space after each name.

(Example: ミーミン Episode 1→ミーミン 01, Last episode of 機能戦士カンタム NEED→キノウセンシカンタムニード 50. As *furigana*, “カンタム” can be specified by omitting “キノウセンシ”)

A business operator can present a character string in which the @type is not specified as “main” in the series/pack list screen using the Title element in which the @type attribute is specified as “seriesTitle” and “episodeTitle.” For instance, as shown below, if the @type in the Title element of GroupInformation is specified as “main,” the title, “LAST Season 1 (total 23 episodes),” which is displayed in a group detail screen, is described, and if the @type is specified as “seriesTitle,” the entire series title, “LAST Season 1,” which is displayed in the series list screen, is described. If the @type in the Title element of ProgramInformation is specified as “main,” the title, “LAST Season 1, Episode 1,” which is displayed in the content detail screen, is described, and if the @type is specified as “episodeTitle,” a unique title in the entire series, “Episode 1,” which is displayed in the series list screen, is described. The Title element in which the @type attributes are “seriesTitle” and “episodeTitle” is not included in a search target.

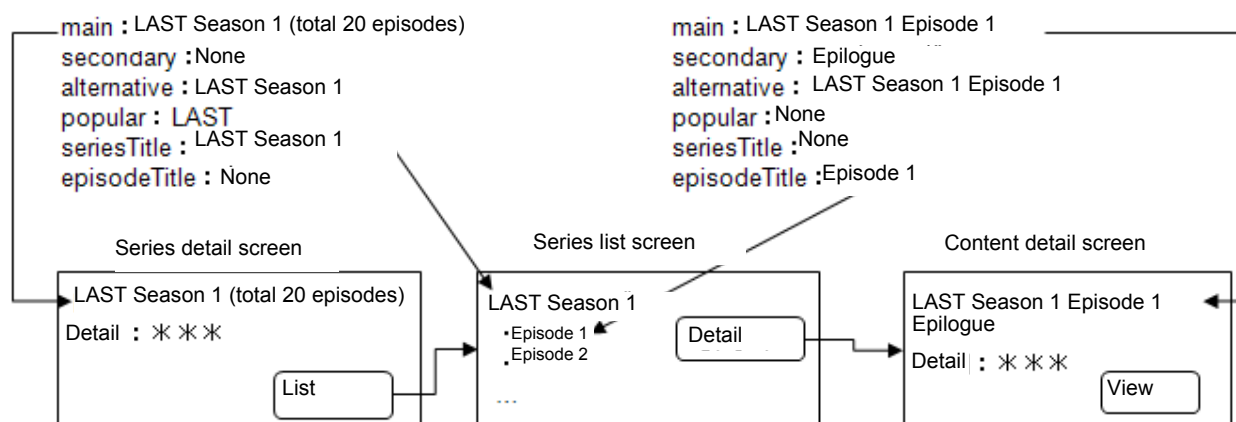


Figure S4-1: Example of screen transition

2 Other Usage Details

2.1 Embedding links/images in an article

If a hyperlink or pictograph is used in a Synopsis element or Title element, the <A> tag or tag is wrapped with a CDATA section and described in each element. The following shows an example.

Example of a hyperlink

```
<Synopsis length="long">article A<![CDATA[<A  
href="link.html">]]>keyword<![CDATA[</A>]]>article B</Synopsis>
```

Example of a pictograph

```
<Title type="main">title A<![CDATA[]]></Title>
```

2.2 Description of various control information using Genre [@type="other"]

A business operator can describe the various control information of content (metadata), especially the search parameters of content (metadata), by using the Genre element in which the @type attribute is specified as "other." Various control information is defined in ISDBTMMIntendedAudienceCS (see 8 in Annex 1), ISDBTMMItemCategoryCS (see 7 in Annex 1), ISDBTMMDeliveryStatusCS (see 6 in Annex 1), and in the business operator's original dictionary.

The following provides a specific example.

- Content type:

In ISDBTMMIntendedAudienceCS, select from "common to ISDB-tmm, contractor's original, receiver business operator A, receiver business operator A's normal terminal, receiver business operator A's multi-function terminal, etc." Content type is a mandatory item for PI. For instance, if ISDBTMMIntendedAudienceCS/1 is applied, it means that applicable content can be used in any terminal. Or, if ISDBTMMIntendedAudienceCS/2 is applied, applicable content consists of the contractor's original content, which means that content can/cannot be used based on the terminal (regardless of the receiver business operator). Whether or not applicable content can be used is determined based on the content type (a ContentTypeCS is specified in the ContentProperties/ContentType/@href) or file format (a FileFormatCS is specified in the ContentProperties/FileProperties/FileFormat/@href). Whether or not such a determination is made, along with the determination method, depend on the implemented receiver.

- Content category:

In ISDBTMMItemCategoryCS, select "RT" (real-time broadcasting), "FD" (storable broadcasting), or "CH" (channel/service). Content category is a mandatory item for PI.

- Extra flag:

ISDBTMMItemCategoryCS/gougai is described in an extra piece of content. This is an optional

item. A service provider can decide how to control the extra flag at the server or terminal. For instance, the schedule of storable broadcasting can be controlled based on whether this attribute is set or not.

- Trailer flag:

ISDBTMMItemCategoryCS/trailer is described in the trailer content. This is an optional item. This item is used to distinguish trailer content from original content. The CRID of trailer content is specified in MediaTitle/TitleVideo or RelatedMaterial/./mpeg7:MediaLocator, which is the original content to be referenced.

- Content distribution status:

Select “scheduled for distribution, distributing, distribution suspended, distribution ended, distribution cancelled, postponed, or delete” in ISDBTMMDeliveryStatusCS. This is a mandatory item. A service provider can decide on the control operation for a terminal based on distribution status. For instance, applicable content can be purchased, viewed, or used only when the status is “distributing.” If the status is “scheduled for distribution, distribution suspended, or distribution ended,” the purchase, view, or use of applicable content is restricted. If the status is “distribution cancelled,” the display in the EPG/ECG is restricted. A message, such as “scheduled for distribution, distribution suspended, and distribution ended,” is described in the EPG/ECG based on the distribution status. In addition, if the metadata indicating “distribution cancelled” or “postponed” has been received, the reservation information of applicable content is deleted. If the status is “postponed,” the metadata (the same CRID is used, and FragmentVersion is updated; the status is updated to “distributing,” etc.), which has a broadcasting date and time that will be changed to a new date and time provided later (after the original broadcasting date and time), is obtained, and then reservation information is updated. If content can no longer be provided due to certain business operator reasons, applicable content stored in a receiver, the corresponding licenses and keys, reservation information, and other data are deleted by providing metadata with the “delete” distribution status. Content distribution status is a mandatory item for PI and GI.

- Content search target flag:

When searching in the EPG/ECG, ISDBTMMItemCategoryCS/search.ok (search target) is described for the content (metadata) that needs to be displayed as a search result.

For instance, if a content is included in a series and if the series is displayed in a keyword search result and if it is necessary to display applicable content on the series detail screen, the following search is performed by adding “search.ok” to the metadata of the series and by not adding “search.ok” to the metadata of the content.

- Keyword search: Keyword (keyword), genre (ISDBTMMItemCategoryCS/search.ok)
⇒ Only metadata (in this case, metadata of a series) with “search.ok” is searched.

* If search is performed using a keyword (keyword), the metadata of a series and the content is searched.

- Searching content included in a series: MemberOf (the CRID of a series GI)
⇒ The metadata of the content included in a series is searched.

- Flag that is not displayed in a UI:

ISDBTMMItemCategoryCS/display.ng is described in a content (metadata) that is not to be displayed on the EPG/ECG.

For instance, if a search is performed using this flag with the “search target flag for content” described above, content is searched but not displayed on a UI.

- Service provider's original genre:

To use a genre (such as “tear-jerker” genre and exciting genre) that is not specified in the ARIBGenreCS and ARIBGenre2, the genre is separately defined in the business operator's original GenreCS.

2.3 Description of various control information using Keyword [@type=“other”]

A business operator can describe various control information of content (metadata), especially the control information that is not used as a content search parameter but that is used only for processing in a terminal or server, by using the Keyword element in which the @type attribute is specified as “other.” As described in Table 3-11, Keyword [@type=“other”] is used to describe supplementary information in metadata and is not presented on a terminal.

【Format】 <control information name>|<control value> * “|” is used as a delimiter between a control information name and a control information value. Multiple <control values> can be described by separating them with a comma (.). When a <control value> is split into multiple elements, they are described in <key>:<value> format. If there are multiple <values>, they can be described by separating them with a slash (/). In addition, if the described control information exceeds the limit of the number of characters set to Keyword, the description is split into multiple parts and described in multiple Keyword elements with the same <control information name>.

The various control information described in Keyword [@type=“other”] is as follows.

- Content control attribute information:

The format of the content control attribute information is “ContAttributes|<control value>,” and the value defined below is used for as a control value.

- manual_complete: OK/NG

This information indicates whether or not the compensation of stored content can be manually performed for applicable content. If this information is not described, it is regarded that the manual compensation of stored content can be performed (OK).

- complete_only: ON

This information indicates that applicable content is not broadcasted and is directly obtained from a stored content compensation server. If this information is described, a receiver obtains applicable content via communication based on the transmission control metadata.

- autoplay: OK/NG

This information indicates whether applicable content can be automatically played back. If this information is not described, it is regarded that the content is not automatically played back (NG). However, the metadata in ISDBTMMContentTypeCS/1.7 is always automatically played back (autoplay: OK) regardless of the specification of this field.

- syncplay: OK/NG

This information indicates whether applicable content is synchronously played back. If this information is not described, it is regarded that the content is not synchronously played back (NG). However, as for the metadata in ISDBTMMContentTypeCS/1.7, the operation, which is performed when syncplay is set to NG, is always executed regardless of the specification of this field.

* Both autoplay and syncplay consist of information regarding playback. They are used based on the combination in Table S4-1.

Table S4-1: Combination of autoplay and syncplay

		Syncplay	
		OK (Note 1)	NG
autoplay	OK	<ul style="list-style-type: none"> Automatically starts playing back at a disclosed date and time Playback cannot be started at an optional date and time. 	<ul style="list-style-type: none"> After download is complete, the processing of the content (such as content update and program update) starts automatically.
	NG	<ul style="list-style-type: none"> Playback is performed in the middle of content when a user starts playing back (same as the real-time broadcasting service). 	Normal storable broadcasting content

Note 1: One synchronous viewing time (the same as distribution date and time) is set to one piece of content. If it is necessary to reuse a synchronous viewing time by changing it, it is used by changing the metadata and license period information.

- force_receive: ON

This information indicates that the reception of applicable content is required. If this information is described, a receiver forcibly reserves the storage of applicable content. However, as for the metadata in ISDBTMMContentTypeCS/1.7, the operation, which is performed when force_receive is set to ON, is always executed regardless of the specification of this field.

- auto_receive: ON

This information indicates that applicable content is automatically reserved. If this information is not described, it is regarded that the content is not automatically reserved. When a receiver performs the auto reservation processing, it evaluates the priority order of the auto reservation and creates a reservation list for the content in which auto_receive:ON is described. However, as for the metadata in ISDBTMMContentTypeCS/1.7, the operation, which is performed when auto_receive is set to ON, is always executed regardless of the specification of this field.

- bookmark: OK/NG

This information indicates whether the bookmark of applicable content can be placed. If this information is not described, it is regarded that a bookmark can be placed (OK).

- comment: OK/NG

This information indicates whether or not comments (including rating and votes) from users can be added to applicable content. If this information is not described, comments can be added (OK). A URI that is used to add comments is described in MediaLocator/mpeg7:MediaUri in which RelatedMaterial/HowRelated/@href is specified as ISDBTMMHowRelatedCS/27.

- encrypted: OFF

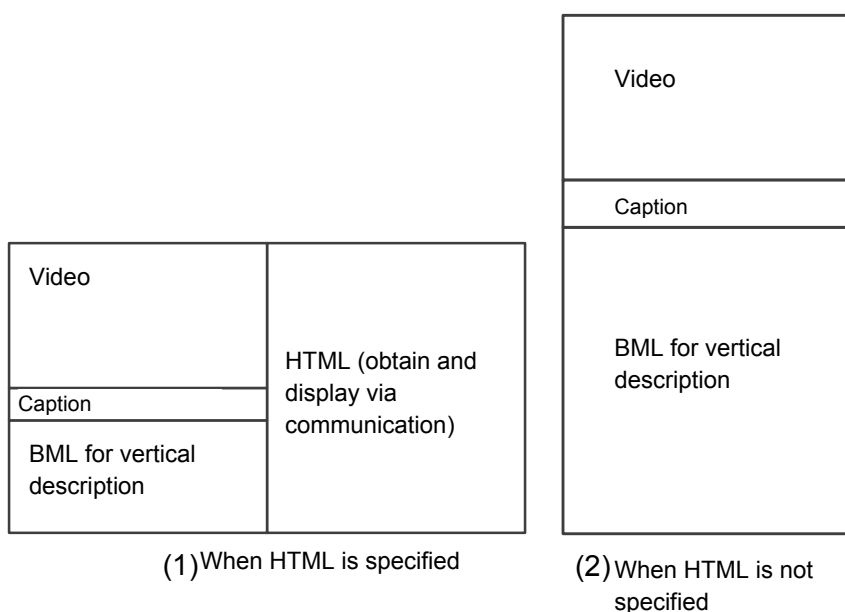
This information indicates that applicable content is not encrypted. If this information is not described, it is regarded that applicable content is encrypted.

- export_outside_cas: ON

This information indicates that applicable content can be exported (see 4.1.2 in Part 2 of ARIB STD-B45). If this information is not described, it is regarded that applicable content is not exported.

- portrait_html: URI/OFF

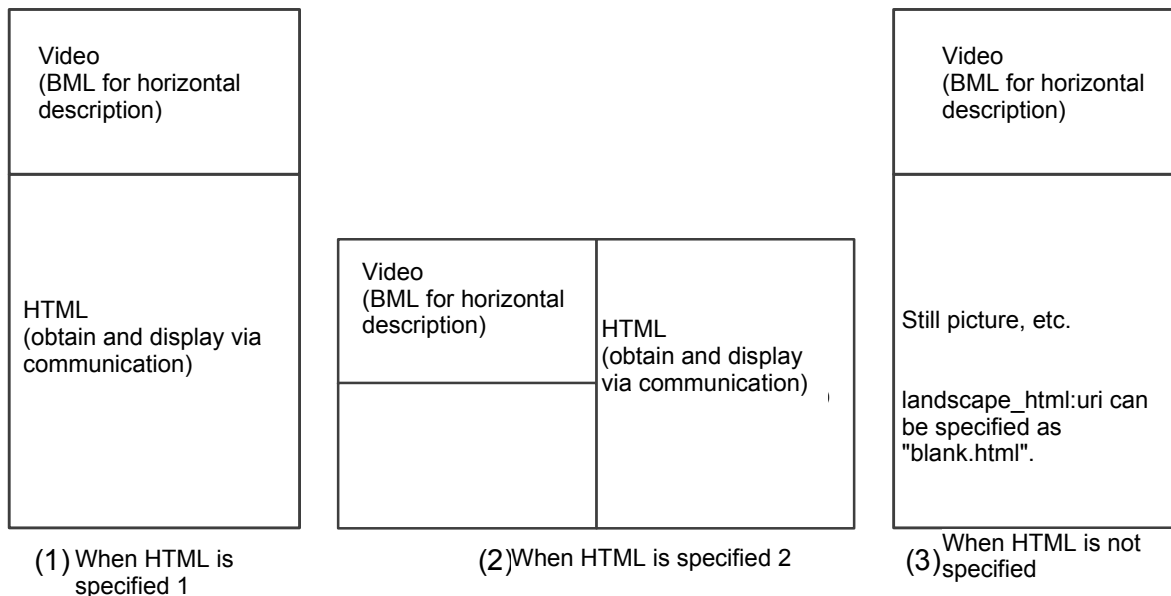
This information indicates whether or not HTML is presented when using a BML that is written vertically. If HTML is presented, the URI of the HTML that is presented after the URI is described. In this case, the format is portrait_html:URI="<URI of HTML>." If HTML is not presented, OFF is described. If this information is omitted, it is regarded that portrait_html is set to OFF.



Implementation example

- landscape_html: URI/OFF

This information indicates whether or not HTML is presented when using a BML that is written horizontally. If HTML is presented, the URI of the HTML that is presented after the URI is described. In this case, the format is landscape_html:URI="<URI of HTML>." If HTML is not used, OFF is described. If this information is omitted, it is regarded that landscape_html is set to OFF.



Implementation example

Example:

```
<Keyword type="other">ContAttributes | manual_comple:NG,complete_only:ON,
syncplay:OK,autoplay:OK,force_receive:ON,auto_receive:ON</Keyword>
```

```
<Keyword type="other">ContAttributes | bookmark:NG,comment:NG,encrypted:OFF,
export_outside_cas:ON</Keyword>
```

```
<Keyword type="other">ContAttributes | portrait_html:OFF,
landscape_html:URI="http://www.example.org"</Keyword>
```

○ Version information of content

Receivers have a variety of functions and specifications (such as with or without a communication function, browser version, and player version). It is expected that new functions and specifications will be added to receivers in the future. The specifications of content (markup language version, the profile and level of video, etc.) also vary widely. In order to deal with this, a version value is assigned based on the specification of content and described in metadata as follows, as an index for assessing whether or not the content can be used by the receiver.

- Base: <version value 1>.<version value 2>.....<version value N>

An integer value that is 0 or greater is described as each version number. A greater version value indicates that the specification of the content has higher accuracy. A version value can be incremented when the content is added. If multiple version values are described and if there is content that does not need a version value specified, then "0" is described to save the order. If all the version values of the succeeding content are also "0," they can be omitted.

If version information is not described, it is regarded that the version of the content is 1 (Base: 1).

Example: Version | Base:3.1

- How to evaluate whether or not content can be used

A receiver compares the content version information (receiver version), which is supported by the receiver and stored separately, and the version information (content version), which is described in metadata. If the receiver version and the content version are the same or if the receiver version is greater than the content version, the content version that corresponds to this condition can be used. If the result of the comparison indicates that any version of common content can be used, then the content that corresponds to the applicable metadata can be used. For instance, if a receiver supports Base:3.1, the content in which the version is Base: 1.0, 1.1, 2.0, 2.1, 3.0, or 3.1 can be used. However, the conditions above are not applied to the metadata in ISDBTMMContentTypeCS/1.7. The metadata is used regardless of the version information.

- Combination with content type

The version information by content type is expressed by combining it with “content type,” described in Section 2.2 in Annex 4. For instance, if a content type is “ISDB-tmm common” and the version information is “Base:1,” the content represented by this metadata is “Ver.1 of the ISDB-tmm common version.” If a content type is “receiver business operator A” and the version information is “Base:1,” the content represented by this metadata is “Ver.1 of the receiver business operator A version.”

For the correspondence between the configuration element of an “ISDB-tmm common version” and the version, see “5.2.7 Content classification in storable broadcasting” in Volume 3 of this operational standard. A content type other than the “ISDB-tmm common version” (such as contractor's original and receiver business operator A. See 8 in Annex 1.) is defined by each business operator and not managed by this operational standard.

- Identifier for banner advertisements:

Taking into account the use of a banner video or image in the EPG/ECG, it is desirable to provide banner advertisement content that supports the specification of a receiver (such as screen size and application version). As multiple banner advertisement content with different sizes exist, it is necessary to have the information to identify which banner advertisement content is used by a receiver. Therefore, the identifier of a banner advertisement is described in metadata as follows. As for the metadata of banner advertisement content, ISDBTMMContentType CS/2.1 is described in :ContentProperties/ContentType/@href, and the destination URI when selecting a banner

advertisement is described in MediaLocator/mpeg7:MediaUri of :RelatedMaterial/HowRelated [href="ARIBHowRelatedCS/10"].

- Version | Banner: <Spec>

In Spec, a character string, which uniquely represents the specification of banner advertisement content, is described as an identifier. For instance, the size of a banner video or image is described as “360 × 240.”

- Evaluation method

A receiver compares the specifications of a banner advertisement, which are held separately and used by the receiver, and the specification information described in the metadata, and uses the corresponding banner advertisement content.

- Identifier of a thumbnail image set:

Taking into account the case of providing thumbnail images (referenced from each metadata) used in the EPG/ECG as one storable piece of broadcasting content, it is desirable to provide a thumbnail image set that supports the specification of a receiver (such as image size and application version). As multiple thumbnail image sets with the same content exist, it is necessary to have information to identify which thumbnail image set is used by a receiver. Therefore, the identifier of a thumbnail image set is described in metadata as follows. As for the metadata of thumbnail image set content, ISDBTMMContent TypeCS/1.9 is described in the :ContentProperties/ContentType/@href.

- Version | Thumbnail: <Spec>

In Spec, a character string that uniquely represents the specification of a thumbnail image set is described as an identifier. For instance, the size of a thumbnail image is described as “360 × 240.”

- Evaluation method

A receiver compares the specification of a thumbnail image, which is held separately and used by the receiver, and the specification information described in the metadata, and then uses the corresponding thumbnail image set.

- Grouping content with different versions

Content with different version information (“Version | Base: <version value>”) has a different content specification but the same content details. Display or reservation in the EPG/ECG can be controlled as follows by grouping the content with different version information. For instance, as shown in Fig. S4-2, “otherChoice” is written as the GroupType of grouped GroupInformation, and the CRID of the GroupInformation is described in the MemberOf of each ProgramInformation with

different version information. If the applicable GroupInformation is selected in the EPG/ECG, content is displayed or reserved by referencing the version information of each ProgramInformation that has children and selecting one ProgramInformation of the latest version that can be used by a receiver.

The same method is applied to group banner advertisement content with different specs (“Version | Banner: <spec>”) and thumbnail image set content with different specs (“Version | Thumbnail: <spec>”).

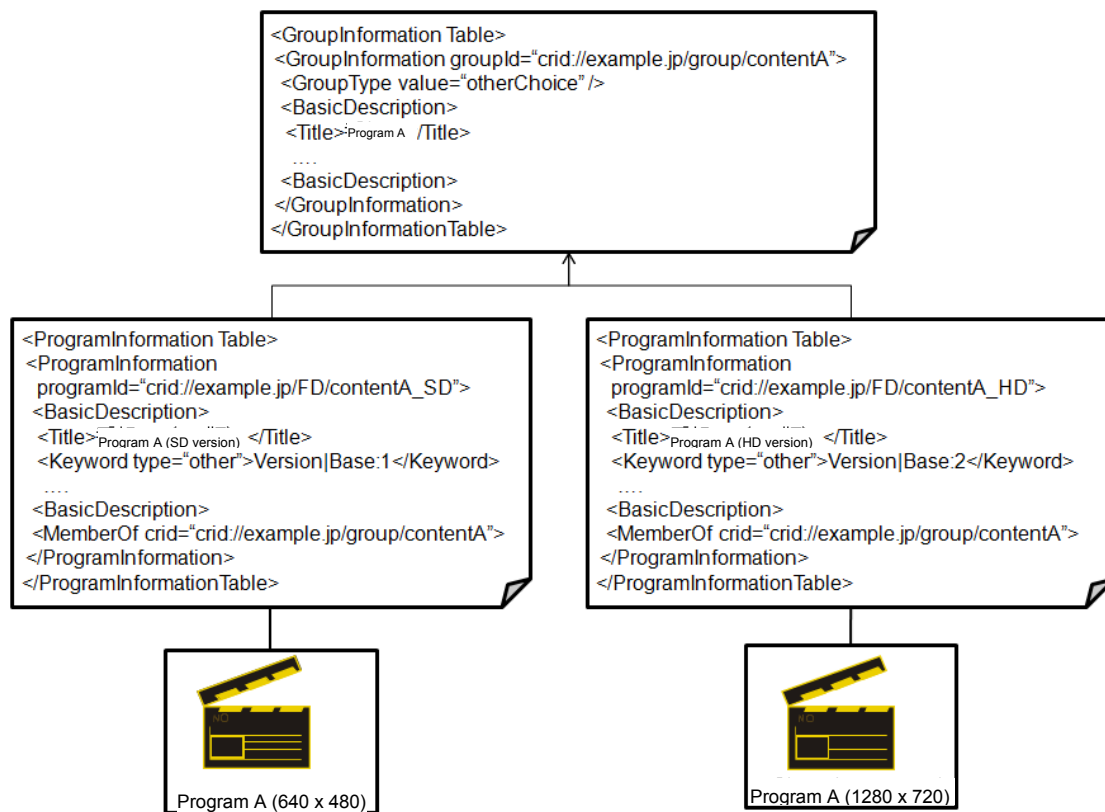


Fig. S4-2: Metadata configuration when grouping content in which versions are different

○ Mapping information from EIT:

The schedule pattern information of a series program that is mapped from the series_descriptor/program_pattern is described (series_pp). This information is described in a program PI. The following shows the conversion table.

The program additional information for digital television broadcasting that is mapped from an item specified in user_nibble is written when the content_nibble_level is 0xE0 in content_descriptor (for details, see Annex B in Volume 4 of ARIB TR-B14). This information is written in a program PI.

When a series is reserved, the series_descriptor/series_id and episode_number are mapped (series_id, episode_number) for searching a program (metadata) to be rebroadcasted. This information is described in a program PI.

Example: EIT|series_pp: 0,content_un: 00,series_id: 0010,episode_number: 2

program_pattern	Schedule pattern	Keyword[@type="other"]
0x0	Irregular (any schedule patterns other than those defined in 0x1 to 0x7)	series_pp:0
0x1	Across-the-board (such as every day, every weekday, and every weekend), scheduled to be broadcasted multiple times in a week	series_pp:1
0x2	Scheduled once a week (every Tuesday, etc.)	series_pp:2
0x3	Scheduled once a month	series_pp:3
0x4	Multiple episodes are scheduled to be broadcasted in the same day.	series_pp:4
0x5	Separating a program that lasts for hours	series_pp:5
0x6	Reserved	series_pp:6
0x7	Reserved	series_pp:7

High-order 4-bit of user_nibble	Low-order 4-bit of user_nibble	Description detail	Keyword[@type="other"]
0x0	*	Fluctuating schedule related additional information	
0x0	0x0	Broadcasting may be cancelled.	content_un:00
0x0	0x1	Broadcasting may be postponed.	content_un:01
0x0	0x2	Broadcasting may be interrupted.	content_un:02
0x0	0x3	Possibility of broadcasting another episode in the same series	content_un:03
0x0	0x4	Programs for which the schedule has not been decided	content_un:04
0x0	0x5	A program may be broadcasted ahead of schedule.	content_un:05
0x0	0x6-0xE	Reserved	—

0x0	0*F	Undefined	—
0x1	*	Digital program characteristic information	
0x1	0x0	Breaking news existing	content_un:10
0x1	0x1	With special service related to the applicable event	content_un:11
0x1	0x2	Reserved	—
0x1	0x3		—
0x1	0x4		—
0x1	0x5		—
0x1	0x6–0xE		—
0x1	0xF		Undefined
0x2	*	3D video broadcasting program identification information	
0x2	0x0	Applicable event with 3D video	content_un:20
0x2	0x1–0xE	Reserved	—
0x2	0xF	Undefined	—
0x3~0xE	0x0–0xE	Reserved	—
0x2~0xE	0xF	Undefined	—
0xF	0x0–0xF	Undefined	—

2.4 Operation when multiple resources (multiple video image files) are included

If multiple resources (video image files, etc.) are included in storable broadcasting content, the resource files are described using a manifest file (see “5.3 Manifest File” in Volume 3 of this operational standard). When describing, the video/audio information and file format of the metadata are used as follows.

- Video/audio information (PI/AVAttributes):

The video/audio information of the main resource file (only one such information is specified in a manifest file) is described.

- Content type (PI/BD/ContentProperties/ContentType):

The type that represents the entire applicable storable broadcasting content (such as video, package, and metadata set) is described.

- File format (PI/BD/ContentProperties/FileProperties/FileFormat):

When the file format of a resource file is the same, the file format is described. If the file format of a resource file is different, the file format of the main resource file is described. However, a file format indicating that multiple resources are included (for details, see 5 in Annex 1) can be described, and a service provider can decide the actual usage.

- File size (PI/BD/ContentProperties/FileProperties/FileSize):

A value obtained by calculating the file size of all resources included in the applicable storable broadcasting content is described.

Annex 5 BiM Encoding Rules

1 Wrapper Schema

For multimedia broadcasting, XML schema for EPG/ECG metadata (see 3.2.1 in ARIB STD-B38. <http://www.arib.or.jp/metadata/tva/2011/03>), XML schema for transmission control metadata (see “2.1.5.4 Transmission control metadata” in Volume 11 of this operational standard; <http://www.arib.or.jp/tmm/fci/2011/03>), and the Wrapper schema to wrap the XML schemas are defined. Figure S5-1 shows the Wrapper schema structure. The XML-based Wrapper schema is also described below.

The BiM decoder can reference the XML schemas of both EPG/ECG metadata and transmission control metadata by referencing the Wrapper schema, which enables decoding both pieces of BiM data.



Figure S5-1: Structure of a Wrapper schema

```

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:tva="http://www.arib.or.jp/metadata/tva/2011/03"
xmlns:tcm="http://www.arib.or.jp/tmm/fci/2011/03"
xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="baseSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <import namespace="http://www.arib.or.jp/metadata/tva/2011/03"
schemaLocation="./STD-B38.xsd"/>
  <import namespace="http://www.arib.or.jp/tmm/fci/2011/03" schemaLocation="./TmmFci.xsd"/>
</schema>
  
```

2 ARIB-TVA Init Message

The method of transferring an ARIB-TVA Init message to a BiM decoder will be defined in a separate usage standard, and the method is not used until then. This section describes the guidelines to use the ARIB-TVA Init message.

A TVA-init message is used to write a parameter to decode a TVA metadata fragment stream (see Section 4.4.1 in ETSI TS 102 822-3-2[15]). In multimedia broadcasting, the ARIB profiles described below are used. In ARIB profile, the value of EncodingVersion is updated based on the TVA MPEG-7 profile defined in Section 4.4.1 in ETSI TS 102 822-3-2[15], which applies some restrictions to the items related to the initialization of a BiM decoder.

Table S5-1 shows the structure of an ARIB-TVA Init message.

Table S5-1: ARIB-TVA Init message

Data structure	Bit number	Value	Meaning of the value
TVA-init {			
EncodingVersion	8	'0xF1'	ETSI TS 102 323 profile (BiM)
IndexingFlag	1	0	Indexing is not used in metadata stream.
reserved	7	1111111	
DecoderInitptr	8	8	Offset from the beginning of the TVA-init message of the decoderInit data
if(EncodingVersion == '0xF1') {			
BufferSizeFlag	1	1	This indicates that a buffer size for Zlib codec is defined.
PositionCodeFlag	1	0	This indicates that a Position Code is not used.
reserved	6	111111	
CharacterEncoding	8	'0x01'	Character coding in UTF-8 (Note 1)
if (BufferSizeFlag=='1') {			
BufferSize	24	0x0186A0	The buffer size of a Zlib codec is set to 100,000 bytes.
}			
}			
reserved	0 or 8+		
DecoderInit()			Decoder initialization message
}			

Note 1: It is recommended that a receiver at least supports UTF-8 character coding in order to maintain compatibility.

The meaning of each item of the TVA-init message defined in ETSI TS 102 822-3-2[15] is applied to the meaning of each item of the ARIB-TVA Init message, excluding EncodingVersion and PositionCodeFlag described below.

EncodingVersion: Specifies an encoding method to indicate a TVAmetadatafragment

Table S5-2: EncodingVersion value

Value	Description
0x00	Reservation
0x01	TVA MPEG_7 profile (BiM) ISO/IEC 15938-1 [16]
0x02 - 0xEF	TVA reserved
0xF1	ETSI TS 102 323 profile
0xF2	ETSI TS 102 471 profile
0xF3 - 0xFF	User definition

PositionCodeFlag: A flag indicating whether the BiM contextPath Position Code is used in an encoded fragment. This field must be set to “0.” When 0 is set, the relative order of fragments is not maintained because the Position Code in a contextPath is ignored.

3 BiM DecoderInit Message

The BiM DecoderInit message is used to write a parameter to decode a TVA metadata fragment that is translated into binary format and to initialize a BiM decoder. The value of NumberOfSchemas is 1 or greater because at least one schema URN must be described in this message. Table S5-3 shows the structure of the BiM Decoder Init message in multimedia broadcasting.

