ARIB STD-B10
Version 5.13-E1

ENGLISH TRANSLATION

SERVICE INFORMATION FOR DIGITAL BROADCASTING SYSTEM

ARIB STANDARD

ARIB STD-B10 Version 5.13

Version 1.0 June 19, 1997
Version 5.13 December 5, 2019

Association of Radio Industries and Businesses

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

1. Notes on Copyright

2. Notes on English Translation

- ARIB Standards and Technical Reports are usually written in Japanese. This docment 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 Standards include "government technical regulations" (mandatory standard) that are set for the purpose of encouraging effective use of frequency and preventing interference with other spectrum users, and "private technical standards" (voluntary standards) that are defined in order to ensure compatibility and adequate quality of radio equipment and broadcasting equipment as well as to offer greater convenience to radio equipment manufacturers, telecommunication operators, broadcasting equipment manufacturers, broadcasters and users.

This ARIB Standard is developed for "SERVICE INFORMATION FOR DIGITAL BROADCASTING SYSTEM". In order to ensure fairness and transparency in the defining stage, the standard 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 Standard will be widely used by radio equipment manufacturers, telecommunication operators, broadcasting equipment manufacturers, broadcasters and users.

NOTE:

Although this ARIB Standard contains no specific reference to any Essential Industrial Property Rights relating thereto, the holders of such Essential Industrial Property Rights state to the effect that the rights listed in the Attachment 1 and 2, which are the Industrial Property Rights relating to this standard, are held by the parties also listed therein, and that to the users of this standard, in the case of Attachment 1, such holders shall not assert any rights and shall unconditionally grant a license to practice such Industrial Property Rights contained therein, and in the case of Attachment 2, the holders shall grant, under reasonable terms and conditions, a non-exclusive and non-discriminatory license to practice the Industrial Property Rights contained therein. However, this does not apply to anyone who uses this ARIB Standard and also owns and lays claim to any other Essential Industrial Property Rights of which is covered in whole or part in the contents of the provisions of this ARIB Standard.

For details, refer to “Guidelins for Treatment of Industrial Property Rights in connection with the ARIB Stantard” posted in the IPR Policy section of the ARIB website (https://www.arib.or.jp/english/).

Attachment 1 (Selection of Option 1)

(N/A)

Attachment 2 (Selection of Option 2)

PATENT HOLDER	NAME OF PATENT	REGISTRATION NO./ APPLICATION NO.	REMARKS
Matsushita Electronic Industrial Co., Ltd.	映像ﾃﾞｰﾀ送信方法、映像ﾃﾞｰﾀ送信装置、及び映像ﾃﾞｰﾀ再生装置*2	特開平 9-327004 号	JP,US,GB, DE,FR,CN, KR
	映像ﾃﾞｰﾀ送信方法及び映像ﾃﾞｰﾀ再生装置及び映像音声ﾃﾞｰﾀ再生装置*2	特願平 9-45599 号	JP,US,GB, DE,FR,CA
	放送送信装置、放送受信装置及びこれらを用いた放送システム*3	特願平 10-127642 号	JP,US,GB, DE,FR,CN, KR,TW,AU, SG
	放送局システム及び受信機 *3	特願平 10-195093 号	
Panasonic Corporation	Submitted comprehensive confirmation of patents for ARIB STD-B10 5.4 *19		
Next Generation Information Broadcasting Research Laboratory Co.(IBLabs) & Japan Broadcasting Corporation(NHK) (joint application)	放送システム*1,*3	特開 2000-13755 号	JP
	ﾃﾞｼﾞﾀﾙ放送の時刻同期方法、ﾃﾞｼﾞﾀﾙ放送送出装置、ﾃﾞｼﾞﾀﾙ放送受信装置、ﾃﾞｼﾞﾀﾙ放送送受ｼｽﾃﾑ、及びﾃﾞｼﾞﾀﾙ放送のﾃﾞｰﾀ構造*1,*3	特開 2000-4210 号	JP
Next Generation Information Broadcasting Research Laboratory Co.	ﾃﾞｨｼﾞﾀﾙ放送に用いられるｲﾝﾃﾞｯｸｽ情報ｻｰﾋﾞｽ提供方法、ﾃﾞｨｼﾞﾀﾙ放送送出装置、ﾃﾞｨｼﾞﾀﾙ放送受信装置、及びﾃﾞｨｼﾞﾀﾙ放送のﾃﾞｰﾀ構造*1	特開 2000-4427 号	JP
Victor Company of Japan, Ltd.	再生プロテクト方法及びプロテクト再生装置*4	特許第 2853727 号	JP,US,DE, GB,FR,KR, IN,CN
	情報記録方法及び情報記録媒体*4	特許第 3102416 号	JP
Sony Corporation	デジタル放送送受信システム及びデジタル放送受信装置 *5	PCT/JP01/07317	JP,AU,BR, CN,US
	Submitted comprehensive confirmation of patents for ARIB STD-B10 3.0 *5		
	Submitted comprehensive confirmation of patents for ARIB STD-B10 3.1 *6		

PATENT HOLDER	NAME OF PATENT	REGISTRATION NO./ APPLICATION NO.	REMARKS
	Submitted comprehensive confirmation of patents for ARIB STD-B10 5.3 *18		
	Submitted comprehensive confirmation of patents for ARIB STD-B10 5.4 *19		
Mitsubishi Electric Corporation	Submitted comprehensive confirmation of patents for ARIB STD-B10 3.1*6		
	Submitted comprehensive confirmation of patents for ARIB STD-B10 4.1 *13		
TOSHIBA Corporation	デジタル放送の送信装置、その受信方法及び受信装置*7	特願 2000-054591	JP
Motorola Japan Ltd.	Submitted comprehensive confirmation of patents for ARIB STD-B10 3.6 *8		
	Submitted comprehensive confirmation of patents for ARIB STD-B10 3.8 *9		
	Submitted comprehensive confirmation of patents for ARIB STD-B10 3.9 *10		
	Submitted comprehensive confirmation of patents for ARIB STD-B10 4.0 *11		
Philips Japan Ltd.	Submitted comprehensive confirmation of patents for ARIB STD-B10 3.8 版*9		
	Submitted comprehensive confirmation of patents for ARIB STD-B10 3.9 *10		
	Submitted comprehensive confirmation of patents for ARIB STD-B10 4.0 *12		
NTT DoCoMo, Inc.	Submitted comprehensive confirmation of patents for ARIB STD-B10 4.1 *13		
QUALCOMM Incorporated	Scheduled downloads: enabling background processes to receive broadcast data*15	JP 2010-528852	US 12/677,272; CN; EP;IN; KR; WO
	Submitted comprehensive confirmation of patents for ARIB STD-B10 4.8*14		
	Submitted comprehensive confirmation of patents for ARIB STD-B10 4.9 *16		
	Submitted comprehensive confirmation of patents for ARIB STD-B10 5.0 *17		

*1: Valid since version 1.0 of ARIB STD-B10
*2: Valid since version 1.1 of ARIB STD-B10
*3: Valid since version 1.2 of ARIB STD-B10
*4: Valid since version 1.3 of ARIB STD-B10
*5: Valid for the revised parts of ARIB STD-B10 Version 3.0
*6: Valid for the revised parts of ARIB STD-B10 Version 3.1
*7: Valid for the revised parts of ARIB STD-B10 Version 3.4
*8: Valid for the revised parts of ARIB STD-B10 Version 3.6
*9: Valid for the revised parts of ARIB STD-B10 Version 3.8
*10: Valid for the revised parts of ARIB STD-B10 Version 3.9
*11: Valid for the revised parts of ARIB STD-B10 Version 4.0 (accepted on November 17, 2004)

*12: Valid for the revised parts of ARIB STD-B10 Version 4.0 (accepted on December 7, 2004)
*13: Valid for the revised parts of ARIB STD-B10 Version 4.1 (accepted on March 6, 2006)
*14: Valid for the revised parts of ARIB STD-B10 Version 4.8 (accepted on April 21, 2010)
*15: Valid for the revised parts of ARIB STD-B10 Version 4.8 (accepted on October 22, 2010)
*16: Valid for the revised parts of ARIB STD-B10 Version 4.9 (accepted on March 18, 2011)
*17: Valid for the revised parts of ARIB STD-B10 Version 5.0 (accepted on November 29, 2011)
*18: Valid for the revised parts of ARIB STD-B10 Version 5.3(accepted on March 11, 2014)
*19: Valid for the revised parts of ARIB STD-B10 Version 5.4 (accepted on July 24, 2014)

TOTAL CONTENTS

Part 1

STRUCTURE OF SERVICE INFORMATION AND OPERATIONAL STANDARD OF IDENTIFIER FOR DIGITAL BROADCASTING

Part 1
STRUCTURE OF SERVICE INFORMATION AND OPERATIONAL STANDARD OF IDENTIFIER FOR DIGITAL BROADCASTING

CONTENTS

<BLANK>

1. Purpose

This standard is established for the structure of Service Information (SI) and operational standard of identifier specified in "Standard transmission system for digital broadcasting among standard television broadcasting and the like" in the Ministerial Ordinance No. 87 (hereinafter refer to as "Ordinance") issued by the Ministry of Internal Affairs and Communications in 2011, and “Structure and delivery procedure of related information, delivery procedure of PES packet, section format, TS packet, IP packet, ULE packet, MMTP packet, compressed IP packet and TLV packet, structure of transmission control signal and identifier, structure of emergency information descriptor and emergency warning broadcasting message” in the Ministerial Notification No. 233 (hereinafter refer to as "Notification") issued by the Ministry of Internal Affairs and Communications in 2014.

2. Scope

This standard is applied to the structure of, types of signal, and the data structure of the Service Information and the operational standard of the identifiers used in digital broadcasting.

3. Structure of Service Information

Service Information includes both "ARIB signals" specified in this standard and "company signals" which companies specify individually under certain conditions. Service information is transmitted by section format data structure specified in MPEG-2 Systems (Rec. ITU-T H.222.0, ISO/IEC 13818-1).

As service information is closely related to other transmission control signals specified in the Ministerial Ordinance, it is also denoted herein.

Two types of signals for service information are specified. One is a "mandatory" signal, which shall be transmitted as minimum information. And the other type is an "optional" signal, which optionally supplies information on the program. The "company signals" are "optional" signals.

Transmission control signals and service information are applied immediately when these are received, except that the targets to be controlled are indicated (for example, these are to be made active by a current_next_indicator field)

4. Types of Service Information

4.1 Types of table

Types of table used for Service Information are shown in table 4-1. Tables used for digital broadcasting other than Service Information are shown in table 4-2.

Tables established by the companies should be registered and released as company signals.

Table 4-1 Names and functions of Service Information tables

Table name	Functions
PAT* (Program Association Table)	Designates packet identifier of TS packet carrying PMT related to broadcasting program.
CAT* (Conditional Access Table)	Designates packet identifier of the TS packet conveying individual information among related information of charged broadcasting.
PMT* (Program Map Table)	Designates packet identifier of TS packets conveying each coded signal comparing a broadcasting program and packet identifier of TS packets conveying common information among related information of charged broadcasting. .
NIT* (Network Information Table)	Conveys information about the transmitting path such as modulation frequency and its relation to broadcasting programs.
SDT (Service Description Table)	Conveys information related to programmed channel such as channel name and broadcaster's name.
BAT (Bouquet Association Table)	Conveys information related to bouquet (set of programmed channels) such as names of bouquet and programmed channels in it.
EIT (Event Information Table)	Conveys information related to program such as program name, broadcasting date and time, and explanation of contents.
RST (Running Status Table)	Indicates program running status.
TDT (Time and Date Table)	Indicates present date and time.
TOT (Time Offset Table)	Indicates present date and time, and designates time difference between present time and indication time for humans.
LIT (Local Event Information Table)	Conveys information related to local event such as discrimination (time), name and explanation of local event (scene etc.) in the program.
ERT (Event Relation Table)	Indicates relationship between programs or local events, such as groups and attributes of programs or local events.
ITT (Index Transmission Table)	Describes information related to program index when sending the program.
PCAT (Partial Content Announcement Table)	Indicates transmission schedule of partial content in data broadcasting.
ST (Stuffing Table)	Makes table invalid.
BIT (Broadcaster Information Table)	Designates unit of broadcaster and SI transmission parameter of each broadcaster unit.
NBIT	Conveys network board information and reference information

(Network Board Information Table)	to gain the network board information.
LDT (Linked Description Table)	Conveys information by which reference information from other tables is collected.
AMT* (Address Map Table)	Conveys information relating service identifier for broadcasting program number to IP packet.
INT* (IP/MAC Notification Table)	Conveys information relating broadcasting programs to the IP/MAC stream composing these programs in IP packet transmission with MPEG-2 TS. INT is conveyed by private section (stream_type=0x05) of MPEG-2 Systems.
Table set by the companies	To be registered and released

*: Table specified in Ministerial Ordinance

Table 4-2 Names and functions of tables used in digital broadcasting (excluding SI.)

Table name	Function
ECM (Entitlement Control Message)*1	Conveys common information consisting of program information (information related to programs and descramble key, etc.) and control information (instruction of compulsory on/off of decoder's descramble function).
EMM (Entitlement Management Message)*1	Conveys individual information including contract information of each subscriber and work key to decrypt common information.
DCT (Download Control Table)*3	Conveys various information to separate and extract DLT.
DLT (DownLoad Table)*3	Conveys software to be downloaded.
DIT (Discontinuity Information Table)*2	Indicates changing point where possible discontinuity of service information exists in a program transmitted by partial Transport Stream.
SIT (Selection Information Table)*2	Conveys information related to programs transmitted by partial Transport Stream.
SDTT*2 (Software Download Trigger Table)	Conveys notification information for download such as service ID, schedule and receiver types for revision.
CDT (Common Data Table) *2	Conveys data commonly required for receivers and stored in non-volatile memory such as company's logo marks.
DSM-CC section *4	Conveys various data in data broadcasting.
AIT (Application Information Table)*5	Conveys dynamic control information concerning application and additional information for the execution.
DCM (Download Control Message)*6	Conveys related information of the key etc. for decrypting transmission encryption used with download.
DMM (Download Management Message)*6	Conveys key related information of download key etc. for decrypting the encryption of DCM.

*1: Table specified in the Notification

*2: Table specified in ARIB STD-B1 and B21

*3: Table specified in ARIB STD-B16

*4: Table specified in ARIB STD-B24

*5: Table specified in ARIB STD-B23 and ARIB STD-B24

*6: Table specified in ARIB STD-B61

4.2 Types of descriptor

Types of descriptor used in Service Information are shown in table 4-3, and descriptors used in digital broadcasting other than Service Information are shown in table 4-4. These descriptors are not used in INT and AIT. Descriptors set by the companies should be registered and released as "Company signal".

Table 4-3 Names and function of descriptors in Service Information

Descriptor name	Function
Conditional Access Descriptor[*1]	Describes PID conveying access controlmethod and its ECM & EMM.
Copyright Descriptor[*1]	Identifies copyright.
Network Name Descriptor	Describes network name.
Service List Descriptor[*1]	Describes programmed channels and their list of type.
Stuffing Descriptor	Secures descriptor space or invalidates descriptor.
Satellite Delivery System Descriptor[*1]	Describe physical characteristics of satellite transmission path
Terrestrial Delivery System Descriptor[*1]	Describes physical characteristics of terrestrial transmission path
Bouquet Name Descriptor	Describes name of bouquet
Service Descriptor	Describes names of programmed channel and company
Country Availability Descriptor	Describes countries intended to be available with the service
Linkage Descriptor	Describes relation to other programmed channels
NVOD Reference Descriptor	Describes a list of time-shifted programmed channels for a reference Near VOD programmed channel.
Time Shifted Service Descriptor	Describes a reference programmed channel for a Near VOD time-shifted programmed channels.
Short Event Descriptor	Describes name and brief explanation of the program.
Extended Event Descriptor	Describes detailed information about the program
Time Shifted Event Descriptor	Describes the reference program for Near VOD time-shifted programs.
Component Descriptor	Describes types and explanation related to program element signal.
Mosaic Descriptor	Describes unit of division related to mosaic (picture division) service and relation with other programmed channels and programs, etc.
Stream Identifier Descriptor	Identifies individual program element signal.
CA Identifier Descriptor	Describes available access control method.
Content Descriptor	Describes program genre.
Parental Rating Descriptor	Describes permitted minimum audience age.

Hierarchical Transmission Descriptor	Describes relation between hierarchical streams in hierarchical transmission.
Digital Copy Control Descriptor	Describes information controlling copy generation in digital recording equipment and maximum transmission rate.
Emergency Information Descriptor*1	Describes information and function necessary for emergency alarm signal.
Data Component Descriptor*1	Identifies data signal format.
System Management Descriptor*1	Identifies broadcasting/non-broadcasting.
Local Time Offset Descriptor	Describes time difference between the present time (UTC + 9 hours) and indication time to human when summer time (daylight saving time) system is introduced.
Audio Component Descriptor	Describes parameters related to audio signal among program elements.
Target Region Descriptor	Describes target region.
Hyperlink Descriptor	Describes links to other programs, program contents and program related information.
Data Content Descriptor	Describes detailed information related to contents of each data program.
Video Deocode Control Descriptor	Controls video decoding at event change.
Basic Local Event Descriptor	Describes information for local event identification.
Reference Descriptor	Describes node reference from programs and local events.
Node Relation Descriptor	Describes relation between nodes.
Short Node Information Descriptor	Describes node name and brief explanation.
STC Reference Descriptor	Describes relation between identification time of local event and STC.
Partial Reception Descriptor*1	Describes service identifier transmitted by partial reception hierarchy on terrestrial transmission path.
Series Descriptor	Describes series information among multiple events.
Event Group Descriptor	Describes grouping information of multiple events.
SI Parameter Descriptor	Describes SI transmission parameter (periodic group and resending period, etc.).
Broadcaster Name Descriptor	Describes broadcaster name.
Component Group Descriptor	Describes grouping information of plural components.
SI Prime TS Descriptor	Describes identifier information of SI prime TS and transmission parameter.
Board Information Descriptor	Describes title and text of board information.
LDT linkage Descriptor	Collects and conveys descriptions referred from other tables.
Connected Transmission Descriptor	Describes physical characteristics of connected transmission in terrestrial transmission path
TS Information Descriptor	Describes information related to TS such as allocation of the remote control key number to the TS and the transmission layer of service in the TS.
Extended Broadcaster Descriptor	Describes broadcaster information of other networks.

Logo Transmission Descriptor	Describes character string for simple logo or pointing to CDT-format logo data.
Content Availability Descriptor	Describes information to control record and output of programs.
Carousel Compatible Composite Descriptor	Applies the descriptive functions of the descriptors defined in the Data Carousel scheme.
Conditional Playback Descriptor*1,*2	Describes PID conveying conditional playback method and its ECM and EMM.
AVC Video Descriptor	Describes basic information on coding of profile, level and the like for ITU-T Rec. H.264|ISO/IEC 14496-10 Video.
AVC timing and HRD descriptor	Describes timing information for decoding ITU-T Rec. H.264|ISO/IEC 14496-10 Video.
Service Group Descriptor	Describes grouping information of multiple services.
MPEG-4 Audio Descriptor	Describes basic information for specifying coding parameters of MPEG-4 audio streams.
MPEG-4 Audio Extension Descriptor	Describes profile, level and individual specification of coding method of MPEG-4 audio streams.
Registration Descriptor	Describes information for identifying private data not specified by ISO/IEC 13818-1.
Data Broadcast Id Descriptor	Describes identification of data broadcasting.
Access Control Descriptor*1	Describes an access control method, PID conveying the ECM and EMM, and transmission information.
Area Broadcasting Information Descriptor	Describes identification, signal format of a delivery station and information related to the delivery station in broadcasting limited to an area.
Material Information Descriptor	Describes basic information of type, material name, material code, link to related information and the like regarding materials composing programs.
HEVC Video Descriptor	Describes basic information such as profile and level for ITU-T Rec. H.265|ISO/IEC 23008-2 Video.
Hierarchy Descriptor*1	Describes information for identifying program elements including hierarchically-coded video stream components.
Hybrid Information Descriptor	Describes information for designating a communication stream used with broadcast services
Scrambler Descriptor*1	Describes information for designating a cypher algorithm of a scramble system.
Descriptor set by the companies	To be registered and released

*1: Descriptor specified in the Notification

*2: Descriptor defined in ARIB STD-B25

Table 4-4 Names and functions of descriptors used in digital broadcasting (excluding Service Information)

Descriptor	Function
Partial Transport Stream Descriptor*1	Describes partial Transport Stream.
Network Identification Descriptor*1	Describes network identifier.
Partial Transport Stream Time Descriptor*1	Describes partial Transport Stream time
Download Content Descriptor*1	Describe attribute information such as size and types of downloaded contents and downloaded ID.
CA EMM TS Descriptor*2	Indicates the specific channel when the EMM transmission is made by the specific channel method
CA Contract Information Descriptor*2	Describes conditional access service type (tear/flat/PPV) of scheduled program and permission of reception and recording.
CA Service Descriptor*2	Describes charged broadcast service provider for presenting automatic indication message.
Carousel Identifier Descriptor*3	Describes Carousel Identifier specified in ISO/IEC 13818-6.
Association Tag Descriptor*3	Describes association Tag information specified in ISO/IEC 13818-6.
Deferred Association tags Descriptor*3	Describes Association Tags information of other broadcasting programs specified in ISO/IEC 13818-6.
Network Download Content Descriptor*1	Describes source information, etc. of contents downloaded via network.
Download Protection Descriptor*4	Describes protection information, etc. regarding download.
CA Startup Descriptor*4	Describes information for starting up CAS programs with conditional access function.
Character Code Descriptor*5	Describes component of text information including SIT

*1: Descriptor specified in ARIB STD-B1, B21

*2: Descriptor specified in ARIB STD-B25

*3: Descriptor used in ARIB STD-B23

*4: Descriptor specified in ARIB STD-B61

*5: Descriptor specified in ARIB STD-B63

4.3 Types of descriptor used in INT

Types of descriptor used in INT are shown in table 4-5. The descriptors defined in tables 4-3 and 4-4 shall not be used in INT.

Table 4-5 Names and functions of descriptors used in INT

Descriptor	Function
Target Smartcard Descriptor	Indicates receiver with smartcard.
Target IP Address Descriptor	Indicates IPv4 address of target.
Target IPv6 Address Descriptor	Indicates IPv6 address of target.
IP/MAC Platform Name Descriptor	Describes name of IP/MAC platform.
IP/MAC Platform Provider Name Descriptor	Describes name of IP/MAC platform provider.
IP/MAC Stream Location Descriptor*1	Describes location information of IP stream

*1: Descriptor specified by Notification

4.4 Types of descriptor used in AIT

Types of descriptor used in AIT are shown in table 4-6. The descriptors defined in tables 4-3 and 4-4 shall not be used in AIT. The descriptors used in AIT are specified in ARIB STD-B24, Part 4, Chapter 5, 5.3.

Table 4-6 Names and functions of descriptors used in AIT

Descriptor	Function
Application descriptor	Describes information of application
Transport protocol descriptor	Indicates transport protocol of broadcasting and communication and the like, and location information of application depending on transport protocol.
Simple application location descriptor	Indicates the detail of acquisition destination of application
Application boundary and permission descriptor	Sets application boundary, and rights of broadcasting resource access for every URL.
Autostart priority descriptor	Indicates priority of application starting up.
Cache control info descriptor	Indicates information of cache which temporally holds resources composing application.
Randomized latency descriptor	Sets timing of application control.
External application control descriptor	Indicates access rights of broadcasting resource with external application.
Playback application descriptor	Describes information of application starting up at playback of stored content.
Simple playback application location descriptor	Indicates the detail of acquisition destination of application starting up at playback of stored content.
Application expiration descriptor	Indicates start-up expiration of application designated by playback application descriptor and simple playback application location descriptor.

5. Transmission of service information

5.1 PID for tables

PID values of the Transport Stream packets carrying tables as specified in table 4-1 and table 4-2 shall be as shown in table 5-1.

PID values of Transport Stream packets carrying tables set by companies can be set to any value as long as they do not prevent transmission of the signal specified by the Ministerial Ordinance and Notification or ARIB signal. The PID values shall be registered and released as the company signals.

Table 5-1 Allocation of PID

Table	PID
PAT*1	0x0000
PMT*1	Indirect designation by PAT
CAT*1	0x0001
ECM*1	Indirect designation by PMT
EMM*1	Indirect designation by CAT
NIT*1	0x0010
SDT	0x0011
BAT	0x0011
EIT	0x0012
EIT (terrestrial digital television broad casting, multimedia broadcasting)*8	0x0012, 0x0026, 0x0027
RST	0x0013
TDT	0x0014
TOT	0x0014
DCT*3	0x0017
DLT*3	Indirect designation by DCT
DIT*2	0x001E
SIT*2	0x001F
LIT	Indirect designation by PMT*6 or 0x0020*5
ERT	Indirect designation by PMT*6 or 0x0021*5
ITT	Indirect designation by PMT
PCAT	0x0022
SDTT*2	0x0023
SDTT (terrestrial digital television broad casting, multimedia broadcasting) *2,*8	0x0023, 0x0028
BIT	0x0024
NBIT	0x0025
LDT	0x0025
CDT	0x0029
Fragmented TLV packet*11	0x002D
AMT	0x002E
Multiple frame header information*7	0x002F

DSM-CC section*4	Indirect designation by PMT
AIT*9	Indirect designation by PMT
ST	Exclude 0x0000, 0x0001, 0x0014
INT*1	Indirect designation by PMT
DCM*10	Indirect designation by PMT
DMM*10	Indirect designation by SDTT
Null packet*1	0x1FFF

*1: According to the Notification
*2: Specified in ARIB STD-B1, B21
*3: Specified in ARIB STD-B16
*4: Specified in ARIB STD-B24
*5: When used as program group index
*6: When used as index within program
*7: In accordance with the Notification No. 95 of Ministry of Internal Affairs and Communications in 2015 and JCTEA STD-002
*8: In accordance with the operational guidelines for the assignment of PID values to each hierarchy
*9: Specified in ARIB STD-B23 and ARIB STD-B24
*10: Specified in ARIB STD-B61
*11: JCTEA STD-002

5.2 Table identifier and transmission standard

Allocation of table ID specified in table 4-1 and table 4-2 is shown in table 5-2. Among them, the transmission level of the Service Information tables is shown in table 5-2 in Part 2.

Table ID value of tables set by the companies can be set in the range of 0x90 to 0xBF. The table ID value shall be registered and released as the company signal.

Table 5-2 Allocation of table_ID values

table_id	Table
0x00	PAT*1
0x01	CAT*1
0x02	PMT*1
0x3A – 0x3F	DSM-CC section*4
0x40	NIT (Actual network)*1
0x41	NIT (Other network)*1
0x42	SDT (Actual stream)
0x46	SDT (Other stream)
0x4A	BAT
0x4C	INT*1
0x4E	EIT (Present and next program of actual stream)
0x4F	EIT (Present and next program of other stream)
0x50 – 0x5F	EIT (Actual stream, schedule)
0x60 – 0x6F	EIT (Other stream, schedule)
0x70	TDT
0x71	RST

table_id	Table
0x72	ST
0x73	TOT
0x74	AIT*4 (AIT controlled application)
0x75	AIT*5 (ARIB-J)
0x7E	DIT*2
0x7F	SIT*2
0x82 – 0x83	ECM*1
0x84 – 0x85	EMM*1
0x87 – 0x88	DCM*6
0x89 – 0x8A	DMM*6
0xC0	DCT*3
0xC1	DLT*3
0xC2	PCAT
0xC3	SDTT*1
0xC4	BIT
0xC5	NBIT (Network board information body)
0xC6	NBIT (Reference information to gain network board information)
0xC7	LDT
0xC8	CDT*2
0xD0	LIT
0xD1	ERT
0xD2	ITT
0x90 – 0xBF	Selectable range for table ID value set by companies
0xFE	AMT

*1: According to the Notification
*2: Specified in ARIB STD-B1, B21
*3: Specified in ARIB STD-B16
*4: Specified in ARIB STD-B24
*5: Specified in ARIB STD-B23
*6: Specified in ARIB STD-B61

5.3 Identifier of descriptors

Tag values of descriptors specified in tables 4-3 and 4-4 are shown in table 5-3. Transmission standard of descriptors for the Service Information is shown in table 6-1 in Part 2. Descriptors used in INT and tag values of descriptors are shown in tables 5-4 and 5-5, respectively.

When the number of usable descriptors needs to be increased, the method of using composite descriptors shown in Part 2 Annex L shall be used with a tag value of 0xDF. The tag value of subdescriptor shall be determined for each composite descriptor.

The tag value of descriptors set by the companies can be set in the range of 0x80 to 0xBF. The tag value shall be registered and released as the company signal.

Table 5-3 Allocation of descriptors-tag values

Tag value	Descriptor
0x04	Hierarchy descriptor*1, 8
0x05	Registration descriptor*8
0x09	Conditional access descriptor*1
0x0D	Copyright descriptor*1
0x13	Carousel identifier descriptor*7
0x14	Association tag descriptor*7
0x15	Deferred association tags descriptor*7
0x1C	MPEG-4 audio descriptor*8
0x28	AVC video descriptor*8
0x2A	AVC timing and HRD descriptor*8
0x2E	MPEG-4 audio extended descriptor*8
0x38	HEVC Video descriptor*8
0x40	Network name descriptor*2
0x41	Service list descriptor*1
0x42	Stuffing descriptor
0x43	Satellite delivery system descriptor*1
0x44	Cable distribution system descriptor*4
0x47	Bouquet name descriptor
0x48	Service descriptor*2
0x49	Country availability descriptor
0x4A	Linkage descriptor
0x4B	NVOD reference descriptor
0x4C	Time shifted service descriptor*2
0x4D	Short event descriptor*2
0x4E	Extended event descriptor
0x4F	Time shifted event descriptor*2
0x50	Component descriptor
0x51	Mosaic descriptor
0x52	Stream identifier descriptor
0x53	CA identifier descriptor
0x54	Content descriptor
0x55	Parental rating descriptor
0x58	Local time offset descriptor
0x63	Partial Transport Stream descriptor*3
0x66	Data Broadcasting Identification descriptor
0x67	Material Information descriptor
0x68	Hybrid Information descriptor
0x80 – 0xBF	Selectable range for tag value of company-defined descriptor
0xC0	Hierarchical transmission descriptor

<table>
<tr><th>Tag value</th><th colspan="2">Descriptor</th></tr>
<tr><td>0xC1</td><td colspan="2">Digital copy control descriptor</td></tr>
<tr><td>0xC2</td><td colspan="2">Network identification descriptor[*3]</td></tr>
<tr><td>0xC3</td><td colspan="2">Partial Transport Stream time descriptor[*3]</td></tr>
<tr><td>0xC4</td><td colspan="2">Audio component descriptor</td></tr>
<tr><td>0xC5</td><td colspan="2">Hyperlink descriptor</td></tr>
<tr><td>0xC6</td><td colspan="2">Target region descriptor</td></tr>
<tr><td>0xC7</td><td colspan="2">Data content descriptor</td></tr>
<tr><td>0xC8</td><td colspan="2">Video decode control descriptor</td></tr>
<tr><td>0xC9</td><td colspan="2">Download content descriptor[*3]</td></tr>
<tr><td>0xCA</td><td colspan="2">CA_EMM_TS descriptor[*5]</td></tr>
<tr><td>0xCB</td><td colspan="2">CA contract information descriptor[*5]</td></tr>
<tr><td>0xCC</td><td colspan="2">CA service descriptor[*5]</td></tr>
<tr><td>0xCD</td><td colspan="2">TS information descriptor</td></tr>
<tr><td>0xCE</td><td colspan="2">Extended broadcaster descriptor</td></tr>
<tr><td>0xCF</td><td colspan="2">Logo transmission descriptor</td></tr>
<tr><td>0xD0</td><td colspan="2">Basic local event descriptor</td></tr>
<tr><td>0xD1</td><td colspan="2">Reference descriptor</td></tr>
<tr><td>0xD2</td><td colspan="2">Node relation descriptor</td></tr>
<tr><td>0xD3</td><td colspan="2">Short node information descriptor</td></tr>
<tr><td>0xD4</td><td colspan="2">STC reference descriptor</td></tr>
<tr><td>0xD5</td><td colspan="2">Series descriptor</td></tr>
<tr><td>0xD6</td><td colspan="2">Event group descriptor</td></tr>
<tr><td>0xD7</td><td colspan="2">SI parameter descriptor</td></tr>
<tr><td>0xD8</td><td colspan="2">Broadcaster name descriptor</td></tr>
<tr><td>0xD9</td><td colspan="2">Component group descriptor</td></tr>
<tr><td>0xDA</td><td colspan="2">SI prime TS descriptor</td></tr>
<tr><td>0xDB</td><td colspan="2">Board information descriptor</td></tr>
<tr><td>0xDC</td><td colspan="2">LDT linkage descriptor</td></tr>
<tr><td>0xDD</td><td colspan="2">Connected transmission descriptor</td></tr>
<tr><td>0xDE</td><td colspan="2">Content availability descriptor</td></tr>
<tr><td rowspan="3">0xDF</td><td colspan="2">For tag value extension</td></tr>
<tr><td>Subdescriptor tag value</td><td>Descriptor</td></tr>
<tr><td>0x00 – 0xFF</td><td>Undefined</td></tr>
<tr><td>0xE0</td><td colspan="2">Service group descriptor</td></tr>
<tr><td>0xE1</td><td colspan="2">Area broadcasting information descriptor</td></tr>
<tr><td>0xE2</td><td colspan="2">Network download content descriptor[*3]</td></tr>
<tr><td>0xE3</td><td colspan="2">Download protection descriptor[*9]</td></tr>
<tr><td>0xE4</td><td colspan="2">CA startup descriptor[*9]</td></tr>
<tr><td>0xE5</td><td colspan="2">Character Code descriptor[*11]</td></tr>
<tr><td>0xE5 – 0xF2</td><td colspan="2">Undefined</td></tr>
<tr><td>0xF3</td><td colspan="2">Cable multiple carrier transmission delivery system descriptor[*10]</td></tr>
<tr><td>0xF4</td><td colspan="2">Advanced cable delivery system descriptor[*10]</td></tr>
<tr><td>0xF5</td><td colspan="2">Scrambler descriptor[*1]</td></tr>
</table>

Tag value	Descriptor
0xF6	Access control descriptor[*1]
0xF7	Carousel compatible composite descriptor[*1]
0xF8	Conditional playback descriptor[*1, *5]
0xF9	Cable TS division system descriptor[*6]
0xFA	Terrestrial delivery system descriptor[*1]
0xFB	Partial reception descriptor[*1]
0xFC	Emergency information descriptor[*1]
0xFD	Data component descriptor[*1]
0xFE	System management descriptor*[*1]

*1: In accordance with the Notification

*2: Can be alternated to descriptor implying this function, set by the company

*3: Descriptor specified in ARIB STD-B1 and B21

*4: In accordance with the Notification No. 312 of the Ministry of Internal Affairs and Communications in 2011

*5: Descriptor specified in ARIB STD-B25

*6: In accordance with the Notification No. 312 of the Ministry of Internal Affairs and Communications in 2011 and JCTEA STD-003

*7: Descriptor used in ARIB STD-B23

*8: In accordance with ITU-T Rec. H.222.0 | ISO/IEC 13818-1

*9: Descriptor specified in ARIB STD-B61

*10: In accordance with the Notification No. 92 of Ministry of Public Management, Home Affairs, Posts and Telecommunications in 2015 and JCTEA STD-003

*11: Descriptor specified in ARIB STD-B63

5.4 Identifier of descriptors used in INT

Tag values of descriptors used in INT are shown in table 5-4. Specifications of delivering these descriptors are shown in table 6-100 in Part 2.

Table 5-4 Allocation of tag values of descriptors used in INT

Tag value	Descriptor
0x06	Target smartcard descriptor
0x09	Target IP address descriptor
0x0A	Target IPv6 address descriptor
0x0C	IP/MAC platform name descriptor
0x0D	IP/MAC platform provider name descriptor
0x13	IP/MAC stream location descriptor

5.5 Identifier of descriptors used in AIT

Tag values of descriptors used in AIT are specified in ARIB STD-B24, Part 4, Chapter 5, 5.3 and shown in table 5-5.

Table 5-5 Allocation of tag values of descriptors used in AIT

Tag value	Descriptor
0x00	Application descriptor
0x02	Transport protocol descriptor
0x15	Simple application location descriptor
0x30	Application boundary and permission descriptor
0x31	Autostart priority descriptor
0x32	Cache control info descriptor
0x33	Randomized latency descriptor
0x34	External application control descriptor
0x35	Playback application descriptor
0x36	Simple playback application location descriptor
0x37	Application expiration descriptor

6. Data structure of Service Information

6.1 Data structure of tables

Tables specified in table 4-1 shall be in accordance with the section format specified in MPEG-2 Systems (Rec. ITU-T H.222.0, ISO/IEC 13818-1), and its data structure shall be in accordance with figures 6.1-1 to 6.1-20.

Data structure of tables specified by companies shall be registered and released as the company signal.

Semantics and usage of each segment of the data structure are specified in Parts 2 and 3 of this standard.

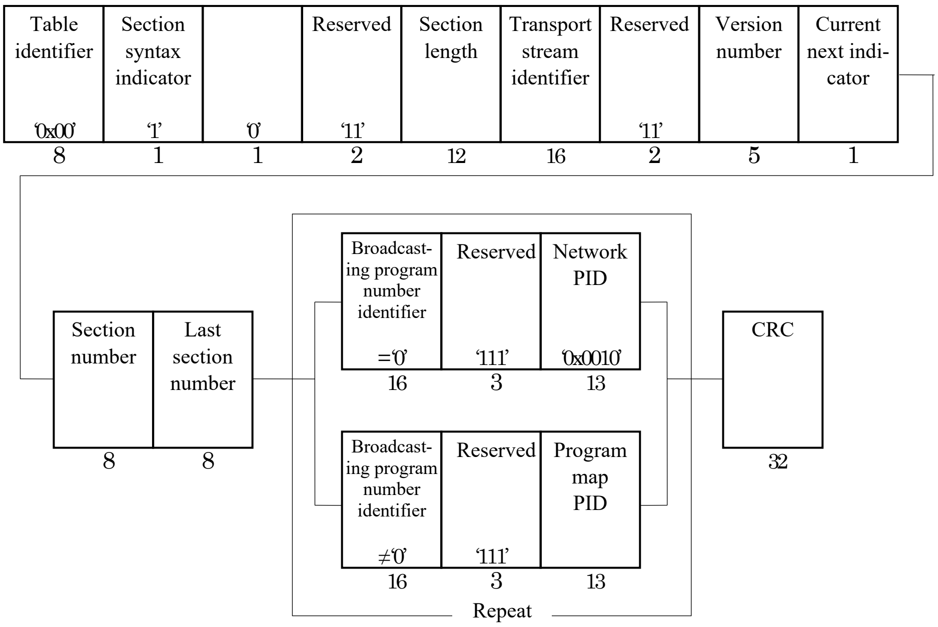


Figure 6.1-1 Data structure of PAT

Table identifier	Section syntax indicator		Reserved	Section length	Broadcast-ing program number identifier	Reserved	Version number	Current next indi-cator
'0x02'	'1'	'0'	'11'			'11'		
8	1	1	2	12	16	2	5	1

Section number	Last section number	Reserved	PCR_PID	Reserved	Program infor-mation length	De-scriptors field 1
		'111'		'1111'		
8	8	3	13	4	12	8xN

Stream type identifier	Reserved	Elemen-tary stream PID	Reserved	ES infor-mation length	De-scriptors field 2	CRC
	'111'		'1111'			
8	3	13	4	12	8xN	32

Repeat

Figure 6.1-2 Data structure of PMT

Table identifier	Section syntax indicator		Reserved	Section length	Reserved		Version number	Current next indi-cator
'0x01'	'1'	'0'	'11'		'111111' '111111' '111111'			
8	1	1	2	12	18	2	5	1

Section number	Last section number	De-scriptors field	CRC
8	8	8xN	32

Figure 6.1-3 Data structure of CAT

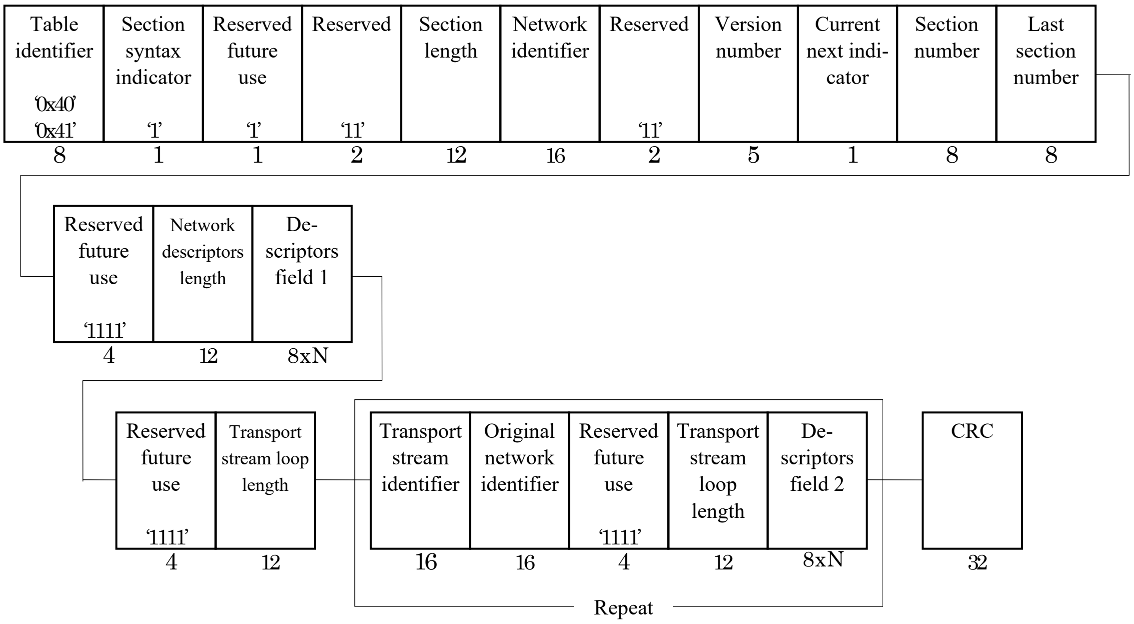


Figure 6.1-4 Data structure of NIT

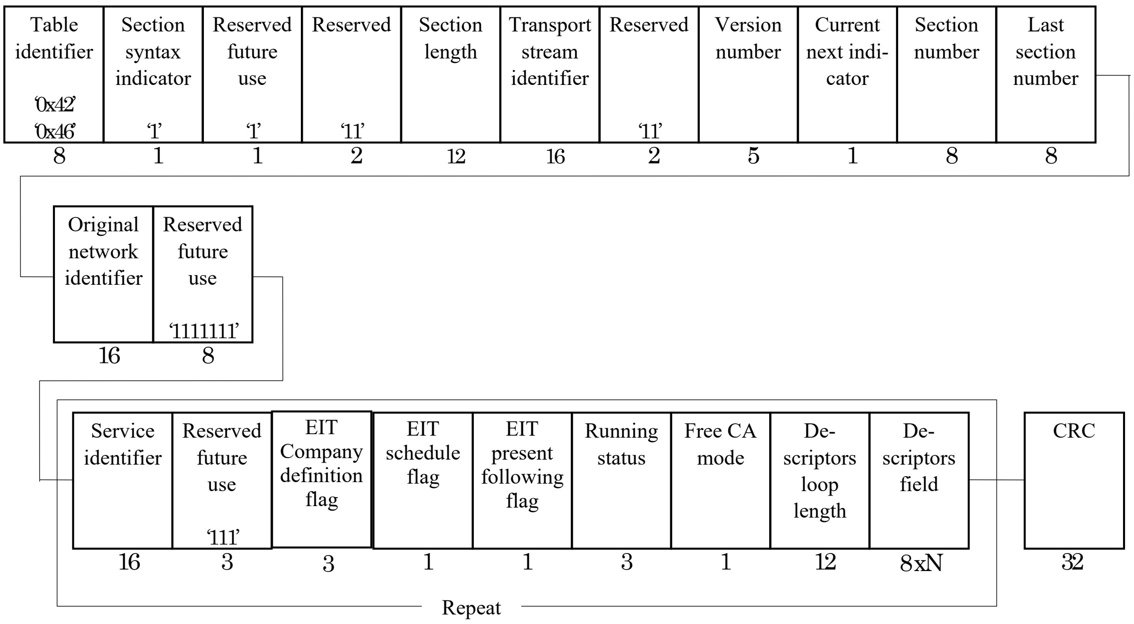


Figure 6.1-5 Data structure of SDT

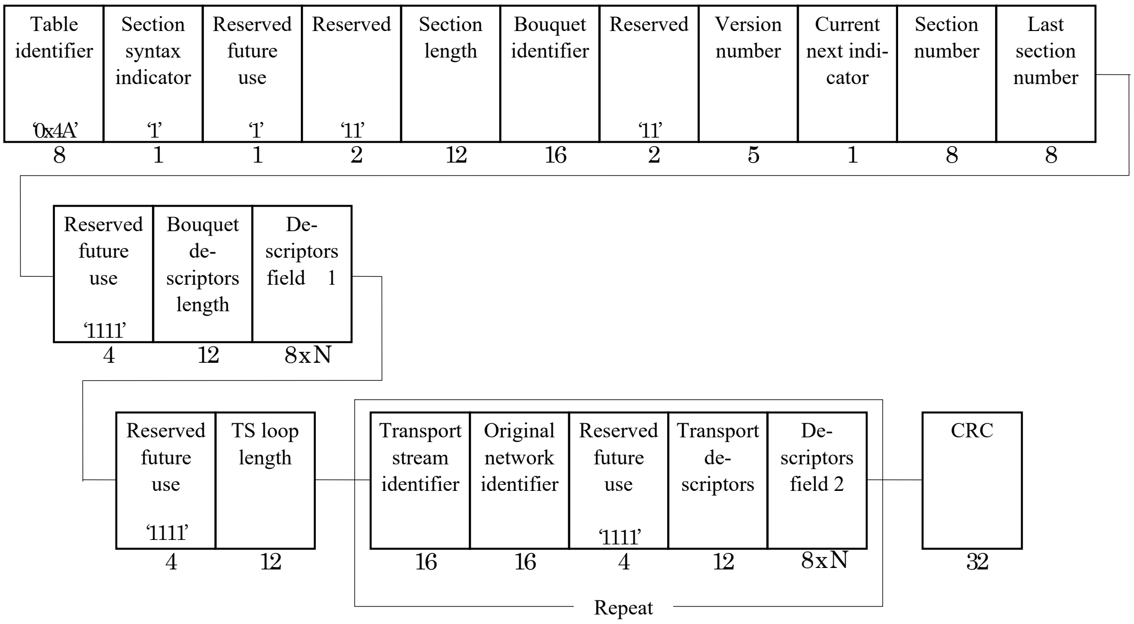


Figure 6.1-6 Data structure of BAT

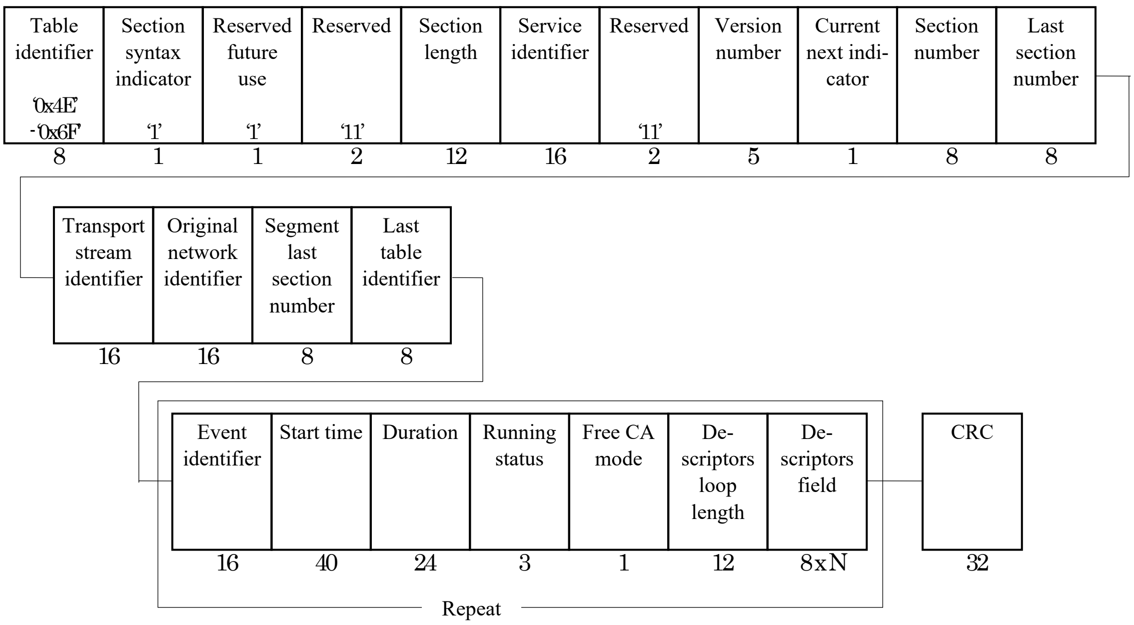


Figure 6.1-7 Data structure of EIT

Table identifier	Section syntax indicator	Reserved future use	Reserved	Section length	JST time
'0x70'	'0'	'1'	'11'		
8	1	1	2	12	40