Table S5-3: BiM DecoderInit message

Data structure	Bit number	Value	Meaning of the value
DecoderInit {			
SystemsProfileLevelIndication	16	'0x80'	Arbitrarily value (encode in uimsbf)
UnitSizeCode	3	000	Default value of unit size
ReservedBits	5	11111	
NumberOfSchemas	8	'0x01'	This value indicates that only one schema is used.
{			
SchemaURI_Length[0]	8	'0x0a	The character string length of a schema URI is set to 10 characters.
SchemaURI [0]		"baseSchema"	URI of a Wrapper schema (defined in 1.1 in Annex 5)
LocationHint_Length[0]	8	'0x00'	This value indicates that a location hint is not provided.
NumberOfTypeCodecs[0]	8	'0x00'	This value indicates that only a default codec is used.
}			
InitialDescription_Length	8	'0x00'	This value indicates that there is no explanation on the initial root.
}			

VOLUME 11

Multimedia Broadcasting Operation of Storable Broadcasting

Contents

Chapter 1 General Terms	1
1.1 Introduction	1
1.2 Operational Guidelines Provided in this Volume	1
1.3 References	1
1.4 Terminology	2
Chapter 2 Operation of Storable Broadcasting	7
2.1 Operation of Data Transmission System	7
2.1.1 Outline.....	8
2.1.2 PSI/SI	11
2.1.3 Transmission of content	14
2.1.4 File block division	15
2.1.5 Operation of FLUTE transmission	15
2.1.6 Operation of AL-FEC transmission	43
2.1.7 Operation of UDP/IP transmission.....	51
2.1.8 Operation of ROHC transmission.....	55
2.1.9 Operation of ULE transmission	61
2.1.10 Operation of communication network transmission.....	61
2.1.11 Effective content transmission method	61
2.1.12 Action for content acquisition	67
2.2 EPG/ECG Metadata Transmission Method	68
2.2.1 Definition of terms related to EPG / ECG metadata transmission and main point of metadata transmission operation	68
2.2.2 Program transmission A.....	71
2.2.3 Program transmission B.....	77
2.2.4 Identification of metadata	81
2.3 Operation of Media Coding	81
2.3.1 Video coding	81
2.3.2 Static image and bitmap graphics coding.....	81
2.3.3 Audio coding.....	81
2.3.4 Other coding.....	81
2.4 Multimedia Coding	82
2.4.1 File format.....	82
2.4.2 Manifest file	82

2.4.3 Media type.....	82
2.5 Operation of Character Encoding.....	82
2.5.1 Operation of HTML version	82
2.5.2 Operation of character code	82
2.5.3 Operation range of media type and monomedia	82
2.5.4 Operation of HTML element	82
2.5.5 Operation of CSS	82
2.5.6 Operation range of DOM	82
2.5.7 Operation range of built-in object.....	82
2.6 Total Compensation of Stored Content	82
2.6.1 Overview	82
2.6.2 Implementation based on FDT instance	83
2.6.3 Implementation based on HTTP directory resource information	85

Chapter 1 General Terms

1.1 Introduction

Multimedia broadcasting transmits the following data:

- Video and audio data: Simple video (H.264 | MPEG-4 AVC), MPEG-2 AAC
- Multimedia data: Data coded in conformance with ARIB STD-B24 “Data encoding and transmission specification for digital broadcasting” Volume 2, Annex 4.
- Subtitles and superimposed text
- Storable broadcasting data: See this volume.

The functions of multimedia receivers are largely categorized into:

- Real-time receiving of broadcast

A function to receive a real-time broadcast

The receiver should display all the broadcast contents, which is a set of video, audio, and data.

- Storable broadcasting receiving

A function to receive, store, display, and play back storable broadcasting contents

In this volume, a standard receiver is considered to be equipped with the storable broadcasting receiving function, and details of the function, which should be implemented in a storage-based receiver, are provided.

1.2 Operational Guidelines Provided in this Volume

- The basic functions that should be implemented in a storable broadcasting receiver are defined.
- The operational guidelines provided in this volume should be observed as a minimum when the basic functions as well as optional functions are implemented in the receiver.

In this volume, any description with the term “should” indicates that the described implementation (specification) is not mandatory, but is desired by the broadcasting companies.

1.3 References

This volume specifies the operation of data broadcast for multimedia broadcast, based on the schemes provided in the following standards:

- (1) “Receiver for Digital Broadcasting” ARIB STD-B21
- (2) “Service Information for the Digital Broadcasting System” ARIB STD-B10
- (3) “Transmission System for Digital Terrestrial Television Broadcasting” ARIB STD-B31
- (4) “Data Coding and Transmission Specification for Digital Broadcasting” ARIB STD-B24
- (5) “Video Coding, Audio Coding, and Multiplexing Specifications for Digital Broadcasting” ARIB STD-B32
- (6) “Conditional Access System Specifications for Digital Broadcasting” ARIB STD-B25

1.4 Terminology

16:9	Horizontal-to-vertical ratio of the display screen: 16 horizontal by 9 vertical
4:3	Horizontal-to-vertical ratio of the display screen: 4 horizontal by 3 vertical
ARIB	Association of Radio Industries and Business: The ARIB is an organization that standardizes technologies in relation to the use of radio in Japan, with participation by broadcasters, telecommunications operators, and equipment manufacturers.
ES	Elementary Stream; corresponds to encoded video, sound, and independent data in a PES packet; one ES is transmitted by a PES packet with an identical stream ID.
EUC-JP	Japanese character code encoded in accordance with ISO 2022
HTTP	Hyper Text Transfer Protocol: An application-layer protocol and a protocol (RFC2616) that is used for forwarding data through the World Wide Web
IP	Internet protocol [RFC791]: Used to define the addressing mechanism of the Internet and network-layer protocol as well as to process data delivery.
ISO	International Organization for Standardization
MPEG-2	Moving Pictures Expert Group-2: MPEG-2 is defined as the compression and coding technology for data (such as moving images and audio data) specified by the International Organization for Standardization (ISO/IEC 13818).
NPT	Normal Play Time: Absolute coordinates on a time axis that show the positional relationship of events in a stream
PES	Packetized Elementary Stream: PES is defined as the packetized video, audio and other data with variable lengths.
PID	Packet ID (Identifier): The 13-bit stream identification information of the TS packet header This indicates the attribute of each stream in the applicable packet.
PNG	Portable Network Graphics: Graphics file format succeeding GIF It is pronounced "PING" and is capable of lossless compression. The file format comprises an 8-byte signature followed by a series of "chunks".
RFC	Request for Comments: Technological information made public to the Internet community by the Network Working Group
SBR	Spectral Band Replication: A technology for the ACC low-bit-rate band
TCP	Transmission Control Protocol: End-to-end protocol for the transport layer Offers highly reliable connection transmission that includes error detection and correction. (RFC793)
TS	Transport Stream; stipulated in the MPEG-2 Systems Standard (ISO/IEC 13818-1); 13-segment broadcasting is used for multimedia broadcasting, with 1 TS for each 1-segment of broadcasting.

Storable broadcasting	One of the terrestrial multimedia broadcasting services based on connected segment transmission, which is provided when downloading
Stream format identification	Stream format identification specified in ISO/IEC 13818-1
Transport stream	See TS.
Subtitle/Caption	A service related to video content among the service that is provided by superimposing textual information over video content distributed in real-time broadcasting.
Product design	Functions and operations that depend on the receiver or item
Superimposition	Subtitling service that is not synchronized with the main video, audio, or data, e.g., news flashes, schedule remarks, time signals, earthquake early warnings, etc.
HTML browser	Browser implemented to view HTML contents
Broadcasting content	Broadcasting materials, such as video, audio, and data, received through a broadcasting signal
Mobile phone	“Specific mobile terminal device” specified by ministerial ordinance (Telecommunications Business Law enforcement regulations) (Note: From the viewpoint of protecting personal information, it is assumed that each device features TLS/SSL and route certificate functions.)
Simple video	Video content with a resolution and/or frame rate lower than those for the TV service.
Virtual plane	Plane on which data broadcast contents are arranged It is defined as a rule, but it is not mandatory to implement it in the receiver.
H.264 MPEG-4 AVC	Advanced encoding/decrypting technology co-developed by the Moving Pictures Expert Group (MPEG) of the International Organization for Standardization/ International Electro-technical Commission U.S., Inc. (ISO/IEC) and the Video Encoding Expert Group (VCEG) of the International Telecommunication Union (ITU)
NIT	Network Information Table: NIT is defined as the table that carries information to relate transmission path information such as frequencies to channels and that lists ID numbers for all the service channels contained in a distribution system.
Monomedia	Single expression media such as video, audio, text, and still image
Multimedia	Expression media consisting of multiple monomedia materials It is associated with the interaction between the monomedia materials, and often associated with a dialog with the user.
Resource	When data is used as a raw material, it is called a resource, which often refers to monomedia as a multimedia component.
Selector (CSS)	Format to select an element to which the CSS property is applied
Default style sheet	A list of CSS properties implicitly determined in the receiver It has the least priority and is overridden by an external style sheet or style metadata.
Compensation of stored content	Function to compensate for missing parts if incomplete content is received via broadcasting waves in storable broadcasting

MIME	Multipurpose Internet Mail Extensions: A protocol on the application layer, which provides a content architecture to enable an Internet mail to include a non-US ASCII format text, audio, image, and other multimedia data.
FLUTE	File Delivery over Unidirectional Transport; file delivery protocol (RFC3926) for one-way communication
UDP	User Datagram Protocol: A communications protocol used for Internet data transmission With this protocol, after the data has been transmitted, data reception at the recipient is not checked.
TSI	Transport Session Identifier: Session identifier of FLUTE
TOI	Transport Object Identifier: Object identifier The TOI is set to 0 when an FDT instance is transmitted.
INT	IP/MAC Notification Table: Specifies the transport stream ID/service ID/component tag/target receiver's IP address that corresponds to an applicable platform ID in a stream that configures the storable broadcasting service.
AL-FEC	Application Layer Forward Error Correction: FEC scheme on the application layer
ROHC	RObust Header Compression: Compression scheme for the headers on the IP and upper layers, specified in RFC3095 and RFC4815
SBN	Source Block Number: When a transmitted object is divided into source blocks based on the transmission distance, source block length, and encode symbol length, an SBN is assigned to each source block.
FDT instance	File Delivery Table instance: Object for transmitting a file delivery table (FDT) in which various attributes of a file distributed using the FLUTE defined in IETF RFC 3926 are described
Transmission control metadata	Data in XML format required for receiving and downloading storable broadcasting content.
User Service Description	One of the three descriptions specified by the transmission control metadata, which includes Session Description and Associated Delivery Procedure Description
Session Description	One of the three descriptions specified by the transmission control metadata
Associated Delivery Procedure Description	One of the three descriptions specified by the transmission control metadata
SDP	Session Description Protocol: The data in SDP format that describes the information specific to the downloading transmission protocol used for the user services
ECG	Electronic Contents Guide: Description-language-type metadata specified in ARIB STD-B38 It is used for the contents guide and contents navigation.
URI	Uniform Resource Identifier; description system used to indicate the resource location; this concept includes the URL.
CRID	Content Reference Identifier: Location-independent content identifier
IANA	Internet Assigned Numbers Authority: One of the IETF member organizations responsible for the standardization of the Internet
LDPC	Low-Density Parity-Check Code: An error correcting code, which uses parity check data to repair a missing part of the received content

UEP	Unequal Error Protection: Effective error correction by unequal error protection
ULE	Unidirectional Lightweight Encapsulation: An IP packet encapsulation method for allowing IP transmission on MPEG-2 systems
CID	Context Identifier: An identifier to distinguish a header compressed IP packet flow, which is a set of IPv4 or IPv6 packets that have five identical location properties: next header, source address, destination address, source port, and destination port
HTML	HyperText Markup Language: A markup language to describe web pages
XHTML	eXtensible HTML: HTML standard extending HTML4.0 to be compatible with the XML grammar
XML	Extensible Markup Language: One of the description languages established by W3C
CSS	Cascading Style Sheets: A markup language used for specifying how HTML and XML elements are rendered (displayed) by being either directly described in an HTML document or read from another document
JavaScript	An object-oriented script language used for dynamically modifying HTML documents by being either directly described in an HTML document or read from another document
Manifest file	A data, which is one of the configuration files for the storable broadcasting contents, describing the information in an XML-format document on the configuration file control and scenario control
Entrusted broadcaster	Same as “certified infrastructural broadcaster”
Entrusting broadcaster	Same as “infrastructural broadcasting provider”

<Blank Page>

Chapter 2 Operation of Storable Broadcasting

Storable broadcasting is a system for broadcasting a content that is transmitted and viewed/used at different times, where the content is delivered to receivers through a broadcast wave, stored in each receiver, and then viewed and used. It allows users to view and use the stored contents at any time and location.

If authentication is required for viewing and using the content, the authentication is assumed to be processed through communications. In addition, if the stored contents are defective due to poor reception of broadcast waves, etc., it is possible to complement the missing data through communications. (This feature is called “repair of stored content”.)

The concept of storable broadcasting is illustrated in Fig. 2-1.

This chapter provides the operational guidelines for the data transmission system and media encoding for storable broadcasting.

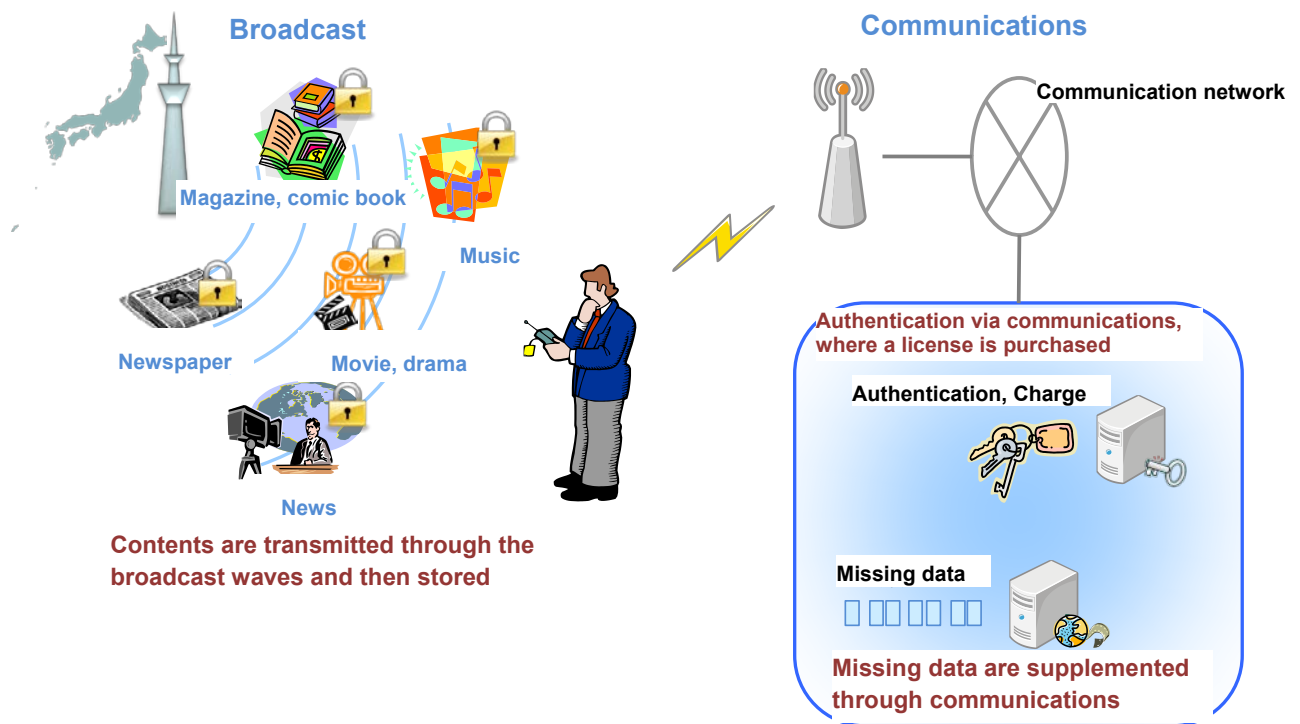


Fig. 2-1: Illustration of the storable broadcasting concept

2.1 Operation of the Data Transmission System

Storable broadcasting can transmit any type of encoded file. The file type can be identified from the MIME. The receiver terminal ignores any file that has an un-interpretable MIME type.

2.1.1 Outline

The storable broadcasting system allows multiplexed transmission of IP packets on the MPEG-2 Systems (ITU-T H.222.0 | ISO/IEC 13818-1).

The contents of storable broadcasting are transmitted using a protocol called FLUTE. In this scheme, the content is divided into segments of appropriate sizes, and after UDP/IP packetization and header compression, they are encapsulated into the MPEG-2 TS. The protocol stack for storable broadcasting is shown in Fig. 2-2.

The receiver uses PSI/SI to obtain the information required for channel selection and gets ready to receive the broadcasted content. Various types of contents are included such as movies, music, electronic books, and games, where each type is encoded using its own coding scheme (see Sections 2.3 and 2.4 in this volume).

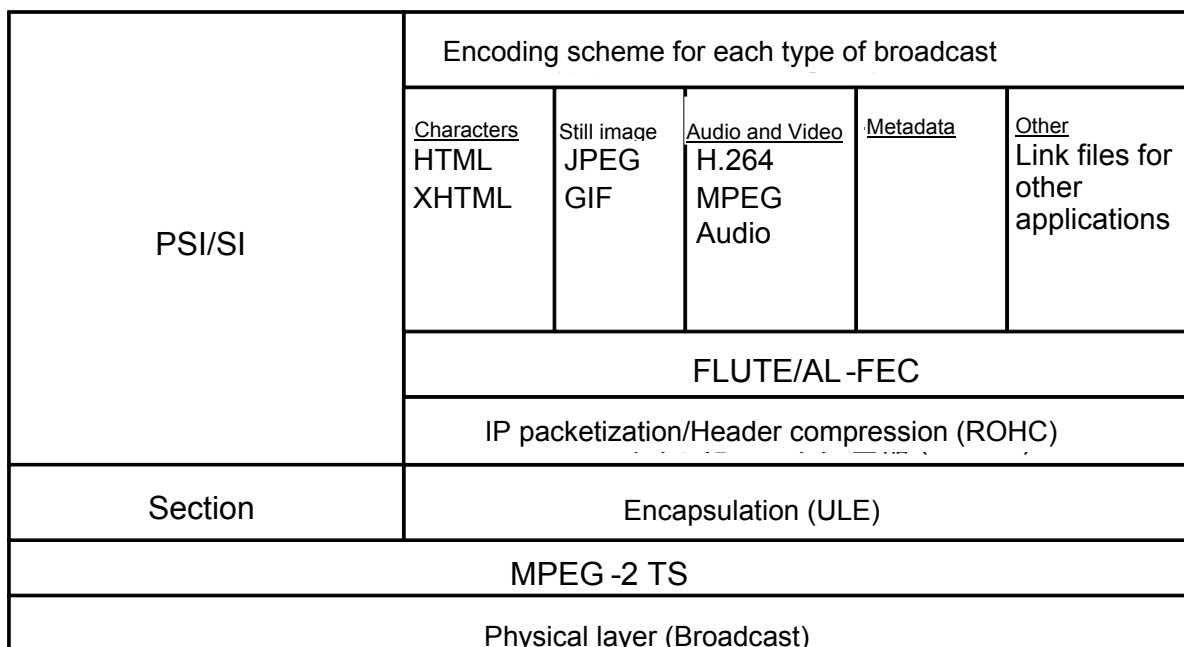


Fig. 2-2: Protocol stack for storable broadcasting

2.1.1.1 Transmission of content

The broadcast content transmitted over the storable broadcasting system is composed of multiple files (objects), and each unit of the broadcast content is transmitted with one FLUTE session identifier (TSI). One session of FLUTE (TSI) is transmitted in the MPEG2-TS, and one PID is assigned to the stream. This PID is associated with one service_id, which is allocated to each entrusted broadcaster. In addition, a TOI is assigned to each of the objects that constitute the broadcast content. A diagram of the concept of transmission of broadcast contents is depicted in Fig. 2-3.

When multiple units of broadcast contents are transmitted with a specific service_id, these contents are transmitted using different FLUTE sessions with the same PID.

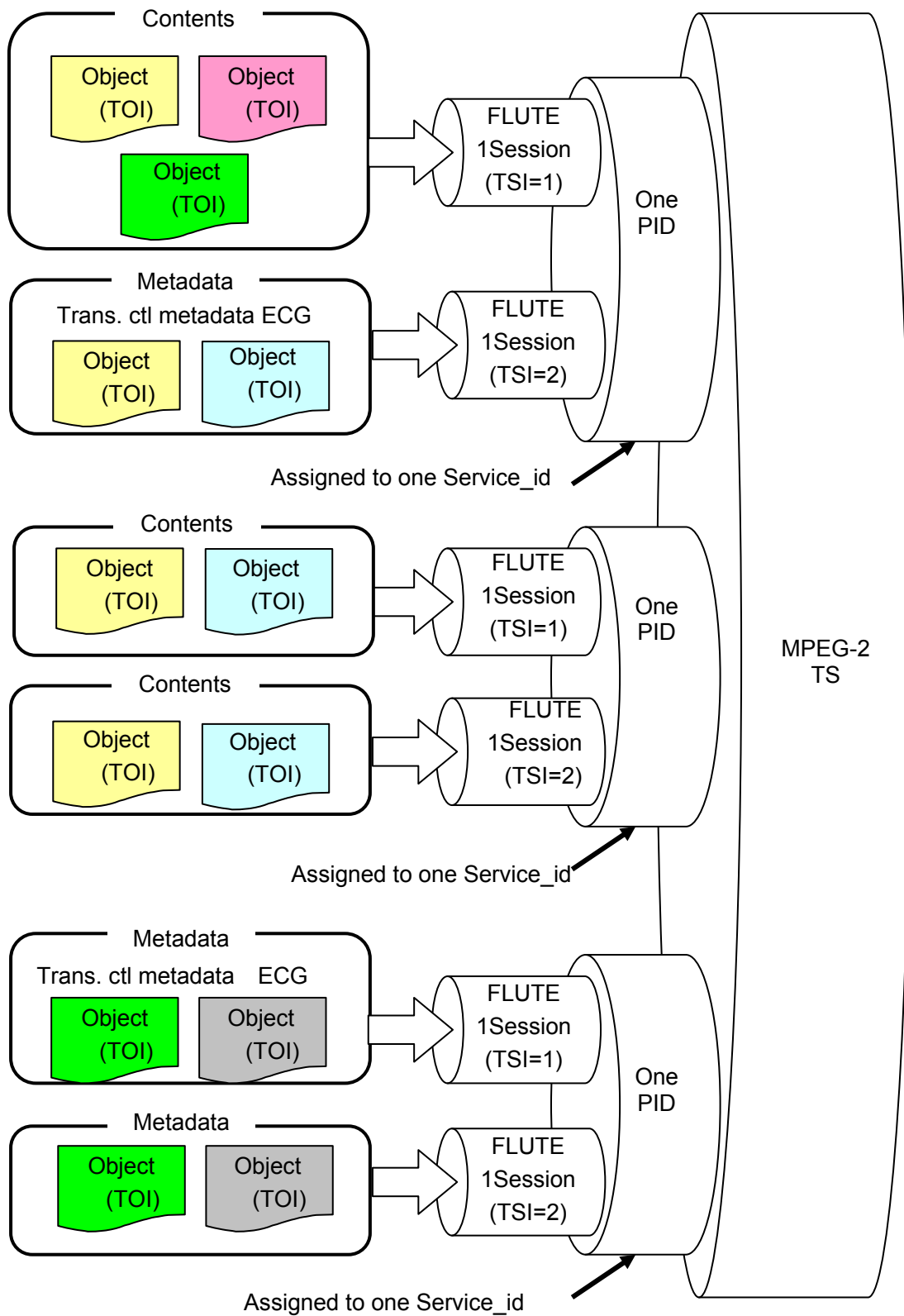


Fig. 2-3: Illustration of transmission of broadcast contents

【Operational policy for transmission of storable broadcasting contents (T.B.D.)】

For each entrusted broadcaster, one or more service_id (the value of the service_id is fixed) is allocated. One PID of MPEG-2 TS is allocated for each of these service_ids, and multiple FLUTE sessions may be constituted within this PID. Through each FLUTE session, one unit of broadcast content is transmitted.

For metadata transmission purposes, another service_id, which is different from the service_id for the entrusted broadcaster, is allocated, and one PID is assigned to this additional service_id. Using this PID, the transmission control metadata and ECG metadata are transmitted through multiple FLUTE sessions. For details of the metadata transmission method, refer to Section 2.2 “EPG/ECG Metadata” in this volume.

An example of assignment of a service_id to an entrusted broadcaster and metadata is depicted in Fig. 2-4 below.

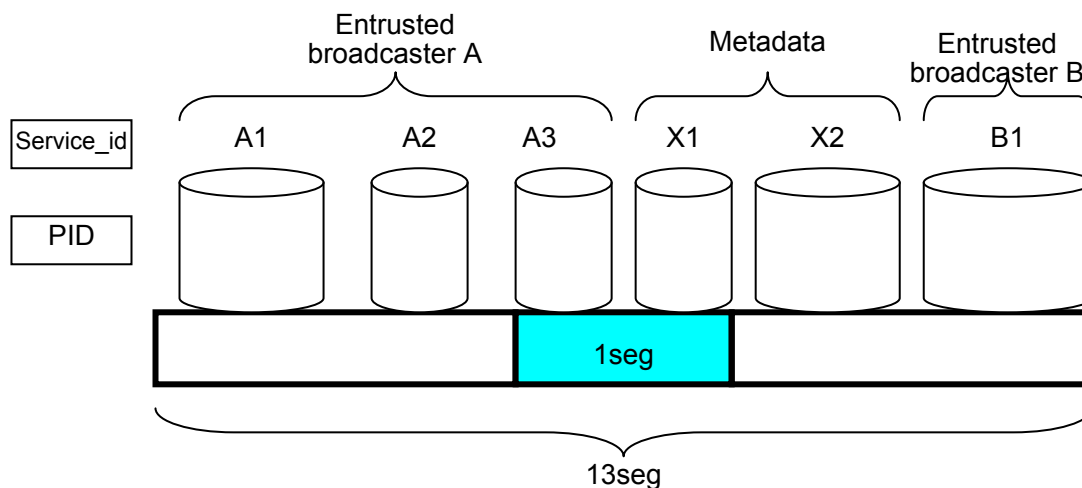


Fig. 2-4: Assignment example of service_ids to entrusted broadcasters and metadata

In addition to the transmission of ordinary storable broadcasting contents, the contents of extra news may also be transmitted on a partial reception layer. In the case of entrusted broadcaster A, such transmission is available as a service with service_id = A3. If entrusted broadcaster B wants to transmit extra news contents, a service with service_id = B2 needs to be prepared on the partial reception layer. However, since ordinary contents cannot be transmitted with service_id = B2, it is not used on a regular basis.

2.1.1.2 Contents delivery schedule

An example for the delivery schedule of storable broadcasting contents is shown below.

When multiple units of broadcast contents are delivered with one service_id (equivalent to the channel), they are delivered in a time-sequential manner. An individual TSI is assigned to each unit of broadcast content so that the receiver can receive the contents on a service_id and time basis. Figure 2-5 shows an example of the delivery schedule of broadcast contents.

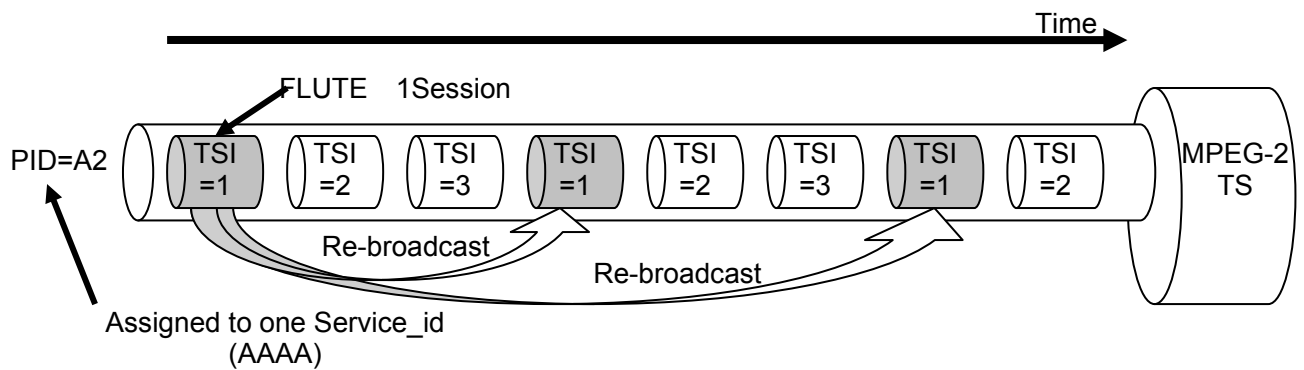


Fig. 2-5: Example of the delivery schedule of broadcast contents

As an example, the contents with TSI = 1 are broadcasted three times at different timings. In the same manner, the contents with TSI = 2 are also re-broadcasted. The TSI distinguishes the contents and is set to a unique value under the same service_id. However, the same TSI value may be allocated under different service_ids.

Meanwhile, the EPG/ECG metadata for the storable broadcasting as well as for the real-time-type broadcasting are both transmitted as a storable broadcasting content. The storable broadcasting data and EPG/ECG metadata are distinguished by the service_id (see Volume 7, Section 2.4 “Service Pattern”).

2.1.2 PSI/SI

The storable broadcasting system allows multiplexed transmission of IP packets on the MPEG-2 Systems (ITU-T H.222.0 | ISO/IEC 13818-1).

For each service (corresponding to a certain broadcast program identifier defined by the PAT and PMT), one and only one FLUTE session shall be operated.

PSI/SI tables used for the storable broadcasting system are listed in Table 2-1.

Table 2-1: PSI/SI Tables Used for Storable Broadcasting

Table name	Purpose
NIT	To acquire service_ids for storable broadcasting and INT
INT	To acquire IP information for storable broadcasting
PAT	To acquire PMT information for the service_id
PMT	To acquire the PID of storable broadcasting.
BIT	To acquire the broadcaster’s information
SDT	To acquire channel description
TOT	To acquire time and date information

For details of the structure and operation of PSI/SI, refer to Volume 4 “PSIO/SI Operational Guidelines”.

2.1.2.1 Detailed description of each table

(1) PAT

PAT specifies the PID of PMT for the `service_id` assigned to each content of the storable broadcasting. Each content can be uniquely identified by the `service_id` and time.

(2) PMT

PMT specifies the PID of the TS packet that transmits a content. It is not allowed to transmit one content with multiple PIDs.

Each content is transmitted as a ULE data stream, and the `stream_type` is set to the storable broadcasting content (refer to Volume 7, Section 3.1.2).

(3) NIT

NIT indicates the modification frequency and other parameters of the transmission line, and `service_ids` of the storable broadcasting service and EPG/ECG metadata service. (For the `service_type`, refer to Volume 7, Section 2.4.) The `service_id` of the INT is also indicated.

(4) INT

INT links the `service_id` to the information about the IP/MAC stream for storable broadcasting.

(5) BIT

BIT specifies the SI transmission parameters.

(6) SDT

SDT specifies the information about the service channel such as the service channel name and broadcaster name. All `service_ids` within the broadcaster's own service network are listed.

(7) TOT

TOT transmits time and date information.

2.1.2.2 Relationship between PSI/SI tables

Figure 2-6 shows the relationship between the PSI/SI tables for storable broadcasting.

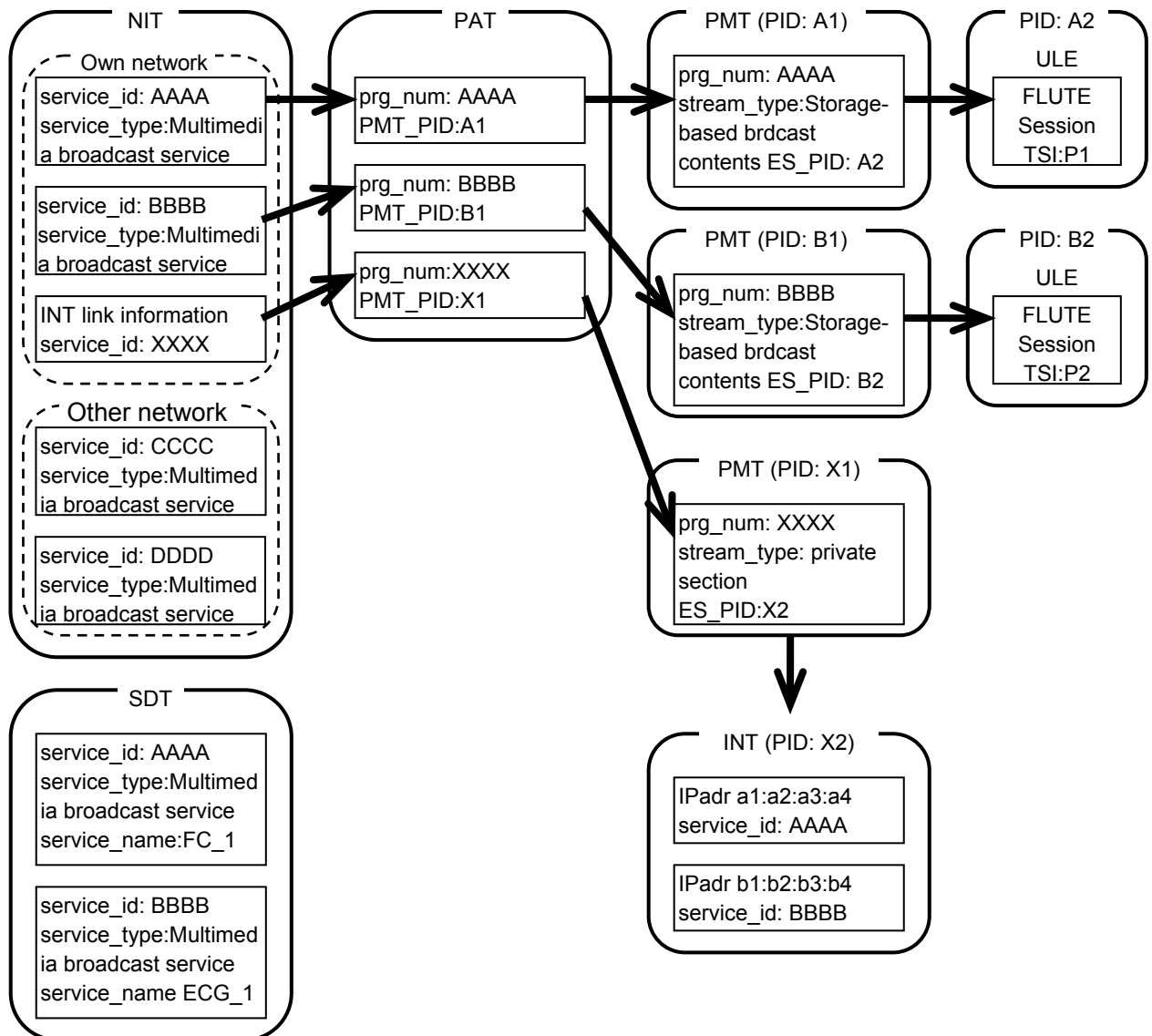


Fig. 2-6: Relationship between PSI/SI tables

In the NIT and SDT, the service_ids for the EPG/ECG metadata services are described (refer to Volume 7, Section 2.4). It should be noted that in the NIT, the service_ids of other networks and the service_ids for the INT link information are also described.