Figure 6.1-8 Data structure of TDT

Table identifier	Section syntax indicator	Reserved future use	Reserved	Section length	JST time	Reserved	De-scriptors loop length	De-scriptors field	CRC
'0x73'	'0'	'1'	'11'			'1111'			
8	1	1	2	12	40	4	12	8xN	32

Figure 6.1-9 Data structure of TOT

Table identifier	Section syntax indicator	Reserved future use	Reserved	Section length	Transport stream identifier	Original network identifier	Service identifier	Event identifier	Reserved future use	Running status
'0x71'	'0'	'1'	'11'						'11111'	
8	1	1	2	12	16	16	16	16	5	3
					Repeat					

Figure 6.1-10 Data structure of RST

Table identifier	Section syntax indicator	Reserved future use	Reserved	Section length	Date byte
'0x72'		'1'	'11'		
8	1	1	2	12	8xN

Figure 6.1-11 Data structure of ST

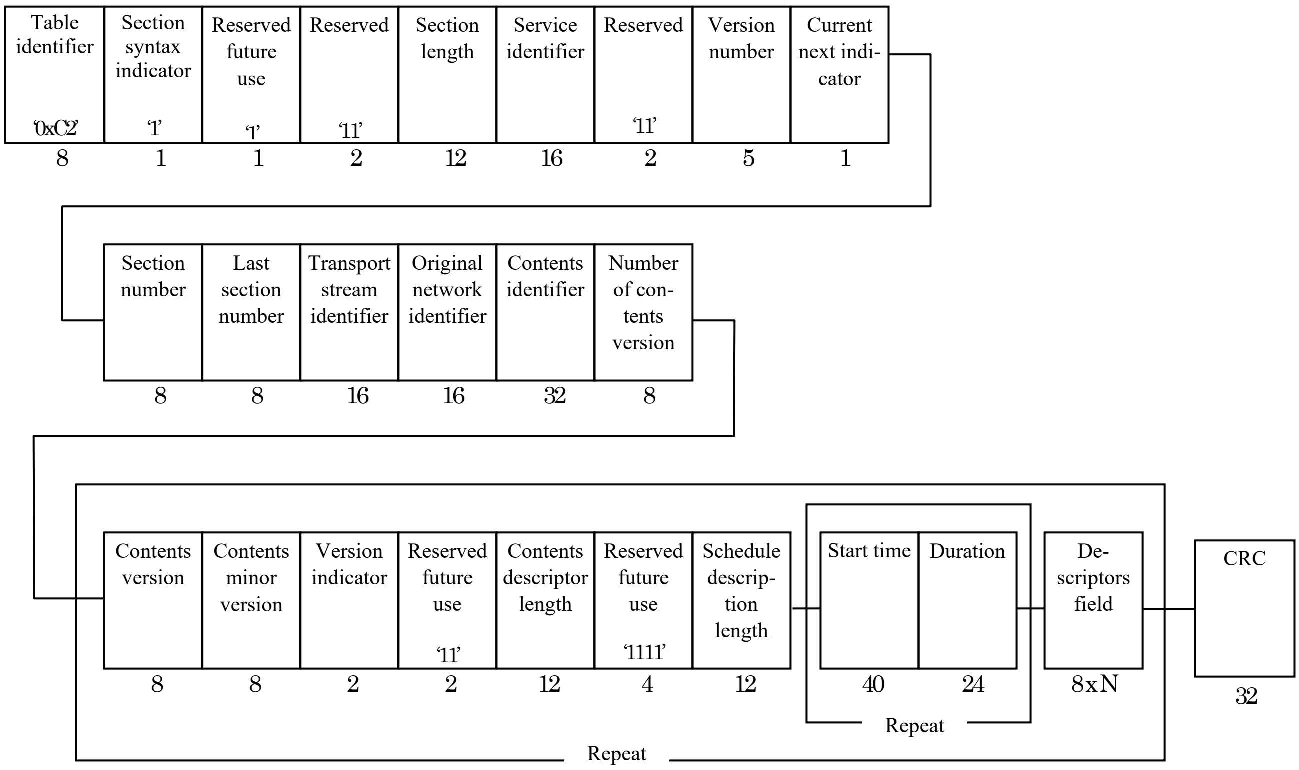


Figure 6.1-12 Data structure of PCAT

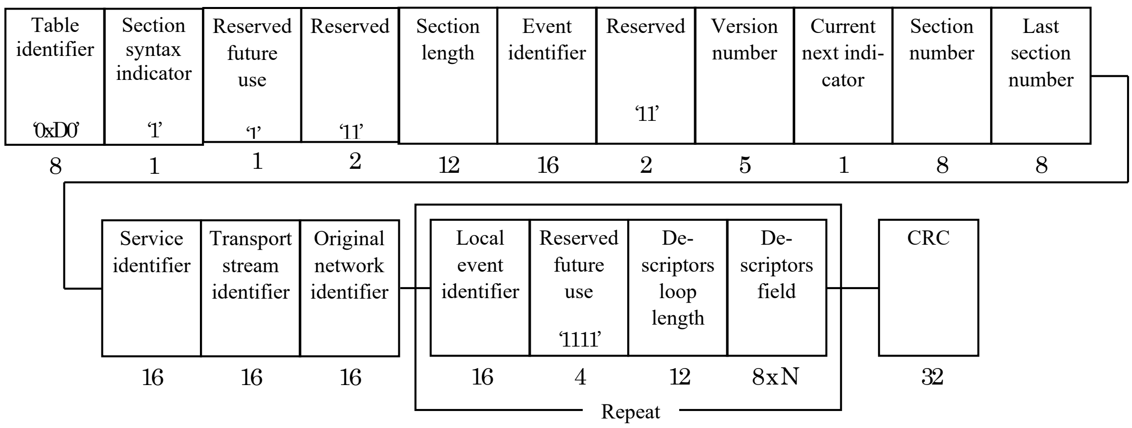


Figure 6.1-13 Data structure of LIT

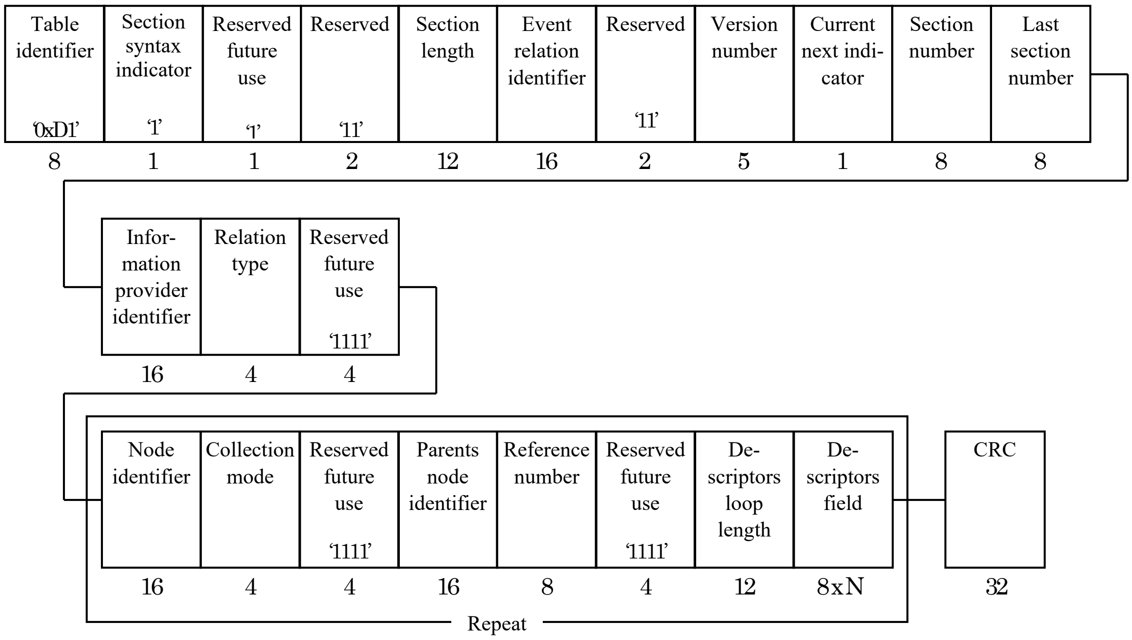


Figure 6.1-14 Data structure of ERT

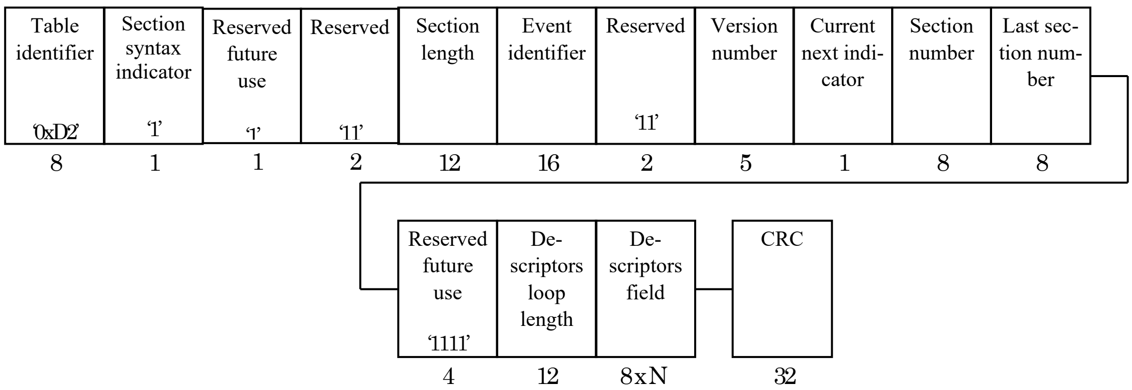


Figure 6.1-15 Data structure of ITT

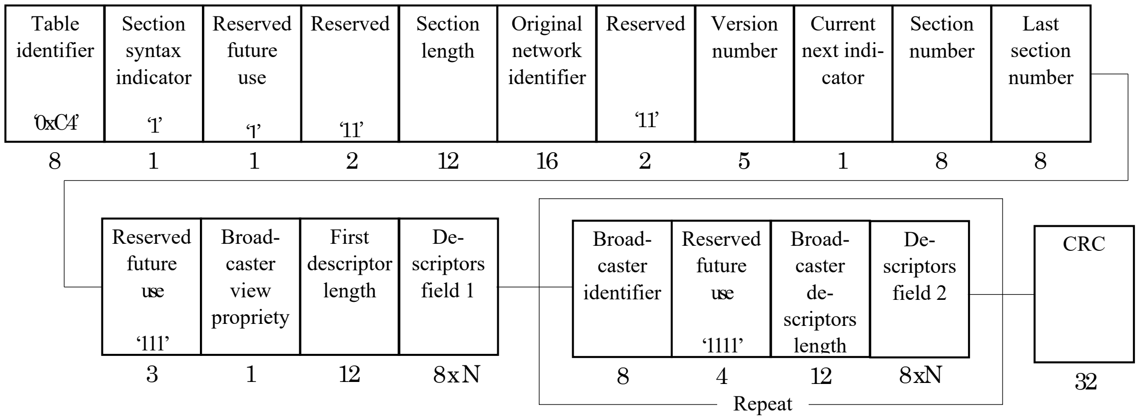


Figure 6.1-16 Data structure of BIT

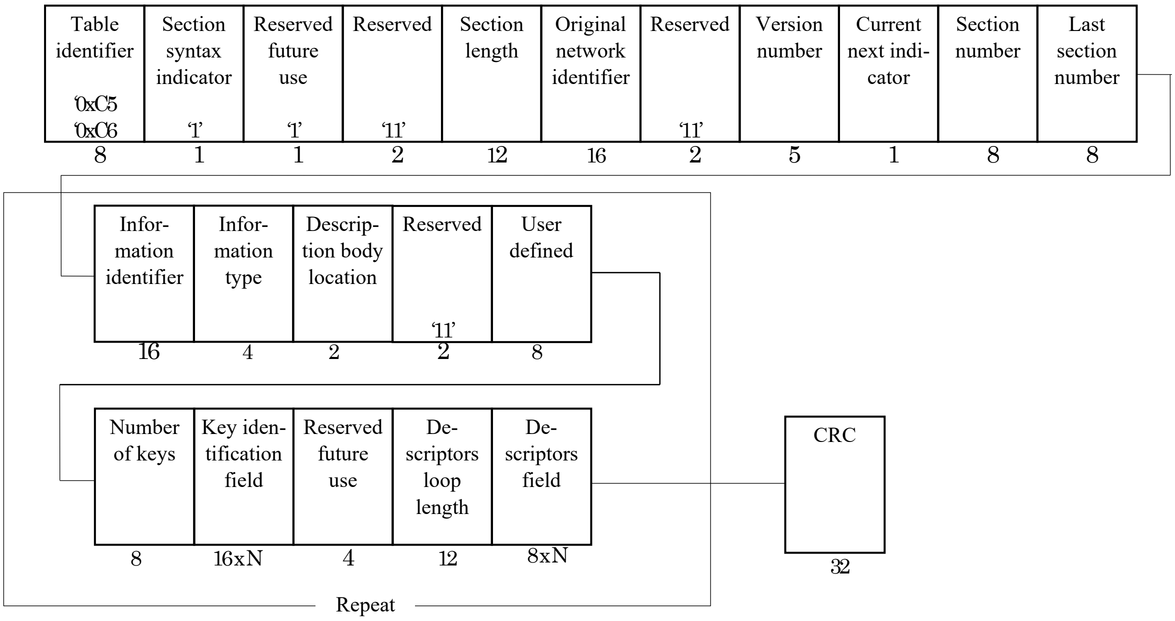


Figure 6.1-17 Data structure of NBIT

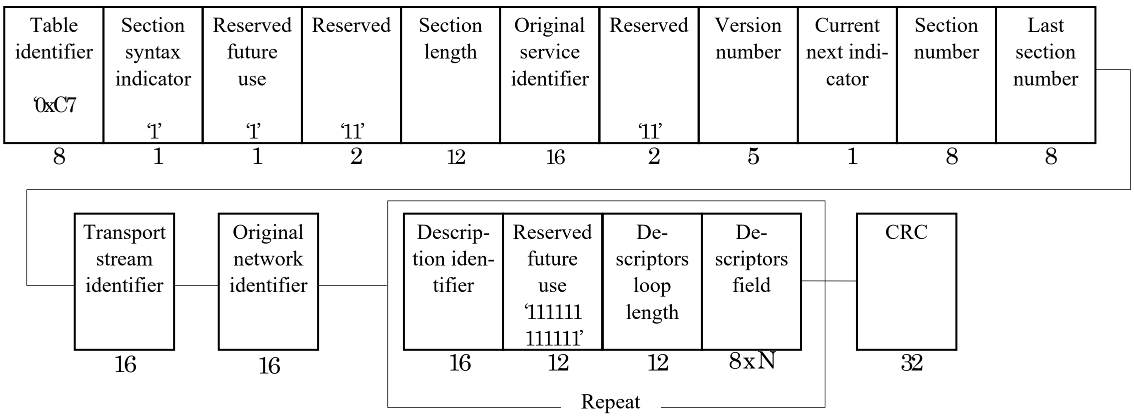


Figure 6.1-18 Data structure of LDT

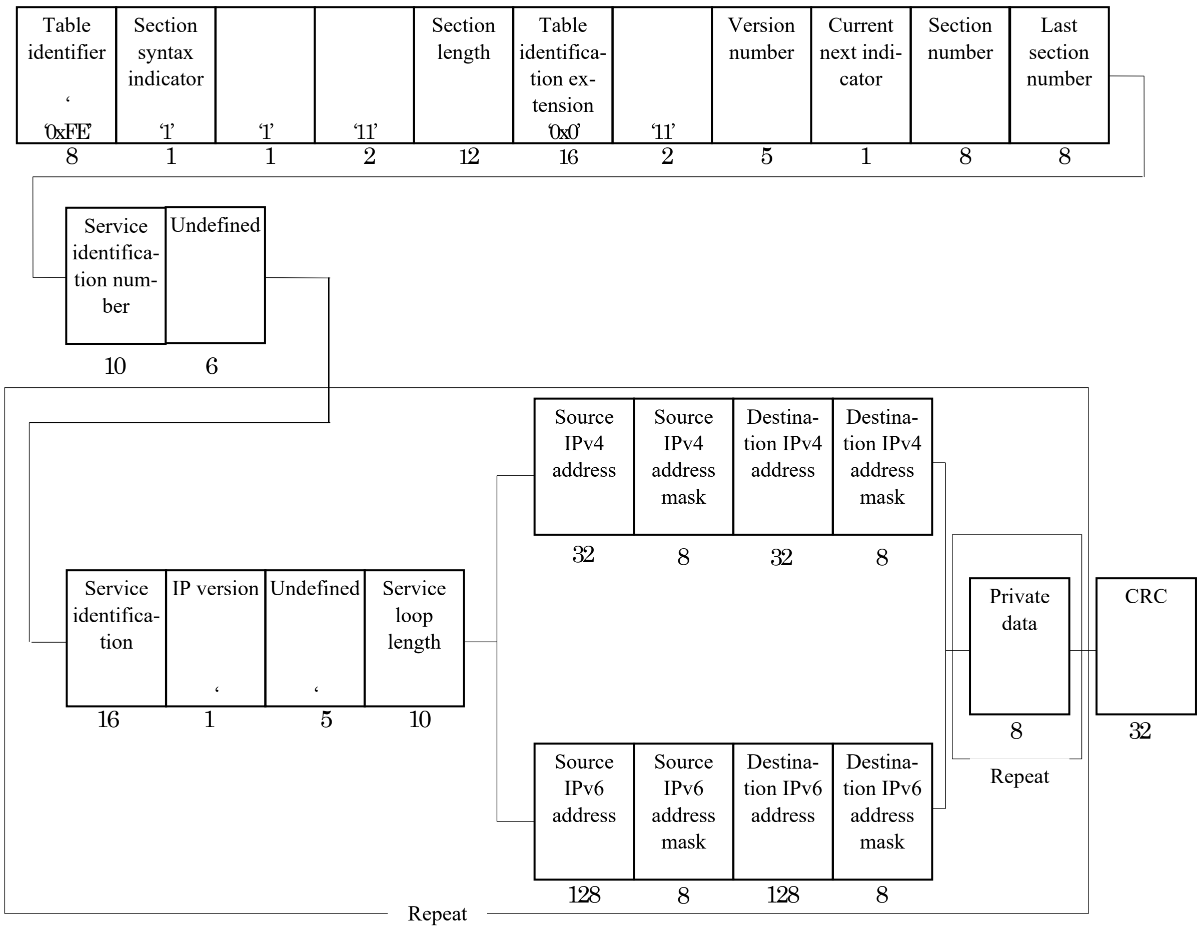


Figure 6.1-19 Data structure of AMT

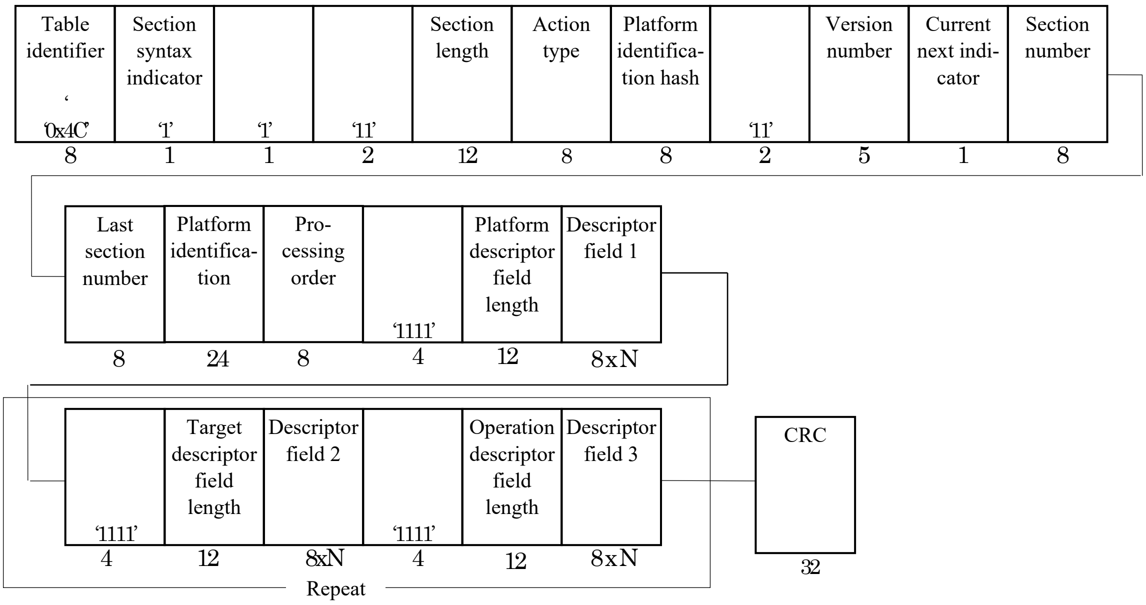


Figure 6.1-20 Data structure of INT

6.2 Data structure of descriptor

Descriptor specified in table 4-3 shall be in accordance with the format specified in MPEG-2 Systems (Rec. ITU-T H.222.0, ISO/IEC 13818-1) and its data structure shall be in accordance with figures 6.2-1 to 6.2-68.

Data structure of descriptors set by the companies shall be registered and released as the company signal.

Semantics and usage of each segment of data structure are specified in Parts 2 and 3 of this standard and in the operational guidelines.