In the PAT, the PMT_PID is listed for each of the service_ids in the broadcaster's own network; and the PMT associated with the service_id holds the PID to which the storable broadcasting data or EPG/ECG metadata are transmitted. The storable broadcasting data and the EPG/ECG metadata are distinguished by the service_id (refer to Volume 7, Section 2.4). In the PMT for INT, the PID of INT is described, and the INT holds the IP address of each service_id.

2.1.2.3 Receiver operation

The example below describes how a receiver acquires a storable broadcasting content.

(1) NIT

The receiver accesses the NIT to search for a `service_id` in the storable broadcasting content or the EPG/ECG data service.

(2) PAT

It accesses the PAT to find the `program_number` that is equal to the `service_id` and obtains its `PMT_PID`.

(3) PMT

It accesses the PMT of the obtained PID to obtain the `ES_PID` from the `stream_type` of either storable broadcasting data or EPG/ECG metadata.

(4) ES

It accesses the obtained `ES_PID` to obtain the ES data (ULE); and from this ES data, it obtains the content.

(5)INT

It obtains the `service_id` of the INT from NIT descriptor; it is then allowed to access the INT through the PAT and PMT. The NIT describes the link between the `service_id` and IP address of the storable broadcasting content.

It should be noted that the metadata is transmitted in the same manner as the contents. The metadata describes the broadcast time, `service_id`, TSI, and other information of each content, and by using these data, the desired content can be obtained.

2.1.3 Transmission of content

Storable broadcasting is able to transmit broadcast contents that consist of multiple files including audio, video, text, still image, and metadata.

These files are divided into segments that have the block size specified by FLUTE and AL-FEC; then, after application-layer FEC processing, UDP/IP packetization, and header compression, they are encapsulated into MPEG-2 TS. A functional block diagram for the transmission of contents (multiple files) is depicted in Fig. 2-7.

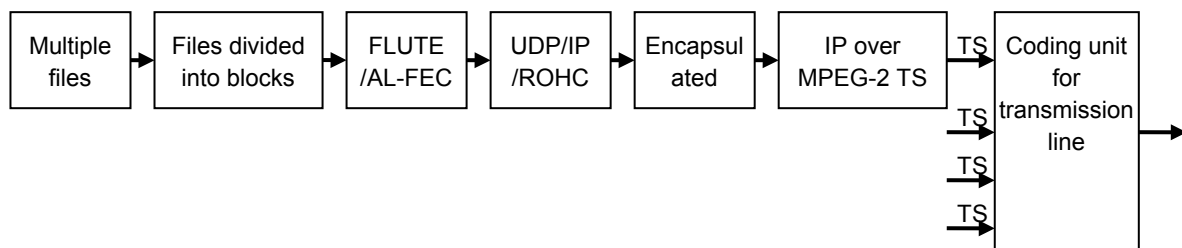


Fig. 2-7: Functional block diagram of the transmission of contents (multiple files)

Figure 2-8 shows the process of transmitting multiple content files as TS packets.

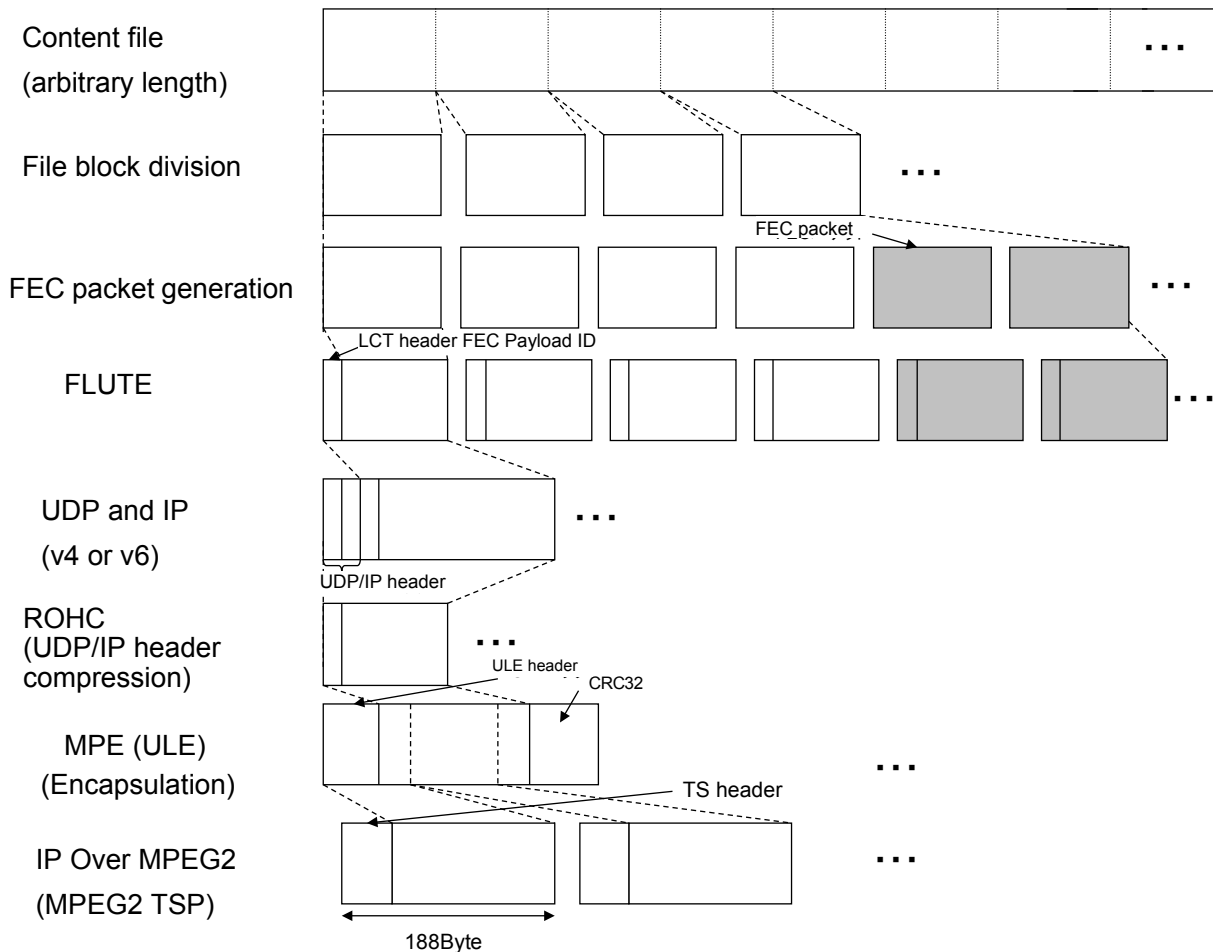


Fig. 2-8: Mapping of contents (multiple files) onto TS packets

When transmitting multiple files that constitute a content through FLUTE, it is possible to modify certain parameters for each file such as the retransmission frequency within the FLUTE session and the FEC error tolerance capability.

It should be noted that while it is possible to use either the v4 or v6 IP protocol, for the time being, only IPv4 is used in the FLUTE session.

2.1.4 File block division

The specifications for the file block division are detailed in Sections 2.1.5 “Operation of FLUTE transmission”, and 2.1.6 “Operation of AL-FEC transmission”.

2.1.5 Operation of FLUTE transmission

2.1.5.1 Packet structure

The FLUTE packet structure is shown below.

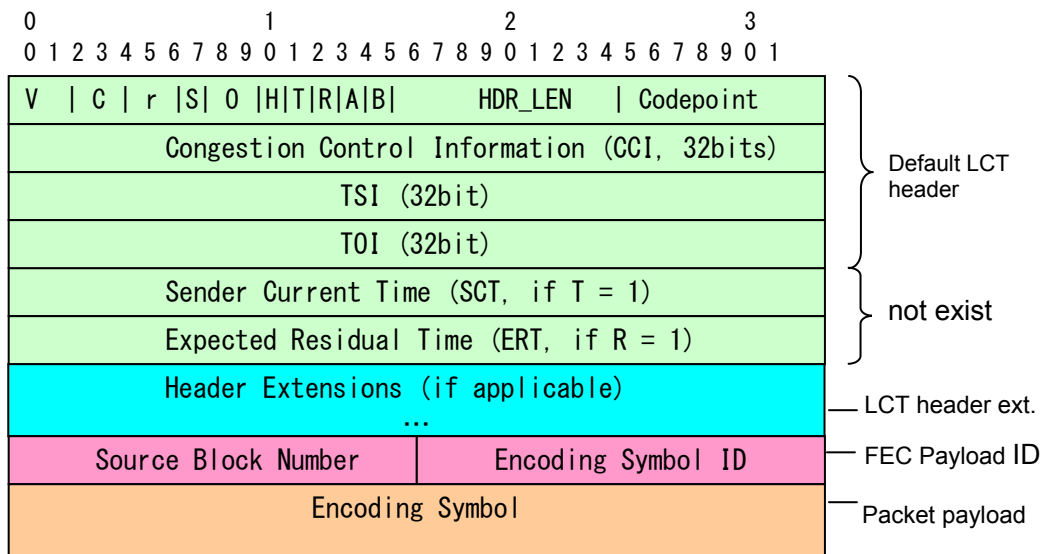


Fig. 2-9: FLUTE packet structure

Details of the fields are given below.

(1) Default LCT header

Table 2-2: Default LCT Header Parameters

Field	Size (bit)	Value	Remarks
V (version)	4	1 LCT version number	
C (congestion control flag)	2	0 CCI field size: 32 bits	
r (reserved)	2	0	
S (TSI flag)	1	1 TSI field size: 32 bits	
O (TOI flag)	2	1 TOI field size: 32 bits	
H (half-word flag)	1	0 TSI,TOI field size: 32 bits	
T (SCT present flag)	1	0 Presence or absence of SCT field	
R (ERT present flag)	1	0 Presence or absence of ERT field	
A (Close Session flag)	1	0 Close session flag	
B (Close Object flag)	1	0 Close object flag	

Field	Size (bit)	Value	Remarks
HDR_LEN (LCT header length)	8	LCT header total length (including default LCT header and LCT header extensions)	Expressed in units of 32 bits
CP (Codepoint)	8	When a FDT instance is being transmitted: 0; otherwise FEC Encoding ID	
CCI (Congestion Control Information)	32	0 CCI information is not present	
TSI (Transport Session Identifier)	32	TSI value (Identifier of the FLUTE session where the content is transmitted.)	
TOI (Transport Object Identifier)	32	TOI value (Identifier of the object within the session)	
SCT (Sender Current Time)	32	Time elapsed at the sender from the start of the session (milliseconds)	Not used
ERT (Expected Residual Time)	32	Residual transmission time for the object packet being transmitted (milliseconds)	Not used

(2) LCT header extensions

In the storable broadcasting system, the following header extensions are used.

(i) EXT_NOP

The EXT_NOP invalidates the header extension. It is set when no additional data is transmitted in the header extension.

Details of the format and fields are described below.

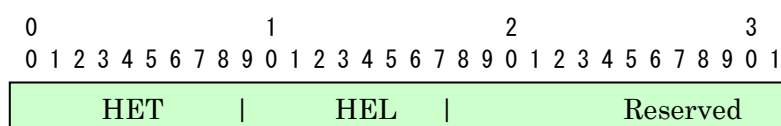


Fig. 2-10: Format of EXT_NOP

Table 2-3: Parameters of EXT_NOP

Field	Size (bit)	Value
HET (Header Extension Type)	8	0 Header type
HEL (Header Extension Length)	8	1 EXT_FTI total length: 32*1 = 32 bits
Reserved	16	Unused area

(ii) EXT_FTI

The EXT_FTI header transmits the information required to reconstruct the FDT instance. Details of the format and fields are provided below.

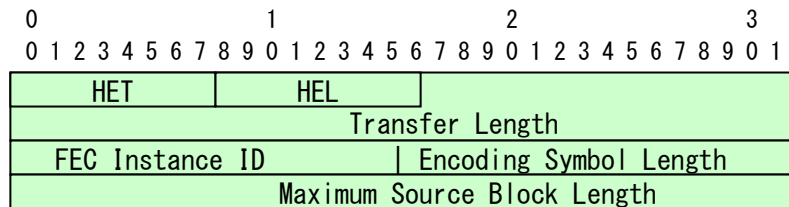


Fig. 2-11: Format of EXT_FTI

Table 2-4: Parameters of EXT_FTI

Field	Size (bit)	Value
HET (Header Extension Type)	8	64 Header type
HEL (Header Extension Length)	8	4 EXT_FTI total length: 32*4 = 128 bits
Transfer Length	48	Length of object transmitted
FEC Instance ID	16	0: Compact No-Code FEC
Encoding Symbol Length	16	Length of encoding symbol
Maximum Source Block Length	32	Maximum number of source symbols for one source block.

(iii) EXT_FDT

The EXT_FDT header transmits the identifier of the FDT instance included in the packet payload.

Details of the format and fields are shown below.

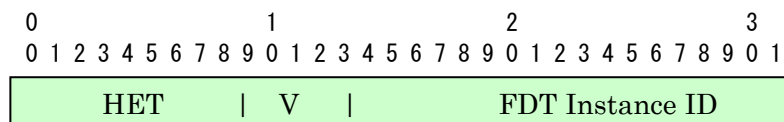


Fig. 2-12: Format of EXT_FDT

Table 2-5: Parameters of EXT_FDT

Field	Size (bit)	Value
HET (Header Extension Type)	8	192 Header type
V (Version)	4	1 Version of FLUTE
FDT Instance ID	20	FDT instance identifier It is incremented by one each time the content of the FDT instance is modified (Initial value is 0)

(iv) FEC payload ID

It identifies the payload information represented by the encoding symbols in the packet payload. The format and fields are determined by the FEC Encoding ID. For more details, refer to Section 2.1.6.1 in this volume.

(3) Packet payload

The packet payload contains encoding symbols. The objects (contents and FDT instances) are divided into source blocks and then converted into encoding symbols by the FEC schema. Each encoding symbol is identified by the FEC Payload ID.

(4) Packet structure for FDT and file object transmission

The following table summarizes the packet structure when an FDT or file object is transmitted.

Table 2-6: Packet Structure for Object Transmission

Packet element	FDT instance	File object	
	FEC Encoding ID: 0	FEC Encoding ID: 0	FEC Encoding ID: 3
Default LCT header	○	○	○
LCT header ext. (EXT_NOP)	×	○	○
LCT header ext. (EXT_FTI)	○	×	×
LCT header ext. (EXT_FDT)	○	×	×
FEC Payload ID	○	○	○
Packet payload	○	○	○

○: exists ×: does not exist

2.1.5.2 Source block division

In the delivery server, an object to be transmitted by downloading is divided into source blocks based on the following information:

- L: Transfer length (bytes)
- B: Source block length (number of source symbols per source block)
- E: Encoding symbol length (bytes)

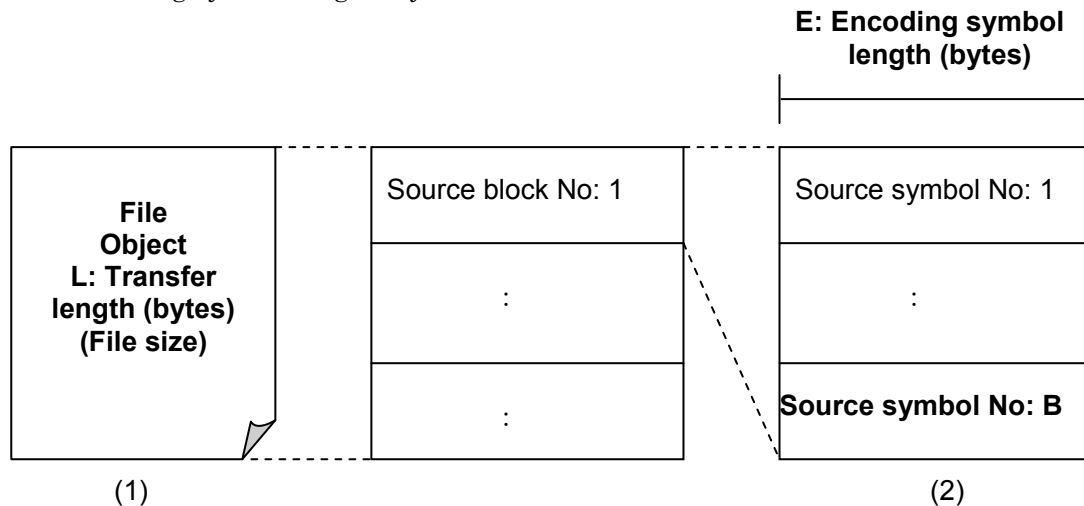


Fig. 2-13: Source block structure

- (1) File object with a transfer length of (L)
- (2) One source block is configured to be encoding symbol length (E) × source symbol length (B).

The logic of source block division is:

1. Number of all source symbols $T = L / E$ (rounded up)
2. Number of source blocks $N = T / B$ (rounded up)
3. Average length of source blocks $A = T / N$
4. $A_{\text{large}} = A$; the fraction is rounded up.
5. $A_{\text{small}} = A$; the fraction is rounded down.
6. $A_{\text{fraction}} = A - A_{\text{small}}$
7. $I = A_{\text{fraction}} * N$

As a result, the first “T” number of source blocks consist of “A_large” number of source symbols per block (E bytes of source symbols). Except for the last block, the rest of “N-I” number of source blocks consisting of “A_small” source symbols carry E bytes, and the last block carries $L - ((L - 1) / E) \leftarrow \text{fraction rounded down} * E$ bytes (see the following figure).

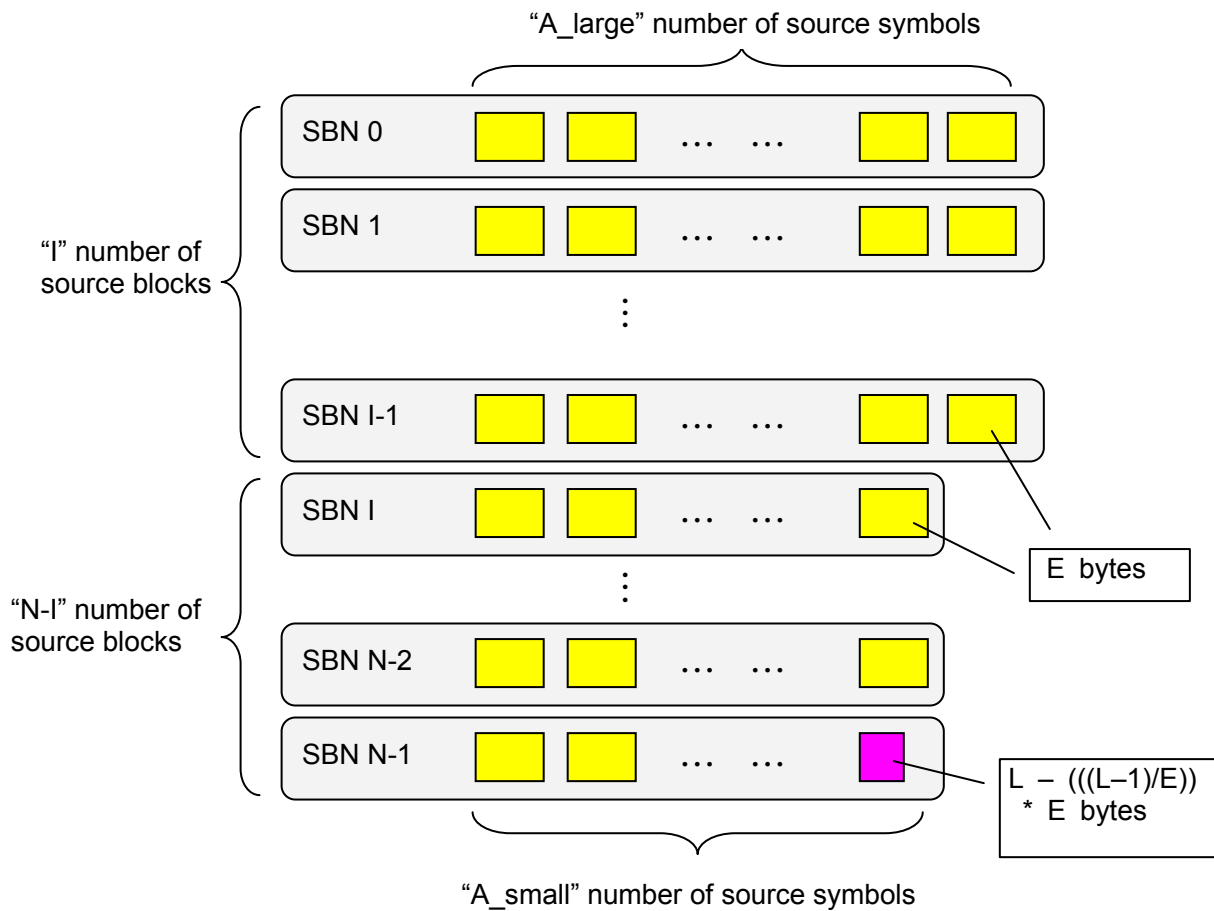


Fig. 2-14: Blocking method

Example) In the case of Transfer length $L = 411$ bytes, Source block length $B = 4$, Encoding symbol length $E = 20$ bytes,

1. $T = 411 / 20 = 20.55 \Rightarrow 21$
2. $N = 21 / 4 = 5.25 \Rightarrow 6$
3. $A = 21 / 6 = 3.5$
4. $A_{\text{large}} = 4$
5. $A_{\text{small}} = 3$
6. $A_{\text{fraction}} = 0.5$
7. $I = 0.5 * 6 = 3$
- 8.

Last symbol length = $411 - (((411-1) / 20) \leftarrow \text{fraction rounded down}) * 20 = 11$ bytes

As shown above, for the Source Block Number (SBN) from 0 through 2, each source block contains four 20-byte-long source symbols; SBN from 3 through 5 except the last block contains three 20-byte-long source symbols; and the last source symbol length is 11 bytes.

2.1.5.3 FDT instance

The FDT instance is the data in XML format that provides detailed information about an object transmitted in the FLUTE session. The information is mapped to an object transmitted according to the TOI value.

2.1.5.3.1 Transmission of the FDT instance

An FDT instance is always sent prior to object transmission (first-time transmission). After the first-time transmission of the object, the FDT instance may be sent again during object transmission. By transmitting the FDT instance again, the receiver is allowed to receive the FDT instance and hence the transmitted object in the middle of the FLUTE session.

2.1.5.3.2 XML schema

The XML schema for the FDT instance is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns="http://www.arib.or.jp/tmm/fdt/2011/03"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.arib.or.jp/tmm/fdt/2011/03"
  elementFormDefault="qualified">

  <xs:element name="FDT-Instance" type="FDT-InstanceType"/>
  <xs:complexType name="FDT-InstanceType">
    <xs:sequence>
      <xs:element name="File" type="File-Type" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="Group" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
      <xs:any namespace="##other" processContents="skip" minOccurs="0"
        maxOccurs="unbounded"/>
    </xs:sequence>

    <xs:attribute name="Expires" type="xs:string" use="required"/>
    <xs:attribute name="Complete" type="xs:boolean" use="optional"/>
    <xs:attribute name="Content-Type" type="xs:string" use="optional"/>
    <xs:attribute name="Content-Encoding" type="xs:string" use="optional"/>
    <xs:attribute name="FEC-OTI-FEC-Encoding-ID" type="xs:unsignedLong"
      use="optional"/>
    <xs:attribute name="FEC-OTI-FEC-Instance-ID" type="xs:unsignedLong"
      use="optional"/>
    <xs:attribute name="FEC-OTI-Maximum-Source-Block-Length"
      type="xs:unsignedLong" use="optional"/>
    <xs:attribute name="FEC-OTI-Encoding-Symbol-Length"
      type="xs:unsignedLong" use="optional"/>
    <xs:attribute name="FEC-OTI-Max-Number-of-Encoding-Symbols"
      type="xs:unsignedLong" use="optional"/>
    <xs:attribute name="FEC-OTI-Scheme-Specific-Info"
      type="xs:base64Binary" use="optional"/>
    <xs:anyAttribute processContents="skip"/>
  </xs:complexType>

  <xs:complexType name="File-Type">
    <xs:sequence>
      <xs:element name="Group" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
      <xs:any namespace="##other" processContents="skip"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```

</xs:sequence>
<xs:attribute name="Content-Location" type="xs:anyURI" use="required"/>
<xs:attribute name="TOI" type="xs:positiveInteger" use="required"/>
<xs:attribute name="Content-Length" type="xs:unsignedLong" use="required"/>
<xs:attribute name="Transfer-Length" type="xs:unsignedLong" use="optional"/>
<xs:attribute name="Content-Type" type="xs:string" use="optional"/>
<xs:attribute name="Content-Encoding" type="xs:string" use="optional"/>
<xs:attribute name="Content-MD5" type="xs:base64Binary" use="optional"/>
<xs:attribute name="FEC-OTI-FEC-Encoding-ID" type="xs:unsignedLong"
use="optional"/>
<xs:attribute name="FEC-OTI-FEC-Instance-ID" type="xs:unsignedLong"
use="optional"/>
<xs:attribute name="FEC-OTI-Maximum-Source-Block-Length"
type="xs:unsignedLong" use="optional"/>
<xs:attribute name="FEC-OTI-Encoding-Symbol-Length"
type="xs:unsignedLong" use="optional"/>
<xs:attribute name="FEC-OTI-Max-Number-of-Encoding-Symbols"
type="xs:unsignedLong" use="optional"/>
<xs:attribute name="FEC-OTI-Scheme-Specific-Info"
type="xs:base64Binary" use="optional"/>
<xs:anyAttribute processContents="skip"/>
</xs:complexType>
</xs:schema>

```

2.1.5.3.3 Semantics of the FDT instance

The semantics of the FDT instance are defined below.

Table 2-7: Operation of the FDT Instance

Element/ Attribute Name	Use	Appearance	Max. character length and value	Definition
FDT-Instance	○			FDT instance information
@Expires	○	1	10 decimal digits (0 - 4294967295)	Describes the expiry time of the FDT instance in UNIX time format. The expiry time is set as the end time of the period while the repair of stored content is available. Once all files that constitute the content have been received, the receiver can delete the FDT instance information.
@Complete	○	0..1		Tells that no new FDT instance will be transmitted.
@Content-Type	○	0..1	Common information within the FDT instance. The contents are the same as those of File-Type element. The value of the common information is applied to any attribute which is not specified by individual File-Type element.	
@Content-Encoding	—	0..1		
@FEC-OTI-FEC-Encoding-ID	○	0..1		
@FEC-OTI-FEC-Instance-ID	○	0..1		
@FEC-OTI-Maximu	○	0..1		

	m-Source-Block-Length				
	@FEC-OTI-Encoding-Symbol-Length	○	0..1		
	@FEC-OTI-Max-Number-of-Encoding-Symbols	○	0..1		
	@FEC-OTI-Scheme-Specific-Info	○	0..1		
	File	○	0..∞		Describes the information about the file object to be transmitted.
	@Content-Location	○	1	Max. 255 bytes	Describes the URI of the content.
	@TOI	○	1	10 decimal digits (0 - 4294967295)	Describes the object identifying information.
	@Content-Length	○	1	10 decimal digits (0-1073741823)	Describes the content length up to 1 GB.
	@Transfer-Length	○	0..1	10 decimal digits (0-1073741823)	Describes the transfer length.
	@Content-Type	○	0..1	Max. 255 bytes	Describes the MIME type.
	@Content-Encoding	—	0..1	Max. 255 bytes	Describes the content encoding information.
	@Content-MD5	○	0..1	Max. 24 bytes	Describes the message digest of the file object. The use of the message digest allows the receiver to check the compatibility of the file object that has been 100% received.
	@FEC-OTI-FEC-Encoding-ID	○	0..1	0: Compact No-Code FEC 3: LDPC Staircase Codes	Describes the FEC Encoding-ID.
	@FEC-OTI-FEC-Instance-ID	○	0..1	Fixed to 0	Describes the FEC Instance ID.
	@FEC-OTI-Maximum-Source-Block-Length	○	0..1	For Compact No-Code FEC : 5 decimal digits (0-65535) For LDPC Staircase Codes FEC: 7 decimal digits (0-1048575)	Describes the maximum number of source symbols within the source block.
	@FEC-OTI-Encoding-Symbol-Length	○	0..1	10 decimal digits (0-4294967295)	Describes the length of the encoding symbol in bytes.
	@FEC-OTI-Max-Number-of-Encoding-Symbols	○	0..1	For Compact No-Code FEC: 5 decimal digits (0-65535)	Describes the maximum number of encoding symbols within the source block.

				For LDPC Staircase Codes FEC: 7 decimal digits (0-1048575)	
	@FEC-OTI-Scheme-Specific-Info	○	0..1	Max. 12 bytes	Describes the seed of random number, etc. Encoding-ID: 0 - Don't Care Encoding-ID: 3 - Random number, degree and multiplexed number of symbols are described.
	Group	○	0..∞	Max. 255 bytes	Describes the group information. ※: It can be used for the grouping of contents, software, etc. consisting of multiple files. If a group is set up, the receiver must receive all files pertaining to the group.
	##other	○	0..∞		FDT extension
	Group	○	0..∞	Max. 255 bytes	Describes the group information. ※: Refer to the Group of the File
	##other	○	0..∞		FDT extension

2.1.5.3.4 Extension of FDT

The FDT instance is extended to convey information about a schedule change in the stored content broadcast, object version, and full FDT.

(1) XML schema

The XML schema of the FDT extension is defined below. It should be noted that the "TmmFdt.xsd" included below is the XML schema file defined in Section 2.1.5.3.2 in this volume.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:fdt="http://www.arib.or.jp/tmm/fdt/2011/03"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.arib.or.jp/tmm/fdt/2011/03"
  elementFormDefault="qualified">

  <xs:redefine schemaLocation="TmmFdt.xsd">
    <xs:complexType name="FDT-InstanceType">
      <xs:complexContent>
        <xs:extension base="fdt:FDT-InstanceType">
          <xs:sequence>
            <xs:element name="ScheduleChange" type="fdt:ScheduleChangeType"
minOccurs="0" />
          </xs:sequence>
          <xs:attribute name="FullFDT" type="xs:boolean" use="optional" default="false"/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
```

```

<xs:complexType name="File-Type">
  <xs:complexContent>
    <xs:extension base="fdt:File-Type">
      <xs:attribute name="Version-ID-Length" type="xs:unsignedLong" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
</xs:redefine>

<xs:complexType name="ScheduleChangeType">
  <xs:attribute name="Cause" type="xs:string" use="required"/>
  <xs:attribute name="Action" type="xs:string" use="optional"/>
  <xs:attribute name="Time" type="xs:string" use="optional"/>
</xs:complexType>

</xs:schema>

```

(2) Semantics of FDT extension

The semantics of FDT extension are defined below.

Table 2-8: Operation of FDT Extension

Element/ Attribute Name	Use	Appearance	Max. character length and value	Definition
FDT-Instance	○			Refer to the FDT-Instance element in Table 2-7.
@FullFDT	○	0..1		A flag to describe the information about all the file objects to be transmitted.
ScheduleChange	○	0..1		Describes the information about the schedule change
@Cause	○	1	Max. 255 bytes	Describes the cause(s) of the schedule change.
@Action	○	0..1	Max. 255 bytes	Describes the receiver's action required in response to the schedule change.
@Time	○	0..1	Specified in the YYYY-MM-DDT hh:mm:ss+09:00 format (25 bytes)	Describes the time when the receiver should take the action described by the Action attribute.
File	○	0..∞		Refer to the File element in Table 2-7.
@Version-ID-Length	○	0..1	3	The number of bits to indicate the file version transmitted. Must be 3 bits, 8 gradations. This attribute is defined for each FDT File element.

(3) Schedule change information

The schedule change information notified by the FDT instance allows the receiver to take the

actions required at the time of schedule change. Specific examples of the cause, receiver's action, and FDT instance are described below.

■ Causes of schedule change

Causes of schedule change that can be notified to receivers and the values of the Cause attribute are listed below.

Table 2-9: Definition of Causes

Cause	Value	Definition
Extension	Extension	Continue the program broadcasting beyond the scheduled time.
Early finish	Shortening	Finish the program earlier than the scheduled time.
Delay	Delay	Start the program at a delayed time.
Interruption	Interruption	Due to certain cause, stop the program being broadcasted and start another program.
Cut-in	Cut-in	During broadcasting, insert an unscheduled program.
Move forward	Forwarding	Start the program earlier than scheduled.

■ Receiver's action at the time of schedule change

The receiver's actions that can be notified to receivers and the values of the Action attribute are listed below.

Table 2-10: Definition of the Receiver's Actions

Receiver's action	Value	Definition
Forced termination	Abort	Forcibly stop receiving the storable broadcasting content.
Program schedule check	Check	Receive program transmission A to check the latest program schedule.
Repair of stored content	Repair	Repair the stored content by communications.
Retry	Retry	Re-try to receive the schedule-changed storable broadcasting content.

■ Example of schedule change operation

An example of an FDT instance sent at the time of schedule change is presented below.

As an example, by transmitting the FDT instance for posting a schedule change as listed below (Delay) 30 seconds before the starting time of the scheduled program, and repeating it every 2 seconds, the receiver is allowed to know the delay of the broadcast (Cause = "Delay") and the time of the retry for reception (Action = "Retry", Time = "2010-06-30T22:00:00+09:00"). Note that this technical report does not provide any rule on the sending time of the schedule change notice.

```
<?xml version="1.0" encoding="UTF-8"?>
<FDT-Instance Expires="3478323603"
```

```
Complete="false"
FullFDT="false"
xmlns="http://www.arib.or.jp/tmm/fdt/2011/03">

<ScheduleChange
  Cause="Delay"
  Action="Retry"
  Time="2010-06-30T22:00:00+09:00"/>
</FDT-Instance>
```

(4) Version information

In order to represent the version of the transmitted file object, the “Version-ID-Length” parameter is added to the FDT instance. The version information of the transmitted file object is represented by the TOI. The “Version-ID-Length” indicates how many least significant bits of the TOI are used as the version information. A specific example is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<FDT-Instance
  xmlns="http://www.arib.or.jp/tmm/fdt/2011/03"
  Expires="2890842807">
  <File
    Content-Location="/user/sample/test1.txt"
    Version-ID-Length="3" ← The lower 3 bits of TOI represent the version information.
    Content-Length="2048"
    Content-MD5="KGdImnJdwImsArJmMpGC2Q=="
    TOI="9"/> ← Binary expression: 0000000000001001 upper 13 bits: Object identifier (1)
                                     lower 3 bits: Version identifier (1)
  <File
    Content-Location="/user/sample/test2.txt"
    Version-ID-Length="3" ← The lower 8 bits of TOI represent the version information.
    Content-Length="4096"
    Content-MD5="Tm6CjilYWMcPWwYUIt2IYw=="
    TOI="17"/> ← Binary expression: 0000000000010001 upper 13 bits: Object identifier (2)
                                     lower 3 bits: Version identifier (1)
  <File
    Content-Location="/user/samtest1.txt"
    Content-Length="8192"
    Content-MD5="LIXUOUKdAnvHZGKolymDPw=="
    TOI="3"/> ← “Version-ID-Length” attribute does not exist; therefore, the whole
                                     represents the Object identifier (3).
</FDT-Instance>
```

(5) FullFDT information

The FullFDT flag is set to indicate that all file objects (content files) transmitted within the same session are described in one FDT instance. If the FullFDT is set “true”, the receiver recognizes that the information about all file objects (contents) transmitted in the current session is described in the received FDT instance.

By setting the FullFDT, the receiver is allowed to recognize the file list constituting the

content with one FDT instance. A specific use of FullFDT is shown below.

(Example) There are four file objects, A, B, C, and D, to be transmitted.

If FullFDT = True: A, B, C, and D are all defined in one FDT instance.

If FullFDT = False or omitted: A, B, C, and D are not defined altogether in one FDT instance, so that they are transmitted in separate FDT instances.

However, only FullFDT = True is used for the time being.

2.1.5.3.5 XML instance of the FDT instance

An XML instance of the FDT instance is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<FDT-Instance
  Expires="3478323603"
  Complete="true"
  FullFDT="true"
  xmlns="http://www.arib.or.jp/tmm/fdt/2011/03">
  <File
    Content-Length="1024"
    Content-Location="/user/sample/test1.txt"
    Content-Type="text/plain"
    Version-ID-Length="3"
    TOI="17"
    Content-MD5="MTIzNDU="
    FEC-OTI-FEC-Encoding-ID="3"
    FEC-OTI-FEC-Instance-ID="0"
    FEC-OTI-Maximum-Source-Block-Length="4"
    FEC-OTI-Encoding-Symbol-Length="32"
    FEC-OTI-Max-Number-of-Encoding-Symbols="8"
    FEC-OTI-Scheme-Specific-Info="MDAxMTIyMzM0NDU1"/>
  </File
</FDT-Instance>
```

2.1.5.4 Transmission control metadata

This section specifies the detailed data format for the transmission control metadata used for the transmission of storable broadcasting contents.

The transmission control metadata consists of the parameters associated with the FLUTE session used for the transmission of contents, the repair of stored contents, and reception reports. Therefore, it is closely related to the EPG/ECG metadata. For the relationship between the transmission control metadata and the EPG/ECG metadata, refer to Volume 10, Section 8.2.3.2.

There are three kinds of elements contained in a transmission control metadata.

- User Service Description
- Session Description
- Associated Delivery Procedure Description

As depicted in Fig. 2-15 below, the User Service Description is configured with the contained Session Description and Associated Delivery Procedure Description, and it is described in XML format.

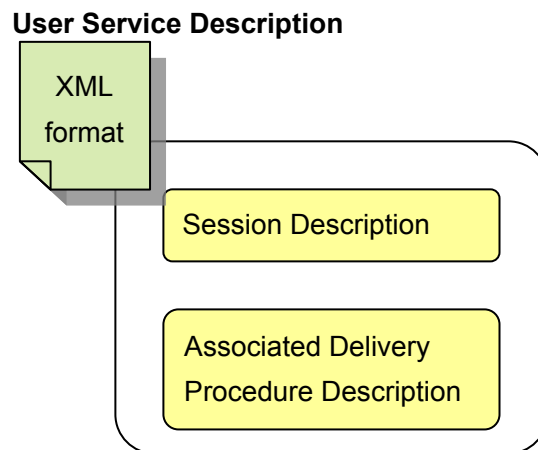


Fig. 2-15: Configuration of User Service Description

2.1.5.4.1 User Service description

The User Service Description holds various kinds of Description information about the contents transmission.

(1) XML schema of User Service Description

The XML schema of the User Service Description is defined below.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns="http://www.arib.or.jp/tmm/fci/2011/03"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

```

targetNamespace="http://www.arib.or.jp/tmm/fci/2011/03"
elementFormDefault="qualified">

<xs:element name="userServiceDescription" type="userServiceDescriptionType"/>
<xs:complexType name="userServiceDescriptionType">
  <xs:sequence>
    <xs:element name="version" type="xs:unsignedLong"/>
    <xs:element name="program" type="CRIDType"/>
    <xs:element name="sessionDescription" type="SDPType" minOccurs="0"/>
    <xs:element name="associatedProcedureDescription"
      type="associatedProcedureDescriptionType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="CRIDType">
  <xs:restriction base="xs:anyURI">
    <xs:pattern value="(c|C)(r|R)(i|I)(d|D):/*.*/"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="SDPType">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:complexType name="associatedProcedureDescriptionType">
  <xs:sequence>
    <xs:element name="postFileRepair" type="repairProcedureType" minOccurs="0"/>
    <xs:element name="postReceptionReport" type="reportProcedureType"
minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="basicProcedureType">
  <xs:sequence>
    <xs:element name="receptionSchedule" type="receptionScheduleType"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="offsetTime" type="xs:unsignedLong" use="optional"/>
  <xs:attribute name="randomTimePeriod" type="xs:unsignedLong" use="optional"/>
</xs:complexType>
<xs:complexType name="receptionScheduleType">
  <xs:sequence>
    <xs:element name="serverURI" type="xs:anyURI" />
    <xs:element name="receptionPeriod" type="receptionPeriodType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="repairProcedureType">
  <xs:complexContent>
    <xs:extension base="basicProcedureType">
      <xs:attribute name="manualRepairStartDate" type="xs:dateTime" use="optional"/>
      <xs:attribute name="manualRepairEndDate" type="xs:dateTime" use="optional"/>
      <xs:attribute name="autoRepairAutoStorePercentage" type="xs:string"
use="optional" default="50"/>
      <xs:attribute name="autoRepairManualStorePercentage" type="xs:string"
use="optional" default="50"/>
      <xs:attribute name="manualRepairAutoStorePercentage" type="xs:string"
use="optional" default="50"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

        <xs:attribute name="manualRepairManualStorePercentage" type="xs:string"
use="optional" default="50"/>
        <xs:attribute name="optionValue" type="xs:string" use="optional"/>
    </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="receptionPeriodType">
    <xs:sequence>
        <xs:element name="receptionCycleStartTime" type="xs:time"/>
        <xs:element name="receptionCycleEndTime" type="xs:time"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="reportProcedureType">
    <xs:complexContent>
        <xs:extension base="basicProcedureType">
            <xs:attribute name="samplePercentage" type="xs:string" use="optional"
default="50"/>
            <xs:attribute name="reportType" type="xs:byte" use="optional" default="15"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
</xs:schema>

```

(2) Semantics of User Service Description

The semantics of User Service Description is defined below.

Table 2-11: Operation of User Service Description

Element/ Attribute Name	Use	Appearance	Max. character length and value	Definition
userServiceDescription	○			User ServiceDescription info.
version	○	1	14 decimal digits (YYYYMMDDhhmmss format)	Describes the version of User ServiceDescription. Equivalent to the fragmentVersion defined in Volume 10 “4.4.2 Operation of fragmentVersion”.
program	○	1	Max. 255 bytes	Describes corresponding CRID.
sessionDescription	○	0..1	Described in CDATA	Describes Session Description Protocol (SDP).
associatedProcedureDescription	○	0..1		Information about Repair of stored contents/ Reception report.
postFileRepair	○	0..1		Control data for Repair of stored contents. If omitted: Repair of stored contents is not operated.
@offsetTime	○	0..1	10 decimal digits (0 - 4294967295)	Describes the offset time [in sec.] associated with automatic repair. If omitted: Automatic repair not operable. For details, refer to Volume 12 Section 4.7.

		@randomTimePeriod	○	0..1	10 decimal digits (0 - 4294967295)	Describes the random time period [in sec.] associated with automatic repair. If omitted: Automatic repair not operable. For details, refer to Volume 12 Section 4.7.
		@manualRepairStartDate	○	0..1	Specified in YYYY-MM-DDT hh:mm:ss+09:00 format (25 bytes)	Describes manual repair start date and time. If omitted: Manual repair is not operable. For details, refer to Volume 12 Section 4.7.
		@manualRepairEndDate	○	0..1	Specified in YYYY-MM-DDT hh:mm:ss+09:00 format (25 bytes)	Describes manual repair end date and time. If omitted: Manual repair is not operable. For details, refer to Volume 12 Section 4.7.
		@autoRepairAutoStorePercentage	○	0..1	0 – 100 (Max. 3 bytes)	Describes the automatic repair threshold value [%] for automatic storage. If omitted: 50%, For details, refer to Volume 12 Section 4.7.
		@autoRepairManualStorePercentage	○	0..1	0 – 100 (Max. 3 bytes)	Describes the automatic repair threshold value [%] for manual storage. If omitted: 50%. For details, refer to Volume 12 Section 4.7.
		@manualRepairAutoStorePercentage	○	0..1	0 – 100 (Max. 3 bytes)	Describes the manual repair threshold value [%] for automatic storage. If omitted: 50%. For details, refer to Volume 12 Section 4.7.
		@manualRepairManualStorePercentage	○	0..1	0 – 100 (Max. 3 bytes)	Describes the manual repair threshold value [%] for manual storage. If omitted: 50%. For details, refer to Volume 12 Section 4.7.
		@optionValue	○	0..1	Max. 255 bytes	Describes option values for the repair of stored contents. This Technical Report does not provide any specific value.
		receptionSchedule	○	1.. ∞		Information about reception schedule for automatic repair. Refer to Table 2-12.
		postReceptionReport	○	0..1		Reception report control information. If omitted: No reception report operated.
		@offsetTime	○	0..1	10 decimal digits (0 - 4294967295)	Describes the offset time [in sec.]. If omitted: 0. For details, refer to Volume12 Section 4.7.
		@randomTimePeriod	○	0..1	10 decimal digits (0 - 4294967295)	Describes the random time period [in sec.]. For details, refer to Volume12 Section 4.7.

		@samplePercentage	○	0..1	0 – 100 (Max. 3 bytes)	Describes the reception report threshold value [%]. If omitted: 0. For details, refer to Volume12 Section 4.7.
		@reportType	○	0..1	3 decimal digits (0-255)	Report-type information. Set the bit corresponding to the type of report. (LSB) bit 1: Content acquisition is completed only with the reception of broadcast. bit 2: After broadcast has been received, the contents are discarded by checking the threshold value. bit 3: Repair of stored contents has failed. bit 4: Others. bit 5 ~ 8: Not used.
		receptionSchedule	○	1.. ∞		Information about reception schedule for the reception report. Refer to Table 2-12.

Table 2-12: Operation of receptionSchedule

Element/ Attribute Name	Use	Appearance	Max. character length and value	Remarks
receptionSchedule	○			Reception schedule information.
serverURI	○	1	Max. 255 bytes	Reception server URL
receptionPeriod	○	0..1		Information about reception period. If omitted: CycleTime for the repair of stored contents/ reception report is not operated.
receptionCycleStartTime	○	1	Specified in hh:mm:ss+09:00 format (14 bytes)	Describe the reception start time within the server reception period. For details, refer to Volume12 Section 4.7.
receptionCycleEndTime	○	1	Specified in hh:mm:ss+09:00 format (14 bytes)	Describe the reception end time within the server reception period. Even if the date is changed from the “receptionCycleStartTime”, the time is specified as a period from “receptionCycleStartTime” to “receptionCycleEndTime” in 24 hours. For details, refer to Volume12 Section 4.7.

(3) XML instance of User Service Description

An XML instance of User Service Description is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<userServiceDescription xmlns="http://www.arib.or.jp/tmm/fci/2011/03">
  <version>20100501090000</version>
  <program>crd://example.com/fc/0011/c00123</program>
  <sessionDescription>
    <![CDATA[
      v=0
      o=example.com 751092616 751111042 IN IP4 192.168.0.128
      s=crd://example.com/fc/0011/c00123
      t=1275145200 1275167100
      m=application 12345 FLUTE/UDP 0
      c=IN IP4 239.0.0.1/128
      a=flute-tsi:777
      a=storageDemands:65536
      a= transmissionSchedule:/user/sample/CM_A.mp4 version=20100501090000
        number=1 2010-05-30T09:00:00+09:00 09:05:00+09:00
        number=2 2010-05-30T15:00:00+09:00 15:05:00+09:00 ]]>
  </sessionDescription>
  <associatedProcedureDescription>
    <postFileRepair
      offsetTime="1500"
      randomTimePeriod="606600"
      manualRepairStartDate="2010-05-30T15:05:00+09:00"
      manualRepairEndDate="2010-06-07T00:00:00+09:00"
      autoRepairAutoStorePercentage="35"
      autoRepairManualStorePercentage="35"
      manualRepairAutoStorePercentage="35"
      manualRepairManualStorePercentage="35">
      <receptionSchedule>
        <serverURI>http://example.com/fileRepair1_Server/</serverURI>
        <receptionPeriod>
          <receptionCycleStartTime>09:00:00+09:00</receptionCycleStartTime>
          <receptionCycleEndTime>16:00:00+09:00</receptionCycleEndTime>
        </receptionPeriod>
      </receptionSchedule>
    </postFileRepair>
    <postReceptionReport
      offsetTime="1500"
      randomTimePeriod="606600"
      reportType="15"
      samplePercentage="25">
      <receptionSchedule>
        <serverURI>http://example.com/fileReport1_Server/</serverURI>
        <receptionPeriod>
          <receptionCycleStartTime>09:00:00+09:00</receptionCycleStartTime>
          <receptionCycleEndTime>16:00:00+09:00</receptionCycleEndTime>
        </receptionPeriod>
      </receptionSchedule>
    </postReceptionReport>
  </associatedProcedureDescription>
</userServiceDescription>
```

2.1.5.4.2 Session Description

The Session Description is given in SDP format data used for a FLUTE session. It is described in a SessionDescription element of the UserServiceDescription.

The Session Description is described in the following order. * indicates optional information.

Session Description

v= (protocol version)
o= (owner/creator and session identifier).
s= (session name)
i=* (session information)
u=* (URI of description)
c=* (connection information - not required if included in all media)
b=* (bandwidth information)
<One or more Time descriptions>
z=* (time zone adjustments)
a=* (zero or more session attribute lines)
<One or more Media descriptions>

Fields in Time description

t= (time the session is active)
r=* (zero or more repeat times)

Fields in Media description

m= (media name and transport address)
i=* (media title)
c=* (connection information - optional if included at session-level)
b=* (bandwidth information)
a=* (zero or more media attribute lines)

Details of the fields are given below.

v (Protocol Version) (Description: Required)

Content : SDP version
Field : v = 0 (fixed)

o (Origin) (Description: Required)

Content : Information about the originator of the Session Description
Field : o=<username> <session id> <version> <network type> <address type>
<address>
Subfield : username: Login name of the originator.
session id: Session identifier (Unix time format)
version: Version of the description within the SDP (Unix time format)
network type: Type of network: "IN"
address type: Type of the address "IP4", "IP6"
address: IP address in accordance with the address type

s (Session Name) (Description: Required)

Content : Session name specified in the Session Description
Field : s=<session name>

i (Session and Media Information) (Description: Optional)

Content : Session or media information
Field : i=<session/media description>

u (URI) (Description: Optional)

Content : Reference to additional information
Field : u=<URI>

c (Connection Data) (Description: Required)

Content : Connection address for a session
Field : c=<network type> <address type> <connection address>/<ttl>[/<number
of addresses>]
Subfield : network type: Network type: "IN"
address type: Address type. "IP4", "IP6"
connection address: IP address in accordance with the address type.
ttl: Time-to-live of packets (number of hops to be relayed), set for
multicast only
number of addresses: Number of multicast groups used, set for multicast
only; If omitted: regarded as "1".

b (Bandwidth) (Description: Optional)

Content : Specifies the bandwidth.
Field : b=<modifier>:<bandwidth-value>
Subfield : modifier: Identifier for the specified bandwidth ("CT", "AS", "RR", etc.).
bandwidth-value: bandwidth used by the modifier in units of kbps.

t (Times) (Description: Required)

Content : Start and end times of the broadcast of contents.
Field : t=<start time> <stop time>
Subfield : start time: Broadcast period start time (Unix time format)
stop time: Broadcast period end time (Unix time format)

r (Repeat Times) (Description: Optional)

Content : Specifies repeat times for a session.
Field : r=<repeat interval> <active duration> <list of offsets from start time >
Subfield : repeat interval: Repeat interval
active duration: Active duration
list of offsets from start time: List of offsets from the start time.

z (Time Zones) (Description: Optional)

Content : Specifies a time zone
Field : z=<adjustment time> <offset> <adjustment time> <offset>
Subfield : adjustment time: Adjustment time from the standard time
offset: Offset from the start time

a (Attributes)

Content : Specifies an attribute
Field : a=<attribute>
a=<attribute>:<value>
Subfield : attribute: Attribute name
value: Attribute value

The main attributes used are as follows:

tsi (Description: Required)

Content : Specifies a TSI
Field : a=flute-tsi:<integer (TSI value)>

FEC (Description: Optional)

Content : Reference to the declaration of FEC information used
Field : a=FEC:<fec-ref>
Subfield : fec-ref: FEC-declaration identifier

FEC-declaration (Description: Optional)

Content : Declaration of FEC information
Field : a=FEC-declaration:<fec-ref> fec-enc-id=<encode id>[:fec-inst-id=<instance id>]

Subfield : fec-ref: Identifier within SDP for declaration of FEC information
 encode id: FEC Encoding ID
 instance id: FEC Instance ID (optional)

FEC-OTI-extension (Description: Optional)

Content : OTI specific to the FEC code required for recipients to reconstruct the FEC payload
 Field : a=FEC-OTI-extension:<fec-ref> <oti-extension>
 Subfield : fec-ref: Identifier within SDP for declaration of FEC information
 oti-extension: Object Transmission Information specific to the FEC code
 BASE64 format

m (Media Announcements) (Description: Required)

Content : Detailed media information
 Field : m=<media> <port>/<number of port> <transport> <fmt list>
 Subfield : media: Media type (Set to “application” in a FLUTE session)
 port: Port number used
 number of port: Number of ports used
 transport: Transmission protocol (Set to “FLUTE/UDP” in a FLUTE session)
 fmt list: List of payload types (Set to “0” in a FLUTE session)

(1) Extension of SDP

The following parameters are added to the Media description to allow a receiver to secure the storage capacity required to store the contents.

StorageDemands (Description: Required)

Content : Capacity of the storage device required for the storage of content
 Field : a=storageDemands: <size>
 Subfield : size: Storage capacity (in units of kilobytes)

The Media description is further extended with the following parameters in order to add information about the version and broadcast schedule of the file object transmitted.

TransmissionSchedule (Description: Optional)

Content : Information about an object transmitted (Version, Broadcast schedule)
 Field : a=transmissionSchedule:<content-location> [version=<version>
 number=<sendnumber> <schedule>]
 Subfield : content-location: Content location
 version: Version of object transmitted
 sendnumber: Order of transmission
 schedule: Transmission start and end times
 Transmission start time <YYYY-MM-DD·Thh:mm:ss+hh:mm>
 Transmission end time <hh:mm:ss+hh:mm>
 ※: Information about objects transmitted may be described together. In this case, each object name is omitted from the content-location and a

representative path name is described. (Example:
a=transmissionSchedule:/image/ version=201208010900 number=1
2012-08-08T01:00:00+09:00 01:01:00+09:00 number=2
2012-08-08T02:00:00+09:00 02:01:00+09:00 number=3
2012-08-08T03:00:00+09:00 03:01:00+09:00).

However, if the operation is performed so that a receiver does not receive anything from the FLUTE session, and all unit contents are obtained by the repair of stored contents, no object name can be omitted.

※: +hh:mm is time zone offset.

※: Parameters within [] are optional.

(2) Parameters of SDP operation

The parameters of SDP operation are shown below.

Table 2-13: Operation of SDP Parameters

Parameter name	Notation	Use	Parameter value
Protocol Version	v=	○	v=0
Origin	o=	○	o = example.com< Unix time format> < Unix time format> IN IP4 ***.***.***.***
Session Name	s=	○	(When transmitting metadata for program transmission A) · Refer to this Volume Section 4.2.2.2.2 (3) (When transmitting a content) s= < CRID of a content transmitted >
Session and Media Information	i=	—	—
URI	u=	—	—
Connection Data	c=	○	c=IN IP4 ***.***.***.***
Bandwidth	b=	—	—
Times	t=	○	(When transmitting metadata for program transmission A)) t=0 0 (When transmitting a content) t=<broadcast start time (Unix time format)> <broadcast end time (Unix time format)>
Repeat Times	r=	—	—
Time Zones	z=	—	—
TSI	a=flute-tsi	○	a= flute-tsi:<tsi value>
FEC	a=FEC	—	FEC information is obtained from FLUTE packet information. FEC information of SDP is not used.
FEC-declaration	a=FEC-declaration	—	
FEC-OTI-extension	a=FEC-OTI-extension	—	
Media Announcements	m=	○	m=application <port number> FLUTE/UDP 0
StorageDemands	a=storageDemands	○	a=storageDemands:<size>

TransmissionSchedule	a=transmissionSchedule	○	a=transmissionSchedule<content-location> version=<version> number =<sendnumber> <schedule>
----------------------	------------------------	---	--

–: not used

(3) TSI

The ranges of TSI values for a FLUTE session to send an ECG metadata, and for a FLUTE session to send a content, are shown below.

Table 2-14: Range of TSI

Use	Range	Remarks
FLUTE session for transmitting an ECG metadata	1 – 40	
FLUTE session for transmitting a content	41 – 4294967295	

2.1.5.4.3 Associated delivery procedure description

The Associated Delivery Procedure Description specifies the action taken by a receiver after a content is transmitted by storable broadcasting. It includes the procedures for repairing a defective packet detected in a FLUTE session and the procedures for reception report performed on the completion of content reception. The Associated Delivery Procedure Description is described in an associated Procedure Description element of UserServicesDescription.

2.1.5.4.4 Reception report procedure in Associated Delivery Procedure Description

A receiver can post a reception report according to the postReceptionReport procedure of an Associated Delivery Procedure Description. However, a reception report is optional and has product-specific features; therefore, this technical report does not provide any XML schema or semantics for the reception report.

The type of reception report is specified by a reportType element of a postReceptionReport, and report(s) are posted if the corresponding bit(s) are set. For example, if all the bits 1 through 4 are ON, all reports must be posted.

2.1.5.4.5 Location solution of transmission control metadata

Location solving of transmission control metadata is performed in the following manner.

The location of a transmission control metadata is solved by obtaining a content reference identifier described after

/TVAMain/ProgramDescription/ProgramInformationTable/ProgramInformation/@programId in a metadata (program information element) that indicates a unit content of storable broadcasting, and obtaining a URI described after

/TVAMain/ProgramDescription/ProgramLocationTable/OnDemandService/OnDemandProgram/ProgramURL in another metadata (program location information element) that describes the

same content reference identifier.

(1) Location solution within a receiver

The storage method within a receiver is dependent on the implementation. However, the location must be uniquely solved from the ProgramURI value in the metadata.

For example, the storage location of transmission control metadata is determined from the content reference identifier (CRID), and the transmission control metadata is placed under its directory. In this case, the path information of the transmission metadata is not required, and only the file name is effective. When a storage location of transmission control metadata is determined by the CRID, the transmission control metadata is generated for each CRID, and the CRID is thus treated as the logical location (directory) of the storage location of transmission control metadata.

(Example) CRID: crid://example.com/fc/12424

ProgramURI: http://example.com/isdb-tmm/fccontrol/c00123.fci (※)

Storage location for transmission control metadata: /example.com/fc/12424/

Storage location of transmission control metadata (c00123.xfci):

/example.com/fc/12424/c00123.fci

※: The underlined path information is deleted and only the file name is made effective.

2.1.5.4.6 Storage control of transmission control metadata

The storage of transmission control metadata (storage method, storage directory, etc.) is dependent on the receiver implementation. However, the valid period of stored transmission control metadata is controlled in accordance with Volume 10, Section 7.3.1 “Valid period of metadata”.

(1) Adding / Updating transmission control metadata

By referring to the version information in the transmission control metadata (equivalent to the fragmentVersion described in Volume 10, Section 4.4.2 “Operation of fragmentVersion”), a receiver is allowed to identify logical addition / update of the transmission control metadata.

If the version information of the transmission control metadata is newly obtained, the receiver determines it to be the addition of transmission control metadata and can take action for an additional storage process. On the other hand, if the version of the transmission control metadata is later than that of the previously obtained version information, the receiver can take action to update the transmission control metadata.

(2) Deleting transmission control metadata

If a certain metadata is deleted, the transmission control metadata linked to it through the CRID of the transmission control metadata must be deleted. For the deletion of metadata, refer to Volume 10, Section 7.3 “Deleting Metadata”.

2.1.6 Operation of AL-FEC transmission

To enhance the data loss tolerance of the transmission line, transmission data is made redundant. Transmission data is divided into multiple source symbols, from which a parity symbol is generated by the AL-FEC coding. The source symbols and the parity symbol are combined to form encoding symbols. Any source symbol lost on the transmission line can be restored from the parity symbol and the source symbols that were successfully received.

In multimedia broadcasting, the algorithm shown in Table 2-15 below is used for AL-FEC encoding.

Table 2-15: FEC Encoding Algorithm

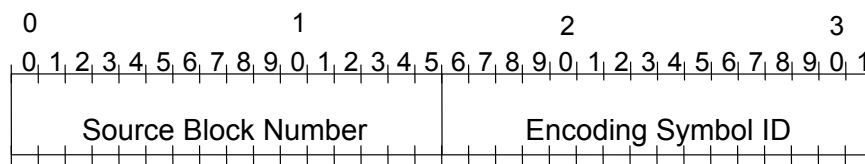
FEC Encoding ID	FEC encoding algorithm
0	Compact No-Code FEC
3	LDPC Staircase Codes

The FEC encoding algorithm used for transmission between a transmitter and a receiver is posted to the receiver. This guideline provides the operation of the AL-FEC encoding algorithms for the Compact No-Code FEC (FEC Encoding ID 0) specified in RFC 3695 and the LDPC-Staircase Codes (FEC Encoding ID 3) specified in RFC 5170.

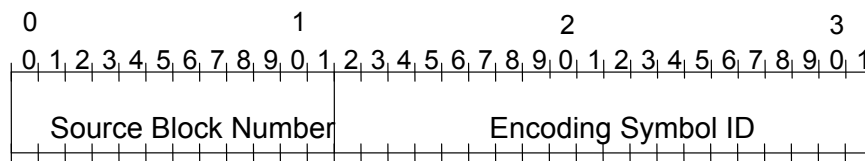
The FEC encoding algorithm may be changed on an object-by-object basis within a FLUTE session. For example, it is desirable to use the Compact No-Code FEC for small FDT instances, text files, and other objects with a small transmission data size, and the LDPC-Staircase Codes for music, video, and other objects with a large transmission data size.

2.1.6.1 FEC payload ID

A sequence number of an encoding symbol is stored in a FEC payload ID in the FLUTE header. In the case of Compact No-Code FEC, the FEC payload ID is composed of a 16-bit Source Block Number and a 16-bit Encoding Symbol ID. In the case of LDPC code, it is composed of a 12-bit Source Block Number and a 20-bit Encoding Symbol ID, which allows a larger block size. The Source Block Number is assigned a unique value for each of the blocks into which a transmission file is divided. The Source Block Number starts from 0 and is incremented by 1, from the top to the consecutive Blocks in order. The Encoding Symbol ID carries the encoding symbol number of the encoding symbol in the Block specified by the Source Block Number. In the case of a Compact No-Code FEC, if there are k partitioned symbols, IDs 0 through $k-1$ are assigned to the top and consecutive symbols. In the case of an LDPC code, for the Encoding Symbol ID, if n encoding symbols are generated from k source symbols, IDs 0 through $k-1$ are assigned to the top and consecutive source symbols, and IDs k through $n-1$ are assigned to the parity symbols in the order in which they were generated.



FEC payload ID (Compact No- Code FEC)



FEC payload ID (LDPC-Staircase)

Fig. 2-16: FEC payload ID

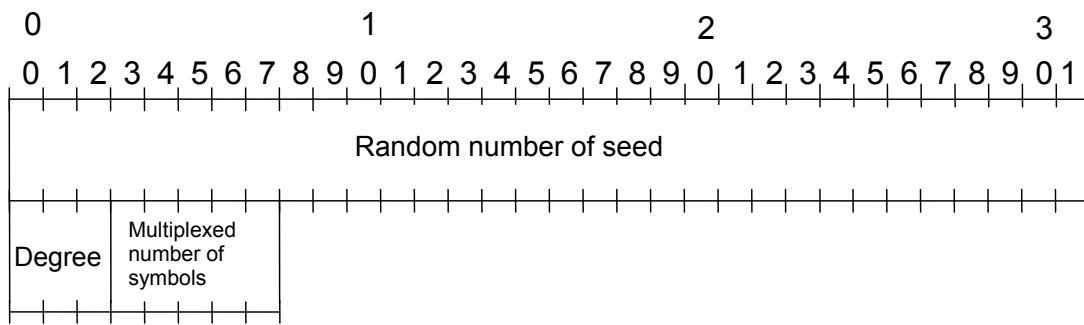
2.1.6.2 LDPC-Staircase Codes

When the LDPC-Staircase Codes algorithm is used, the information that needs to be shared between the sender and receiver is stored in an FDT of FLUTE as an FDT instance. Table 2-16 below shows the FDT instances as common parameters.

Table 2-16: FDT Instances as Common Parameters

FEC-OTI-FEC-Encoding ID	FEC encoding algorithm ID (3)
Transfer-Length	Length of the file transferred (bytes)
FEC-OTI-Encoding-Symbol-Length	Symbol length (bytes)
FEC-OTI-Maximum-Source-Block-Length	Maximum number of source symbols within one source block
FEC-OTI-Max-Number-of-Encoding-Symbols	Maximum number of encoding symbols within one source block
FEC-OTI-Scheme-Specific-Info	Random number seed, degree, multiplexed number of symbols

The FEC-OTI-Scheme-Specific-Info contains specific parameters by each FEC encoding algorithm. In the case of LDPC-Staircase Codes, a random number seed, degree, and multiplexed number of symbols are contained. These parameters are contained in the 5-byte field as shown in Table 2-17 below, and are transformed by Base64 encoding into a character string, which becomes the value of the FEC-OTI-Scheme-Specific-Info.



FEC-OTI-Scheme-Specific-Info

Fig. 2-17: FEC-OTI-Scheme-Specific-Info

The random number seed is specified in the 32-bit field, and it is used to generate a series of random numbers when a parity-check matrix is created.

The degree, which is specified in the 3-bit field, indicates how many 1s are placed in each column of the parity-check matrix (left). The number of 1s subtracted by 3 is stored in this 3-bit field.

The multiplexed number of symbols, which is defined in the 5-bit field, indicates how many symbols are multiplexed in one packet. In this guideline, since one symbol is carried in one packet, the multiplexed number of symbols is set to 1.

(1) Random number generation

For the LDPC codes, a series of pseudo-random numbers is used to create a parity-check matrix, and the same series must be used on both the sender and receiver sides. This section describes the method of generating a series of pseudo-random numbers that is uniquely determined by the random number seed (initial value).

Random numbers are generated by using the Park-Miller-Carta Pseudo Random Number Generator. This random number generator outputs a 31-bit random number. The algorithm of this generator is described below.