Descrip-tion tag	Descrip-tion length	Conditional access method identifier		Condi-tional access PID	Private data
'0x09'			'111'		
8	8	16	3	13	8xN

Figure 6.2-1 Data structure of Conditional access method descriptor

Descriptor tag	Descrip-tion length	char (network name)
'0x40'		
8	8	8xN

Figure 6.2-2 Data structure of Network name descriptor

Descriptor tag	Descrip-tion length	Service id	Service type
'0x41'			
8	8	16	8
		Repeat	

Figure 6.2-3 Data structure of Service list descriptor

Descriptor tag	Descrip-tion length	Stuffing byte
'0x42'		
8	8	8xN

Figure 6.2-4 Data structure of Stuffing descriptor

Descriptor tag	Descriptor length	Frequency	orbital position	west east flag	Polariza-tion	Modula-tion	Symbol rate	FEC inner
'0x43'								
8	8	32	16	1	2	5	28	4

Figure 6.2-5 Data structure of Satellite delivery system descriptor

Descriptor tag	Descriptor length	char (bouquet name)
'0x47'		
8	8	8xN

Figure 6.2-6 Data structure of Bouquet name descriptor

Descriptor tag	Descriptor length	Service type	service provider name length	char (service provider name)	Service name length	char (service name)
'0x48'						
8	8	8	8	8xN	8	8xN

Figure 6.2-7 Data structure of Service descriptor

Descriptor tag	Descriptor length	country ability flag (1: possi-ble, 0: impossible)	Reserved Future use	Country code (Repeat)
'0x49'				
8	8	1	7	24

Figure 6.2-8 Data structure of Country availability descriptor

Descriptor tag	Descriptor length	Transport stream id	Original network id	Service id	Linkage type	Private data byte
'0x4A'						
8	8	16	16	16	8	8xN

Figure 6.2-9 Data structure of Linkage descriptor

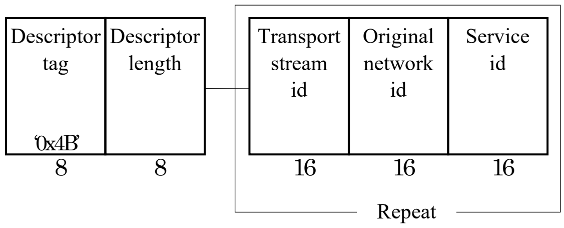


Figure 6.2-10 Data structure of NVOD reference descriptor

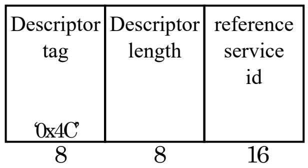


Figure 6.2-11 Data structure of Time shifted service descriptor

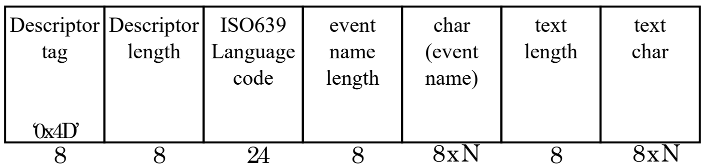


Figure 6.2-12 Data structure of short Event descriptor

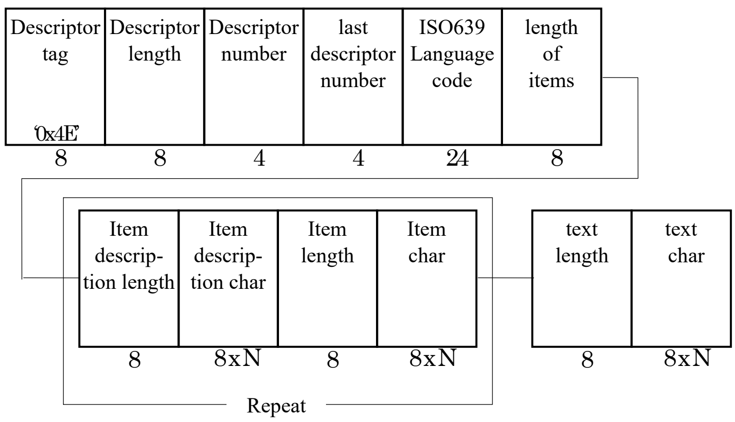


Figure 6.2-13 Data structure of Extended event descriptor

Descriptor tag ‘0x4F’	Descriptor length	reference service id	reference event id
8	8	16	16

Figure 6.2-14 Data structure of Time shifted event descriptor

Descriptor tag ‘0x50’	Descriptor length	Reserved future use	Stream content	Compo-nent type	Compo-nent tag	ISO639 Language code	text char
8	8	4	4	8	8	24	8xN

Figure 6.2-15 Data structure of Component descriptor

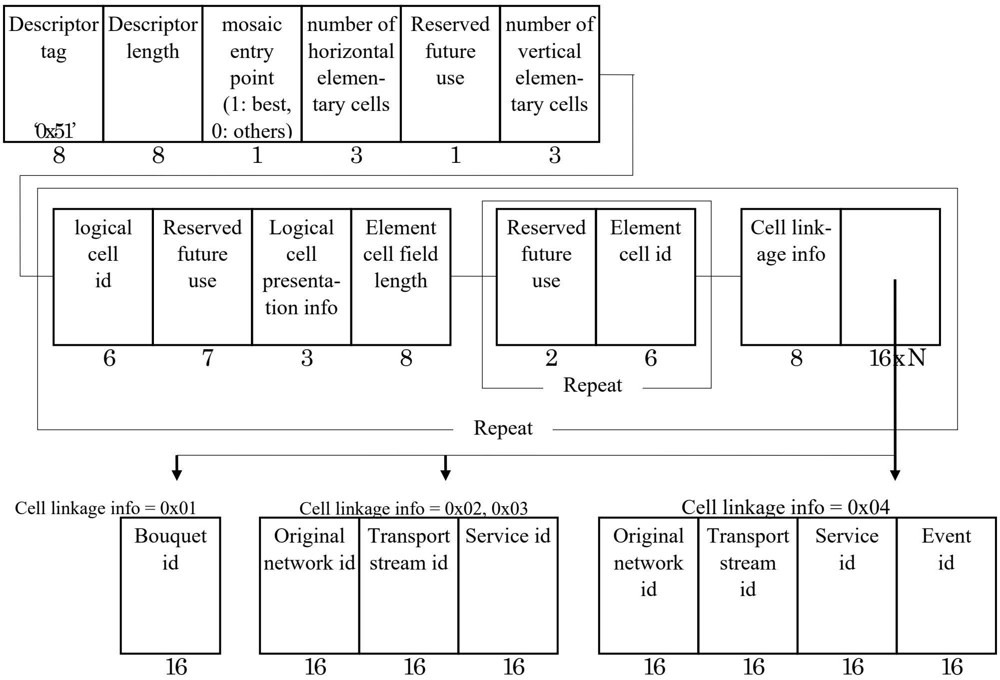


Figure 6.2-16 Data structure of Mosaic descriptor

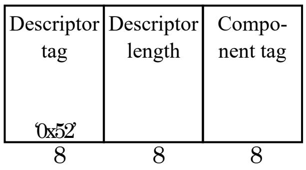


Figure 6.2-17 Data structure of Stream identifier descriptor

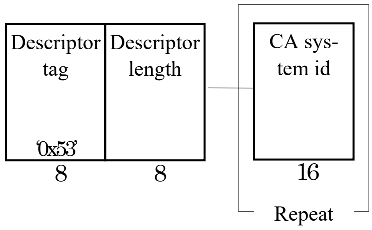


Figure 6.2-18 Data structure of CA identifier descriptor

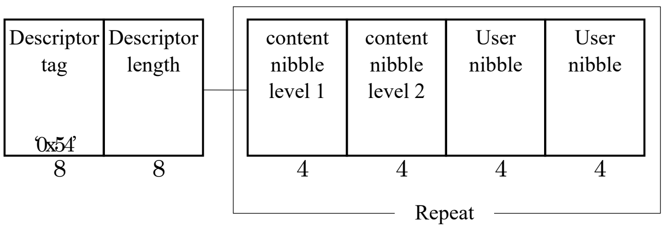


Figure 6.2-19 Data structure of Content descriptor

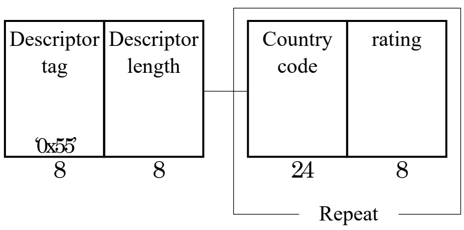


Figure 6.2-20 Data structure of Parental rating descriptor

Descriptor tag	Descriptor length	Data component id	Additional identifier info
'0xFD'			
8	8	16	8xN

Figure 6.2-21 Data structure of Data component descriptor

Descriptor tag	Descriptor length	System management id	Additional identifier info
'0xFE'			
8	8	16	8xN

Figure 6.2-22 Data structure of System management descriptor

Descriptor tag	Descriptor length	Copyright id	Copyright additional info
'0x0D'			
8	8	32	8xN

Figure 6.2-23 Data structure of Copyright descriptor

Descriptor tag	Descriptor length	Reserved future use	Hierarchical level	Reserved future use	Reference PID
'0xC0'					
8	8	7	1	3	13

Figure 6.2-24 Data structure of Hierarchical transmission descriptor

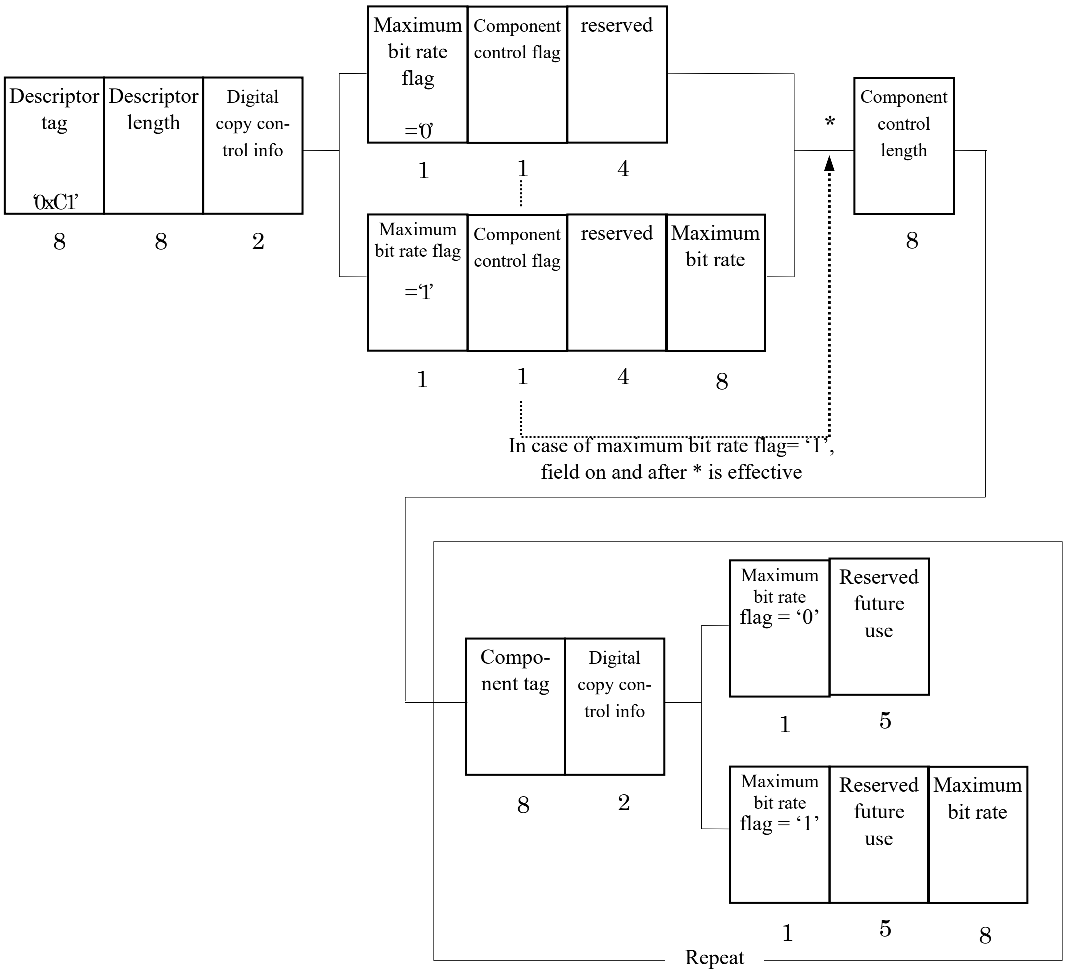


Figure 6.2-25 Data structure of Digital copy control descriptor

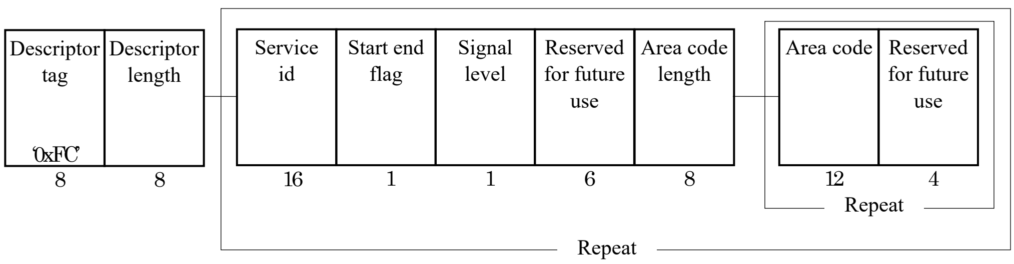


Figure 6.2-26 Data structure of Emergency information descriptor

Descriptor tag	Descriptor length	Country code	country region id		Local time offset polarity	Local time offset	Changing date	time of change	next time offset
'0x58'				'1'					
8	8	24	6	1	1	16	16	24	16
		Repeat							

Figure 6.2-27 Data structure of Local time offset descriptor

Descriptor tag	Descriptor length	Reserved future use	stream content	Component type	Component tag	Stream type	Simulcast group tag
'0xC4'							
8	8	4	4	8	8	8	8

ES multi-language flag	Main component flag	quality indicator	Sampling rate	Reserved future use	ISO639 Language code (first audio)	ISO639 Language code	text char
1	1	2	3	1	24	24	8xN

ES multi-language flag = 0

ES multi-language flag = 1

Figure 6.2-28 Data structure of Audio component descriptor

Descriptor tag	Descriptor length	Hyper linkage type	link destination type	Selector length	Selector byte	Private area
'0xC5'						
8	8	16	8	8	8xN	8xN

Figure 6.2-29 Data structure of Hyperlink descriptor

Descriptor tag	Descriptor length	Region spec type	Target region spec
'0xC6'			
8	8	8	8xN

Figure 6.2-30 Data structure of Target region descriptor

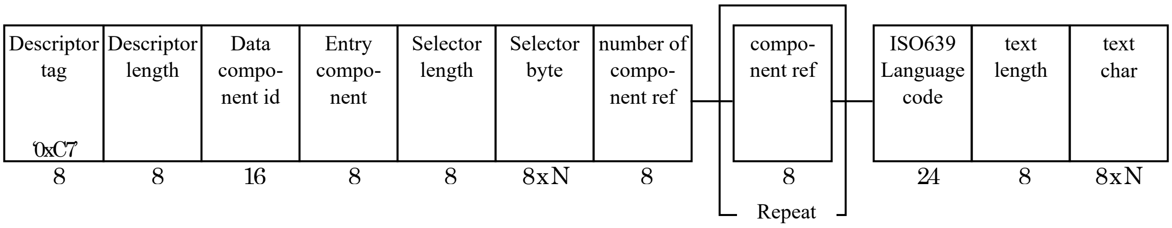


Figure 6.2-31 Data structure of Data contents descriptor

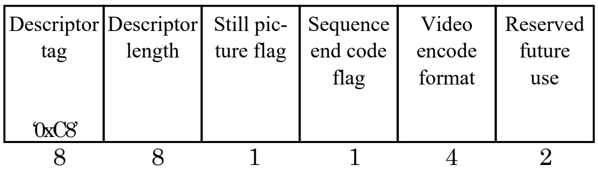


Figure 6.2-32 Data structure of Video decode control descriptor

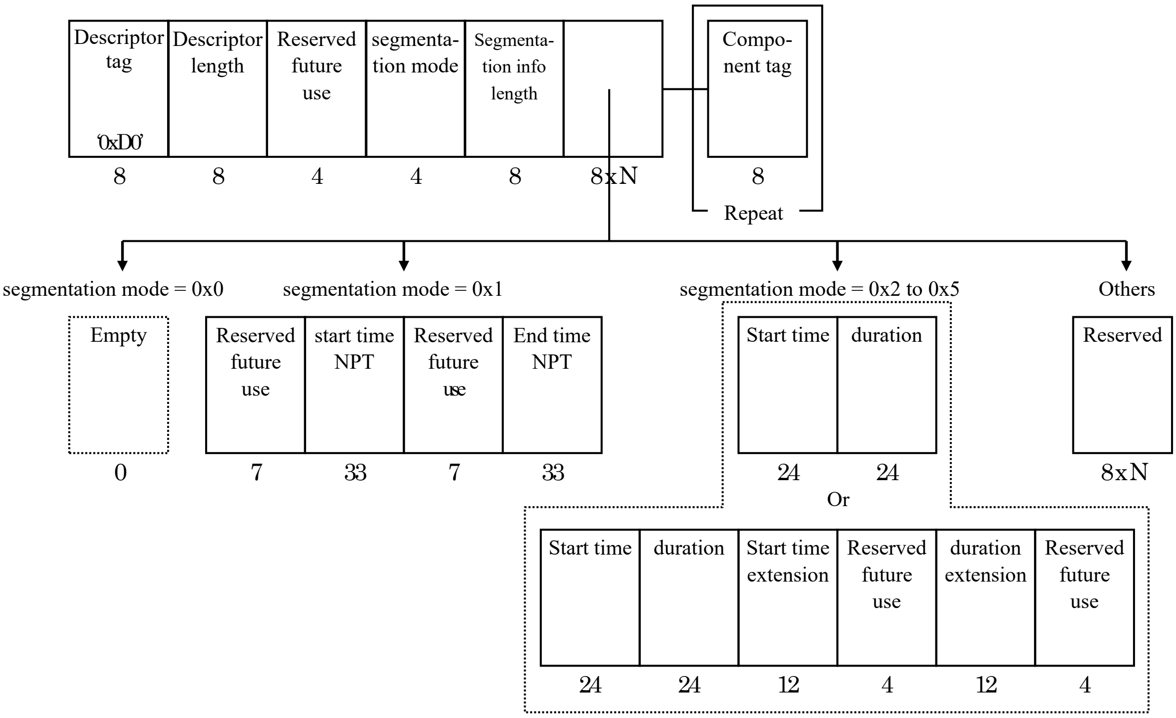


Figure 6.2-33 Data structure of Basic local event descriptor

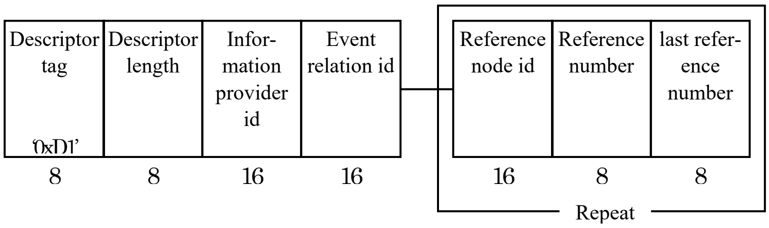


Figure 6.2-34 Data structure of Reference descriptor

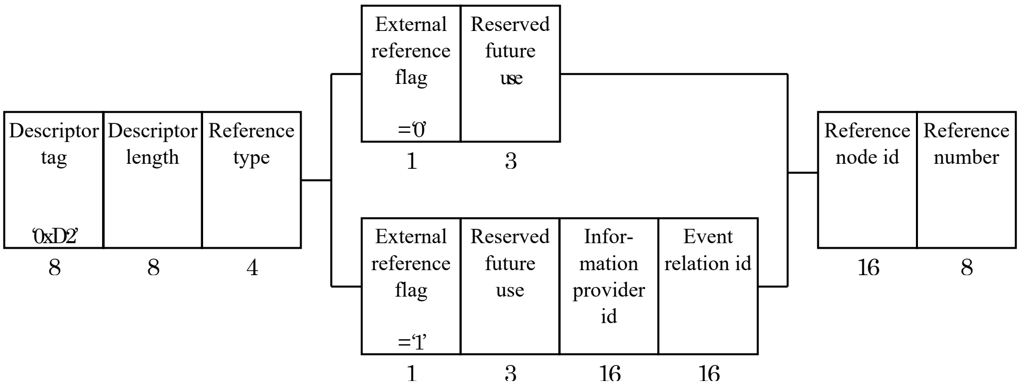


Figure 6.2-35 Data structure of Node relation descriptor

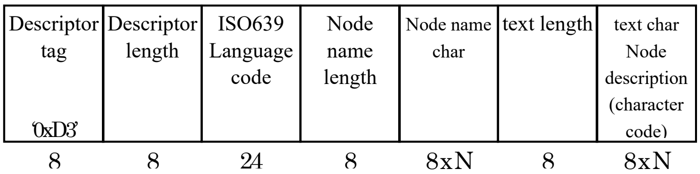


Figure 6.2-36 Data structure of Short node information descriptor

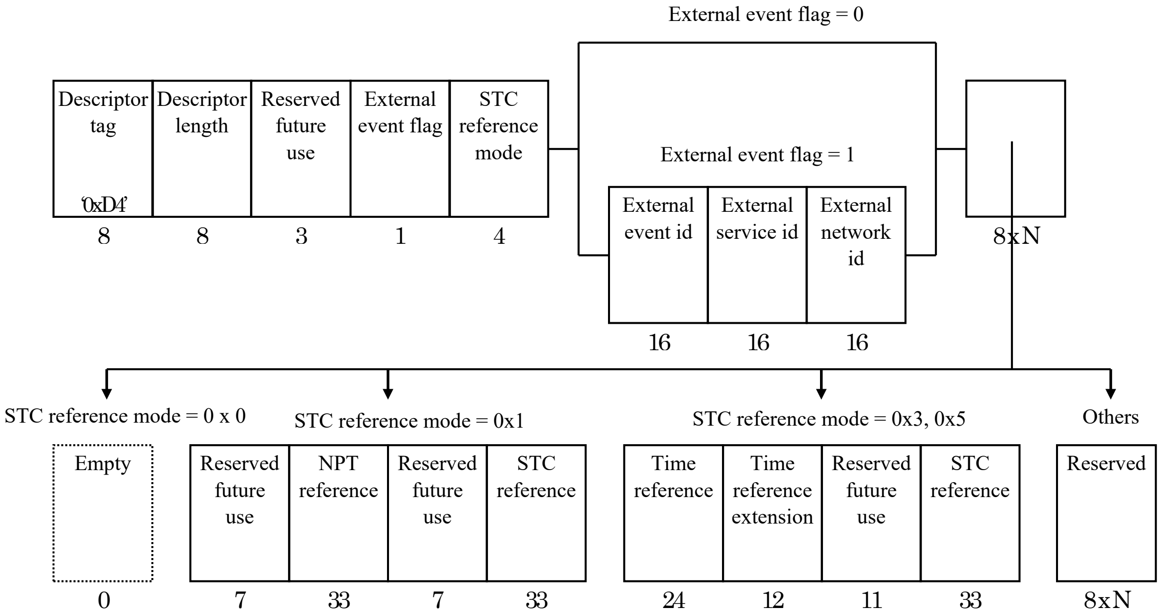


Figure 6.2-37 Data structure of STC reference descriptor

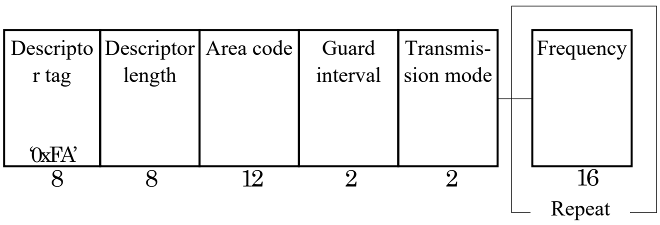


Figure 6.2-38 Data structure of Terrestrial delivery system descriptor

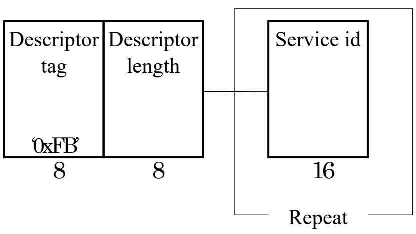


Figure 6.2-39 Data structure of Partial reception descriptor

Descriptor tag	Descriptor length	Series id	repeat label	program pattern	expire date valid flag	expire time	expire name	last expire number	Series name char
'0xD5'									
8	8	16	4	3	1	16	12	12	8xN

Figure 6.2-40 Data structure of Series descriptor

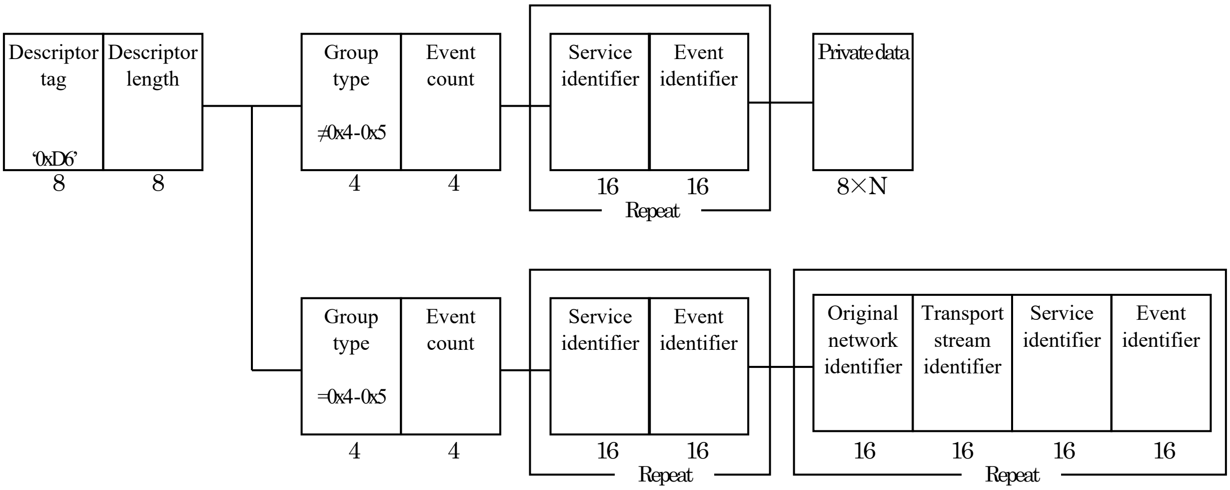


Figure 6.2-41 Data structure of Event group descriptor

Descriptor tag	Descriptor length	Parameter version	update time	Table id	Table descrip-tion length	Table descrip-tion byte
'0xD7'						
8	8	8	16	8	8	8xN
				Repeat		

Figure 6.2-42 Data structure of SI transmission parameter descriptor

Descriptor tag	Descriptor length	Char (Broadcast-er name
'0xD8'		
8	8	8xN

Figure 6.2-43 Data structure of Broadcaster name descriptor

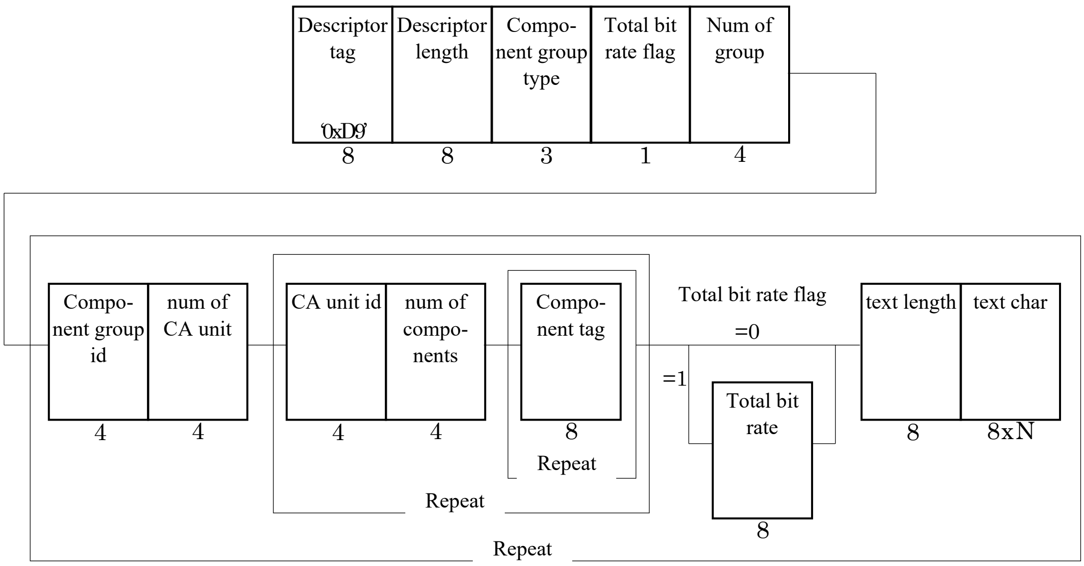


Figure 6.2-44 Data structure of Component group descriptor

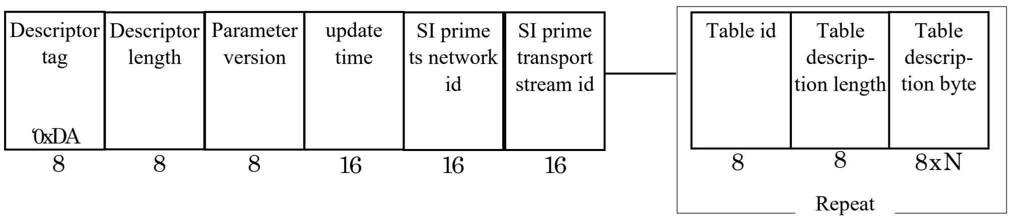


Figure 6.2-45 Data structure of SI prime TS descriptor

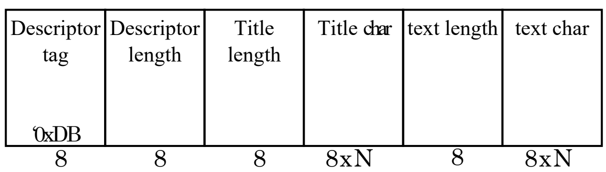


Figure 6.2-46 Data structure of Board information descriptor

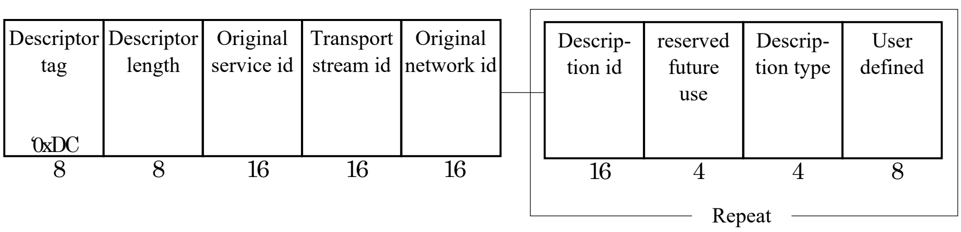


Figure 6.2-47 Data structure of LDT linkage descriptor

Descriptor tag '0xDD'	Descriptor length	Connected transmission group id	Segment type	Modulation type A	Modulation type B	Modulation type C	Additional connected transmission info
8	8	16	2	2	2	2	8xN

Figure 6.2-48 Data structure of Connected transmission descriptor

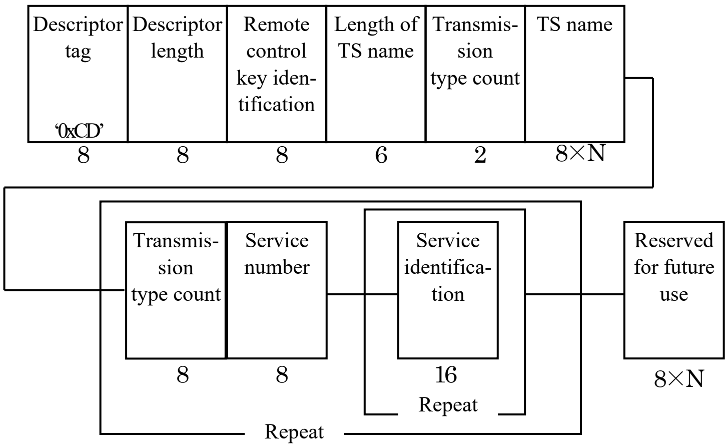


Figure 6.2-49 Data structure of TS information descriptor

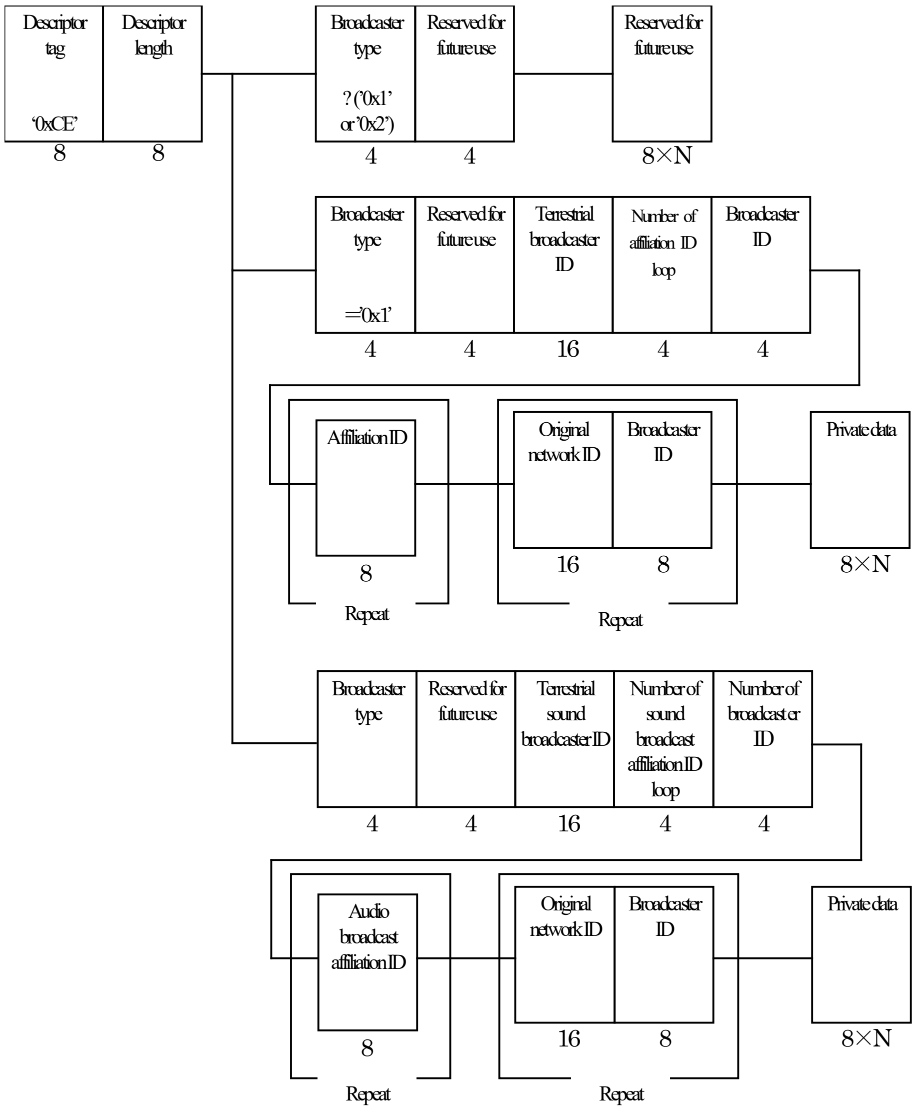


Figure 6.2-50 Data structure of extended broadcaster descriptor

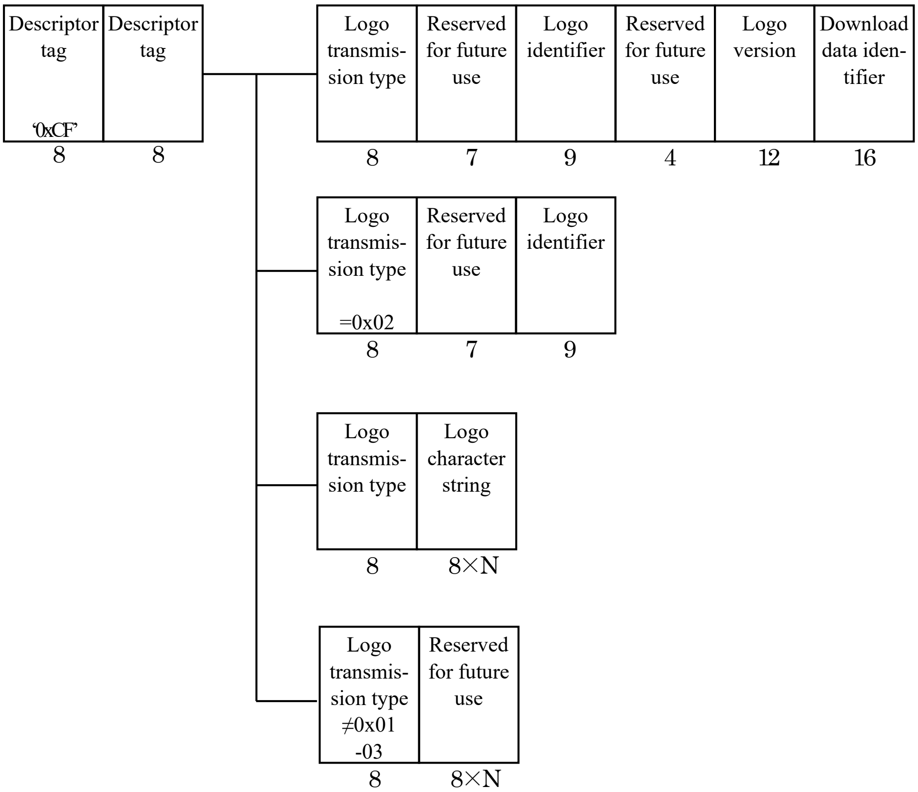


Figure 6.2-51 Data structure of logo transmission descriptor

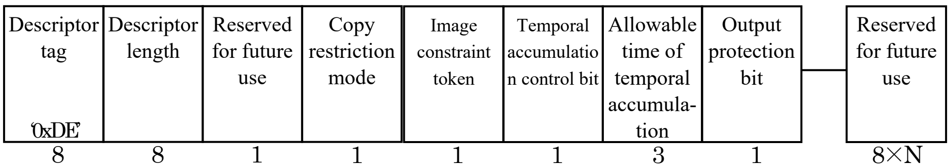


Figure 6.2-52 Data structure of content availability descriptor

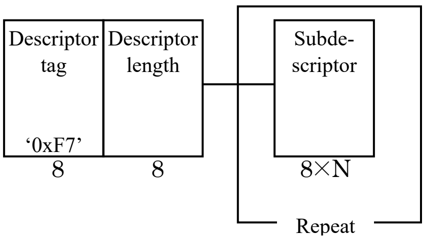


Figure 6.2-53 Data structure of Carousel compatible composite descriptor

Descriptor tag	Descriptor length	Conditional playback system identifier		Conditional playback PID	Private data
'0xF8'			'111'		
8	8	16	3	13	8×N

Figure 6.2-54 Data structure of restricted playback descriptor

Descriptor tag	Descriptor length	AVC profile	Set 0 constraint flag	Set 1 constraint flag	Set 2 constraint flag	Set 3 constraint flag	Set 4 constraint flag	Set 5 constraint flag	AVC compati-ble flag
'0x28'									
8	8	8	1	1	1	1	1	1	2

AVC level	AVC still pic-ture	AVC 24 hour video flag	Frame packing SEI non-use flag	Reserved for future use
8	1	1	1	5

Figure 6.2-55 Data structure of AVC video descriptor

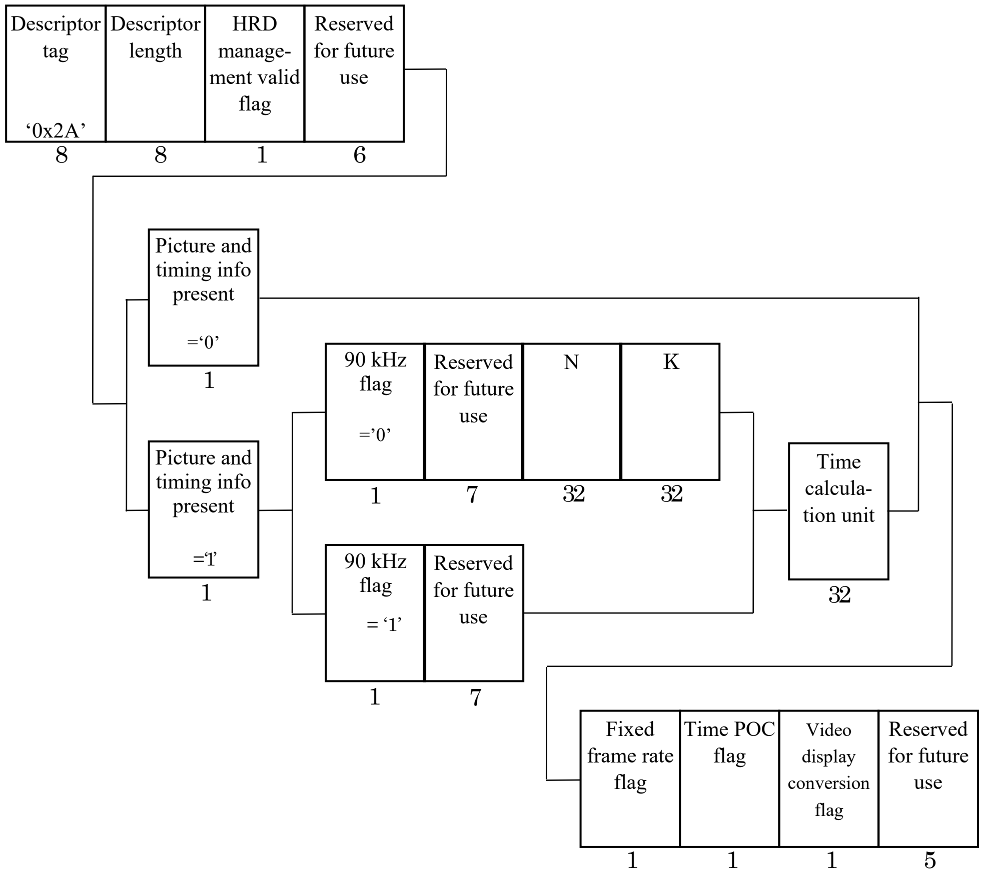


Figure 6.2-56 Data structure of AVC timing HRD descriptor

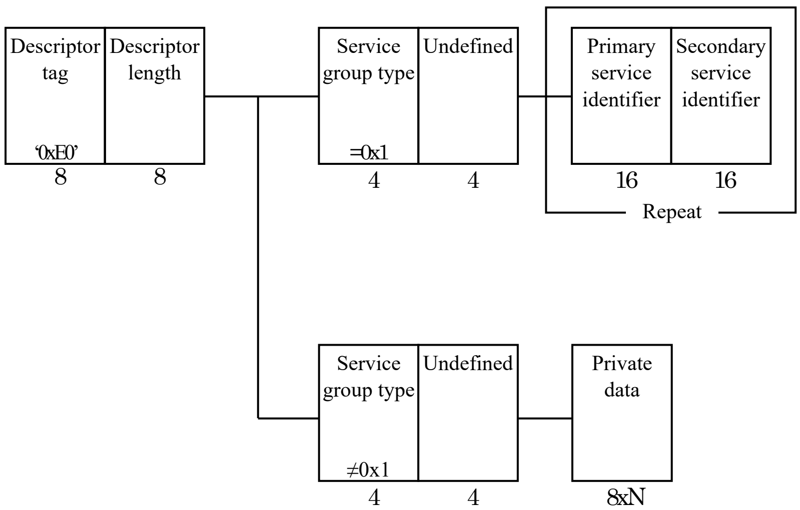


Figure 6.2-57 Data structure of service group descriptor

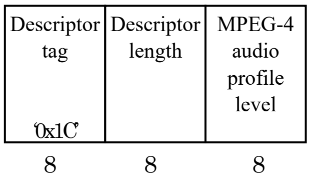


Figure 6.2-58 Data structure of MPEG-4 audio descriptor

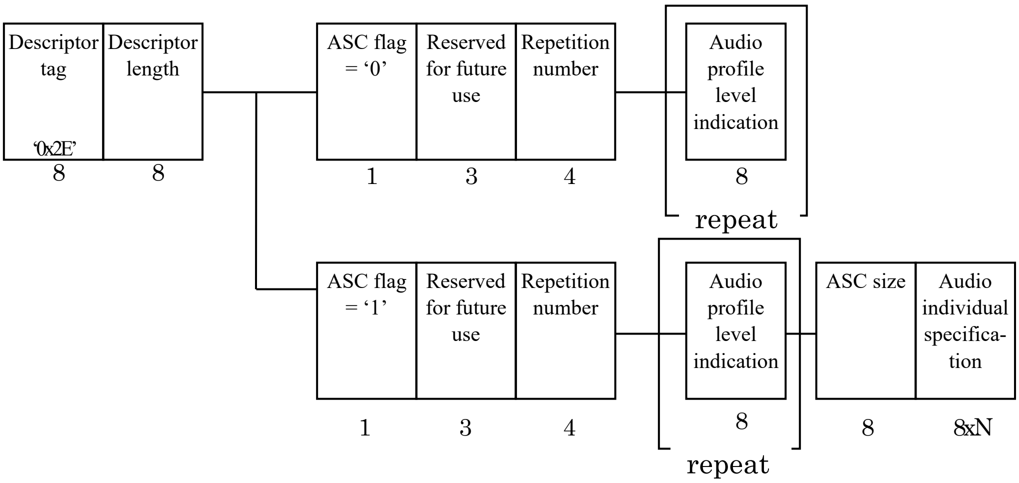


Figure 6.2-59 Data structure of MPEG-4 audio extension descriptor

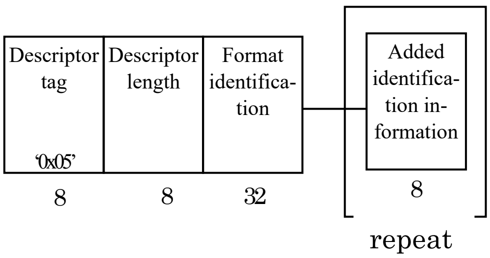


Figure 6.2-60 Data structure of registration descriptor

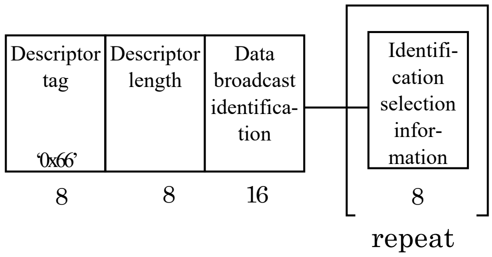


Figure 6.2-61 Data structure of data broadcast identification descriptor

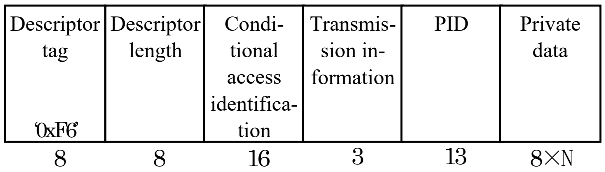


Figure 6.2-62 Data structure of access control descriptor

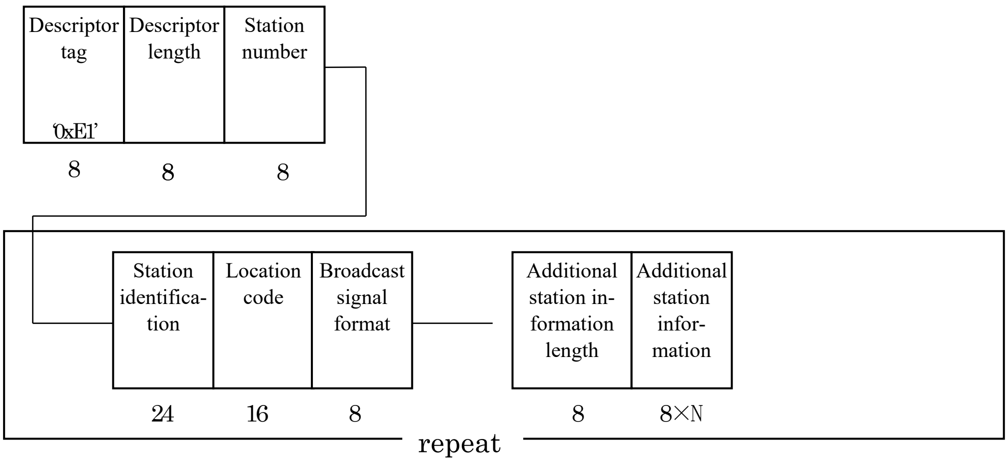


Figure 6.2-63 Data structure of area broadcasting information descriptor

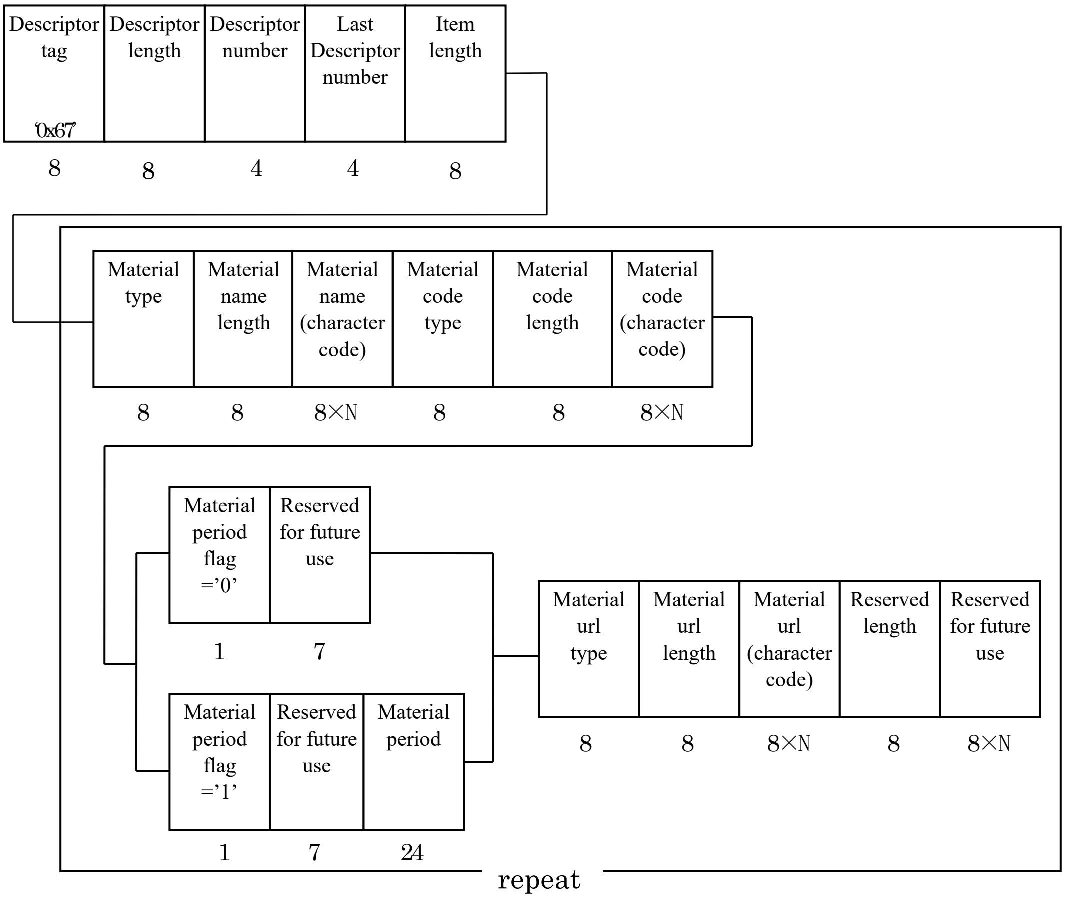


Figure 6.2-64 Data structure of material information descriptor

Descriptor tag '0x38'	Descriptor length	Profile space	Tier flag	HEVC profile	Profile compati-bility indication	Progressive scanning source flag	Interlace scanning source flag	Un-pack limitation flag	Frame specifica-tion limi-tation flag	Copy 44bits	HEVC level
8	8	2	1	5	32	1	1	1	1	44	8