```

unsigned long rand31 ()
{
    unsigned long hi, lo;
    lo = 16807 * (seed & 0xFFFF);
    hi = 16807 * (seed >> 16);
    lo += (hi & 0x7FFF) << 16;
    lo += hi >> 15;
    if (lo > 0x7FFFFFFF)
        lo -= 0x7FFFFFFF;
    return (seed = (long) lo);
}

```

Each time the above function is called, one random number is output and the seed value is updated. The seed value used when calling the function for the first time is the seed of the random number, and by using the same seed value on the sender and receiver sides, the same series of random numbers can be used. To transform a 31-bit random number to another random number within an arbitrary range, the following formula is used for scaling:

$$\text{Scaled_value} = ((\text{double})\text{maxv} * (\text{double})\text{rand31}) / 0x7FFFFFFF$$

“maxv” is the maximum value of the range of random numbers. A 31-bit random number is scaled to an arbitrary range, by being multiplied by the maximum value of the range of random numbers, and is then divided by 0x7FFFFFFF.

(2) Parity-check matrix

The parity-check matrix of the LDPC Codes is created from a series of pseudo-random numbers generated as described above. The parity-check matrix is composed of two matrices: the left-side and right-side matrices. The left-side matrix shows the check equations in which the source symbols are involved. The right-side matrix shows the equations in which the parity symbols are involved. In the left-side matrix, 1s are inserted into the matrix elements chosen by using the series of random numbers, and the number assigned by the degree or more 1s are inserted into each column and each row. The right-side matrix is created by inserting 1s into the elements (i-1, i) of the unit matrix. An example of the parity-check matrix for the degree of 3 is shown below.

$$\begin{array}{c}
 \text{LDPC-Staircase Codes check matrix} \\
 \left(\begin{array}{cc|cccc}
 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
 \end{array} \right) \begin{array}{l}
 \left(\begin{array}{c}
 s_1 \\
 s_2 \\
 s_3 \\
 s_4 \\
 s_5 \\
 s_6 \\
 p_1 \\
 p_2 \\
 p_3 \\
 p_4 \\
 p_5 \\
 p_6
 \end{array} \right) = 0
 \end{array}
 \end{array}$$

Left-side check matrix
Right-side check matrix

This example shows a parity-check matrix when the number of source symbols is 6 and the number of parity symbols is 6. The check equation associated with the third row of this parity-check matrix is $s_2+s_4+s_5+p_2+p_3=0$, where s and p denote the source symbol and parity

symbol, respectively, and the suffix denotes the number of each symbol.

The parity-check matrix is created by a series of pseudo-random numbers generated from a random number seed shared between the sender and the receiver. The algorithm for creating a parity-check matrix is given below.

```

void left_matrix_init(int k, int n, int N1)
{
    int i, j, h, t, u[N1 * k];

    for(h = N1 * k - 1; h >= 0; h--){
        u[h] = h % (n - k);
    }
    t = 0;
    for (j = 0; j < k; j++) {
        for (h = 0; h < N1; h++) {
            for (i = t; i < N1*k && matrix_has_entry(u[i], j); i++);
            if (i < N1*k) {
                do {
                    i = t + pmms_rand(N1*k-t);
                } while (matrix_has_entry(u[i], j));
                matrix_insert_entry(u[i], j);
                u[i] = u[t]; t++;
            } else {
                do {
                    i = pmms_rand(n-k);
                } while (matrix_has_entry(i, j));
                matrix_insert_entry(i, j);
            }
        }
    }
}

```

The above function `left_matrix_init` is the algorithm to create the left-side matrix of the parity-check matrix. The arguments `k`, `n`, and `N1` denote the number of source symbols, the number of encoding symbols, and the number of 1s inserted into each row and column (degree), respectively. A matrix of size $(n-k) * k$ has already been created and all elements are set to 0, and `matrix_insert_entry(i, j)` sets the element at the *i*-th row and *j*-th column to 1. The function

`matrix_has_entry(i, j)` checks whether the element at the *i*-th row and *j*-th column is already set to 1. `pmms_rand(n)` generates a random number between 0 and *n*-1, using the previously described 31-bit random number generation algorithm.

The above function inserts *N*₁ number of 1s in each column of the left-side matrix. If the number of rows is greater than the number of columns, in some rows, the number of 1s inserted may be fewer than 2. In this case, the function `degree_of_row` is called to check how many 1s are inserted in each row by using the algorithm given below. If the number of 1s is fewer than 2 in any row, 1 is inserted into a randomly chosen element until the number of 1s reaches 2.

```
for (i = 0; i < n-k; i++) {
    if (degree_of_row(i) == 0) {
        j = pmms_rand(k);
        matrix_insert_entry(i, j);
    }
    if (degree_of_row(i) == 1) {
        do {
            j = pmms_rand(k);
        } while (matrix_has_entry(i, j));
        matrix_insert_entry(i, j);
    }
}
```

Although the number of source symbols in each source block is divided almost equally as described in “2.1.5.2 Source block division”, the number of source symbols in posterior blocks is sometimes lower by one than that in anterior blocks. Therefore, the parity-check matrix is created independently for each source block to perform encoding and decoding. The number of source symbols in each source block can be calculated in the same manner as described in “2.1.5.2 Source block division”.

2.1.6.3 UEP (Unequal Error Protection)

In some cases, certain source symbols are of higher importance. In particular, source symbols within the top part of the content are likely to be of higher importance. Meanwhile, when source symbols are being transmitted, certain transmission conditions make it desirable to enhance the data loss tolerance for specific symbols. To deal with these cases, the left-side matrix may be configured so that the distribution of 1s is biased toward specific columns on some rows. When using the technique called Unequal Error Protection (UEP) for the LDPC-Staircase Codes, recipients must be notified of the number of rows with a higher density of 1s (UEP rows count), and the leftmost and rightmost column numbers of the column range where the distribution of 1s is set higher, by adding them to the FEC-OTI-Scheme-Specific-Info, which is the FDT instance in the FLUTE session. The extended FEC-OTI-Scheme-Specific-Info is shown in Fig. 2-18 below. The 16-bit UEP rows count field, 20-bit UEP leftmost column number field, and

20-bit rightmost column number field are added to the FEC-OTI-Scheme-Specific-Info.

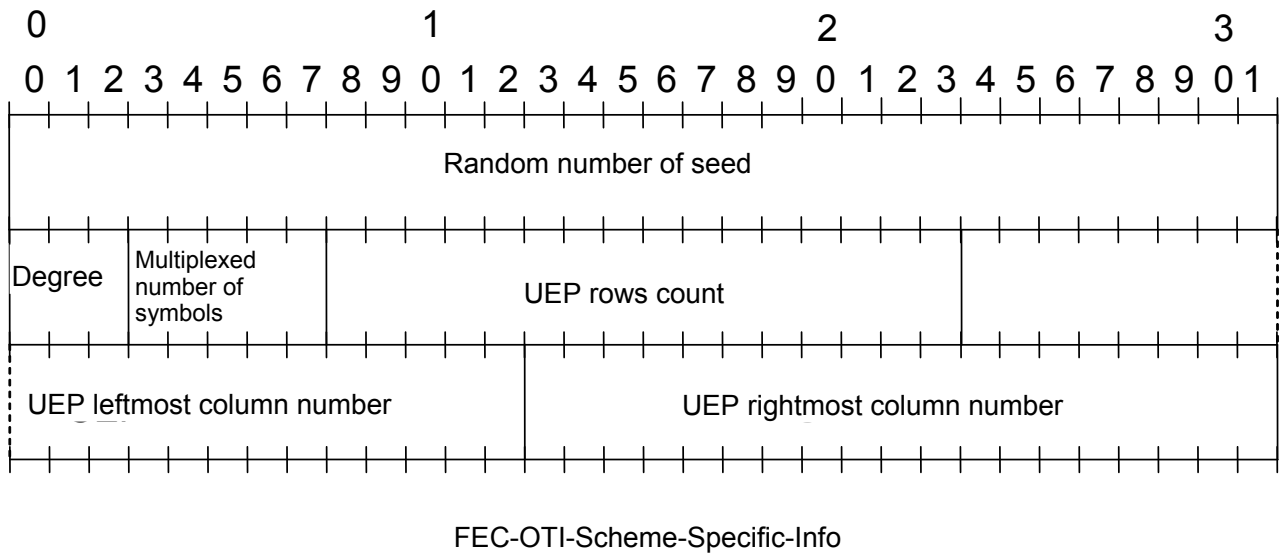


Fig. 2-18: FEC-OTI-Scheme-Specific-Info

When using UEP, the `left_matrix_init` function used for creating a parity-check matrix is extended as shown below. When the receiver has received an FDT instance within a FLUTE session, it specifies whether an ordinary parity-check matrix or a UEP check matrix is to be created depending on the presence of the UEP extension fields in the FEC-OTI-Scheme-Specific-Info.

However, the UEP is not used for the time being.

```

void left_matrix_init(int k, int n, int N1, int m, int low, int high)
{
    int i, j, h, t, u[N1 * k], uep;

    for(h = N1 * k - 1; h >= 0; h--){
        u[h] = h % (n - k - m);
    }
    t = 0;
    for (j = 0; j < k; j++) {
        for (h = 0; h < N1; h++) {
            for (i = t; i < N1*k && matrix_has_entry(u[i], j); i++){
                if (i < N1*k) {
                    do {
                        i = t + pmms_rand(N1*k-t);
                    } while (matrix_has_entry(u[i], j));
                    matrix_insert_entry(u[i], j);
                    u[i] = u[t]; t++;
                } else {
                    do {
                        i = pmms_rand(n - k - m);
                    } while (matrix_has_entry(i, j));
                    matrix_insert_entry(i, j);
                }
            }
        }
    }

    uep = high - low + 1;
    for(h = N1 * uep - 1; h >= 0; h--){
        u[h] = (n - k - m) + (h % m);
    }
    t = 0;
    for (j = low; j < high; j++) {
        for (h = 0; h < N1; h++) {
            for (i = t; i < N1*uep && matrix_has_entry(u[i], j); i++){
                if (i < N1*uep) {
                    do {
                        i = t + pmms_rand(N1*uep-t);
                    } while (matrix_has_entry(u[i], j));
                    matrix_insert_entry(u[i], j);
                    u[i] = u[t]; t++;
                } else {
                    do {
                        i = (n - k - m) + pmms_rand(m);
                    } while (matrix_has_entry(i, j));
                    matrix_insert_entry(i, j);
                }
            }
        }
    }
}

```


The above function inserts N1 number of 1s into each column of the left-side matrix. If the number of rows is greater than the number of columns, in some rows, the number of 1s inserted may be fewer than 2. In this case, the function `degree_of_row` is called to check how many 1s are inserted in each row by using the algorithm described below. If the number of 1s is fewer than 2 in any row, 1 is inserted into a randomly chosen element until the number of 1s reaches 2.

```

for (i = 0; i < n-k-m; i++) {
    if (degree_of_row(i) == 0) {
        j = pmms_rand(k);
        matrix_insert_entry(i, j);
    }
    if (degree_of_row(i) == 1) {
        do {
            j = pmms_rand(k);
        } while (matrix_has_entry(i, j));
        matrix_insert_entry(i, j);
    }
}

for (i = n-k-m; i < n-k; i++) {
    if (degree_of_row(i) == 0) {
        j = low + pmms_rand(high - low + 1);
        matrix_insert_entry(i, j);
    }
    if (degree_of_row(i) == 1) {
        do {
            j = low + pmms_rand(high - low + 1);
        } while (matrix_has_entry(i, j));
        matrix_insert_entry(i, j);
    }
}

```

2.1.7 Operation of UDP/IP transmission

Storable broadcasting can transmit not only contents through the FLUTE session but also arbitrary IP packets. To transmit arbitrary IP packets, the packets are not processed on the FLUTE layer. Various IP broadcast services are provided by assigning a specific IP address and port number to each application.

While IPv4 and IPv6 schemes are described below for FLUTE transmission and arbitrary IP packet transmission, only IPv4 is used for FLUTE transmission for the time being.

2.1.7.1 Packet structure

To transmit an arbitrary file, UDP, IP v4 or IPv6, and IP header compression is used.

2.1.7.2 UDP

The UDP specified in RFC768 is used. The UDP header structure is shown in Fig. 2-19.

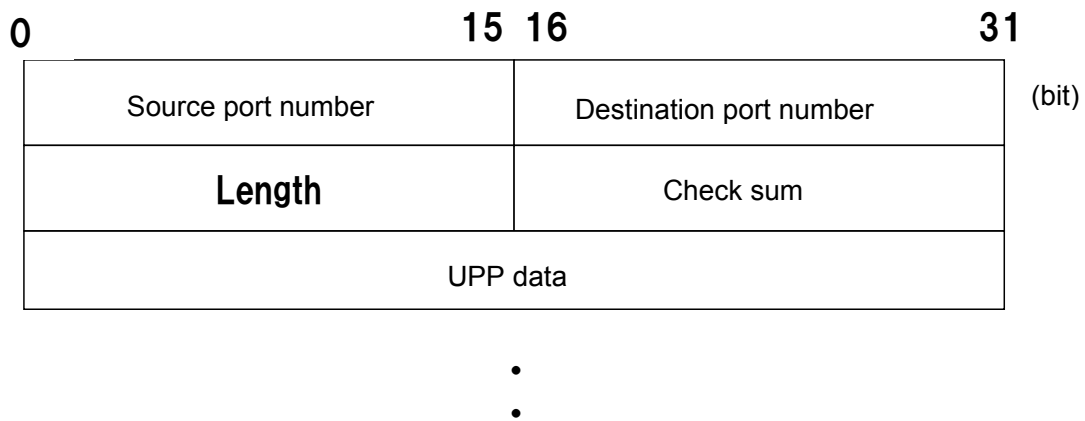


Fig. 2-19: UDP header structure

Each field carries the following information. However, the values listed in Table 2-17 are mapped in a fixed manner assuming that the ROHC always uses the UO-0 header.

- Source port number: Expresses the source port number
- Destination port number: Expresses the destination port number
- Length: The size of the area in octets for the UDP header length plus UDP data length
- Check sum: Not used for IPv4; Based on RFC 768, all bits are set to 0 and transmitted. Used for IPv6; A value calculated based on RFC 768 is set and transmitted.

Table 2-17: Example of Mapping between Addresses, Port Numbers, and CIDs When UO-0 is Always Used

	Source address	Destination address	Source port number	Destination port num.	CID
For FLUTE transmission	0.0.0.0 / 0::0 ※	IP address specified by INT	65527	65527	1
		255.255.255.255 / ff02::1 ※	65528	65528	2
Area freely usable by broadcaster		IP address specified by INT	65529	65529	3
			65530	65530	4
			65531	65531	5

※ Indicates IPv4 / IPv6 addresses, respectively.

(Explanation)

In a partial address space, the source / destination addresses and source / destination port numbers are mapped to CIDs in a fixed manner. By doing so, an entire ROHC header is not transmitted; thus, it allows the broadcaster to set up specific applications freely. One example is shown in Table 2-17.

2.1.7.3 IP

Either IPv4 (RFC791) or IPv6 (RFC2460) is used for the operation.

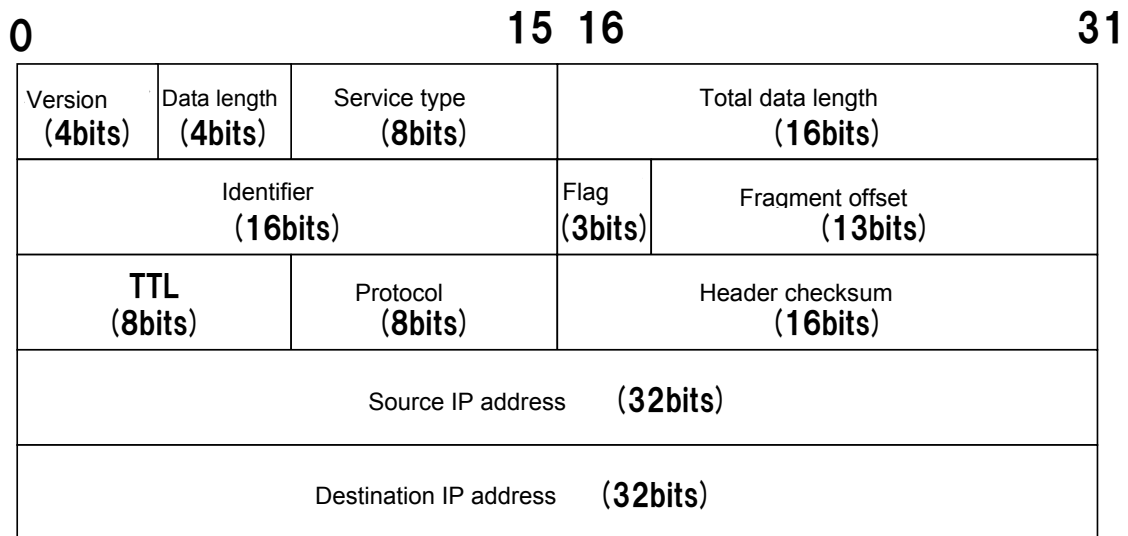


Fig. 2-20: IPv4 header format

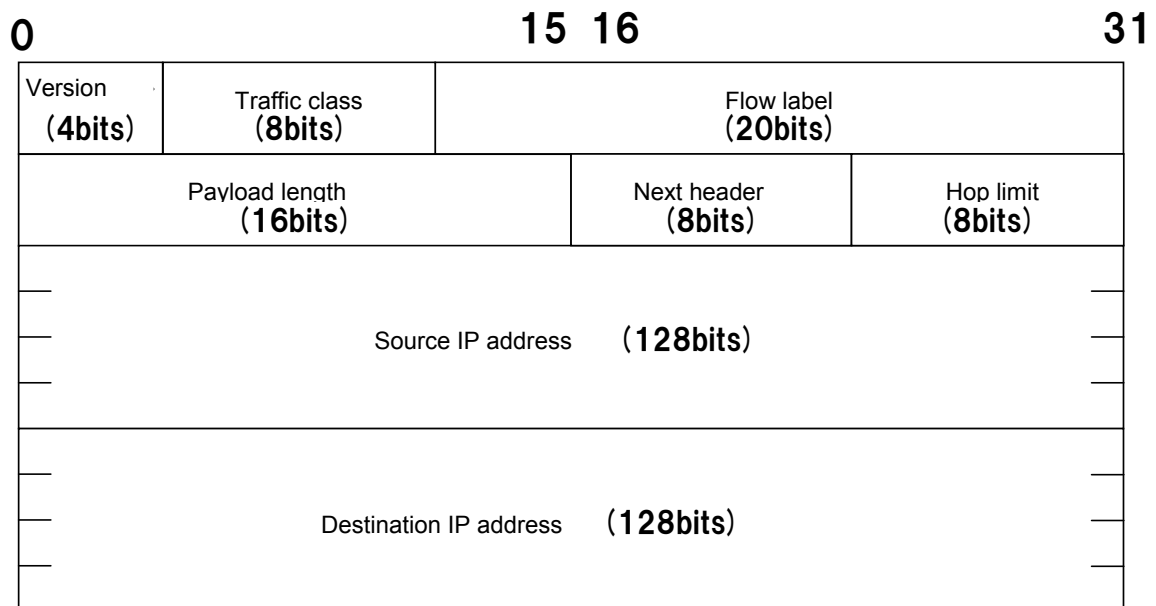


Fig. 2-21: IPv6 header format

For IPv4, the following parameters are used. However, the values listed in Table 2-17 are mapped in a fixed manner assuming that the ROHC always uses the UO-0 header.

- Version: 4 (0100)
- Data length: Set to 5 (0101) (IP header length in units of 32 bits)
- Service type: Set to 0 (0000 0000)
- Total data length: Indicates the total data length, i.e., the sum of the IP header length and payload length

- Identifier: 0000 0000 0000 0000
- Flag: 000
- Fragment offset: 0000 0000 0000 0
- TTL: 1000 0000 (128)
- Protocol: 0001 0001 (17) UDP
- Header checksum: Calculated based on RFC 791
- Source IP address: Specifies the source IP address
- Destination IP address: Expresses the destination IP address

For IPv6, the following parameters are used. However, the values listed in Table 2-17 are mapped in a fixed manner assuming that the ROHC always uses the UO-0 header.

- Version: 6 (0110)
- Traffic class: Set to 0(0000 0000)
- Flow label: 1(0000 0000 0000 0000 0001)
- Payload length: Indicates the payload length in units of 32 bits
- Hop limit: 1000 0000 (128)
- Next header: 0001 0001 (17) UDP
- Source IP address: Specifies the source IP address
- Destination IP address: Expresses the destination IP address

An IP packet sent out may be duplicated or lost on the transmission line. Therefore, the receiver must operate properly even if an IP packet received is duplicated or lost.

2.1.8 Operation of ROHC transmission

A receiver is required to operate assuming three types of IP packets to be transmitted: (1) uncompressed, (2) ROHC UDP U-mode, and (3) ROHC UDP U-mode (always with the UO-0 header). The receiver distinguishes the packet type based on the Packet Type field of the ULE.

The receiver selects the packet type from the above-mentioned (1) ~ (3).

When using CIDs 0x0001 through 0x0005, the receiver is operated in ROHC UDP U-mode, with the mapping of the destination port numbers as shown in Table 2-17.

2.1.8.1 Uncompressed transmission

The UDP header and IP header as respectively described in 2.1.7.2 and 2.1.7.3 are added to every packet and then transmitted. It should be noted that the ROHC uncompressed (profile 0x0000) specified in RFC 3095 is not used.

2.1.8.2 ROHC UDP U-mode transmission

Based on RFC 3095, transmission is performed in ROHC UDP U-mode. The header type is selected depending on the packet error condition from: IR, IR DYN, and UO-0.

i) IR packet

The IR packet format is defined as follows. Details of the fields will be provided later in this section.

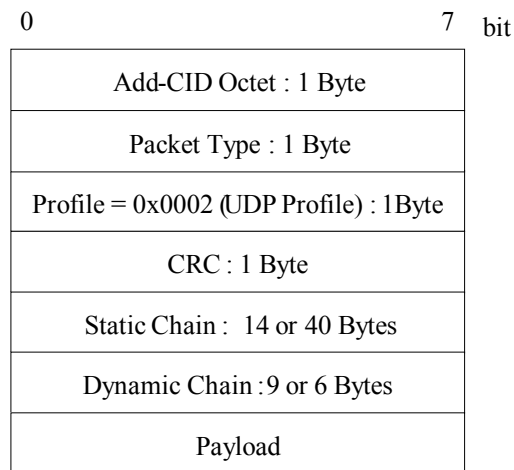


Fig. 2-22: IR packet

ii) IR-DYN packet

The IR packet format is defined as follows. Details of the fields will be provided later in this section.

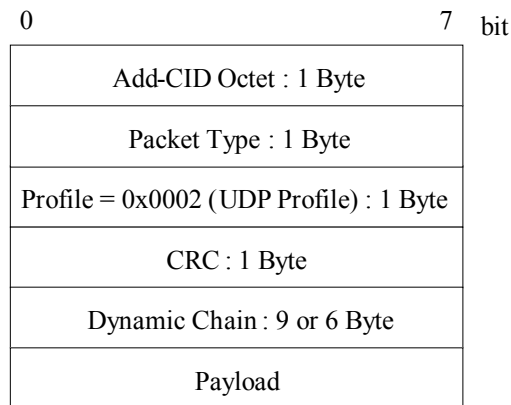


Fig. 2-23: IR-DYN packet

iii) UO-0 packet

The UO-0 packet format is defined as shown in Fig. 2-24 for IPv4, and as shown in Fig. 2-25 for IPv6. Details of the fields will be provided later in this section.

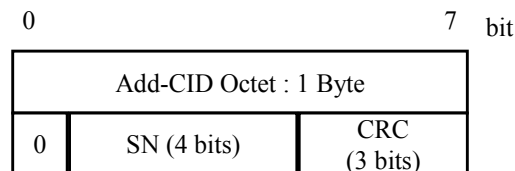


Fig. 2-24: UO-0 packet (IPv4)

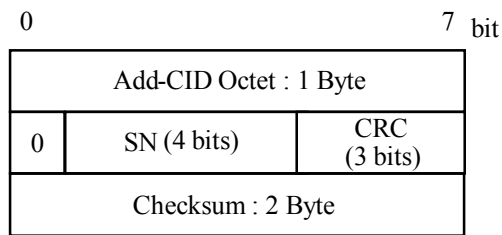


Fig. 2-25: UO-0 packet (IPv6)

Add-CID octet field

The Add-CID octet field is defined in the Small CID configuration as shown in Fig. 2-26. When using CIDs 0x0001 through 0x0005, ROHC UDP U-mode (always use the UO-0 header) is used. The other area is used for multiplexed transmission at the IP level.

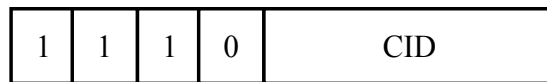


Fig. 2-26: Add-CID octet

Packet Type field

The Packet Type field in the IR packet is defined and operated according to Fig. 2-27. The D flag is always set to “1” (dynamic chain always exists).

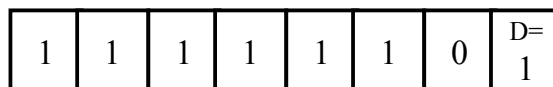


Fig. 2-27: Packet Type field (IR packet)

The Packet Type field in the IR-DYN packet is defined and operated according to Fig. 2-28.

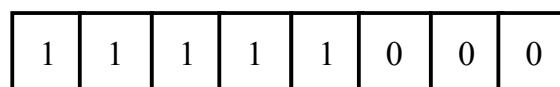


Fig. 2-28: Packet Type field (IR-DYN packet)

SN field

The SN field is operated according to RFC 3095.

CRC field

The CRC field is operated according to RFC 3095.

Static chain

The static chain is operated according to Fig. 2-29 and Fig. 2-30.

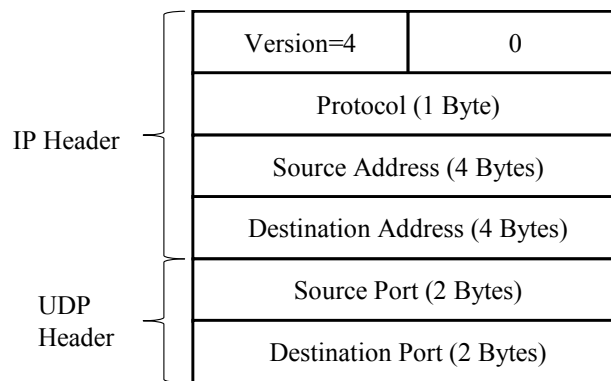


Fig. 2-29: Static chain (IPv4)

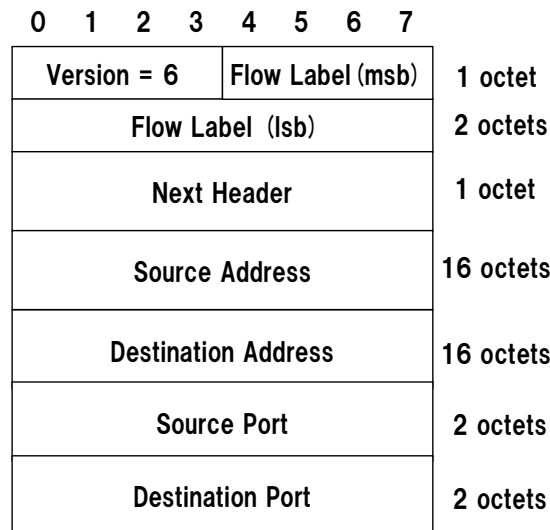


Fig. 2-30: Static chain (IPv6)

IPv4 and IPv6 are operated using the following values.

- ✓ Protocol field
Set to UDP (17)
- ✓ Source / Destination address fields
Operated according to Section 2.1.7.3
- ✓ Source / Destination port fields
Operated according to Section 2.1.7.2

Dynamic chain

The dynamic chain is operated according to Fig. 2-31 for IPv4, and according to Fig. 2-32 for IPv6. For IPv4, the Checksum field in the UDP header is not used; and the Type of Service field, Time-to-Live field, and IP-Identification field are operated according to Fig. 2-20. For IPv6, the Checksum field in the UDP header is operated according to Fig. 2-19; and the Traffic Class field

and HOP Limit field are operated according to Fig. 2-21.

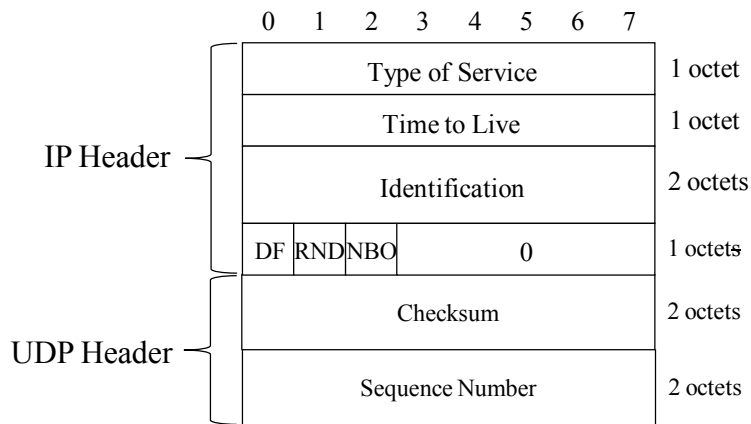


Fig. 2-31: Dynamic chain (IPv4)

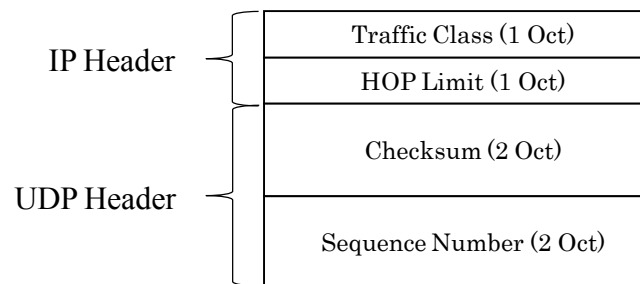


Fig. 2-32: Dynamic chain (IPv6)

IPv4 is operated using the following values.

- DF (Don't Fragment) flag: 0
- RND flag: Set to "0".
- NBO flag: Set to "0".

[Reference] Guidelines for the transmission frequency of IR packets and IR-DYN packets, and the repair process when receiving IR packets

The IR and IR-DYN packets are used for the mapping of the CIDs (Context ID) used for identifying the IP packet stream and the values essentially contained in the IP and UDP headers. Considering those receivers that enter the area in the middle of contents transmission and packets that may be lost due to the radio transmission conditions, it is recommended to transmit the IR packet and IR-DYN packet at the following frequencies:

IR packet: When the content transmission is started and every 10 seconds (TBD)

IR-DYN packet: Every 10 seconds (TBD)

In addition, if an IR-DYN or UO-0 packet is received before receiving an IR packet, it is recommended to hold the received packet for a while and repair the IR-DYN or UO-0 packet that has the same CID when an IR packet is received.

2.1.8.3 ROHC UDP U-mode transmission (always use UO-0 packets)

For all packets, IPv4 packets are transmitted with the smallest ROHC header configuration shown in Fig. 2-24 (without IP-ID or UDP Checksum), and IPv6 packets are transmitted with the smallest ROHC header configuration shown in Fig. 2-25 (with UDP Checksum).

(Explanation) Fundamentally, the ROHC sender makes dynamic transitions between three states (IR, FO, and SO) according to the radio link conditions, etc.; and it initializes and updates the Context using the IR packet and IR-DYN packet, and transmits compressed packets using the UO-0 packets. If multiplexed contents transmission at the IP level is not used, the operation can be performed with UO-0 packets only. In this case, the source/destination addresses and source/destination port numbers are uniquely mapped to the CIDs as shown in Table 2-17.

If UO-0 packets are always used for transmission, it is assumed in the operation that the IR packet has set the fields to the following values:

- Values assigned to each field
 - Add-CID: Operated with small CID according to Fig. 2-26
 - Profile: ROHC UDP (0x0002)
 - Protocol: UDP
 - IP header: In accordance with Section 2.1.7.3
 - UDP header: In accordance with Section 2.1.7.2

[Reference] Overhead length

The overhead length for each ROHC transmission scheme is as follows:

Table 2-18: Overhead Length

	Uncompressed	ROHC (IR Packet)	ROHC (IR-DYN Packet)	ROHC (UO-0)
UDP header (IPv4 / IPv6)	8 bytes	8 bytes	4 bytes	0 / 2 bytes
IP header (IPv4 / IPv6)	20 / 40 bytes	15 / 38 bytes	5 / 2 bytes	0 bytes
ROHC-relevant header	0 bytes	4 bytes	4 bytes	2 bytes
Total (IPv4 / IPv6)	28 / 48 bytes	27 / 50 bytes	13 / 10 bytes	1 / 4 bytes

2.1.9 Operation of ULE transmission

2.1.9.1 Encapsulation of IP datagrams

In order to transmit an IP datagram over an MPEG-2 TS packet, the IP datagram is encapsulated using ULE (Unidirectional Lightweight Encapsulation).

The ULE header format is shown below.

Destination flag (1)	Data length (15)	Packet type (16)	Data (8×N)	CRC (32)
-------------------------	---------------------	---------------------	---------------	-------------

Fig. 2-33: ULE header format

- Destination flag: 1 (The destination MAC address is not used.)
- Data length: This field carries the byte length of the data following the Packet Type field.
- Packet Type: This field carries the type of packet stored in the data field, as specified in IETF RFC 4326.
 - IPv4: 0x0800
 - IPv6: 0x86DD
 - ROHC: 0x22F1
- CRC shall be made in accordance with IETF RFC 4326.

An IP datagram encapsulated using ULE is divided into 184-byte-long segments, each of which is stored in an MPEG-2 TS packet having the same PID.

2.1.10 Operation of communication network transmission

In the storable broadcasting system, if any content is not received properly from the broadcast waves, the missing packet may once again be obtained over the communication network as described in Volume 12, Chapter 2.

2.1.11 Effective content transmission method

When storable broadcasting delivers a content, since the content data must be perfectly transmitted, the data delivered over the broadcast waves is given a redundancy to enhance the error tolerance and hence to achieve efficient transmission. For a content with a small data size, the error tolerance can be enhanced by transmitting the same data several times; on the other hand, for a content with a large data size, the error tolerance is drastically enhanced, not by repeating the same data transmission, but by using Forward Error Correction (FEC) to transmit data that is recoverable.

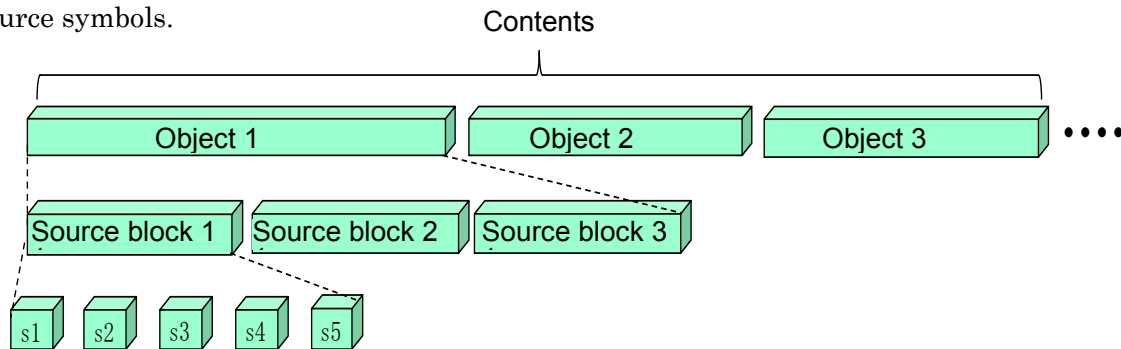
In addition, by transmitting the data in multiple time zones, the receiving success rate of moving receivers can be improved. For example, if data transmission is divided into three sessions and transmitted in the morning, in the daytime, and at night, both the receivers that are outside the area in the morning and daytime but inside the area at night, and the receivers that are inside the area in the daytime but outside the area in the morning and at night could perfectly receive the data. In addition, even if a receiver is in an area with poor reception in the

morning, in the daytime, and at night, if the combined data from the morning, daytime, and night transmissions covers the required amount of data, the whole data can be restored.

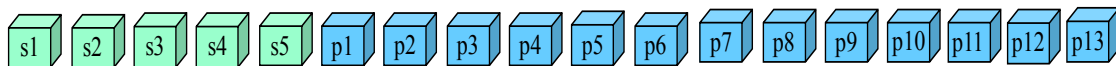
In the following two examples, redundancy data is generated by using the FEC and the data is broadcasted three times.

(Example 1)

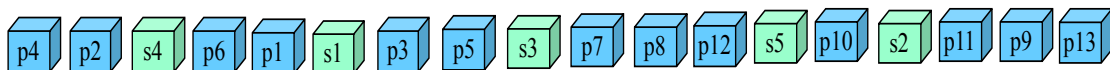
(i) An object is divided into source blocks, and each source block is divided into multiple source symbols.



(ii) Parity symbols are generated, and encoding symbols are created with 3.6 times the source symbols.

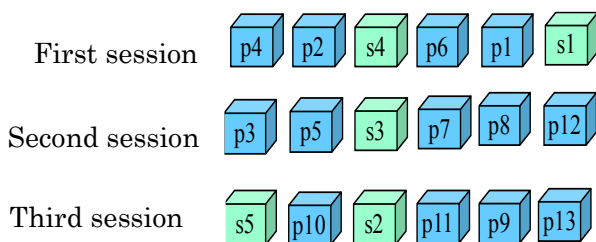


(iii) Interleaved in random order



(iv) Divided into three segments, and individually transmitted in three different time zones

* In each time zone, encoding symbols with 1.2 times the source symbols are transmitted.



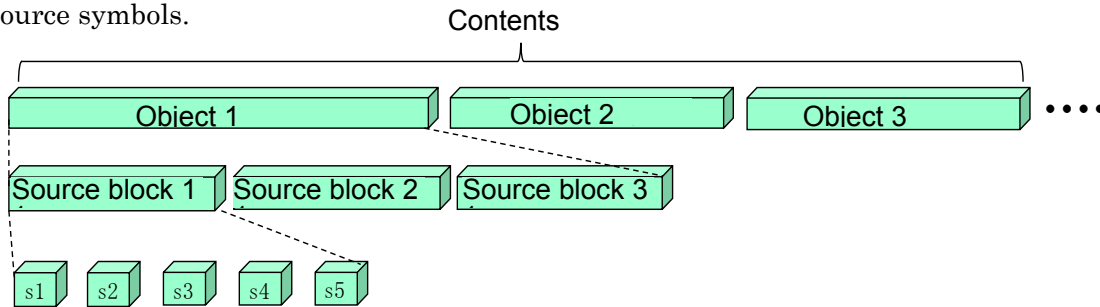
First, an object constituting the content is divided into multiple source blocks; in this example, it is divided into 3 source blocks. Each source block is then divided into multiple source symbols (s1 ~ s5). From these source symbols, parity symbols (p1 ~ p13) are generated by FEC encoding. In this example, 13 parity symbols are generated from 5 source symbols, and thus the number of encoding symbols is 18, which means that the encoding symbols are generated with

3.6 times the source symbols. These source symbols and parity symbols are combined to make 18 encoding symbols, which are then interleaved in random order. The interleaved encoding symbols are divided into three segments each consisting of six encoding symbols. Each set of encoding symbols is transmitted at the first, second, and third broadcast. In this example, at each broadcast, the encoding symbols are transmitted with 1.2 times the source symbols. As a result, if a receiver has successfully received all the symbols from the first broadcast, it can perfectly restore the content from only the first broadcast. In the case where the first broadcast has not been received at all, if almost all symbols are received from the second broadcast, the content data can be perfectly restored. Even in the case where the symbols received from the first broadcast are not sufficient to restore the content, if some symbols are received from the second and/or third broadcast, the content can be perfectly restored.

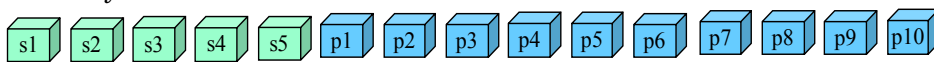
The second example presents a case where the encoding symbols with the same number as the source symbols are transmitted in each broadcast session.

(Example 2)

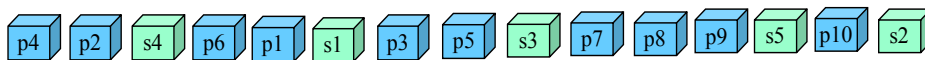
(i) An object is divided into source blocks, and each source block is divided into multiple source symbols.



(2) Parity symbols are generated, and encoding symbols are created with 3 times the source symbols.

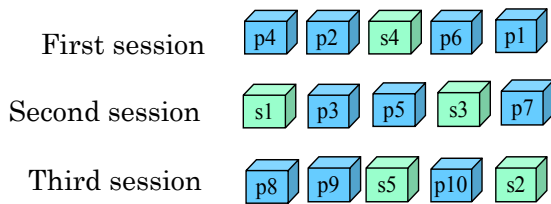


(3) Interleaved in random order



(iv) Divided into three segments, and individually transmitted in three different times zones

* In each time zone, encoding symbols with same number as the source symbols are transmitted.



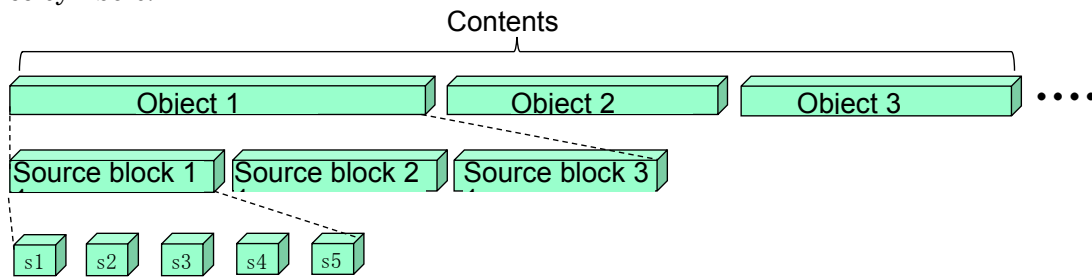
In this case, the total amount of transmission is equal to the object size \times the number of broadcasts. However, even if one broadcast is fully received, decoding may not be completed; thus, it is recommended to transmit encoding symbols with about 1.2 times the source symbols in each broadcast, provided that there is a sufficient transmission capacity.

It is recommended to use these transmission methods when transmitting large-size object data, whereas the Compact No-Code FEC method without FEC encoding is recommended for the transmission of small-size object data.

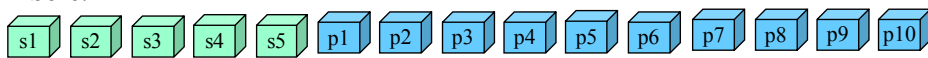
Another example is also presented, where transmission is made with two broadcasts. In this example, about 1.5 times more encoding symbols are transmitted.

(Example 3)

(i) An object is divided into source blocks, and each source block is divided into multiple source symbols.



(2) Parity symbols are generated, and encoding symbols are created with 3 times the source symbols.

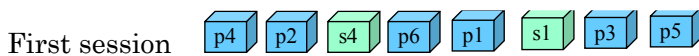


(3) Interleaved in random order



(4) Divided into two segments, and individually transmitted in two different time zones

* In each time zone, encoding symbols with 1.5 times the source symbols are transmitted. If there are excessive symbols remaining after division, they are transmitted in the first session.



If a content (object) to be transmitted is divided into multiple source blocks, encoding symbols are randomly interleaved and transmitted in multiple broadcast sessions by each source block. The symbols shall not be interleaved across the blocks. For each transmitted content (object), the division processing of encoding symbols is made possible by selecting the FEC encoding scheme, and defining the number of broadcasts and the number of multiples that the encoding symbols should be transmitted in a broadcast compared to the source symbols.

In addition, the encoding symbol length and the number of source blocks, as the FEC encoding parameters, also need to be specified for the source block division as described in Section 2.1.5.2. If the FEC encoding uses the LDPC-Staircase Codes, the “degree” and “random number seed” also need to be specified as described in Section 2.1.6.2. These FEC parameters need to be transmitted to and shared with the receiver as an FDT instance as described in Section 2.1.5.3. This FDT instance is transmitted as an object with TOI=0. The FDT instance is not FEC encoded, and only source symbols are repeatedly transmitted. The FDT instance must be transmitted at the beginning of a session (program) followed by a periodic transmission every 1 to 60 seconds. The FDT instance can be transmitted as one XML file for a group of sessions, or as XML files divided into several object groups. If the message size of the FDT instances for all

sessions is 10 Kbytes or less, it is recommended to transmit them in one XML file for each session. If the message size exceeds 10 Kbytes, it is recommended to divide them into object groups and to make each group 10 Kbytes or less, and then to transmit these object groups. (If the message size of the FDT instance of one object exceeds 10 Kbytes, it must not be divided.)

Table 2-19 below shows the recommended parameters of broadcast data division, FEC encoding, and FDT instance transmission.

Table 2-19: Recommended Parameters of Broadcast Data Division, FEC Encoding, and FDT Instance Transmission

(Parameters of broadcast data division)

	Explanation of parameter	Parameter value
Number of broadcasts	Broadcasted in how many time zones?	Set to 1 ~ 3
Symbol length transmitted per broadcast	The number of multiples that the encoding symbols should be transmitted in each broadcast compared to the source symbols.	Set to 1.0 ~ 2.0 times. Set to 1.0 time for Compact No-Code FEC

(Parameters of FEC encoding)

	Explanation of parameter	Parameter value
FEC encoding algorithm	FEC encoding algorithm (Described in 5.1.6)	Choose ① Compact No Code FEC or ② LDPC Staircase codes. If the number of source symbols is 100 or less, ① Compact No Code FEC is recommended.
Encoding symbol length	Length of symbol in bytes	Set to a value within the range of 1 ~ 1200 bytes. For an object of 1200 bytes or less, the symbol size is set to the object size. For an object of 1201 bytes or more, set to the byte count of each segment when the object is divided as equally as possible into segments of the number divided by 1200 bytes.
Source block count	The number of blocks into which an object is divided (Block : a unit of FEC encoding)	Divided into source blocks so that the number of source symbols in each block is as equal as possible and does not exceed the upper limit of 2000. (Refer to 2.1.5.2 Source Block Division.)
Degree of parity	The number of elements in each	Set to 3.

check matrix (in the case of LDPC Codes)	row of the left-side matrix, into which 1s are inserted.	
Random number seed (in the case of LDPC Codes)	A random number to create a parity check matrix, shared between the sender and receiver.	Set to a 31 bit integer.

(Parameter of FDT instance transmission)

	Explanation of parameter	Parameter value
Transmission period	Every how many seconds the FDT instance is transmitted.	Set to a value within the range of 1 ~ 60 seconds (TBD).

2.1.12 Action for content acquisition

An example is presented for a series of receiver actions to acquire a storable broadcasting content.

2.1.12.1 Content acquisition by ECG

(1) Acquisition of metadata including ECG

First, the receiver accesses the NIT to acquire all service_ids that have the service_type of “Multimedia broadcast service” and the service_id of EPG/ECG metadata service (see Volume 7, Section 2.4). Then, by refereeing to the PAT, it acquires the PIDs of the PMTs associated with the service_id within its own network, and accesses each PMT.

The receiver accesses the PID of the ES described in the PMT, and processes the ULE, ROHC, and FLUTE protocols to acquire the metadata. Information about the delivery of contents is extracted from the metadata acquired, and the available contents (storable broadcasting data) acquired by the receiver are displayed for the user.

In addition, the receiver also acquires the metadata through the PATs and PMTs of other network service_ids described in the NIT and acquires the information used for receiving the contents of those service_ids.

(2) Selection of receiving content

Based on the metadata acquired, a content (storable broadcasting data) to be received is selected by the user’s storage reservation command or receiver’s automatic reservation function.

The receiver stores the broadcast start time, service_id, and session_ID (TSI) and stands by until the broadcast start time is reached.

(3) Acquisition of content

The receiver wakes up a short time before the broadcast start time of the selected content; and from the PAT, it acquires the PID of the PMT for the service_id of selected content and accesses the PMT to acquire the PID of the content to be broadcasted.

Once the receiver has received the storable broadcasting data associated with the PID, it processes the ULE, ROHC, and FLUTE protocols to acquire the session_id (TSI). If the TSI matches the value given in the metadata, the receiver identifies it as the selected content and continues to receive the data transmitted with the PID.

If error correction codes are added to the content, the receiver attempts to repair the contents

using the codes while it continues to receive the data. If the repair process is successfully completed for the whole content, the receive process is terminated even in the middle of the process.

(4) Re-acquisition of content

If the whole content was not received in one broadcast session, the receiver attempts to acquire the content at another broadcast start time indicated in the metadata. If all the broadcast start times have already passed, and depending on the conditions, the receiver restores the stored content through communications. For details, refer to Volume 12.

2.2 EPG/ECG Metadata Transmission Method

This section provides detailed explanation on the operation of EPG/ECG metadata transmission.

2.2.1 Definition of terms related to EPG/ECG metadata transmission and main point of metadata transmission operation

The following provides definitions of terms related to the EPG/ECG metadata transmission, and explanation of basic metadata transmission.

2.2.1.1 EPG/ECG metadata

For details, refer to Volume 10, Section 2.1.

2.2.1.2 Real-time broadcasting metadata

Metadata used for real-time broadcasting, including group metadata related to the contents of real-time broadcasting; hereinafter expressed as real-time broadcasting metadata

2.2.1.3 Storable broadcasting metadata

Metadata used for storable broadcasting, including group metadata related to the contents of storable broadcasting; hereinafter expressed as storable broadcasting metadata

2.2.1.4 Program transmission A

Transmission method using the partial reception layer for the transmission of metadata

2.2.1.5 Program transmission B

Transmission method other than that using the partial reception layer for the transmission of metadata

2.2.1.6 FLUTE session for metadata transmission

FLUTE sessions used for program transmission A

Mainly used for the transmission of metadata

2.2.1.7 FLUTE session for content transmission

FLUTE sessions used for the program transmission B

Same as the FLUTE session used for transmitting storable broadcasting contents

2.2.1.8 Transmission control metadata

For details, refer to Section 2.1.5.4 in this volume.

2.2.1.9 FLUTE session open parameters

Parameters used when opening a FLUTE session

The open parameters are divided into common open parameters and individual open parameters. A metadata FLUTE session has no transmission control metadata, and thus the receiver uses these parameters when it opens a metadata FLUTE session. Note that the receiver holds these parameters as fixed parameters.

2.2.1.10 Common FLUTE session open parameters

Parameters used when opening a FLUTE session for metadata transmission and common to all FLUTE sessions

2.2.1.11 Individual FLUTE session open parameters

Parameters used when opening a FLUTE session for metadata transmission and specific to each FLUTE session

2.2.1.12 Transmission group

A group for transmitting metadata

ECG metadata always belongs to a transmission group and is transmitted according to the transmission protocol defined for each group.

2.2.1.13 Receiver

The receiver has a function of reception and viewing operation while moving, including mobile phone, automotive receiver, PDA, etc.

2.2.1.14 Rough concept of metadata transmission operation

Metadata is transmitted by either program transmission A or program transmission B. Due to the use of layer A, the bit rate of program transmission A is lower than that of program transmission B. Therefore, they can be selectively used depending on the priority of the metadata (difference between the current time and the broadcast start time). For example, if the metadata for two days (i.e., 48 hours) from the current time is transmitted with top priority, the metadata for two days (i.e., 48 hours) from the current time is transmitted by program transmission A, whereas the metadata for the third day (72 hours) and later is transmitted by program transmission B.

In addition, regardless of whether the storable broadcasting service exists or not, since program transmission A can be identified by the `service_id` defined in accordance with the guideline provided in Volume 7, Section 7.1.4, it is assumed that the receiver first receives the program transmission A and then, based on the EPG/ECG metadata contained, it acquires program transmission B.

2.2.1.15 Rough concept of metadata transmission operation for program transmission A

Metadata is transmitted using a FLUTE session for metadata transmission. A different content is transmitted depending on the metadata class. The EPG/ECG metadata transmitted and the transmission control metadata are separately encoded in BiM format for each CRID, and no manifest file is defined. In addition, the metadata is transmitted for all service providers together. Therefore, the metadata is not transmitted for each service provider individually.

Table 2-20: Relationship between Metadata Class and Transmission Content

Metadata class	Transmission content	
	EPG / ECG metadata	Transmission control metadata
Real-time broadcast metadata	○	-
Storable broadcasting metadata	○	○

○: File transmitted, -: File not transmitted

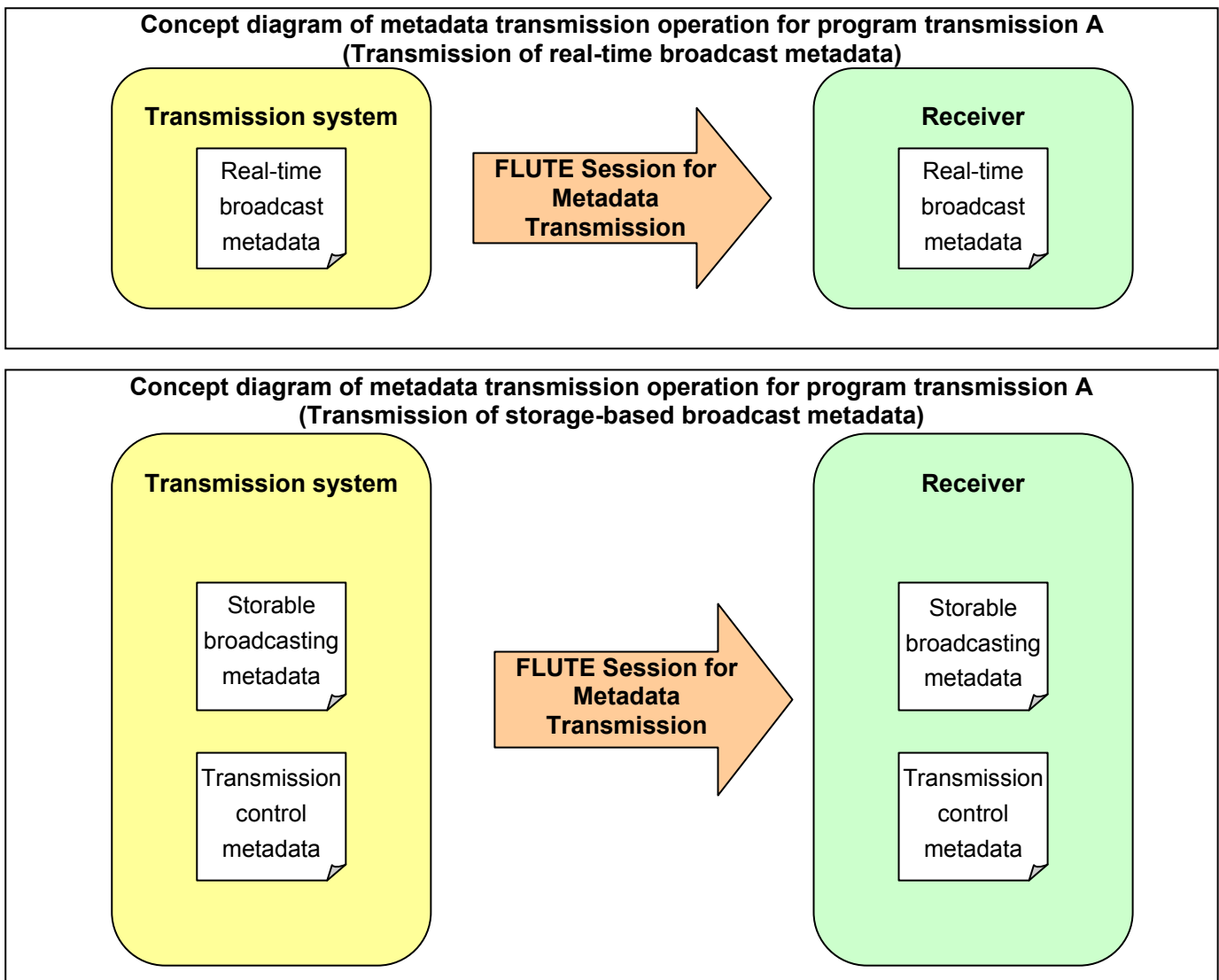


Fig. 2-34: Concept diagram of metadata transmission operation for program transmission A

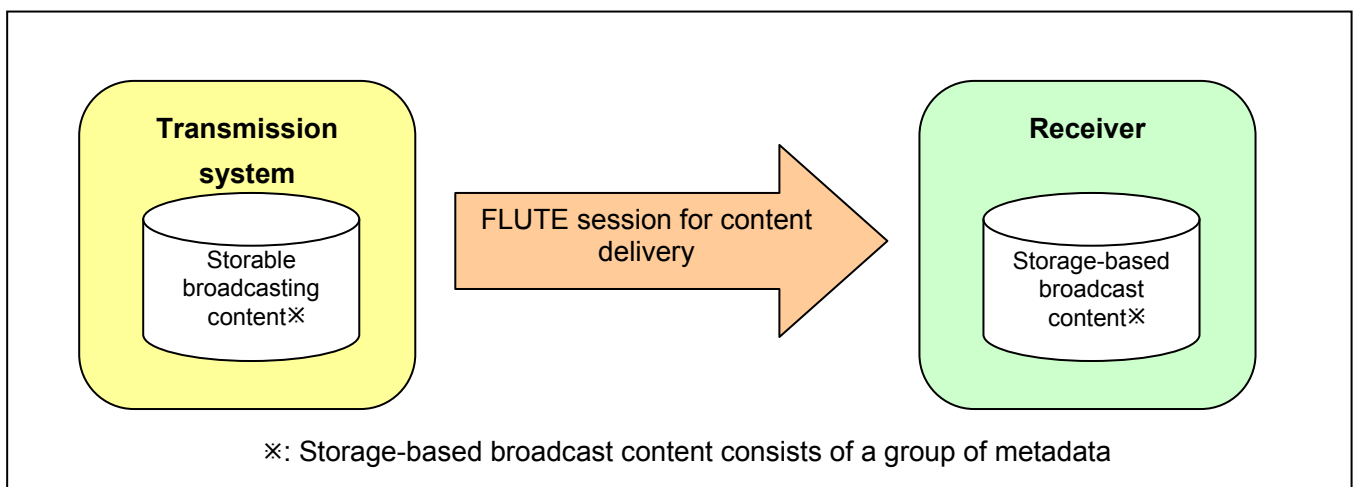
2.2.1.15.1 Rough concept of metadata transmission operation for program transmission B

Metadata is transmitted using a FLUTE session for content transmission. A different content is transmitted depending on the metadata class. The EPG/ECG metadata transmitted and the transmission control metadata are separately encoded in BiM format for each CRID, and no

manifest file is defined. In addition, in program transmission B, metadata is transmitted as a storable broadcasting content.

Table 2-21: Relationship between Metadata Class and Transmission Content

Metadata class	Transmission content	
	EPG / ECG metadata	Transmission control metadata
Real-time broadcast metadata	○	—
Storable broadcasting metadata	○	○



○: Metadata transmitted, —: Metadata not transmitted

Fig. 2-35: Concept diagram of metadata transmission operation for program transmission B

2.2.2 Program transmission A

Metadata is transmitted using a FLUTE session for metadata transmission. Mainly used for the transmission of high-priority metadata with a small time difference between the current time and the program broadcast start time.

2.2.2.1 Metadata transmission model

Metadata is categorized into groups according to the metadata class and the time difference between the current time and the program start time defined in the metadata, and it is repeatedly transmitted by the FLUTE session defined for each group.

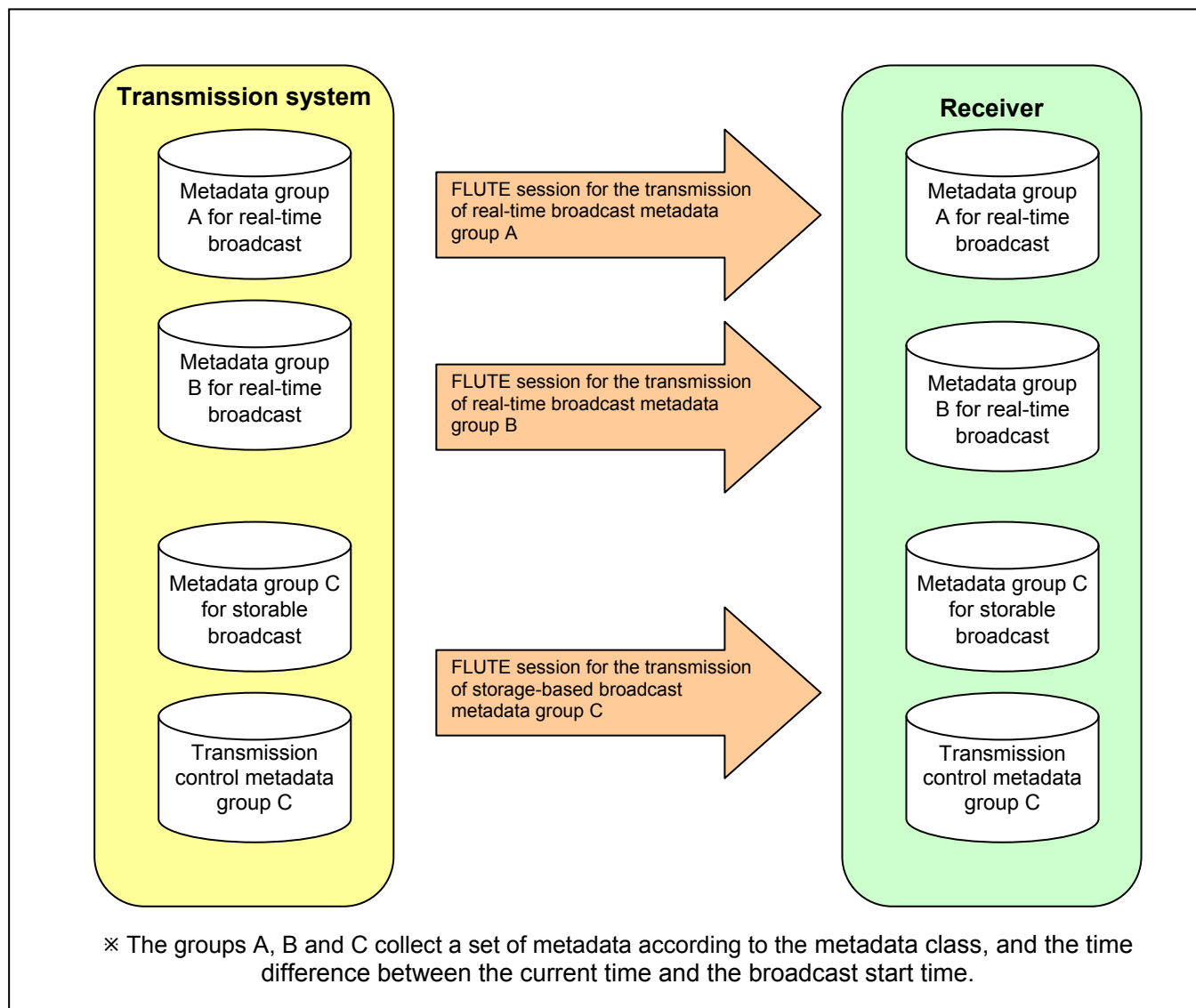


Fig. 2-36: Metadata transmission model

2.2.2.2 Operation of metadata transmission

The following explains the specific operation of metadata transmission.

2.2.2.2.1 Setting metadata transmission group

Metadata is transmitted in a FLUTE session for metadata transmission. While it is possible to transmit all metadata in a single FLUTE session, it is more efficient to set up multiple FLUTE sessions having different transmission speeds and transmit groups of metadata according to the priority of each metadata. Consequently, metadata is grouped according to the

metadata class and the time difference between the current time and the program start time defined in the metadata. An operation example is described below.

(Operation example: Two transmission groups of real-time broadcasting metadata, and two transmission groups of storable broadcasting metadata)

Name of transmission group: Metadata transmission for real-time broadcasting (0 - 9)

Metadata class: Real-time broadcasting metadata

Difference between current time and broadcast start time: Current time ~ current time +9:00

Explanation: Metadata for real-time broadcasts from the current time until 9 hours later

Name of transmission group: Metadata transmission for real-time broadcasting (9 - 48)

Metadata class: Real-time broadcasting metadata

Difference between the current time and broadcast start time: Current time +9:00 ~ current time +48:00

Explanation: Metadata for real-time broadcasts from the current time + 9 hours until the current time + 48 hours

Name of transmission group: Metadata transmission for storable broadcasting (0 - 48)

Metadata class: Storable broadcasting metadata

Difference between the current time and broadcast start time: Current time ~ current time +48:00

Explanation: Metadata for storable broadcasts from the current time until 48 hours later

Name of transmission group: Metadata transmission for storable broadcasting (48 -)

Metadata class: Storable broadcasting metadata

Difference between current time and broadcast start time: Current time +48:00 ~

Explanation: Metadata for storable broadcasts from the current time +48 hours or later

(1) Reference destination of the broadcast schedule defined in metadata

The following shows, for each metadata class, the reference destination of the metadata broadcasting schedule required when grouping metadata transmissions.

Table 2-22: Relationship between Metadata Class and Broadcast Schedule Reference Destination

Metadata class	Reference destination	Remarks
Real-time broadcast metadata	ProgramLocationTable/BroadcastEvent/ PublishedStartTime	
Storable broadcasting metadata	ProgramLocationTable/OnDemandService/ OnDemandProgram/StartOfAvailability	

2.2.2.2.2 Open parameters of the FLUTE session for metadata transmission

The FLUTE session open parameters are explained in this section. There are two categories

of FLUTE session open parameters: those common to all groups and those individually defined for each group. The FLUTE session open parameters are listed below along with their category, Common or Individual.

Table 2-23: List of FLUTE Session Open Parameters for Metadata Transmission

Parameter name	Category	Remarks
Protocol Version	Common	
Origin	Common	
Session Name	Individual	
Session and Media Information	–	
URI	–	
Connection Data	Common	
Bandwidth	–	
Times	Common	
Repeat Times	–	
Time Zones	–	
TSI	Individual	
FEC	–	
FEC-declaration	–	
FEC-OTI-extension	–	
Media Announcements	Common	
TransmissionSchedule	–	

–: Not used

(1) Common FLUTE session open parameters

The FLUTE session open parameters common to all groups are listed below.

Table 2-24: Common FLUTE Session Open Parameters

Open parameter (SDP parameter)	Parameter value	Remarks
Protocol Version	v=0	
Origin	o=nottv.jp <Unix time format> <Unix time format> IN IP4 10.0.0.216	
Connection Data	c=239.255.101.1 c=239.255.101.3	
Times	t=0 0	
Media Announcements	m=application 65527 FLUTE/UDP 0	

(2) Transmission of common FLUTE session open parameters to receivers

The common parameters are fixed and not transmitted to receivers.

(3) Individual FLUTE session open parameters (Session Name)

The naming rules for the Session Name are as follows:

(Naming rules)

“META” + “Metadata class” + “Session number”

+: Represented by the single-byte character “_”

Metadata class: Represented as “ST” for real-time broadcasting metadata, and as “FC” for storable broadcasting metadata

Session number: Session number for each metadata class

An example of the transmission group described in Section 2.2.2.2.1 is shown below.

Table 2-25: Examples of Session Names

Group name	Parameter value	Remarks
Metadata transmission for real-time broadcast (0 - 9)	s=META_ST_1	
Metadata transmission for real-time broadcast (9 - 48)	s=META_ST_2	
Metadata transmission for storable broadcasting (0 - 48)	s=META_FC_1	
Metadata transmission for storable broadcasting (48 -)	s=META_FC_2	

(4) Individual FLUTE session open parameters (TSI)

For each metadata transmission FLUTE session, the TSI is assigned two consecutive values within the range specified for each metadata class. Each time the content of the transmitted metadata is updated, the TSI is also updated. For example, if a metadata transmission FLUTE session is assigned TSI: 1 and 2, a FLUTE session is opened with TSI: 1 for the first transmission. Eventually, if the metadata is updated (addition or deletion of data), the session with TSI: 1 is closed, and another FLUTE session is opened with TSI: 2 to start the transmission of the updated metadata. Then, each time the transmitted metadata is updated, RSI: 1 and 2 are used alternately.

When a receiver detects a change of TSI, it recognizes that the content transmitted in the metadata FLUTE session has been updated. The metadata class and the range of TSI values are listed below.

Table 2-26: Metadata Classes and TSIs

Metadata class	Parameter value	Remarks
Metadata transmission for real-time broadcast	TSI: 1 ~ 20	For 10 sessions
Metadata transmission for storable broadcasting	TSI: 21 ~ 40	For 10 sessions

An example of the transmission group described in Paragraph (1) is shown below.

Table 2-27: Examples of TSI Operation

Group name	Parameter value	Remarks
Metadata transmission for real-time broadcast (0 - 9)	a=flute-tsi:1 a=flute-tsi:2	
Metadata transmission for real-time broadcast (9 - 48)	a=flute-tsi:3 a=flute-tsi:4	
Metadata transmission for storable broadcasting (0 - 48)	a=flute-tsi:21 a=flute-tsi:22	
Metadata transmission for storable broadcasting (48 -)	a=flute-tsi:23 a=flute-tsi:24	

(5) Transmission of individual FLUTE session open parameters to receivers

The individual parameters are fixed and not transmitted to receivers.

2.2.2.3 FDT extension (version information) and update of metadata

As a means of notifying receivers of a metadata update, the version information in the FDT extension is used (details in Section 2.1.5.3.4 (4)). To that end, a unique metadata object identifier in the range of 1 ~ 536870911 is assigned to each metadata (real-time broadcasting metadata, storable broadcasting metadata, transmission control metadata, and group metadata). The validity period of the metadata object identifier is set equivalent to that of the metadata. A metadata object identifier that has become invalid is allowed to be reassigned. In addition, when metadata is updated, the version identifier in the FTD extension is incremented by 1 and transmitted.

By using the metadata object identifier and the version information in the FDT extension (Version-ID-Length), the receiver recognizes the metadata update from the TOI of the FDT instance.

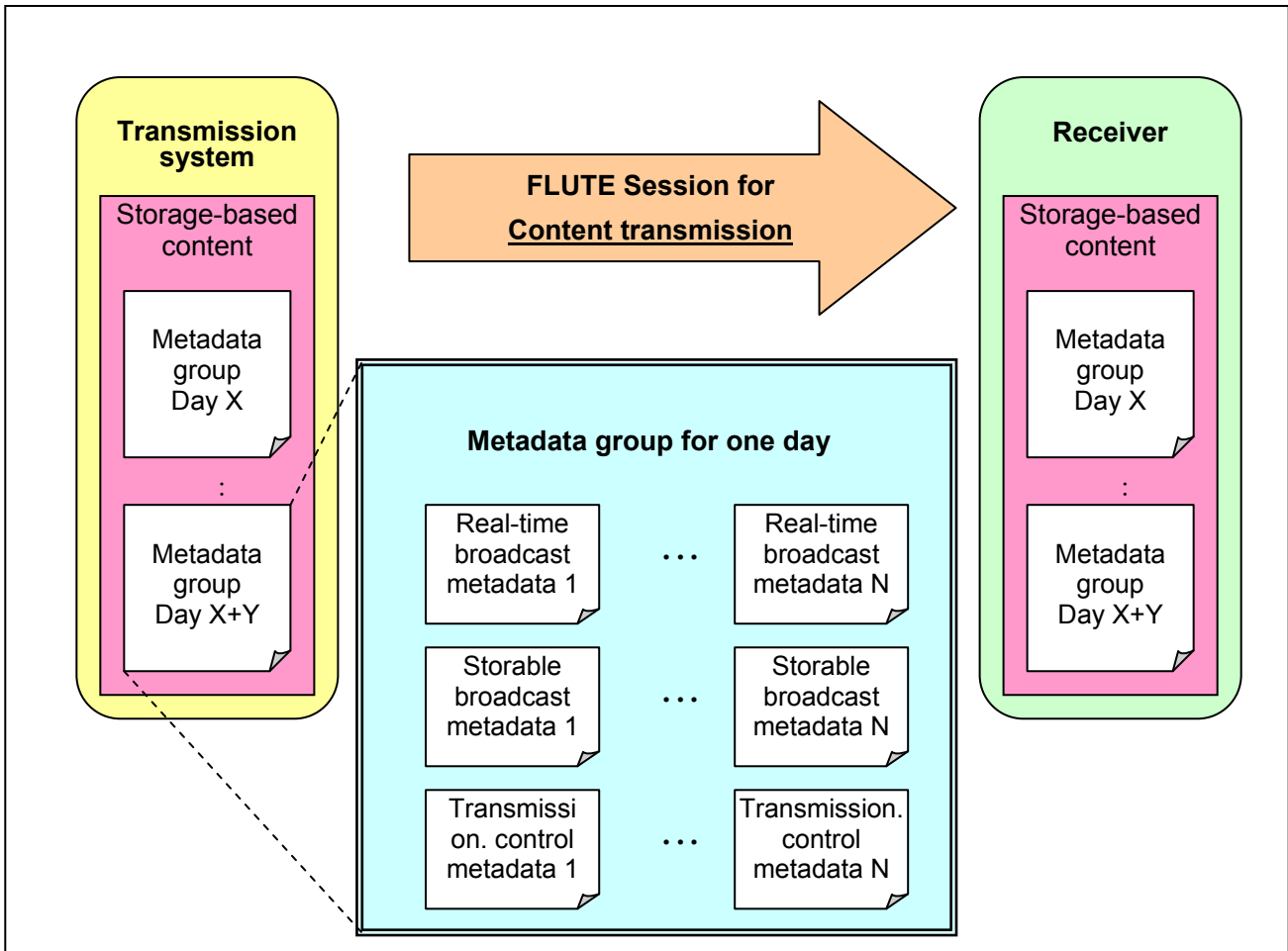
2.2.2.4 Metadata transmission order in a FLUTE session

An example of data transmission in a FLUTE session is described below.

If there are multiple metadata, it is possible to transmit them in the ascending order of the time difference between the “current time and broadcast start time”. In addition, they are transmitted in a carousel until the transmission content is updated.

The FDT instance is always transmitted, as a minimum, at the top of the transmission sequence. After that, it is possible to transmit the FDT instance between the transmission of

metadata) is transmitted using program transmission A in the same manner as other storable



broadcasting contents.

Fig. 2-38: Metadata transmission model (Metadata for one day grouped in one unit)

2.2.3.2 Operation of metadata transmission

The following explains the specific operation of metadata transmission.

2.2.3.2.1 Conversion of metadata into storable broadcasting content

The metadata for program transmission B is transmitted as a storable broadcasting content using a FLUTE session for content transmission. Metadata is categorized into metadata groups according to the metadata class and the difference between the current time and the program start time defined in the metadata, and all metadata groups are collected and converted into a storable broadcasting content. In addition, if the metadata contains a program start time that ranges across two groups, this metadata may be included either in both groups or only in the group with a smaller time difference, and the choice is left up to the entrusted broadcaster's operating policy. An operation example is described below.

(Operation example: When transmitting metadata for 3 days later ~ 8 days later)

Name of group: Metadata (48 - 72)

Metadata class: Metadata for real-time broadcasting/storable broadcasting

Difference between the current time and broadcast start time: Current time +48:00 ~ current time +72:00

Explanation: Metadata for broadcasts starting at the current time +48 hours until +72 hours

Name of group: Metadata (72 - 96)

Metadata class: Metadata for real-time broadcasting/storable broadcasting

Difference between current time and broadcast start time: Current time +72:00 ~ current time +96:00

Explanation: Metadata for broadcasts starting at the current time +72 hours until +96 hours

Name of group: Metadata (96 - 120)

Metadata class: Metadata for real-time broadcasting/storable broadcasting

Difference between current time and broadcast start time: Current time +96:00 ~ current time +120:00

Explanation: Metadata for broadcasts starting at the current time +96 hours until +120 hours

Name of group: Metadata (120 - 144)

Metadata class: Metadata for real-time broadcasting/storable broadcasting

Difference between current time and broadcast start time: Current time +120:00 ~ current time +144:00

Explanation: Metadata for broadcasts starting at the current time +120 hours until +144 hours

Name of group: Metadata (172 - 196)

Metadata class: Metadata for real-time broadcasting/storable broadcasting

Difference between current time and broadcast start time: Current time +172:00 ~ current time +196:00

Explanation: Metadata for broadcasts starting at the current time +172 hours until +196 hours

A storable broadcasting content is created from the above-described six groups that consist of metadata (48 - 72), metadata (72 - 96), metadata (96 - 120), metadata (120 - 144), metadata (144 - 172), and metadata (172 - 196).

2.2.3.2.2 Reference destination of the broadcasting schedule defined in EPG/ECG metadata

The reference destinations of the metadata for the broadcasting schedule (broadcast start time) that are required for grouping of metadata transmission are the same as those for

program transmission A. Refer to Section 2.2.2.2.1 (1) in this volume.

2.2.3.2.3 FLUTE session open parameters

The metadata is transmitted as a storable broadcasting content following the SessionDescription of the transmission control metadata that is associated with the storable broadcasting content.

2.2.4 Identification of metadata

There are three kinds of metadata: real-time broadcasting metadata, storable broadcasting metadata, and transmission control metadata. For a receiver, these metadata need to be distinguished to achieve a location solution and a storage process.

Metadata and storable broadcasting content are distinguished by the media type in the FDT instance or transmission control metadata. However, in the case of program transmission A, they are distinguishable by the service_id defined in accordance with the guideline provided in Volume 7, Section 7.1.2. In the case of program transmission B, it is mandatory to describe ContentProperties/ContentType/@href as described in Volume 10, Section 5.4.2; therefore, in addition to being distinguished by the media type, the presence of ContentProperties/ContentType/@href can also be used to distinguish it as metadata.

If the type of metadata (transmission control metadata or EPG/ECG metadata) needs to be judged, in addition to the media type, the file extension is used. Metadata classes and file extensions are listed below.

Table 2-28: Metadata Class and File Extension

Metadata class	File extension	Remarks
EPG/ECG metadata	ecg (XML format)	Metadata for real-time broadcast and for storable broadcasting use the same file extension.
	becg (BiM format)	
	becgx (MIME multipart format)	
Transmission control metadata	fci (XML format)	
	bfcj (BiM format)	
	bfcix (MIME multipart format)	

2.3 Operation of Media Coding

2.3.1 Video coding

For video coding, refer to Volume 3, Section 5.2.1 “Video coding”.

2.3.2 Static image and bitmap graphics coding

For the operation of static image and bitmap graphics coding, refer to Volume 3, Section 5.2.2 “Static image and bitmap graphics coding”.

2.3.3 Audio coding

For the audio coding, refer to Volume 3, Section 5.2.3 “Audio coding”.

2.3.4 Other coding

2.3.4.1 Character encoding

For the operation of character encoding, refer to Volume 3, Section 5.2.5.1 “Character encoding”.

2.3.4.2 Classification of storable broadcasting content

For the classification of storable broadcasting contents, refer to Volume 3, Section 5.2.7

“Classification of storable broadcasting contents”.

2.4 Multimedia Coding

2.4.1 File format

For the file format, refer to Volume 3, Section 5.2.4 “File format”.

2.4.2 Manifest file

For the operation of the manifest file, refer to Volume 3, Section 5.3 “Operation of the Manifest File”.

2.4.3 Media type

For the storable broadcasting media types covered in these operational guidelines, refer to Volume 3, Section 5.2.6 “Media type”.

2.5 Operation of Character Encoding

This section describes the operation of character encoding.

2.5.1 Operation of the HTML version

For the operation of HTML version, refer to Volume 3, Section 5.4.1 “Operation of the HTML version”.

2.5.2 Operation of the character code

For the operation of the character code, refer to Volume 3, Section 5.4.2 “Operation of the character code”.

2.5.3 Operation range of the media type and monomedia

Refer to Section 2.1 “Operation of Data Transmission System” in this volume.

2.5.4 Operation of the HTML element

For the operation of the HTML element, refer to Volume 3, Section 5.4.4 “Operation of the HTML element”.

2.5.5 Operation of CSS

For the operation of CSS, refer to Volume 3, Section 5.4.5 “Operation of CSS”.

2.5.6 Operation range of DOM

For the operation range of DOM, refer to Volume 3, Section 5.4.6 “Operation range of DOM”.

2.5.7 Operation range of the built-in object

For the operation range of the built-in object, refer to Volume 3, Section 5.4.7 “Operation range of the built-in object”.

2.6 Total Compensation of Stored Content

2.6.1 Overview

If the repair threshold in the transmission control metadata is set to 0%, it is possible to repair the whole stored content completely. However, if none of the content is received from the storable broadcasting, a list of files constituting the content is not available and the stored

content cannot be repaired even though the repair threshold is set to 0%. This section defines two systems that enable the repair of stored content even when none of the content was received from the broadcast.

2.6.2 Implementation based on the FDT instance

To implement a full repair system for stored content using the FDT instance, the FDT instance is returned by means of a stored content repair server or a metadata aggregation server (refer to Volume 0, Section 11.2 “System Model”). It should be noted that when returning the FDT instance, it is not divided and the whole file is returned in response to the request.

In this system, which obtains the FDT instance by communications, a list of file parameters of the stored content to be repaired is created from the FDT content URI, content length, hash value (message digest), etc., whereas information about the AL-FEC is not required. Therefore, operation of the FDT instance is defined as follows.

2.6.2.1 XML schema of the FDT instance

As defined in Section 2.1.5.3.2 “XML schema”

2.6.2.2 Semantics of the FDT instance

The semantics of the FDT instance is defined below.

Table S1-1: Operation of the FDT Instance

Element / Attribute Name	Use	Appearance	Max. character length and value	Definition
FDT-Instance	○			FDT instance information
@Expires	○	1	10 decimal digits (0 - 4294967295)	Describes the expiry time of the FDT instance in UNIX time format. The expiry time is set as the end time of the period while the repair of stored content is available. Once all files that constitute the content have been received, the receiver can delete the FDT instance information.
@Complete	○	0..1		Tells that no new FDT instance will be transmitted.
@Content-Type	○	0..1		Common information within the FDT instance. The contents are the same as those of File-Type element. The value of the common information is applied to any attribute which is not specified by individual File-Type element.
File	○	0..∞		Describes the information about the file object to be transmitted.

	@Content-Location	○	1	Max. 255 bytes	Describes the URI of the content.
	@TOI	○	1	10 decimal digits (0 - 4294967295)	Describes the object identifier.
	@Content-Length	○	1	10 decimal digits (0-1073741823)	Describes the content length up to 1 GB.
	@Transfer-Length	○	0..1	10 decimal digits (0-1073741823)	Describes the content length.
	@Content-Type	○	0..1	Max. 255 bytes	Describes the MIME content.
	@Content-MD5	○	0..1	Max. 24 bytes	Describes the message digest of the file object. The use of the message digest allows the receiver to check the compatibility of the file object that has been received for 100%.
	##other	○	0..∞		FDT extension

※ This table extracts only used items.

2.6.2.3 XML schema of FDT extension

As defined in Section 2.1.5.3.4 (1) “XML schema”

2.6.2.4 Semantics of FDT extension

The semantics of FDT extension used for this system is defined below.

Table S1-2: Operation of FDT Extension

Element / Attribute Name	Use	Appearance	Max. character length and value	Definition
FDT-Instance	○			Refer to the FDT-Instance element in Table S1-1.
@FullFDT	○	0..1		A flag to describe the information about all the file objects to be transmitted.

※ This table extracts only used items.

2.6.2.5 XML instance for the FDT instance

The following describes an example of the XML instance for the FDT instance that can be obtained in this system through the communications.