Time layer subset flag = 0	HEVC still pic-ture flag	HEVC 24 hour moving picture flag	Sub pic hrd params not present flag	Undefined	HDR/WCG
1	1	1	1	2	2

Time layer subset flag = 1	HEVC still pic-ture flag	HEVC 24 hour moving picture flag	Sub pic hrd params not present flag	Undefined	HDR/WCG	Temporal ID mini-mum value	Undefined	Temporal ID maxi-mum value	Undefined
1	1	1	1	2	2	3	5	3	5

Figure 6.2-65 Data structure of HEVC video descriptor

Descriptor tag '0x04'	Descriptor length	Reserved for future use	Temporal hierarchy coding flag	Spatial hierarchy coding flag	Quality hierarchy coding flag	Hierarchy coding type
8	8	1	1	1	1	4

Reserved for future use	Layer index	TREF flag	Reserved for future use	Basic layer index	Reserved for future use	Layer channel
2	6	1	1	6	2	6

Figure 6.2-66 Data structure of hierarchy descriptor

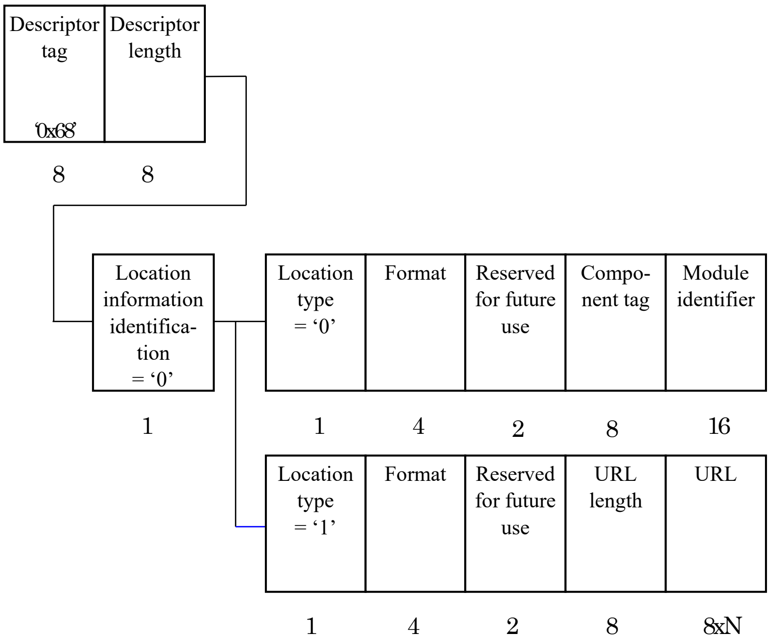


Figure 6.2-67 Data structure of hybrid information descriptor

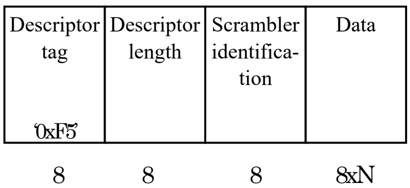


Figure 6.2-68 Data structure of scrambler descriptor

6.3 Data structure of descriptor used in INT

The data structure of descriptor specified in table 4-5 shall be in accordance with figures 6.3-1 to 6.3-6.

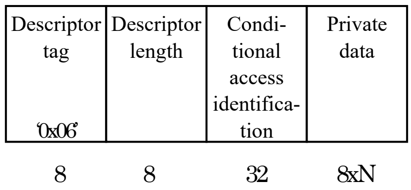


Figure 6.3-1 Data structure of target smartcard descriptor

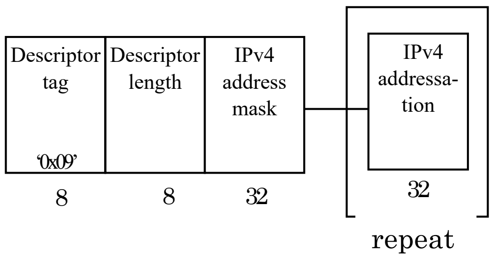


Figure 6.3-2 Data structure of target IP address descriptor

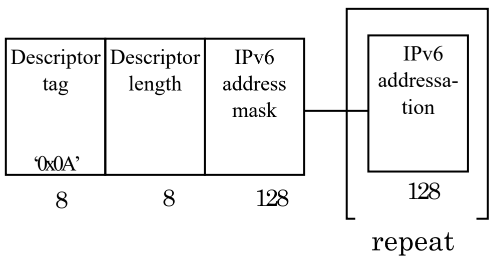


Figure 6.3-3 Data structure of target IPv6 address descriptor

Descriptor tag '0x0C'	Descriptor length	Language code	Platform name (character code)
8	8	24	8xN

Figure 6.3-4 Data structure of IP/MAC platform name descriptor

Descriptor tag '0x0D'	Descriptor length	Language code	Platform provider name (character code)
8	8	24	8xN

Figure 6.3-5 Data structure of IP/MAC platform provider name descriptor

Descriptor tag '0x13'	Descriptor length	Network identification	Original network identification	Transport stream identification	Service identification	Component tag
8	8	16	16	16	16	8

Figure 6.3-6 Data structure of IP/MAC stream location descriptor

7. Operation of identifiers

Allocation of each identifier used in service information shall be as shown in table 7-1. Range of value in the table includes reserved value, which will be specified in the future.

Therefore, company specification may contain reserved values; however, it shall be registered and released as the company signal including the reserved values.

Company signal is valid only within the network identifier in the company. However, it is possible to use the same company signal among plural companies.

Table 7-1 Operational standard of identifiers

Identifier	Corresponding portions of STD-B10				Bit	Range of value	Type of definition	Remarks
	Part	Table	Section	Descriptor name etc.				
Packet identifier (PID)	1	5-1	5.1		13	0x0000 – 0x0010, 0x1FFF	Specified by the Ministry of Internal Affairs and Communications	Specified by the Notification
	2	5-1	5.1.3			0x0011 – 0x002F	Specified by standardization organization (0x0015 and 0x0016 have been specified and operated by the company since before this standard was developed.)	Registered and released after deliberation
	3	6-1	6.1					
		6-2	6.2			Range which does not interfere with the above values	Specified and operated by the company	
Reference PID (reference_PID)	2		6.2.22	Hierarchical transmission descriptor				
Table identifier (table_id)	1	5-2	5.2		8	0x00 – 0x41, 0x82 – 0x86, 0xFF	Specified by the Ministry of Internal Affairs and Communications	Specified by the Notification
	2	5-2	5.1					
	3	6-1	6.1			0x42 – 0x81, 0x87 – 0x8F, 0xC0 – 0xFE	Specified by standardization organization	Registered and released after deliberation
						0x90 – 0xBF	Specified and operated by the company	
		6-2	6.2					

Descriptor tag (descriptor_tag)	1	5-3	5.3		8	0x00 – 0x3F, 0x41, 0x43, 0x44, 0xF5 – 0xFE	Specified by the Ministry of Internal Affairs and Communications	Specified by the Notification
						0x40, 0x42, 0x45 – 0x7F, 0xC0 – 0xF4, 0xFF	Specified by standardization organization	Registered and released after deliberation
						0x80 – 0xBF	Specified and operated by the company	
Descriptor tag for INT (descriptor_tag to be used in INT)	1	5-4	5.4		8	0x13	Specified by the Ministry of Internal Affairs and Communications	Specified by the Notification
						0x00 – 0x12, 0x14 – 0x79	Specified by standardization organization	Registered and released after deliberation
						0x80 – 0xFE	Specified and operated by the company	
						0xFF	Not used	
Descriptor tag for AIT (descriptor_tag to be used in AIT)	1	5-5	5.4		8	0x00 – 0x7F	Specified by standardization organization	Registered and released after deliberation
						0x80 – 0xFE	Specified and operated by the company	
						0xFF	Not used	
Transport Stream identifier (transport_stream_id)	2		5.2.4 etc.	Network Information Table (NIT) etc.	16		Specified and operated by the company	Unique within original network identifier
	3		5.1.1	Local Event Information Table (LIT)				
SI prime Transport Stream identifier (SI_prime_transport_stream_id)	2		6.2.38	SI prime TS descriptor				
Network identifier (network_id)	2	N-1	5.2.4	Network Information Table (NIT)	16		Specified by standardization organization	Registered and released on application
Original network identifier (original_network_id)	2		5.2.4 etc.	Network Information Table (NIT) etc.				
	3		5.1.1	Local Event Information Table (LIT)				
SI prime TS network identifier (SI_prime_ts_network_id)	2		6.2.38	SI prime TS descriptor				
External network identifier (external_network_id)	3		5.2.5	STC reference descriptor				
Bouquet identifier (bouquet_id)	2		5.2.5 etc.	Bouquet Association Table (BAT) etc.	16		Specified by standardization organization	Registered and released after deliberation. Unique within Japan.
Service identifier (service_id)	2		5.2.6 etc.	Service Description Table (SDT) etc.	16		Specified and operated by the company	Unique within Japan for the same broadcasting medium
	3		5.1.1	Local Event Information Table (LIT)				

Original service identifier (original_service_id)	2		5.2.15 etc.	Link Description Table etc.				
Reference service identifier (reference_service_id)	2		6.2.18 etc.	Time shift event descriptor etc.				
Primary service identifier (primary_service_id)	2		6.2.49	Service group descriptor				
Secondary service identifier (secondary_service_id)	2							
External service identifier (external_service_id)	3		5.2.5	STC reference descriptor				
Broadcast event number identifier (program_number)	2		Annex E	PAT, PMT				
Running status (running_status)	2	5-6	5.2.6 etc.	Service Description Table (SDT) etc.	3		Specified by standardization organization	Registered and released after deliberation
Event identifier (event_id)	2		5.2.7 etc.	Event Information Table (EIT) etc.	16		Specified and operated by the company	
	3		5.1.1 etc.	Local Event Information Table (LIT) etc.				
Reference event identifier (reference_event_id)	2		6.2.18	Time shift event descriptor				
Local event identifier (local_event_id)	3		5.1.1	Local Event Information Table (LIT)				
External event identifier (external_event_id)	3		5.2.5	STC reference descriptor				
Content identifier (content_id)	2		5.2.12 etc.	Partial Content Announcement Table etc.	32		Specified and operated by the company	
Broadcaster identifier (broadcaster_id)	2		5.2.13 etc.	Broadcaster Information Table etc.	8		Specified and operated by the company	
Terrestrial broadcaster identifier (terrestrial_broadcaster_id)	2		6.2.43	Extended broadcaster descriptor	16		Specified and operated by the company	
Terrestrial sound broadcaster identifier (terrestrial_sound_broadcaster_id)	2		6.2.43	Extended broadcaster descriptor	16		Specified and operated by the company	
Information identifier (information_id)	2		5.2.14	Network Board Information Table (NBIT)	16		Specified and operated by the company	
Information type (information_type)	2	5-15	5.2.14	Network Board Information Table (NBIT)	4		Specified by standardization organization	Registered and released after deliberation
Description body location (description_body_location)	2	5-16	5.2.14	Network Board Information Table (NBIT)	2		Specified by standardization organization	Registered and released after deliberation
Description identifier (description_id)	2		5.2.15 etc.	Link Description Table etc.	16		Specified and operated by the company	
Action type (action_type)	2	5-22	5.2.18	IP/MAC notification table (INT)	8		Specified by standardization organization	Registered and released after deliberation
Platform identifier (platform_id)	2	P-1	5.2.18	IP/MAC notification table (INT)	24	0x000001 - 0xFFEFFF	Specified by registration organization	Released
						0xFFF000 -	Specified by standard-	Registered and

						0xFFFFFE	ization organization	released on application
						0x000000, 0xFFFFFF	Reserved	
Processing or-der(processing_order)	2	5-23	5.2.18	IP/MAC notifi-cation table (INT)	8		Specified by standard-ization organization	Registered and released after deliberation
Conditional access system identifier (CA_system_id)	2	M-1		CA descriptor	16		Specified by standard-ization organization	Registered and released on application
			6.2.2	CA identifier descriptor				
			6.2.54	Access control descriptor				
Conditional access sub-system identifier (CA_subsystem_id)	2		6.3.2.1	Target smart-card descriptor	16		Specified and operated by the company	
Stream content (stream_content)	2	6-5	6.2.3	Component descriptor	4	0x0 – 0xB	Specified by standard-ization organization	Registered and released after deliberation
			6.2.26	Audio compo-nent descriptor		0xC – 0xF	Specified and operated by the company	

Component type (component_type)	2	6-5	6.2.3	Component descriptor	8	when stream_content=0 xC – 0xF	Specified and operated by the company	
		6-43	6.2.26	Audio component descriptor		Other than above	Specified by standardization organization	Registered and released after deliberation
Component tag (component_tag)	2		6.2.3 etc.	Component descriptor etc.	8		Specified and operated by the company	
	3		5.2.1	Basic local event descriptor				
Large genre classification (content_nibble_level_1)	2	Annex H	6.2.4	Content descriptor	4		Specified by standardization organization	Registered and released after deliberation
Middle genre classification (content_nibble_level_2)	2	Annex H	6.2.4	Content descriptor	4		Specified by standardization organization	Registered and released after deliberation
User genre (user_nibble)	2		6.2.4	Content descriptor	4+4		Specified and operated by the company	
Polarization (polarization)	2	6-9	6.2.6	Satellite delivery system descriptor	2		Specified by standardization organization	Allocated
Modulation (modulation)	2	6-10	6.2.6	Satellite delivery system descriptor	5		Specified by standardization organization	Registered and released after deliberation
Inner FEC scheme (FEC_inner)	2	6-11	6.2.6	Satellite delivery system descriptor	4		Specified by standardization organization	Registered and released after deliberation
Linkage type (linkage_type)	2	6-14	6.2.8	Link descriptor	8	0x00 – 0x7F, 0xC0 – 0xFF	Specified by standardization organization	Registered and released after deliberation
						0x80 – 0xBF	Specified and operated by the company	
Number of horizontal elementary cells (number_of_horizontal_elementary_cells)	2	6-16	6.2.9	Mosaic descriptor	3		Specified by standardization organization	Allocated
Number of vertical elementary cells (number_of_vertical_elementary_cells)	2	6-17	6.2.9	Mosaic descriptor	3		Specified by standardization organization	Allocated
Logical cell identifier (logical_cell_id)	2		6.2.9	Mosaic descriptor	6		Specified and operated by the company	
Logical cell presentation information (logical_cell_presentation_info)	2	6-18	6.2.9	Mosaic descriptor	3		Specified by standardization organization	Registered and released after deliberation
Elementary cell identifie2r (elementary_cell_id)	2		6.2.9	Mosaic descriptor	6		Specified and operated by the company	
Cell linkage information (cell_linkage_information)	2	6-19	6.2.9	Mosaic descriptor	8		Specified by standardization organization	Registered and released after deliberation
Parental rating (rating)	2	6-23	6.2.12	Parental rating descriptor	8	0x00 – 0x0F	Specified by standardization organization	Registered and released after deliberation
						0x10 – 0xFF	Specified and operated by the company	
Service type (service_type)	2	6-25	6.2.13 etc.	Service descriptor etc.	8	0x00 – 0x7F, 0xC0 – 0xFF	Specified by the Ministry of Internal Affairs and Communications	Specified by the Notification

						0xA1 – 0xBF	Specified by standard-ization organization	Registered and released after deliberation
	3	6-4	6.3.3	Identifier used for transmission of program index		0x80 – 0xA0	Specified and operated by the company	
Data component identifier (data_component_id)	2	J-1	6.2.20 etc.	Data component descriptor	16		Specified by standard-ization organization	Registered and released on application
	3		6.3.2	Identifier used for transmission of program index				
System management identifier (system_management_id)	2	6-35	6.2.21	System man-agement de-scriptor	16	Upper 8 bits	Specified by the Min-istry of Internal Affairs and Communications	Specified by the Notifica-tion
		6-36				Lower 8 bits	Specified and operated by the company	
Digital recording control data (digital_recording_ control_data)	2	6-39	6.2.23	Digital copy control de-scriptor	2	0x01	Specified and operated by the company	
						Other than above	Specified by standardi-zation organization	Allocated
Start/end flag (start_end_flag)	2		6.2.24	Emergency information descriptor	1		Specified by the Min-istry of Internal Affairs and Communications	Specified by the Notifica-tion
Signal type (signal_level)	2	D-1	6.2.24	Emergency alarm signal	1		Specified by the Min-istry of Internal Affairs and Communications	Specified by the Notifica-tion
Area code (area_code)	2	D-2	6.2.24	Emergency information descriptor	12		Specified by the Min-istry of Internal Affairs and Communications	Specified by the Notifica-tion
Country region identifier (country_region_id)	2		6.2.25	Local time off-set descriptor	6		Specified by standard-ization organization	Registered and released after deliberation

Stream identifier (stream_type)	2	E-4	6.2.26	Audio component descriptor	8	0x00 – 0x7F	Specified by the Ministry of Internal Affairs and Communications	Specified by the Notification
						0x80 – 0xC3	specified by standardization organization	Registered and released after deliberation
			Annex E	Program Map Table		0xC4 – 0xFF	Specified and operated by the company	
Simulcast group tag (simulcast_group_tag)	2		6.2.26	Audio component descriptor	8	0x00 – 0xFE	Specified and operated by the company	
						0xFF	Specified by standardization organization	Registered and released after deliberation
Quality indicator (quality_indicator)	2	6-44	6.2.26	Audio component descriptor	2		Specified by standardization organization	Registered and released after deliberation
Sampling rate (sampling_rate)	2	6-45	6.2.26	Audio component descriptor	3		Specified by standardization organization	Registered and released after deliberation
Region description method designation (region_spec_type)	2	6-47	6.2.27	Target region descriptor	8		Specified by standardization organization	Registered and released after deliberation
Hyperlink descriptor (hyper_linkage_type)	2	6-50	6.2.29	Hyperlink descriptor	8	0x00 – 0x7F	Specified by standardization organization	Registered and released after deliberation
						0x80 – 0xFF	Specified and operated by the company	
Link destination type (link_destination_type)	2	6-51	6.2.29	Hyperlink descriptor	8	0x00 – 0x7F, 0xFF	Specified by standardization organization	Registered and released after deliberation
						0x80 – 0xFE	Specified and operated by the company	
Information provider identifier (information_provider_id)	2		6.2.29	Hyperlink descriptor	16			Unique within Japan
	3		5.1.2 etc.	Event Relation Table (ERT) etc.				
Node identifier (node_id)	2		6.2.29	Hyperlink descriptor	16		Specified and operated by the company	
Parent node identifier (parent_node_id)	3		5.1.2	Event Relation Table (ERT)				
Reference node identifier (reference_node_id)	3		5.2.2 etc.	Reference descriptor etc.				
Module identifier (moduleId)	2		6.2.29	Hyperlink descriptor	16		Specified and operated by the company	
Event relation identifier (event_relation_id)	2		6.2.29	Hyperlink descriptor	16		Specified and operated by the company	
	3		5.2.2 etc.	Reference descriptor etc.				
Video encode format (video_encode_format)	2	6-60	6.2.30	Video decode control descriptor	4		Specified by standardization organization	Registered and released after deliberation
Area code (area_code)	2		6.2.31	Terrestrial delivery system descriptor	12		Specified by the Ministry of Internal Affairs and Communications	
Guard interval (guard_interval)	2	6-62	6.2.31	Terrestrial delivery system descriptor	2		Specified by standardization organization	Allocated
Transmission mode	2	6-63	6.2.31	Terrestrial de-	2		Specified by standard-	Registered and

(transmission_mode)				livery system descriptor			ization organization	released after deliberation
Series identifier (series_id)	2		6.2.33	Series de-scriptor	16		Specified and operated by the company	Unique within terrestrial broadcaster identifier or broadcaster identifier
Repeated broadcast label (repeat_label)	2		6.2.33	Series de-scriptor	4	0x0	Specified by standard-ization organization	Allocated
						Other than above	Specified and operated by the company	Defined for each series identifier
Program pattern (program_pattern)	2	6-66	6.2.33	Series de-scriptor	3		Specified by standard-ization organization	Registered and released after deliberation
Group type (group_type)	2	6-68	6.2.34	Event group descriptor	4		Specified by standard-ization organization	Registered and released after deliberation
Component group type (component_group_type)	2	6-72	6.2.37	Component group descriptor	3		Specified by standard-ization organization	Registered and released after deliberation
Component group identifier (component_group_id)	2	6-73	6.2.37	Component group descriptor	4		Specified by standard-ization organization	Registered and released after deliberation
CA unit identifier (CA_unit_id)	2	6-74	6.2.37	Component group descriptor	4		Specified by standard-ization organization	Registered and released after deliberation
Description identifier (description_id)	2		6.2.40	LDT linkage descriptor	16		Specified and operated by the company	
Description type (description_type)	2	6-78	6.2.40	LDT linkage descriptor	4		Specified by standard-ization organization	Registered and released after deliberation
Connected transmission group identifier (connected_transmission_ group_id)	2		6.2.41	Connected transmission descriptor	16		Specified and operated by the company	
Segment type (segment_type)	2	6-80	6.2.41	Connected transmission descriptor	2		Specified by standard-ization organization	Registered and released after deliberation
Modulation type A (mod-ulation_type_A)	2	6-81	6.2.41	Connected transmission descriptor	2		Specified by standard-ization organization	Registered and released after deliberation
Modulation type B (mod-ulation_type_B)	2	6-81	6.2.41	Connected transmission descriptor	2		Specified by standard-ization organization	Registered and released after deliberation
Modulation type C (mod-ulation_type_C)	2	6-81	6.2.41	Connected transmission descriptor	2		Specified by standard-ization organization	Registered and released after deliberation
Additional connected transmission information (additional_connected_ transmission_info)	2		6.2.41	Connected transmission descriptor	8		Specified and operated by the company	
Remote control key iden-tifier (remote_control_key_id)	2		6.2.42	TS information descriptor	8		Specified and operated by the company	
Transmission type infor-mation (transmission_type_info)	2		6.2.42	TS information descriptor	8		Specified and operated by the company	
Broadcaster type	2	6-84	6.2.43	Extended	4		Specified by standard-	Registered and

(broadcaster_type)				broadcaster descriptor			ization organization	released after deliberation
Affiliation identifier (affiliation_id)	2		6.2.43	Extended broadcaster descriptor	8		Specified and operated by the company	
Sound broadcasting affiliation identifier (sound_broadcast_affiliation_id)	2		6.2.43	Extended broadcaster descriptor	8		Specified and operated by the company	
Logo transmission type (logo_transmission_type)	2	6-86	6.2.44	Logo transmission descriptor	8		Specified by standardization organization	Registered and released after deliberation
Logo identifier (logo_id)	2		6.2.44	Logo transmission descriptor	9		Specified and operated by the company	
Download data identifier (download_data_id)	2		6.2.44	Logo transmission descriptor	16		Specified and operated by the company	
Copy restriction mode (copy_restriction_mode)	2		6.2.45	Content availability descriptor	1		Specified and operated by the company	
Allowable time of temporal accumulation (retention_state)	2	6-88	6.2.45	Content availability descriptor	3		Specified by standardization organization	Allocated
Subdescriptor tag placed in carousel compatible composite descriptor	2	K-1	6.2.46	Carousel compatible composite descriptor	8	0x01 – 0x7F, 0xC0 – 0xFE	Specified by standardization organization	Registered and released after deliberation
						0x80 – 0xBF	Specified and operated by the company	
Service group type (service_group_type)	2	6-93	6.2.49	Service group descriptor	4		Specified by standardization organization	Registered and released after deliberation
Format identifier(format_identifier)	2		6.2.52	Registration descriptor	32		Specified by registration organization and operated by the company	Released
Data broadcast identifier (data_broadcast_id)	2	6-98	6.2.53	Data broadcast identification descriptor	16		Specified by standardization organization	Registered and released after deliberation
Transmission information	2	6-101	6.2.54	Access control descriptor	3	'000' – '011'	Specified and operated by the company	
						'100' – '111'	Specified by standardization organization	Registered and released after deliberation
Station identifier (station_id)	2	6-103	6.2.55	Area broadcast information descriptor	24		Allocated by standardization organization	Registered and released after deliberation
							Specified and operated by the company based on allocation	
Location code (location_code)	2		6.2.55	Area broadcast information descriptor	16		Specified by the Ministry of Internal Affairs and Communications	
Broadcast signal format (broadcast_signal_format)	2	6-104	6.2.55	Area broadcast information descriptor	8		Specified by standardization organization	Registered and released after deliberation
Location information flag (has_location)	2		6.2.58	Hybrid information descriptor	1		Specified by standardization organization	Registered and released after deliberation
Location type (location_type)	2		6.2.58	Hybrid information descriptor	1		Specified by standardization organization	Allocated

Format (format)	2	6-109	6.2.58	Hybrid information descriptor	4		Specified by standardization organization	Registered and released after deliberation
Prefectural bit map (prefecture_bitmap)	2	G-2	Annex G	Target region descriptor	56		Specified by standardization organization	Registered and released after deliberation
Time_mode (time_mode)	2	K-5	Annex K	Expire descriptor used in carousel compatible composite descriptor	8		Specified by standardization organization	Registered and released after discussion
Private scope type (private_scope_type)	2	K-7	Annex K	Provider Private descriptor used in carousel compatible composite descriptor	8		Specified by standardization organization	Registered and released after deliberation
Subdescriptor tag for tag value extension	2		Annex L	Composite descriptor for tag value extension	8		Specified by standardization organization	Registered and released after deliberation
Relation type (relation_type)	3	5-3	5.1.2	Event Relation Table (ERT)	4		Specified by standardization organization	Registered and released after deliberation
Collection mode (collection_mode)	3	5-4	5.1.2	Event Relation Table (ERT)	4		Specified by standardization organization	Registered and released after deliberation
Segmentation mode (segmentation_mode)	3	5-7	5.1.2	Basic local event descriptor	4		Specified by standardization organization	Registered and released after deliberation
Reference type (reference_type)	3	5-10	5.2.3	Node relation descriptor	4		Specified by standardization organization	Registered and released after deliberation
STC reference mode (STC_reference_mode)	3	5-13	5.2.5	STC reference descriptor	4		Specified by standardization organization	Registered and released after deliberation
Material type (material_type)	3	5-15	5.2.6	Material information descriptor	8	0x00 – 0x7F	Specified by standardization organization	Registered and released after deliberation
						0x80 - 0xFF	Specified and operated by the company	
Material code type (material_code_type)	3	5-16	5.2.6	Material information descriptor	8	0x00 – 0x7F	Specified by standardization organization	Registered and released after deliberation
						0x80 - 0xFF	Specified and operated by the company	
Material url type (material_url_type)	3	5-17	5.2.6	Material information descriptor		0x00 – 0x7F	Specified by standardization organization	Registered and released after deliberation
						0x80 - 0xFF	Specified and operated by the company	
Enabling information type (enable_info_type)	3	A-2	Annex A	Index enabling information	4		Specified by standardization organization	Registered and released after deliberation

Part 2

DATA STRUCTURE AND DEFINITION OF BASIC INFORMATION OF SERVICE INFORMATION

Part 2
DATA STRUCTURE AND DEFINITION OF BASIC INFORMATION OF SERVICE INFORMATION

CONTENTS

<BLANK>

1. Purpose

Part 2 of this standard is established to specify detail data structure of basic information related to the Service Information as specified in "Standard transmission system for digital broadcasting among standard television broadcasting and the like" in Ministerial Ordinance No. 87[(10)] (hereinafter refer to as "Ordinance") issued by the Ministry of Internal Affairs and Communications in 2011, and “Structure and delivery procedure of rerated information, delivery procedure of PES packet, section format, TS packet, IP packet, ULE packet, MMTP packet, compressed IP packet and TLV packet, structure of transmission control signal and identifier, structure of emergency information descriptor and emergency warning broadcasting” in the Ministerial Notification No. 233[(11)] (hereinafter refer to as "Notification") issued by the Ministry of Internal Affairs and Communications in 2014..

2. Scope

Part 2 of this standard is applied to basic structure of the Service Information specified in part 1.

3. Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

bouquet: collection of services marketed as a single entity

broadcaster: organization which assemble a sequence of events or programs to be delivered to the viewer based upon a schedule

component: one or more entities which together make up an event

Conditional Access (CA) system: system to control subscriber access to services, programs and events

delivery system: physical medium by which one or more multiplexes are transmitted

Entitlement Management Message (EMM): private Conditional Access information which specify the authorization levels or the services of specific decoders. They may be addressed to individual decoder or groups of decoders.

event: grouping of elementary broadcast data streams with a with a defined start and end time belonging to a common service

forbidden: when used in the clause defining the coded bit stream, indicates that the value shall never be used.

MPEG-2: See ISO/IEC 13818

multiplex: stream of all the digital data carrying one or more services within a single physical channel

network: collection of MPEG-2 Transport Stream (TS) multiplexes transmitted on a single delivery system

original_network_id: unique identifier of a network.

reserved: when used in the clause defining the coded bit stream, indicates that the value may be used in the future for ISO defined extensions. Unless otherwise specified within the present document, all "reserved" bits shall be set to "1".

reserved_future_use: when used in the clause defining the coded bit stream, indicates that the value may be used in the future for ARIB defined extensions. Unless otherwise specified within the present document all "reserved_future_use" bits shall be set to "1".

section: syntactic structure used for mapping all service information defined in ARIB STD-B10 into ISO/IEC 13818-1 TS packets

service: sequence of programs under the control of a broadcaster which can be broadcast as part of a schedule

service_id: unique identifier of a service within a TS

Service Information (SI): digital data describing the delivery system, content and scheduling/timing of broadcast data streams, etc.

sub_table: collection of sections with the same value of table_id and:

for a NIT: the same table_id extension (network_id) and version_number;

for a BAT: the same table_id extension (bouquet_id) and version_number;

for a SDT: the same table_id extension (transport_stream_id), the same original_network_id and version_number;

for a EIT: the same table_id extension (service_id), the same transport_stream_id, the same original_network_id and version_number;

The table_id_extension field is equivalent to the fourth and fifth byte of a section when the section_syntax_indicator is set to a value of "1".

table: comprised of a number of sub_tables with the same value of table_id

Transport Stream (TS): data structure defined in ISO/IEC 13818-1

transport_stream_id: unique identifier of a TS within an original network.

JST (Japanese Standard Time): "UTC +9" hour, irrespective of summer time, etc.

MJD (Modified Julian Date) (Japan Time): date indication denoted in accordance with Annex C. Time shall refer to "UTC + 9" hour.

The relationships of some of these definitions are illustrated in the service delivery model in figure 3-1.

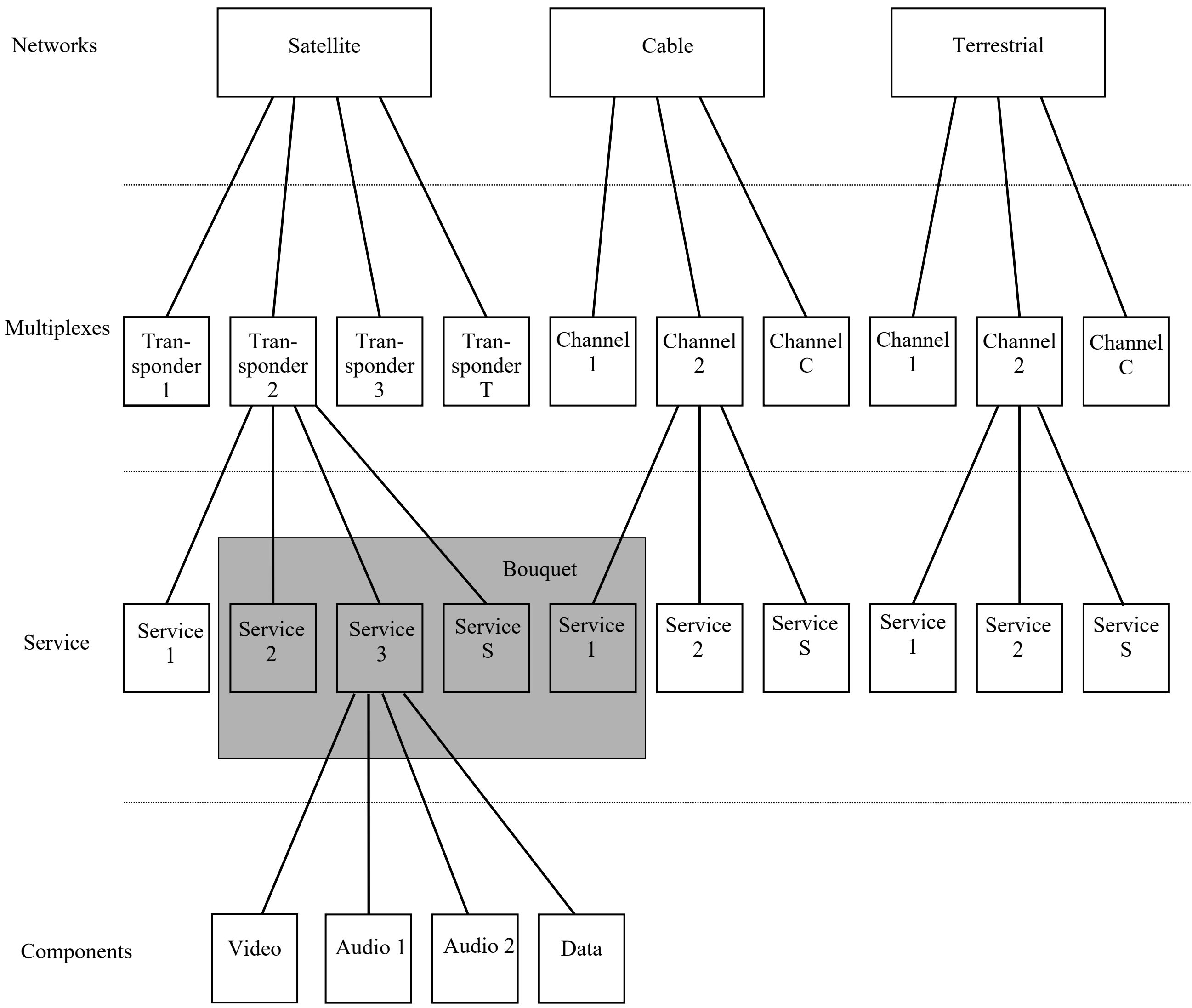


Figure 3-1 Digital broadcasting, service delivery model

3.2 Abbreviations

For the purposes of this standard, the following abbreviations apply:

AMT	Address Map Table
BAT	Bouquet Association Table
BCD	Binary Coded Decimal
BIT	Broadcaster Information Table
CA	Conditional Access
CAT	Conditional Access Table
CRC	Cyclic Redundancy Check
ECM	Entitlement Control Message
EIT	Event Information Table
EMM	Entitlement Management Message
EPG	Electronic Program Guide
FEC	Forward Error Correction
HLG	Hybrid Log-Gamma
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
INT	IP/MAC Notification Table
IRD	Integrated Receiver Decoder
ISO	International Organization for Standardization
JTC	Joint Technical Committee
LDT	Linked Description Table
LSB	Least Significant Bit
MJD	Modified Julian Date
MPEG	Moving Pictures Expert Group
NBIT	Network Board Information Table
NIT	Network Information Table
NVOD	Near Video On Demand
PAT	Program Association Table
PCAT	Partial Content Announcement Table
PID	Packet Identifier
PMT	Program Map Table
PQ	Perceptual Quantization
PSI	Program Specific Information
QPSK	Quadrature Phase Shift Keying
RS	Reed Solomon
RST	Running Status Table
SDT	Service Description Table
SI	Service Information
SMPTE	Society of Motion Picture and Television Engineers
ST	Stuffing Table

TDT	Time and Date Table
TOT	Time Offset Table
ULE	Unidirectional Lightweight Encapsulation
bslbf	bit string, left bit first
rpchof	remainder polynomial coefficients, highest order first
uimsbf	unsigned integer most significant bit first

3.3 Terminology used in Ministerial Ordinances and Notifications

Terminology used in the present document and in Ministerial Ordinances and Notifications is listed in table 3-1.

Table 3-1 Terminology comparison table

Where the terminology is used	Terminology used in the present document	Terminology used in Ministerial Ordinances and Notifications
Everywhere	Identification	Identifier
Everywhere	Descriptor area length	Descriptor length
Everywhere	Stream type	Stream type identifier
Service list descriptor	Service type	Service type identifier

4. SI description

ISO/IEC 13818-1 specifies SI which is referred to as PSI. The PSI data provides information to enable automatic configuration of the receiver to demultiplex and decode the various streams programs within the multiplex.

The PSI data is structured as four types of table. The tables are transmitted in sections.

1) Program Association Table (PAT):

   - for each service in the multiplex, the PAT indicates the location (the PID values of the Transport Stream packets) of the corresponding Program Map Table (PMT). It also gives the location of the Network Information Table (NIT). the ST is used to invalidate existing sections, for example at delivery system boundaries.

2) Conditional Access Table (CAT):

   - the CAT provides information on the Conditional Access (CA) systems used in the multiplex; the information is private (not defined with this standard) and dependent on the CA system, but includes the location of the EMM stream, when applicable.

3) Program Map Table (PMT):

   - the PMT identifies and indicates the locations of the streams that make up each service, and the location of the Program Clock Reference fields for a service.

4) Network Information Table (NIT):

   - the location of the NIT is defined in this standard in compliance with ISO/IEC 13818-1[21] specification, but the data format is outside the scope of ISO/IEC 13818-1[21]. It is identified to provide information about the physical network. The syntax and semantics of the NIT are defined in this standard.

Transmission control signals except for the PSI of MPEG-2 are specified as below. The tables are transmitted in section format.

1) Address map table (AMT):

   - AMT provides information for relating service identifier identifying program numbers with IP packets.

In addition data are needed to provide identification of services and events for the user. The coding of this data is defined in this standard. In contrast with the PAT, CAT, and PMT of the PSI, which give information only for the multiplex in which they are contained (the actual multiplex), the additional information defined within this standard can also provide information on services and events carried by different multiplexes, and even on other networks. This data is structured as eleven tables:

1) Bouquet Association Table (BAT):

   - the BAT provides information regarding bouquets. As well as giving the name of the bouquet, it provides a list of services for each bouquet.

2) Service Description Table (SDT):

   - the SDT contains data describing the services in the system e.g. names of services, the service provider, etc.

3) Event Information Table (EIT):

   - the EIT contains data concerning events or programs such as event name, start time, duration, etc.
   - the use of different descriptors allows the transmission of different kinds of event information e.g. for different service types.

4) Running Status Table (RST):

   - the RST gives the status of an event (running/not running). The RST updates this information and allows timely automatic switching to events.

5) Time and Date Table (TDT):

   - the TDT gives information relating to the present time and date. This information is given in a separate table due to the frequent updating of this information.

6) Time Offset Table (TOT):

   - the TOT gives information relating to the present time and date and local time offset. This information is given in a separate table due to the frequent updating of the time information.

7) Partial Content Announcement Table (PCAT):

- the PCAT includes starting time and continuing time of partial content in accumulated data broadcasting.

8) Stuffing Table (ST):

- the ST is used to invalidate existing sections, for example at delivery system boundaries.

9) Broadcaster Information Table (BIT):

- the BIT includes broadcaster unit comprising network or SI transmitting parameter information for each broadcaster.

10) Network Board Information Table (NBIT):

- the NBIT includes board information in network and reference information for acquiring the board information.

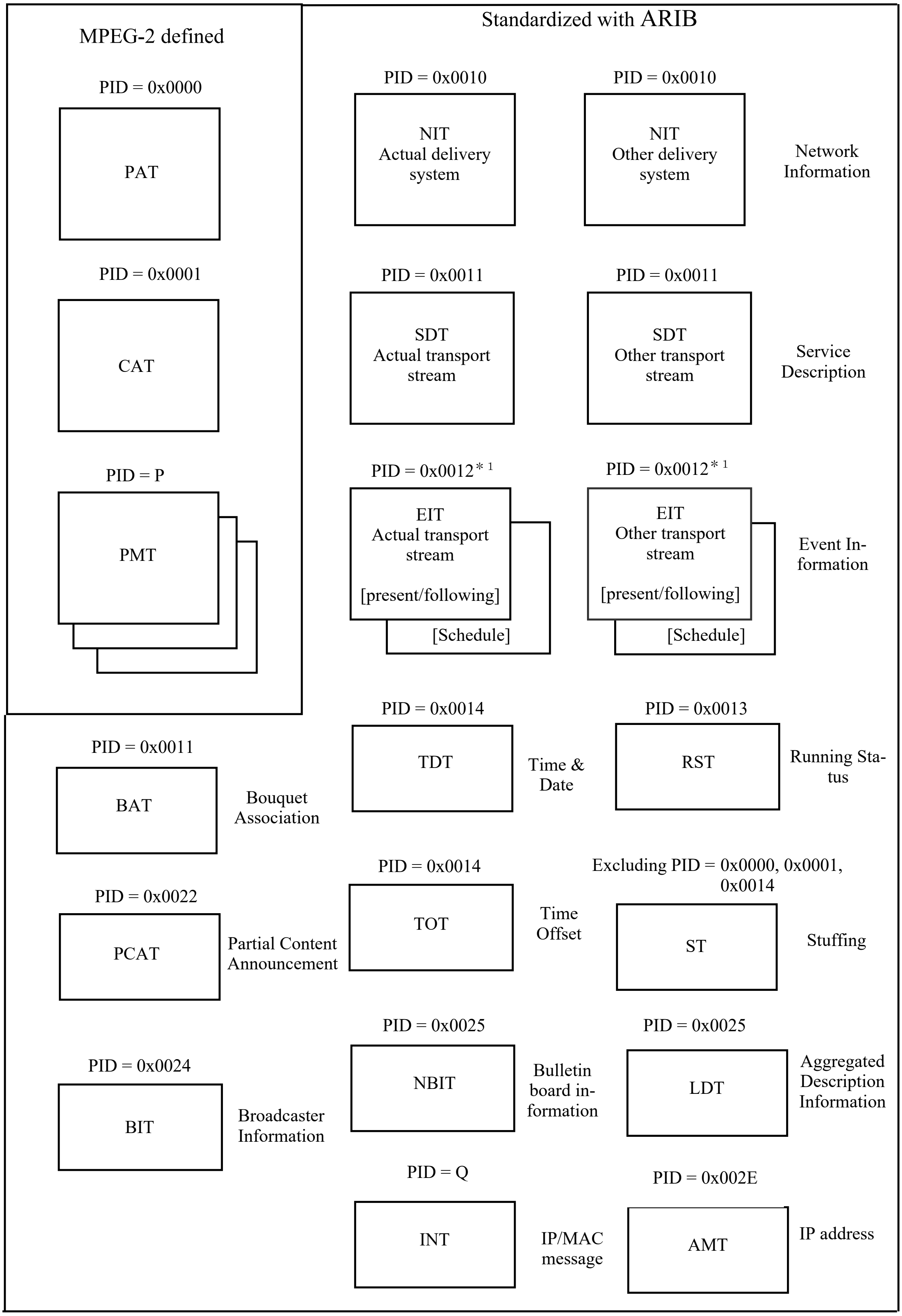
11) Link Description Table (LDT):

- the LDT includes various collected data for reference from other tables.

12) IP/MAC Notification Table (INT):

- the INT provides the relational information for the broadcasting programs and the IP/MAC streams which compose them.

Applicable use of descriptors allows a flexible approach to the structure of the tables and allows for future compatible extensions.


MPEG-2 defined
PID = 0x0000
PAT
PID = 0x0001
CAT
PID = P
PMT
Standardized with ARIB
PID = 0x0010
NIT
Actual delivery system
PID = 0x0010
NIT
Other delivery system
Network Information
PID = 0x0011
SDT
Actual transport stream
PID = 0x0011
SDT
Other transport stream
Service Description
PID = 0x0012* 1
EIT
Actual transport stream
[present/following]
[Schedule]
PID = 0x0012* 1
EIT
Other transport stream
[present/following]
[Schedule]
Event Information
PID = 0x0011
BAT
Bouquet Association
PID = 0x0014
TDT
Time & Date
PID = 0x0013
RST
Running Status
PID = 0x0022
PCAT
Partial Content Announcement
PID = 0x0014
TOT
Time Offset
Excluding PID = 0x0000, 0x0001, 0x0014
ST
Stuffing
PID = 0x0024
BIT
Broadcaster Information
PID = 0x0025
NBIT
Bulletin board information
PID = 0x0025
LDT
Aggregated Description Information
PID = Q
INT
IP/MAC message
PID = 0x002E
AMT
IP address

*1: Using the PID value 0x0012,0x0026,0x0027 in digital terrestrial television broadcasting and multimedia broadcasting

Figure 4-1 Structure of transmission control signal in TS transmission

5. SI tables

5.1 SI table mechanism

The SI specified in this standard and MPEG-2 PSI tables shall be segmented into one or more sections before being inserted into Transport Stream packets. The tables listed in clause 4 are conceptual in that they need never be regenerated in a specified from within an IRD. The tables, when transmitted shall not be scrambled, with the exception of the EIT, which maybe scrambled if required (see subclause 5.1.5). A section is a syntactic structure that shall be used for mapping all MPEG-2 tables and SI tables specified in this standard, into Transport Stream packets or TLV packets. These SI syntactic structures conform to the private section syntax defined in ISO/IEC 13818-1[24].

5.1.1 Explanation

Sections may be variable in length. The sections within each table are limited to 1024 bytes in length, except for sections within the EIT, which are limited to 4096 bytes. Each section is uniquely identified by combinations of the following elements:

a) table_id:
   - the table_id identifies to which table the section belongs.
   - some table_ids have been defined by ISO and others by the present document. Other values of the table_id can be allocated by the user for private purposes. The list of values of table_id is contained in table 5-2.

b) table_id_extension:
   - the table_id_extension is used for identification of a sub_table.
   - the interpretation of each sub_table is given in subclause 5.2.

c) section_number:
   - the section_number field allows the sections of a particular sub_table to be reassembled in their original order by the decoder. It is recommended that sections be transmitted in numerical order, unless it is desired to transmit some sections of the sub_table more frequently than others, e.g. due to random access considerations.
   - for the SI tables as specified in the present document, section numbering applies to sub_tables.

d) version_number:

- when the characteristics of the TS or the TLV described in the SI given in the present document change (e.g. new events start, different composition of elementary streams for a given service), then new SI data shall be sent containing the updated information. A new version of the SI data is signaled by sending a sub_table with the same identifiers as the previous sub_table containing the relevant data, but with the next value of version_number.
- for the SI tables specified in the present document, the version_number applies to all sections of a sub_table.

e) Current_next_indicator:

- each section shall be numbered as valid "now" (current), or as valid in the immediate future (next). This allows the transmission of a future version of the SI in advance of the change, giving the decoder the opportunity to prepare for the change. There is however, no requirement to transmit the next version of a section in advance, but if it is transmitted, then it shall be the next correct version of that section.