```
<?xml version="1.0" encoding="UTF-8"?>
<FDT-Instance
  Expires="3478323603"
  Complete="true"
  FullFDT="true"
  xmlns="http://www.arib.or.jp/tmm/fdt/2011/03">
  <File
    Content-Length="1024"
    Transfer-Length="1024"
```

```
Content-Location="/user/sample/test1.txt"  
Content-Type="text/plain"  
TOI="17"  
Content-MD5="MTIzNDU=" />  
</FDT-Instance>
```

2.6.3 Implementation based on HTTP directory resource information

There is another way to implement the full repair of stored content using HTTP directory resource information. This system also allows the automatic creation of an index file in the stored content repair server using its automatic indexing function.

In this case, the receiver obtains the directory resource information in response to the request described below, and then obtains a list of the files that constitute the content (depending on the implementation and operation of the stored content repair server, the receiver may obtain a list of the files by analyzing the directory resource information). Once a files list is obtained, the rest of the process is the same as the ordinary stored content repair. In this system, however, hash values of the files cannot be obtained. Therefore, completion of file repair cannot be determined by checking the file hash values; rather, completion is done when the response from the stored content repair server is completed.

The directory resource information can be obtained by issuing the request below, for example, the contents are stored in the directory named (/path/to/content/) on the stored content repair server.