5.1.2 Mapping of sections into Transport Stream (TS) packets

Sections shall be mapped directly into Transport Stream packets. Sections may start at the beginning of the payload of a Transport Stream packet, but this is not a requirement, because the start of the first section in the payload of a Transport Stream packet is pointed to by the pointer_field. There is never more than one pointer_field in a Transport Stream packet, as the start of any other section can be identified by counting the length of the first and any subsequent sections, since no gaps between sections within a Transport Stream packet are allowed by the syntax.

Within Transport Stream packets of any single PID value, one section is finished before the next one is allowed to be started, or else it is not possible to identify to which section header the data belongs. If a section finishes before the end of a Transport Stream packet, but it is not convenient to open another section, a stuffing mechanism may be used to fill up the space.

Stuffing may be performed by filling each remaining byte of the Transport Stream packet with the value "0xFF". Consequently the value "0xFF" shall not be used for the table_id. If the byte immediately following the last byte of a section takes the value of "0xFF", then the rest of the Transport Stream packet shall be stuffed with "0xFF" bytes. These bytes may be discarded by a decoder. Stuffing may also be performed using the adaptation_field mechanism.

For more detailed description of the mechanism and functionality, specifically refer to section 2.4.4 and Annex C of ISO/IEC 13818-1[21].

5.1.3 Coding of PID and table_id fields

Table 5-1 lists the PID values which shall be used for the TS packets which carry SI sections.

Table 5-1 PID allocation for SI

Table	PID
PAT*1	0x0000
PMT*1	Indirect designation by PAT
CAT*1	0x0001
NIT*1	0x0010
SDT	0x0011
BAT	0x0011
EIT	0x0012
EIT(digital terrestrial TV broadcasting, multimedia broad-casting) *2	0x0012, 0x0026, 0x0027
RST	0x0013
TDT	0x0014
TOT	0x0014
PCAT	0x0022
BIT	0x0024
NBIT	0x0025
LDT	0x0025
AMT*1	0x002E
ST	Exclude 0x0000, 0x0001, 0x0014
INT*1	Indirect designation by PMT
Null packet *1	0x1FFF

*1: In accordance with the Notification

*2: In accordance with the operating guidelines for the PID allocation to each hierarchy

Table 5-2 lists the values, which shall be used for table_id and transmission level for the SI, defined in the present document.

The value specified as sending frequency in table 5-2 is only a criterion of operation and is not the standard value.

Table 5-2 Allocation of table_id values and transmission level

Table_id	Table	Transmission level	Transmission frequency
0x00	PAT*1	Mandatory	Once or more/100m sec.
0x01	CAT*1	Mandatory	Once or more/1 sec.
0x02	PMT*1	Mandatory	Once or more/100m sec.
0x40	NIT (Actual network)*1	Mandatory	Once or more/10 sec.
0x41	NIT (Other network)*1	Optional	Once or more/10 sec.
0x42	SDT (Actual stream)	Mandatory	Once or more/2 sec.
0x46	SDT (Other stream)	Optional	Once or more/10 sec.
0x4A	BAT	Optional	Once or more/10 sec.
0x4C	INT*1	Optional	Once or more/30 sec.
0x4E	EIT (Present and following program of the actual stream)	Mandatory	Once or more/2 sec.
0x4F	EIT (Present and following program of the other stream)	Optional	Once or more/10 sec.
0x50 – 0x5F	EIT (Program within 8 days of the actual stream)	Optional	Once or more/10 sec.
	EIT (Program after 8 days of the actual stream)	Optional	Once or more/30 sec.
0x60 – 0x6F	EIT (Program within 8 days of the other stream)	Optional	Once or more/10 sec.
	EIT (Program after 8 days of the other stream)	Optional	Once or more/30 sec.
0x70	TDT	Mandatory *2	Once or more/30 sec.
0x71	RST	Optional	Optional
0x72	ST	Optional	Optional
0x73	TOT	Mandatory *2	Once or more/30 sec.
0xC2	PCAT	Optional	Optional
0xC4	BIT	Optional	Once or more/20 sec.
0xC5	NBIT (Board information body)	Optional	Once or more/20 sec.
0xC6	NBIT (Reference information to gain board information)	Optional	1 sec. or more 10 sec.
0xC7	LDT	Optional	1 sec. or more 20 sec.
0x90 – 0xBF	Selectable range as table_id value set by companies		
0xFE	AMT	Optional	Once or more/10 sec.

*1: In accordance with the Notification

*2: Transmitting either TDT or TOT is mandatory.

5.1.4 Repetition rates and random access

In systems where random access is a consideration, it is recommended to re-transmit SI sections specified within the present document several times, even when changes do not occur in the configuration. For SI specified within the present document, multi-sectional availability in the same sub_table section shall be 4KB at maximum. (Multi-sectional availability herein means continuous allocation to TS packets.)

Moreover, TS packets of the same PID is transmitted within the range of 4KB±100% in 32msec each. The rule of "4KB in 32msec" is a detailed specification of 1Mbit per 1 sec. for every PID.

This Limit applies for TSs with a total data rate of up to 100Mbit/s.

5.1.5 Scrambling

With the exception of the EIT carrying schedule information, no tables specified in the present document shall be scrambled. One method for scrambling the EIT schedule table is given in the appendix of the present document. If a scrambling method operating over TS packets is used, it may be necessary to use a stuffing mechanism to fill from the end of a section to the end of a packet so that any transitions between scrambled and unscrambled data occur at packet boundaries.

In order to identify the CA streams which control the descrambling of the EIT data, a scrambled EIT schedule table shall be identified in the PSI. Service_id value 0xFFFF is allocated to identifying a scrambled EIT, and the program map section for this service shall describe the EIT as a private stream and shall include one or more CA_descriptors (defined in ISO/IEC 13818-1[24]) which give the PID values and optionally, other private data to identify the associated CA streams. Service_id value 0xFFFF shall not be used for any other service.

5.2 Table definitions

The following subclauses describe the syntax and semantics of the different types of table.

[Note] The symbols and abbreviations, and the method of describing syntax used in this standard are the same as those defined in sections 2.2 and 2.3 of ISO/IEC 13818-1[24].

5.2.1 Program Association Table (PAT)

PAT designates packet identifier of TS packet that transmits PMT related to broadcasting program.

For details, refer to sub-clause 2.4.4 of ISO/IEC 13818-1 [24] (see Annex E)

5.2.2 Conditional Access Table (CAT)

CAT designates packet identifier of TS packet that transmits related information of charged broadcasting.
For details, refer to clause 2.4.4 of ISO/IEC 13818-1 [24] (see Annex E)

5.2.3 Program Map Table (PMT)

PMT designates packet identifier of TS packet that transmits each coded signal constructing broadcasting program.
For details, refer to clause 2.4.4 of ISO/IEC 13818-1 [24] (see Annex E)

5.2.4 Network Information Table (NIT)

[Note] This item is also explained in Notification.

The NIT (see table 5-3) conveys information relating to the physical organization of the multiplexes/TSs carried via a given network, and the characteristics of the network itself. The combination of original_network_id and transport_stream_id allow each TS to be uniquely identified throughout the present document application area. Networks are assigned individual network_id values, which serve as unique identification codes for networks. The standardization organization shall specify the allocation of these codes. In the event that the NIT is transmitted on the network on which the TS was originated, the network_id and the original_network_id shall take the same value.

Guidelines for the processing of SI at transitions between delivery media boundaries, e.g. from satellite to cable, shall be specified otherwise.

IRDs may be able to store the NIT information in non-volatile memory in order to minimize the access time when switching between channels ("channel hopping"). It is also possible to transmit an NIT for other networks in addition to the actual network. Differentiation between the NIT for the actual network and the NIT for other networks is achieved using different table_id values (see table 5-2).

The NIT shall be segmented into network_information_sections using the syntax of table 5-3. Any sections forming part of NIT shall be transmitted in TS packets with a PID value of 0x0010. Any sections of the NIT which describe the actual network (that is, the network of which the TS containing NIT is a part) shall have the table_id 0x40 with the same table_id_extension (network_id).

The network_id field takes the value assigned to the actual network specified by the standardization organization. Any sections of NIT which refer to a network other than the actual network shall take table_id value of 0x41 and the network_id shall take the value allocated to the other network specified by the standardization organization.

Table 5-3 Network information section

Syntax	No. of bits	Identifier
network_information_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
network_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
reserved_future_use	4	bslbf
network_descriptors_length	12	uimsbf
for(i=0;i<N;i++){		
descriptor()		
}		
reserved_future_use	4	bslbf
transport_stream_loop_length	12	uimsbf
for(i=0;i<N;i++){		
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
reserved_future_use	4	bslbf
transport_descriptors_length	12	uimsbf
for(j=0;j<N;j++)		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

Semantics for the network information section:

table_id: See table 5-2.

section_syntax_indicator: The section_syntax_indicator is a 1-bit field which shall be set to "1".

section_length: This is a 12-bit field, the first two bits of which shall be "00". It specifies the number of bytes of the section, starting immediately following the section_length field and including the CRC. The section_length shall not exceed 1021 so that the entire section has a maximum length of 1024 bytes.

network_id: This is a 16-bit field which serves as a label to identify the delivery system, about which the NIT informs, from any other delivery system. The standardization organization shall specify allocation of the value of this field. (See Annex N)

version_number: This 5-bit field is the version number of the sub_table. The version_number shall be incremented by 1 when a change in the information carried within the sub_table occurs. When it reaches value 31, it wraps around to 0. When the current_next_indicator is set to "1", then the version_number shall be that of the currently applicable sub_table defined by the table_id and network_id. When the current_next_indicator is set to "0", then the version_number shall be that of the next applicable sub_table defined by the table_id and network_id.

current_next_indicator: This 1-bit indicator, when set to "1" indicates that the sub_table is the currently applicable sub_table. When the bit is set to "0", it indicates that the sub_table sent is not yet applicable and shall be the next sub_table to be valid.

section_number: This 8-bit field gives the number of the section. The section_number of the first section in the sub_table shall be "0x00". The section_number shall be incremented by 1 with each additional section with the same table_id and network_id.

last_section_number: This 8-bit field specifies the number of the last section (that is, the section with the highest section_number) of the sub_table of which this section is part.

network_descriptors_length: This 12-bit field gives the total length in bytes of the following network descriptors.

transport_stream_loop_length: This is a 12-bit field specifying the total length in bytes of the Transport Stream loops that follows, ending immediately before the first CRC-32 byte.

transport_stream_id: This is a 16-bit field which serves as a label for identification of this Transport Stream from any other multiplex within the delivery system.

original_network_id: This 16-bit field gives the label identifying the network_id of the originating delivery system.

tranpost_descriptors_length: This is a 12-bit field specifying the total length in bytes of Transport Stream descriptors that follow.

CRC_32: This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in Annex B after processing the entire section.

5.2.5 Bouquet Association Table (BAT)

The BAT (see table 5-4) provides information regarding bouquets. A bouquet is a collection of services, which may traverse the boundary of a network.

The BAT shall be segmented into bouquet_association_sections using the syntax of table 5-4. Any sections forming part of a BAT shall be transmitted in TS packets with a PID value of 0x0011. The sections of a BAT sub_table describing a particular bouquet shall have a bouquet_id field taking the value assigned to the bouquet described otherwise.
All BAT sections shall take a table_id value of 0x4A.

Table 5-4 Bouquet association selection

Syntax	No. of bits	Identifier
bouquet_association_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
bouquet_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
reserved_future_use	4	bslbf
bouquet_descriptors_length	12	uimsbf
for(i=0;i<N;i++){		
descriptor()		
}		
reserved_future_use	4	bslbf
transport_stream_loop_length	12	uimsbf
for(i=0;i<N;i++){		
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
reserved_future_use	4	bslbf
transport_descriptors_length	12	uimsbf
for(j=0;j<N;j++)		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

Semantics for the bouquet association section:

table_id: See table 5-2.

section_syntax_indicator: The section_syntax_indicator is a 1-bit field which shall be set to "1".

section_length: This is a 12-bit field, the first two bits of which shall be "00". It specifies the number of bytes of the section, starting immediately following the section_length field and including the CRC. The section_length shall not exceed 1021 so that the entire section has a maximum length of 1024 bytes.

bouquet_id: This is a 16-bit field which serves as a label to identify the bouquet. Allocations of the value of this field are specified otherwise.

version_number: This 5-bit field is the version number of the sub_table. The version_number shall be incremented by 1 when a change in the information carried within the sub_table occurs. When it reaches value 31, it wraps around to 0. When the current_next_indicator is set to "1", then the version_number shall be that of the currently applicable sub_table defined by the table_id and network_id. When the current_next_indicator is set to "0", then the version_number shall be that of the next applicable sub_table defined by the table_id and bouquet_id.

current_next_indicator: This 1-bit indicator, when set to "1" indicates that the sub_table is the currently applicable sub_table. When the bit is set to "0", it indicates that the sub_table sent is not yet applicable and shall be the next sub_table to be valid.

section_number: This 8-bit field gives the number of the section. The section_number of the first section in the sub_table shall be "0x00". The section_number shall be incremented by 1 with each additional section with the same table_id and bouquet_id.

last_section_number: This 8-bit field specifies the number of the last section (that is, the section with the highest section_number) of the sub_table of which this section is part.

bouquet_descriptors_length: This 12-bit field gives the total length in bytes of the following descriptors.

transport_stream_loop_length: This is a 12-bit field specifying the total length in bytes of the Transport Stream loops that follows.

transport_stream_id: This is a 16-bit field which serves as a label for identification of this Transport Stream from any other multiplex within the delivery system.

original_network_id: This 16-bit field gives the label identifying the network_id of the originating delivery system.

transport_descriptors_length: This is a 12-bit field specifying the total length in bytes of Transport Stream descriptors that follow.

CRC_32: This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in Annex B after processing the entire private section.

5.2.6 Service Description Table (SDT)

Each sub_table of the SDT (see table 5-5) shall describe services that are contained within a particular TS. The services may be part of the actual TS or part of other TSs, these being identified by means of the table_id (see table 5-2).

The SDT shall be segmented into service_description_sections using the syntax of table 5-5. Any sections forming part of an SDT shall be transmitted in TS packets with a PID value of 0x0011. Any sections of the SDT which describe the actual TS (that is, the TS containing the SDT) shall have the table_id value 0x42 with the same table_id_extension (transport_stream_id) and with the same original_network_id. Any sections of an SDT which refer to a TS other than the actual TS shall take a table_id value of 0x46.

Table 5-5 Service description section

Syntax	No. of bits	Identifier
service_description_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
transport_stream_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
original_network_id	16	uimsbf
reserved_future_use	8	bslbf
for(i=0;i<N;i++){		
service_id	16	uimsbf
reserved_future_use	3	bslbf
EIT_user_defined_flags	3	bslbf
EIT_schedule_flag	1	bslbf
EIT_present_following_flag	1	bslbf
running_status	3	uimsbf
free_CA_mode	1	bslbf
descriptors_loop_length	12	uimsbf
for(j=0;j<N;j++)		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

Semantics for the service description section:

table_id: See table 5-2.

section_syntax_ indicator: The section_syntax_indicator is a 1-bit field which shall be set to "1".

section_length: This is a 12-bit field, the first two bits of which shall be "00". It specifies the number of bytes of the section, starting immediately following the section_length field and including the CRC. The section_length shall not exceed 1021 so that the entire section has a maximum length of 1024 bytes.

transport_stream_id: This is a 16-bit field which serves as a label for identification of the Transport Stream, about which the SDT informs, from any other multiplex within the delivery system.

version_number: This 5-bit field is the version number of the sub_table. The version_number shall be incremented by 1 when a change in the information carried within the sub_table occurs. When it reaches value 31, it wraps around to 0. When the current_next_indicator is set to "1", then the version_number shall be that of the currently applicable sub_table defined by the table_id and network_id. When the current_next_indicator is set to "0", then the version_number shall be that of the next applicable sub_table.

current_next_indicator: This 1-bit indicator, when set to "1" indicates that the sub_table is the currently applicable sub_table. When the bit is set to "0", it indicates that the sub_table sent is not yet applicable and shall be the next sub_table to be valid.

section_ number: This 8-bit field gives the number of the section. The section_number of the first section in the sub_table shall be "0x00". The section_number shall be incremented by 1 with each additional section with the same table_id, transport_stream_id, and original_network_id.

last_section_number: This 8-bit field specifies the number of the last section (that is, the section with the highest section_number) of the sub_table of which this section is part.

original_network_id: This 16-bit field gives the label identifying the network_id of the originating delivery system.

service_id: This is a 16-bit field which serves as a label to identify this service from any other service within the Transport Stream. The service_id is the same as the program_number in the corresponding program_map_section.

EIT_user_defined_flags: Each broadcaster can define this 3 bits field individually as the extension to indicate whether it transmits EIT or not. If it is set to "111", it means no use.

EIT_schedule_flag: This is a 1-bit field which, when set to "1", indicates that EIT schedule information for the service is present in the current Transport Stream (see the guideline for information on maximum time interval between occurrences of an EIT schedule sub_table). If the flag is set to 0 then the EIT schedule information for the service should not be present in the Transport Stream.

EIT_present_following_flag: This is a 1-bit field which, when set to "1", indicates that EIT_present_following information for the service is present in the current Transport Stream (see the guideline for information on maximum time interval between occurrences of an EIT present/following sub_table). If the flag is set to 0 then the EIT present/following information for the service should not be present in the Transport Stream.

running_status: This is a 3-bit field indicating the status of the service as defined in table 5-6.

Table 5-6 SDT running_status

Value	Meaning
0	undefined
1	not running
2	starts in a few seconds (e.g. for video recording)
3	pausing
4	running
5 – 7	reserved for future use

free CA mode: This 1-bit field, when set to "0" indicates that all the component streams of the service are not scrambled. When set to "1" it indicates that access to one or more streams may be controlled by a CA system.

descriptors loop length: This 12-bit field gives the total length in bytes of the following descriptors.

CRC 32: This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in Annex B after processing the entire private section.

5.2.7 Event Information Table (EIT)

The EIT (see table 5-7) provides information in chronological order regarding the events contained within each service. Four classifications of EIT have been identified, distinguishable by the use of different table_ids (see table 5-2):

1) actual Transport Stream, present/following event information = table_id = "0x4E";

2) other Transport Stream, present/following event information = table_id = "0x4F";

3) actual Transport Stream, event schedule information = table_id = "0x50" to "0x5F";

4) other Transport Stream, event schedule information = table_id = "0x60" to "0x6F".

The present/following table shall contain only information pertaining to the present event and the chronologically following event carried by a given service on either the actual Transport Stream or another Transport Stream, except in the case of a Near Video On Demand (NVOD) reference service, where it may have more than two event descriptions. The event schedule tables for either the actual Transport Stream or other Transport Streams contain a list of events, in the form of a schedule, including events taking place at some time beyond the next event. The EIT schedule tables are optional. The event information shall be chronologically ordered.

The EIT shall be segmented into event_information_sections using the syntax of table 5-7. Any sections forming part of an EIT shall be transmitted in Transport Stream packets with a PID value of 0x0012.

Table 5-7 Event information section

Syntax	No. of bits	Identifier
event_information_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
service_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
segment_last_section_number	8	uimsbf
last_table_id	8	uimsbf
for(i=0;i<N;i++){		
event_id	16	uimsbf
start_time	40	bslbf
duration	24	uimsbf
running_status	3	uimsbf
free_CA_mode	1	bslbf
descriptors_loop_length	12	uimsbf
for(j=0;j<N;j++)		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

Semantics for the event information section:

table_id: See table 5-2.

section_syntax_indicator: The section_syntax_indicator is a 1-bit field which shall be set to "1".

section_length: This is a 12-bit field. It specifies the number of bytes of the section, starting immediately following the section_length field and including the CRC. The section_length shall not exceed 4093 so that the entire section has a maximum length of 4096 bytes.

service_id: This is a 16-bit field which serves as a label to identify this service from any other service within a Transport Stream. The service_id is the same as the program_number in the corresponding program_map_section.

version_number: This 5-bit field is the version number of the sub_table. The version_number shall be incremented by 1 when a change in the information carried within the sub_table occurs. When it reaches value 31, it wraps around to 0. When the current_next_indicator is set to "1", then the version_number shall be that of the currently applicable sub_table defined by the table_id and service_id. When the current_next_indicator is set to "0", then the version_number shall be that of the next applicable sub_table.

current_next_indicator: This 1-bit indicator, when set to "1" indicates that the sub_table is the currently applicable sub_table. When the bit is set to "0", it indicates that the sub_table sent is not yet applicable and shall be the next sub_table to be valid.

section_ number: This 8-bit field gives the number of the section. The section_number of the first section in the sub_table shall be "0x00". The section_number shall be incremented by 1 with each additional section with the same table_id, transport_stream_id, and original_network_id. In this case, the sub_table may be structured as a number of segments. Within each segment the section_number shall increment by 1 with each additional section, but a gap in numbering is permitted between the last_section of a segment and the first section of the adjacent segment.

last_section_number: This 8-bit field specifies the number of the last section (that is, the section with the highest section_number) of the sub_table of which this section is part.

transport_stream_id: This is a 16-bit field which serves as a label for identification of the Transport Stream, about which the EIT informs, from any other multiplex within the delivery system.

original_network_id: This 16-bit field gives the label identifying the network_id of the originating delivery system.

segment_last_section_number: This 8-bit field specifies the number of the last section of this segment of the sub_table. For sub_tables which are not segmented, this field shall set to the same value as the last_section_number field.

last_table_id: This 8-bit field identifies the last table_id used. If only one table is used this is set to the table_id of this table. The chronological order of information is maintained across successive table_id values.

event_id: This 16-bit field contains the identification number of the described event (uniquely allocated within a service definition).

start_time: This 40-bit field contains the start time of the event in Japan Standard Time (JST) and Modified Julian Date (MJD) (see Annex C). This field is coded as 16 bits giving the 16 LSBs of MJD followed by 24 bits coded as 6 digits in 4-bit Binary Coded Decimal (BCD). If the start time is undefined (e.g., for an event in a NVOD reference service), all bits of the field are set to "1".
Example 1:93/10/13 12:45:00 is coded as "0xC079124500".

duration: A 24-bit field containing the duration of the event in hours, minutes, seconds. When duration is not defined, (such as emergency news, the end time of which is not known), all bits in this field are set to "1".
format: 6 digits, 4-bit BCD = 24 bit.
Example 2: 01:45:30 is coded as "0x014530".

running_status: This is a 3-bit field indicating the status of the event as defined in table 5-6.

free_CA_mode: This 1-bit field, when set to "0" indicates that all the component streams of the event are not scrambled. When set to "1" it indicates that access to one or more streams is controlled by a CA system.

descriptors_loop_length: This 12-bit field gives the total length in bytes of the following descriptors.

CRC_32: This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in Annex B after processing the entire private section.

5.2.8 Time and Date Table (TDT)

The TDT (see table 5-8) carries only the JST-time and date information.

The TDT shall consist of a single section using the syntax of table 5-8. This TDT section shall be transmitted in TS packets with a PID value of 0x0014, and the table_id shall take the value 0x70.

Table 5-8 Time and date section

Syntax	No. of bits	Identifier
time_date_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
JST_time	40	bslbf
}		

Semantics for the time and date section:

table_id: See table 5-2.

section_syntax_indicator: This is a 1-bit indicator which shall be set to "0".

section_length: This is a 12-bit field, which shall be "0x005". It specifies the number of bytes of the section, starting immediately following the section_length field and up to the end of the section.

JST_time: (Current time and date): This 40-bit field contains the current time and date in Japan Standard Time (JST) and MJD (see Annex C). This field is coded as 16 bits giving the 16 LSBs of MJD followed by 24 bits coded as 6 digits in 4-bit BCD.

Example: 93/10/13 12:45:00 is coded as 0xC079124500

[