```
GET /path/to/content/ HTTP/1.1  
Host: www.example.com
```


Volume 12

Compensation of Stored Content

Contents

Chapter 1 General Terms.....	1
1.1 Introduction	1
1.2 References	1
1.3 Terminology.....	1
Chapter 2 Overview of the Compensation of Stored Content.....	3
2.1 Compensation of Stored Content: Execution Types.....	3
2.2 Stored Content Compensation System.....	4
2.3 Overview of Receiver Functions.....	5
2.4 Supporting Network.....	6
Chapter 3 Operation of the Stored Content Compensation Server.....	7
3.1 Requirements for the Stored Content Compensation Server.....	7
3.1.1 HTTP	7
3.1.2 Content-Type	7
3.1.3 Content-Range.....	7
3.2 File Format	7
3.3 User Authentication	8
3.4 Load Distribution Method.....	8
3.4.1 Distributed deployment of the stored content compensation server.....	8
3.4.2 Scheduling the server for stored content compensation	8
3.4.3 Notification to the receiver or the update of time for performing the compensation of stored content.....	8
3.4.4 Available period for the compensation of stored content.....	9
Chapter 4 Detailed Operation of Receivers When Performing the Compensation of Stored Content.....	10
4.1 Overview of File Recovery	11
4.2 Specifying Missing Objects.....	11
4.3 Specifying Missing Symbols.....	12
4.4 Estimating Required Supplemental Packets	13
4.5 Request for Stored Content Compensation Data.....	15
4.6 Merge Processing.....	18
4.7 Determination of the Stored Content Compensation Schedule	18
4.8 Processing When the Compensation of Stored Content has Failed.....	21
4.8.1 Failure of stored content compensation when performed based on the schedule for	

the compensation of stored content.....	21
4.8.2 Failure of stored content compensation when performed by user operation	21
4.9 Avoidance of Congestion	22
4.9.1 Congestion countermeasures.....	22
4.9.2 Congestion countermeasures taken by broadcasting stations.....	22
4.9.3 Receiver functions	23
4.9.4 Avoidance of congestion in the stored content compensation server	24
4.9.5 Countermeasures for emergency	24
4.9.6 Applicable laws and rights acquisition status	24

Chapter 1 General Terms

1.1 Introduction

The purpose of this document is to define the operation method for compensating stored content, which is used especially in storable broadcasting, from among the operation methods related to communication for realizing multimedia broadcasting services.

The compensation of stored content is a function used to create a complete file by compensating for missing data from a stored content compensation server via communication when the entire file could not be received via broadcasting waves in storable broadcasting. There are two methods for compensating stored content: Automatic compensation and allowing the user to perform manual compensation.

1.2 References

The related documents are as follows.

- (1) IETF RFC 1864, "The Content-MD5 Header Field"
- (2) IETF RFC 2616, "Hypertext Transfer Protocol – HTTP/1.1"
- (3) IETF RFC 3926, "FLUTE - File Delivery over Unidirectional Transport"

1.3 Terminology

Content-MD5 field	Generated by encoding the MD5 digest of a check data (128 bits), which is used to check that the obtained data is the same as the source data, in BASE64 (RFC1864)
FDT instance	Object for transmitting a file delivery table (FDT) in which various attributes of a file distributed using the FLUTE defined in IETF RFC 3926 are described
FLUTE	File Delivery over Unidirectional Transport; file delivery protocol (RFC3926) for one-way communication
HTTP	Hyper Text Transfer Protocol: An application-layer protocol and a protocol (RFC2616) that is used for forwarding data through the World Wide Web
LDPC code	Low-density Parity Check Code: An error-correcting code that enables recovering data that couldn't be received using data for recovery

Broadcasting content	Broadcasting materials, such as video, audio, and data, received through a broadcasting signal
Storable broadcasting	One of the terrestrial multimedia broadcasting services based on connected segment transmission, which is provided when downloading
Compensation of stored content	Function to compensate for missing parts if incomplete content is received via broadcasting waves in storable broadcasting

Chapter 2 Overview of the Compensation of Stored Content

Receivers compensate for missing data from a stored content compensation server if part of the content broadcasted in storable broadcasting could not be received. The receiver identifies the necessary data for recovering the content and downloads the data from the stored content compensation server. There are two types of compensation of stored content: Auto compensation, which is a method that compensates stored content at different times in order to reduce the burden on a communication system; and manual compensation, which is a method that compensates stored content when a user uses the content. The subsequent sections define the receivers used for the compensation of stored content and the stored content compensation server.

2.1 Compensation of Stored Content: Execution Types

There are two types of compensations for stored content: Auto compensation and manual compensation. Each type has their own characteristics, as per below.

(1) Auto compensation

Auto compensation is a method used to compensate for stored content according to the determined schedule so that the receiver compensates for stored content by taking an interval of time in order to reduce the burden on a communication system.

Auto compensation is performed for a period of time after broadcasting the last piece of content. Whether or not auto compensation can be performed and the schedule for the compensation of stored content are described in the transmission control metadata. The receiver performs auto compensation based on that information.

(2) Manual compensation

Manual compensation is a method used to compensate for stored content when users use a certain piece of content. During the disclosure period of broadcasting content described in the EPG/ECG metadata, the compensation of stored content is performed freely at any time selected by a user from a period that is specified by the start and end time of the manual compensation described in the transmission control metadata.

A user performs manual compensation when the content is incomplete while playing it back. The receiver plays back the content while compensating for the incomplete content. Whether or not manual compensation is performed is described in the transmission control metadata. If manual compensation is prohibited, incomplete content is not played back.

Figure 2-1 shows the relationship between the broadcasting schedule and the compensation of stored content.

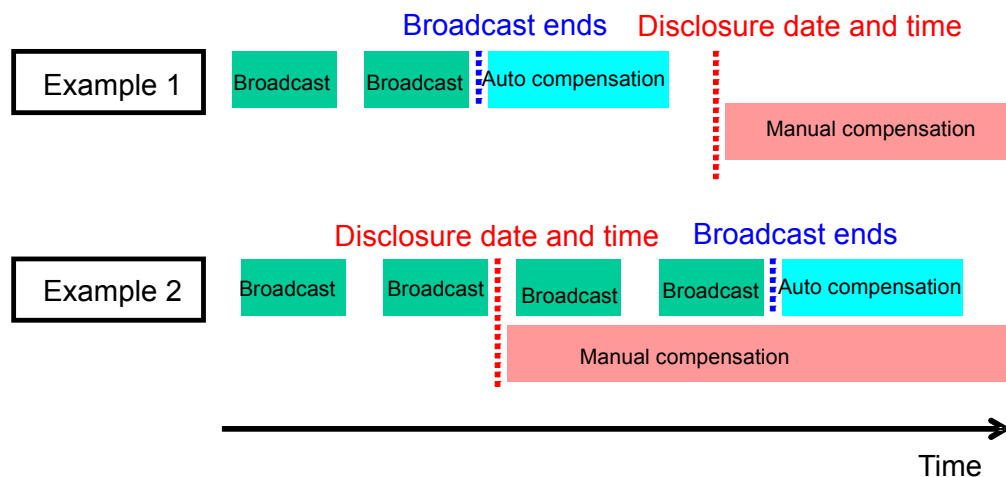


Fig. 2-1: Relationship between the broadcasting schedule and the compensation of stored content

In Example 1, the content disclosure date and time is set after broadcasting the last piece of content. In this example, auto compensation is performed after broadcasting the content, and manual compensation is performed after the disclosure date and time. In Example 2, the content disclosure date and time is set before the end of broadcasting the last piece of content. In this case, manual compensation can be performed any time after the disclosure date and time, even during the rebroadcasting of the content.

The operation of compensating for stored content varies depending on the content reservation method. As for manual storage reservation that is made by a user, auto compensation or manual compensation is executed if content is incomplete.

On the other hand, as for auto storage reservation on recommendation, neither auto compensation nor manual compensation is performed, and incomplete content is discarded after the end of the last broadcast if the reception rate of the content does not exceed a certain threshold value. The threshold value of the reception rate is only used for the compensation of stored content that is performed based on auto storage reservation. The threshold value is set for each piece of content and is notified to the receiver using the transmission control metadata.

2.2 Stored Content Compensation System

Figure 2-2 shows the example of the stored content compensation system used in storage-based services.

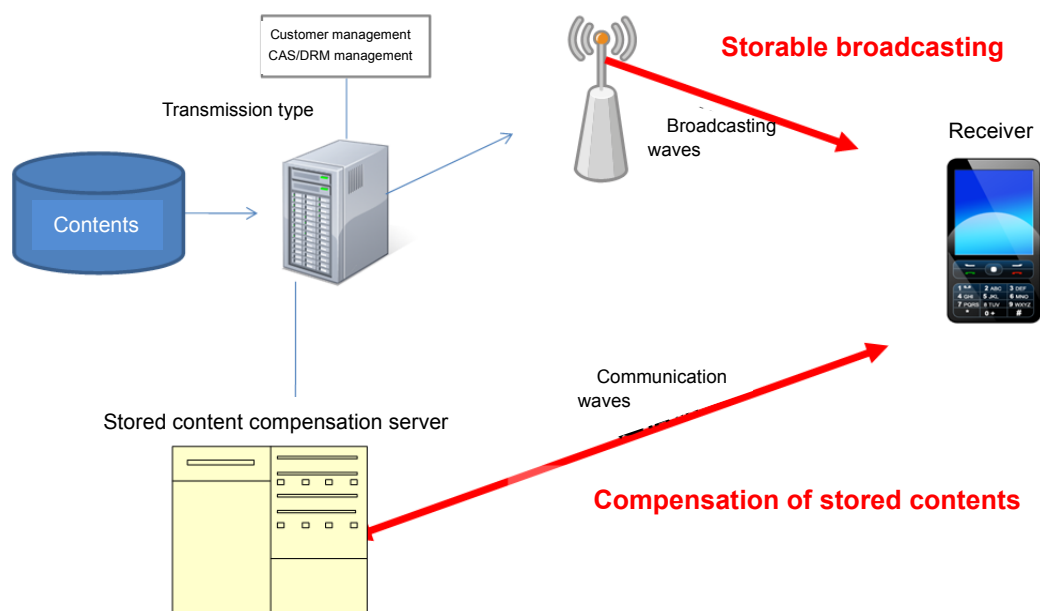


Fig. 2-2: Stored content compensation system

2.3 Overview of Receiver Functions

If the receiver could not normally receive content via broadcasting waves, it performs the compensation of stored content via communication based on the conditions defined in "2.1 Compensation of Stored Content: Execution Types" in this volume. The meaning of the abnormal reception of content indicates that the content, which is received through broadcasting as incomplete content with an error-correcting code, cannot be recovered using the error-correcting code.

The required supplemental packet is specified for each stored content that needs to be compensated for based on "2.1 Compensation of Stored Content: Execution Types" in this volume. After calculating the amount of necessary compensation data, the schedule to compensate stored content is determined based on "2.1 Compensation of Stored Content: Execution Types" in this volume.

When overseas mode is used, both the auto and manual compensation of stored content are not available (allowed). For details on overseas mode, see "4.19 Functional Restrictions in Countries other than Japan" in Vol. 2 in this technical document.

2.4 Supporting Network

The compensation of stored content can be performed using various communications networks.

Chapter 3 Operation of the Stored Content Compensation Server

3.1 Requirements for the Stored Content Compensation Server

When performing the operation of the compensation of stored content, the receiver sends a request message to the stored content compensation server via communication. Then, the stored content compensation server can identify the missing content (missing content, object, or data packet) from the message and sends the missing content to the receiver via communication based on the transmission schedule for data for the compensation of stored content. The receiver receives the missing content via communication and reconfigures the content by combining the normally received content and the received missing content to create normal desired content. The subsequent sections define the operation of the compensation of stored content.

The stored content compensation server supports the following functions.

3.1.1 HTTP

Receivers request the download of missing content to the stored content compensation server via a communications network for the compensation of stored content. Then, the stored content compensation server distributes the missing content via the communications network based on HTTP1.1. Unless otherwise specified, the distribution method conforms to "RFC2616 Hypertext Transfer Protocol-HTTP/1.1." The receiver sends a GET method request, in which the path of a content file to be used is specified, to the stored content compensation server.

3.1.2 Content-Type

The media types defined in "5.2.6 Media types" in Vol. 3 in this technical document are used as Content-Type.

3.1.3 Content-Range

The stored content compensation server sends the missing content requested from the receiver using the Content-Range header via a communications network.

3.2 File Format

The content for the compensation of stored content sent from the stored content compensation server is the encrypted content. In broadcasting waves, FEC data is added to the content. However, the FEC data is not added for the compensation of stored content.

3.3 User Authentication

Basically, the stored content compensation server does not manage customer management information. If a request to which customer management information is added in a terminal or gateway is issued, access to the stored content compensation server can be restricted based on the information.

3.4 Load Distribution Method

3.4.1 Distributed deployment of the stored content compensation server

Even if a single host name included in a URL for the compensation of stored content is used, the compensation of stored content can be performed using multiple HTTP servers by applying a load distribution system using a name server, etc.

3.4.2 Scheduling the server for stored content compensation

In order to reduce the influence on a communications network, when auto compensation is selected, the compensation of stored content is scheduled so that communication traffic volume is not heavily distributed over a specified time. For details, see "4.7 Determination of the Stored Content Compensation Schedule."

The schedule for the compensation of stored content is defined based on the offset time and random back-off time for starting the compensation as described in "2.1.5.4.1 (2) Semantics of User Service Description" in Vol. 11 of this technical document. The compensation of stored content starts any time between the time calculated by adding the offset time to the last broadcasting time of the content and the random back-off time.

As receivers can freely select any time within the random back-off period for starting compensation, the operation of the compensation of stored content does not concentrate at a specific time. In addition, the overlapping of the time for compensating for multiple pieces of stored content can be avoided by using the same offset time to start the compensation regardless of the content and by setting the random back-off time for starting the compensation to the same time as the broadcasting time of the content. If the information on traffic load status in a communications network or the load information of a complementation server can be obtained, the offset time for starting compensation can be adjusted depending on the load information. However, the details are not described in this operational standard.

3.4.3 Notification to the receiver or the update of time for performing the compensation of stored content

If a request for the compensation of stored content cannot be normally processed due to

congestion on the stored content compensation server or due to the maintenance of the server, the stored content compensation server notifies the updated time for performing the compensation of stored content to the receiver using a Retry-After message.

If content has many missing parts and if many requests for compensation are required, the time for finishing broadcasting the content is set (or a future time may be set) as the time for updating the content when registering the content so that rebroadcasting via broadcasting waves can be notified. If the broadcasting time schedule of content is changed due to rebroadcasting or other reasons, the stored content compensation server resets the time for updating the content to the time for finishing broadcasting based on the changed broadcasting time schedule even when the content is not updated. The receiver can identify that the broadcasting time schedule of the content was changed due to rebroadcasting and so on by issuing a request for the compensation of stored content with a condition on the time to update the content stored in the stored content compensation server. If reception is not necessary for rebroadcasting, a request for the compensation of stored content is issued without a condition on the time for updating the content.

3.4.4 Available period for the compensation of stored content

The stored content compensation server retains the necessary information so that the compensation of stored content can be performed during the viewable/usable period of each broadcasted content.

<Blank Page>

Chapter 4 Detailed Operation of Receivers When Performing the Compensation of Stored Content

4.1 Overview of File Recovery

Receivers receive content with an error-correcting code via broadcasting waves. If missing data cannot be recovered using an error-correcting code, the receiver performs the compensation of stored content. The receiver identifies the missing object in the received data and a missing symbol for reconfiguring content data. Then, the receiver repairs the content by downloading the necessary source symbol from the stored content compensation server. The following figure shows the processing procedure, from broadcast reception to file recovery. The processing of the compensation of stored content starts based on "4.7 Determination of the Stored Content Compensation Schedule."

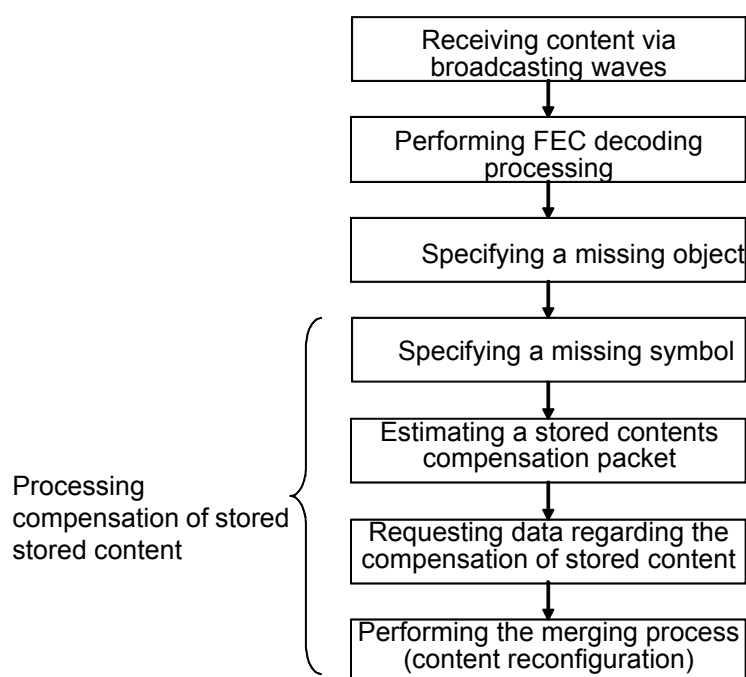


Fig. 4-1: Processing compensation of stored content

4.2 Specifying Missing Objects

Receivers identify missing objects that have not been completely received by referencing the list of objects that configure content from the information included in the transmission control metadata. The receiver also checks each object, which could be received completely, for integrity,

using the "Content-MD5" field checksum that is stored in the FDT instance. If an object lacking integrity is found in an integrity check, which is performed for each object, it is handled as an object that misses all.

4.3 Specifying Missing Symbols

Receivers identify which coding symbol corresponds to the received data based on the FEC payload ID included in the FLUTE header of a reception packet.

(1) Compact No-Code FEC

If the Compact No-Code FEC is used, the coding symbols that are received in duplicate are discarded and the reception of the symbols is ended when all source symbols are gathered. Then, content data is reconfigured using the source symbol and is saved in the receiver as content data. After that, the received symbol data is discarded. If all source symbols are not gathered, all received source symbols are saved.

If the influence on the capacity of a communication path is small (such as a small-capacity object) even when the number of transmission times is increased more than once, this coding is used to reduce the load of decoding processing in the receiver.

(2) LDPC code staircase

If the LDPC code is used, when receiving a new symbol, the receiver determines whether the new symbol can be decoded using the already received symbols and decoded symbols. If the symbol can be decoded, then the receiver decodes the symbol as much as possible. If already decoded symbols are received, they are discarded as invalid symbols. The reception of the symbols ends when all source symbols are decoded. Then, content data is reconfigured using a source symbol and is saved in the receiver as content data. After that, the received symbol data is discarded. If all source symbols cannot be decoded, then all received source symbols and parity symbols are saved. Invalid symbols can be discarded.

If an object is received for which the UEP function of the LDPC code staircase is valid, even if all the source symbols are not decoded at the time of the end of broadcasting, the object is reconfigured from its head to the part where all source symbols are collected. All source symbols are saved.

This coding is used for transmission that requires high error correction ability. As this coding can effectively perform compensation operation while playing back by focusing on correcting an error in the head part of a file, it is desirable to validate the UEP function when transmitting moving image files.

4.4 Estimating Required Supplemental Packets

(1) Compact No-Code FEC

When the Compact No-Code FEC is used, receivers request compensation for all missing source symbols. If it is necessary to calculate the ratio of missing data amount, the ratio is obtained by calculating the ratio of the number of the missing source symbols to the total number of source symbols.

(2) LDPC code staircase

When the LDPC code is used, the burden on the communications network increases if the receiver requests the compensation of all missing source symbols. Therefore, it is necessary to reduce the number of symbols that require compensation using the received redundant symbols. In order to guarantee the minimum number of symbols required for compensation, the calculation of $O(2n)$ is necessary. The figure below provides an example of an algorithm to effectively select the symbols required for compensation using $O(n)$. If an algorithm can reduce the number of symbols required for compensation to the number that is equal to or more than the number obtained using the algorithm below, the algorithm can be implemented in the receiver.

When the LDPC code is used, receivers calculate the number of valid symbols by subtracting the number of invalid symbols from the number of received symbols. When the receiver performs the compensation of stored content, it selects the necessary source symbols from the number of valid symbols. The number of missing symbols required for compensation can be obtained by subtracting the number of valid symbols from the number of all source symbols. If it is necessary to calculate the ratio amounts of missing data, this can be obtained by calculating the ratio of the number of missing symbols to the total number of source symbols.

The source symbols that request compensation are selected for only the number of missing symbols using the following algorithm.

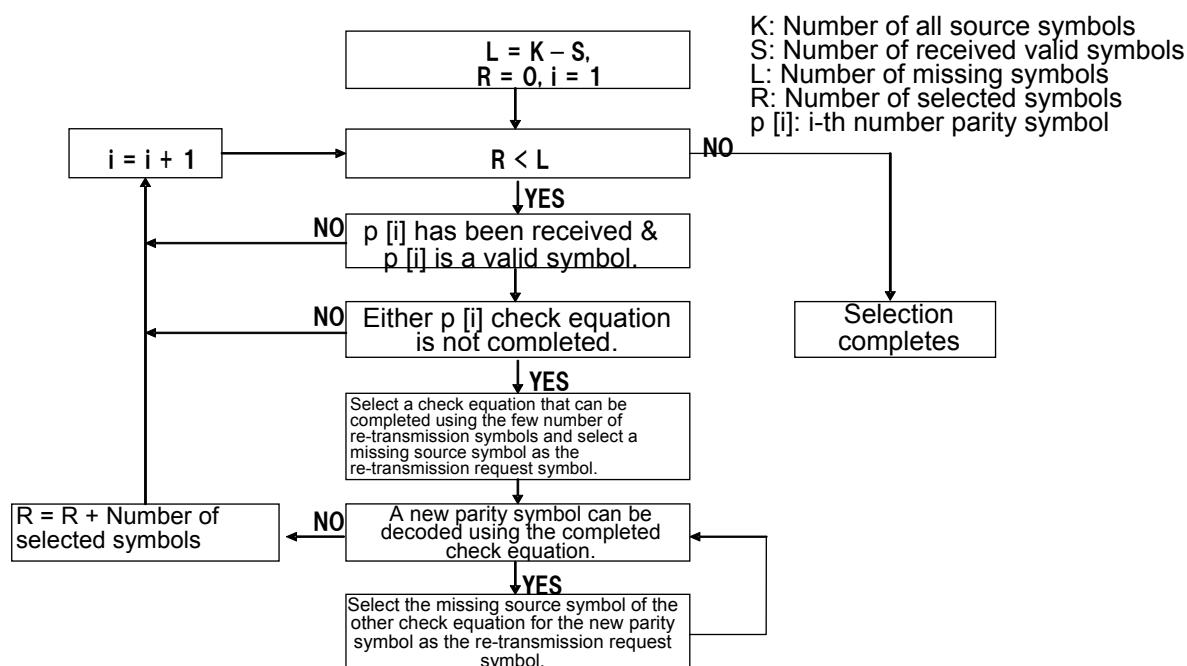


Fig. 4-2: Method for selecting a request symbol

Receivers closely investigate each check equation in the check matrix of the LDPC code in order and select the necessary source symbols for decoding each check equation. When the number of selected symbols reaches the number of missing symbols, the receiver finishes the selection algorithm and requests the resending of the selected source symbols.

If the number of symbols for requesting resending is high, all the symbols may not be described using a single HTTP request. In this case, the resend request is made by selecting the number of symbols that can be described using a single HTTP request. After performing decoding processing using resent symbols, the receiver performs the same symbol selection algorithm again and requests resending.

All source symbols may not be decoded depending on the decoding algorithm for the LDPC code, even if a resend request is made by selecting the calculated number of missing symbols. In this case, the same symbol selection algorithm is performed again, and a resend request is made. After setting the number of lacking symbols to 10 or less if the source symbols cannot be decoded when the number of missing symbols becomes 0, the same symbol selection algorithm is performed again, and a resend request can be made.

4.5 Request for Stored Content Compensation Data

As the symbols requested for the compensation of stored content are all source symbols, part of the content that corresponds to a source symbol to be requested can be specified using the symbol-dividing algorithm. Receivers determine that the data in which a byte range in the content is requested using the source symbol for requesting compensation and request the applicable data using the HTTP Byte Range Request message.

The data to be requested can be calculated using the following formula based on the encoding symbol ID to be requested and the symbol length (number of bytes) specified using the FEC-OTI-Encoding-Symbol-Length value that can be obtained using the FDT instance.

$$\text{(Byte range to be requested)} = \{(\text{Symbol length}) \times (\text{Encoding Symbol ID})\} \text{ to } \{(\text{Symbol length}) \times (\text{Encoding Symbol ID} + 1) - 1\}$$

Receivers can collectively request the byte range data that corresponds to multiple missing symbols.

For instance, when a symbol size is 300 bytes and when the 0, 4th, and 8th source symbols are requested, the following HTTP request protocols are used.

```
GET /sample/sample.mp4 HTTP/1.1
Host: www.example.com
Range: bytes= 0-299, 1200-1499, 2400-2699
```

If the version information is included in the Content-Location in the transmission control metadata or FDT instance, a GET method that requests a URL with the version information is sent. If version information is not included in the Content-Location above, a GET method is sent without including the version information.

When the receiver requests data as described above, access is made to the stored content compensation server described in the Associated Delivery Description in the transmission control metadata. If the Full FDT has been received, the Content-Location in the FDT instance is used. If transmission is made using a divided FDT, the Content-Location in the transmission control metadata can be used because all Content-Locations included in the FDT instance may not be able to be obtained. However, in this case, taking into account the possibility that the obtained transmission control metadata may not be the latest one because the schedule was changed immediately before the transmission or for other reasons, it is desirable to re-obtain the transmission control metadata if possible. It is also preferable to request the download of supplemental data after the broadcast of applicable content ends. Receivers reference the

obtained program guide and wait to perform the compensation of stored content until the next broadcasting time if the applicable content is scheduled to be rebroadcast.

Even if the same content is not scheduled to be broadcast in the program guide obtained by the receiver, the same content may be rebroadcast due to the rescheduling of the program guide. Receivers determine whether the applicable content is rebroadcast by adding the following request header with a condition to a HTTP request.

If Unmodified Since: (Time when the receiver received content in broadcasting) GMT

Receivers set request conditions using the broadcasting time (the time when the receiver received the content in broadcasting) even if they couldn't receive any content data within the broadcasting time.

The content data specified using a GET method is transmitted if a requested piece of content is not updated after the time specified in the header above.

If a requested piece of content is updated after the time specified in the header above, then the following error response is returned from the stored content compensation server.

412 Precondition Failed

The time at which to update the content stored in the stored content compensation server is set to the time at which to finish broadcasting. Therefore, when the receiver receives the error response above, it determines that the content will be rebroadcasted and obtains the rebroadcasting time by re-accessing the program guide, etc. The figure below shows an example of the relationship between the time at which to update data for compensation stored in the stored content compensation server and the time at which to request the compensation of stored content.

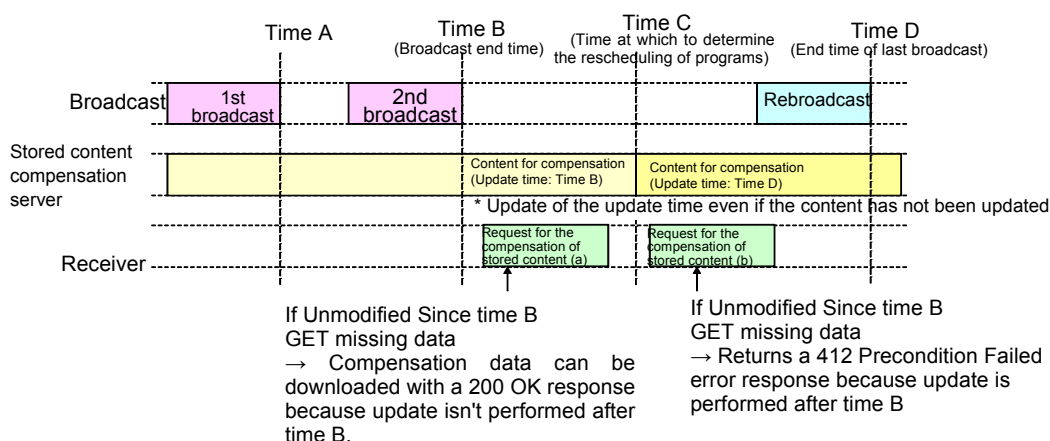


Fig. 4-3: Time at which to request the compensation of stored content

The storable broadcasting content shown in the example above is scheduled to be broadcast twice in the original program schedule. The end time of the first broadcasting is time A, and the end time of the second broadcasting is time B. The program of the content is rescheduled at time C and determines that this content will be rebroadcast. The rebroadcasting ends at time D. In this case, the time at which to update the content for compensation stored in the stored content compensation server is set to time B, which is the end time of the second broadcasting, until time C. However, after the program is rescheduled at time C, the time at which to update the content for compensation is changed to time D. As the applicable content is broadcast until time B, receivers can receive the content in broadcasting without compensating for the stored content. If the compensation of stored content is requested after time B, the request is made at time B, which is the end time of the second broadcasting, as the time set in the If Unmodified Since request header. If the compensation of stored content is requested (a) between time B and time C, as the time at which to update the content for compensation stored in the stored content compensation server is time B, update is not performed after time B. Therefore, the stored content compensation server replies a 200 OK response and transmits the requested missing data. If the receiver requests (b) the compensation of stored content between time C and time D, the content for compensation stored in the stored content compensation server is updated at time D, and another update is performed after time B. Therefore, the stored content compensation server replies with a 412 Precondition Failed error response. When the receiver receives this error response, it waits until the missing data is compensated for. Then, the receiver obtains the next broadcasting time by re-accessing the program guide to schedule

receiving the next broadcast.

4.6 Merge Processing

Receivers use the data resent from the stored content compensation server as a source symbol and restart decoding the LDPC code. If a source symbol can be completely decoded using the resent data, the receiver re-specifies any missing symbol and performs the compensation of stored content again. When the source symbol can be completely decoded, the receiver reconfigures the content and discards the received decoded symbol.

As for objects such as moving image files, the compensation of stored content can be processed while viewing using the head part of a source symbol even if all the data is not collected. In this case, a part of the object from its head to the part where all source symbols are collected is reconfigured as a partial object. The received coded symbols need to be saved until the object can be completely decoded.

4.7 Determination of the Stored Content Compensation Schedule

Receivers start the processing of the compensation of stored content by providing a viewing start action if the viewing action occurs after the content disclosure date and if the content has not been completely received (manual compensation).

As for the content that is scheduled for the performance of manual storage, the processing of the compensation of stored content is automatically started after storable broadcasting based on the transmission control metadata (auto compensation).

The control information regarding the processing of the compensation of stored content is set in the Associated Delivery Procedure Description in the transmission control metadata. The URI of the stored content compensation server can be obtained by referencing the server URI element of the postFileRepair item.

As for manual compensation, the compensation of stored content can be requested during the period of time between the time set in the manualRepairStartDate attribute (manual compensation start date and time) and the time set in the manualRepairEndDate attribute (manual compensation end date and time) of the postFileRepair item in the disclosure period of the content.

As for auto compensation, receivers randomly select a time from the period between the time obtained by adding the number of seconds specified in the offsetTime attribute of the postFileRepair item to the start time of the last broadcasting (auto compensation start date and time) and the time obtained by adding the randomTimePeriod attribute (auto compensation end date and time). Then, receivers request the compensation of stored content. If auto

compensation cannot be performed during this period, the receiver determines not to execute auto compensation and allows performing only manual compensation. To randomly select a time, a uniformly distributed random number, which is normalized using the number of seconds set in the `randomTimePeriod` attribute, is generated in the receiver. Then, calculation is made to determine the time by adding the generated random number (variable value) as the number of seconds to the time (auto compensation start date and time), which is obtained by adding the fixed number of seconds set in the `offsetTime` attribute to the start time of the last broadcasting.

If the `receptionCycleStartTime` element and `receptionCycleEndTime` element of the `postFileRepair` item are set, the period of time during which auto compensation can be started each day is limited to the period between the time set in the `receptionCycleStartTime` element and the time set in the `receptionCycleEndTime` element. In this case, to randomly select a time, calculation is made in the receiver to obtain the number of seconds in the limited period in which auto compensation can be started based on the time set in the `receptionCycleStartTime` element and the time set in the `receptionCycleEndTime` element in the period set in the `randomTimePeriod` attribute. Then, a uniformly distributed random number, which is normalized using the number of seconds in the period during which auto compensation can be started, is generated, and calculation is made by forwarding the generated random number of seconds in the period during which auto compensation can be started.

The following figure shows an example of an auto compensation schedule.

Auto compensation is scheduled to be performed after the final broadcasting time of the applicable content in the broadcasting schedule. When scheduling auto compensation for multiple pieces of content, the congestion of a communications network can be reduced by setting the content pieces so as not to overlap. The communication volume when performing the compensation of stored content can be stabilized by setting the auto compensation period in proportion to the content broadcasting time or content data amount.

The communication volume can also be stabilized by estimating the communication volume for auto compensation based on the expected content viewing rate or current capacity of the communications network.

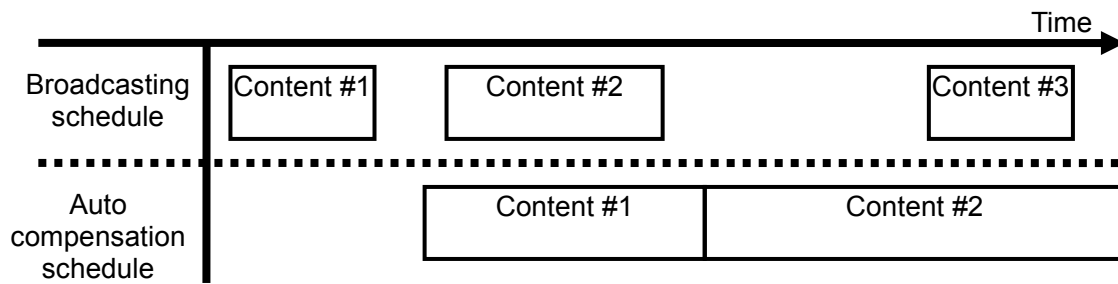


Fig. 4-4: Stored content compensation schedule example 1

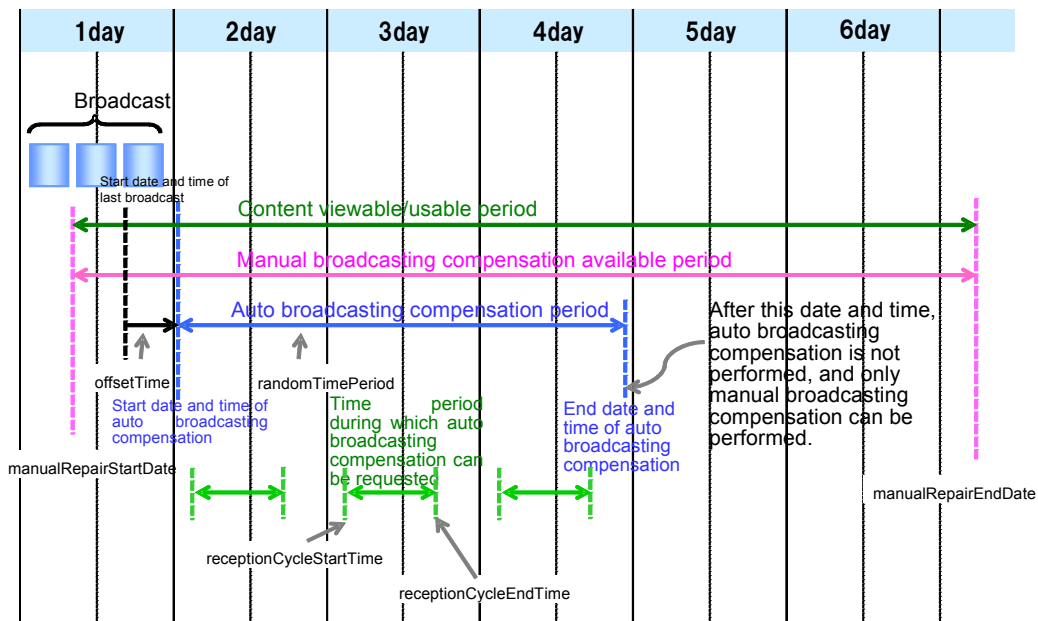


Fig. 4-5: Stored content compensation schedule example 2

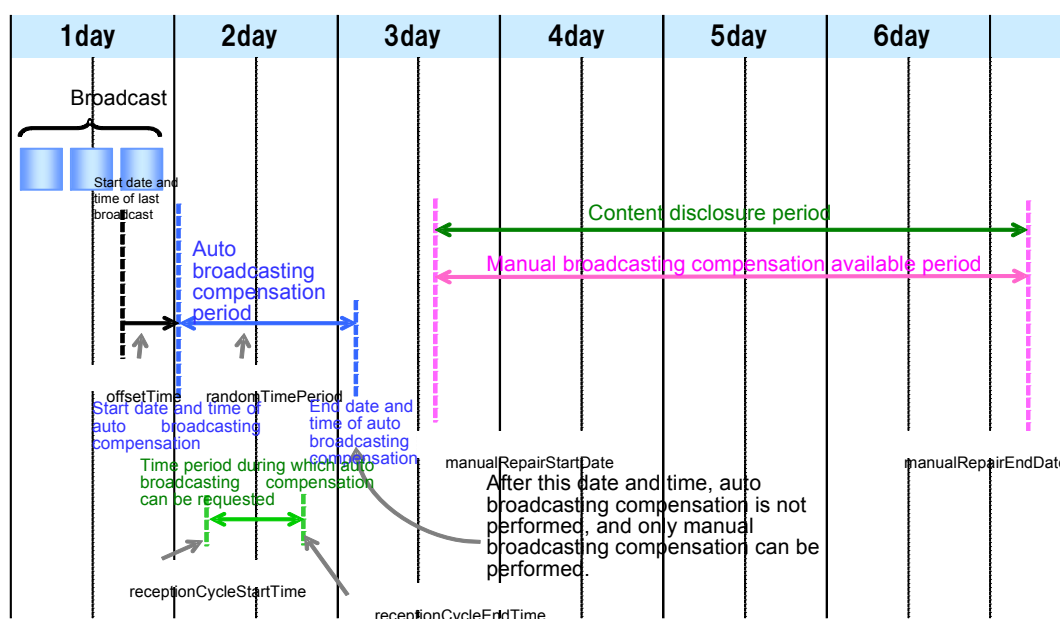


Fig. 4-6: Stored content compensation schedule example 3

4.8 Processing When the Compensation of Stored Content has Failed

4.8.1 Failure of stored content of auto compensation when performed based on the schedule for the compensation of stored content

If the compensation of stored content, which is performed based on a specified schedule, is not normally completed (e.g., the receiver is out of communication range, the communications network is congested, or the stored content compensation server is in busy status, etc.), the compensation of stored content is processed at regular intervals until the last date of auto compensation.

If the compensation of stored content is not normally completed until the end date and time of auto compensation, then auto compensation is no longer performed and the viewer manually performs compensation when viewing.

4.8.2 Failure of stored content compensation when performed by user operation

If the compensation of stored content is not normally completed in manual operation by a viewer (e.g., the receiver is out of communication range, the communications network is congested, or the stored content compensation server is in busy status, etc.), an error message is displayed to the viewer to prompt a retry after waiting a while (for instance, the following message is displayed: "The server is busy now. Please retry the compensation operation later.").

4.9 Avoidance of Congestion

4.9.1 Congestion countermeasures

As a large amount of communication data is used for the compensation of stored content performed in a storable broadcasting service, networks are likely to become congested because communication is concentrated on a specific stored content compensation server in a short period of time. If the network is congested, the multimedia broadcasting service cannot be properly provided; for instance, user communication will be incomplete. This must be prevented because this also affects other communication.

4.9.2 Congestion countermeasures taken by broadcasting stations

Broadcasting stations have systems to disperse requests for the compensation of stored content sent from terminals per storable broadcasting content based on the instruction described in 3.4.2. However, it is necessary to create storable broadcasting content by designing communication requests for the compensation of stored content sent from terminals that are not heavily concentrated.

Specifically, it is desirable to avoid congestion using any of the following methods (multiple methods can be used together) if control is determined necessary in order to avoid congestion after estimating communication traffic taking into account the storage rate of each content, the communication time required for compensation of stored content, and the time when the congestion on a communications network is maximized.

4.9.2.1 Control of compensation start time

The control information regarding the processing of the compensation of stored content is set in the Associated Delivery Procedure Description in the transmission control metadata. The information can be specified for each piece of storable broadcasting content. Specifically, the receiver requests the compensation of stored content by randomly selecting a time from the period between the time obtained by adding the number of seconds of the `offsetTime` attribute in the `postFileRepair` item to the start time of the last broadcasting and the time calculated by adding the time set in the `offsetTime` attribute and the time set in the `randomTimePeriod` attribute.

The time for starting the compensation of stored content is set so as to avoid concentrating the operation in a specified time period by performing the procedures below on a terminal.

1. Random number generation
2. Timer specification
3. Compensation of stored content registration calling

4.9.2.2 Restriction on transmitting the compensation of stored content

The available time for the compensation of stored content is set in the Associated Delivery Procedure Description in the transmission control metadata. The compensation of stored content can be requested during the period between the time set in the receptionCycleStartTime element and the time set in the receptionCycleEndTime element of the postFileRepair item. Receivers cannot request the compensation of stored content at any time outside this time period by specifying auto compensation.

Thus, this function can prohibit requesting auto compensation in a specified time period. As this function prohibits the auto compensation requested from any terminals, it is effective only when the traffic related to auto compensation needs to be completely controlled to perform an operation, such as the maintenance of the stored content compensation server.

4.9.2.3 Providing information to telecom carriers in advance

It is preferable to provide information to telecom carriers beforehand regarding the compensation of stored content when bulk traffic is expected to occur.

If a communications network is congested even though countermeasures such as any of the above are taken, the consideration of applying the following countermeasures should be made: application of the traffic load-balancing setting for the next compensation of stored content and a preventive measure taken by cooperating with telecom carriers.

4.9.2.4 Reducing traffic load for the compensation of stored content by scheduling rebroadcasting

If requests for the compensation of stored content are concentrated on a specific piece of content, the traffic load can be reduced by scheduling the rebroadcasting of the content via broadcasting waves. For details about how to notify the rebroadcasting schedule to terminals, see Section 3.4.3.

4.9.3 Receiver functions

Receivers must have a function to generate the random number that is necessary to control the start time to perform the compensation of stored content.

If a request for the compensation of stored content failed because the receiver is out of communication range, etc., recalling must be performed twice or less within three minutes so as not to burden the receiver.

4.9.4 Avoidance of congestion in the stored content compensation server

If a response from the stored content compensation server is delayed, the main cause is a lack of server performance for the requests to be sent or a lack of performance in a device on the route.

It is desirable to conduct the following countermeasures to avoid congestion.

- (1) Improvement of server processing ability
- (2) Server load balancing
- (3) Implementation of a cash server

If traffic load is concentrated temporarily, the use of the Retry-After message described in 3.4.3 should be taken into account.

4.9.5 Countermeasures for emergency

4.9.5.1 Functions for emergency

This section describes how to ensure the important communications that are necessary for disaster prevention along with a function that enables viewers to shift to emergency communication when an emergency occurs, such as large-scale disaster, when the compensation of stored content is being performed or when it is scheduled.

Table 4-1 shows the functions for emergency, such as disasters, etc.

Table 4-1 Functions for emergency, such as disasters, etc.

	Function
Broadcasting stations	It is desirable that a station can control the interrupt or stop of the compensation of stored content via broadcasting waves. It is desirable that a station can control new communications not being performed via broadcasting waves.
Receivers	It is desirable that receivers can control new communications not being performed by controlling broadcasting waves.

4.9.6 Applicable laws and rights acquisition status

4.9.6.1 Applicable laws

The subsequent sections describe the applicable laws & regulations that need to be taken into account when compensating for stored content.

4.9.6.1.1 Laws to be taken into account when dealing with an emergency

(1) Telecommunications Business Act

- Article 8 Securing of Essential Communications

4.9.6.1.2 Laws related to congestion in communications systems

(1) Ordinance Concerning Terminal Facilities Etc

- Articles 11 and 18 Transmission Functions

OPERATIONAL GUIDELINES FOR
TERRESTRIAL MOBILE MULTIMEDIA BROADCASTING BY
TRANSMISSION SYSTEM BASED ON CONNECTED SEGMENTS
FOR VHF-HIGH BAND

ARIB TECHNICAL REPORT

ARIB TR-B33 VERSION 2.3-E3
(Fascicle 3)
(September 30th, 2015)

This Document is based on the ARIB technical report of “Operational Guidelines for Terrestrial Multimedia Broadcasting by Transmission System Based on Connected Segments for VHF-HIGH Band” in Japanese edition and translated into English in October 2015.

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
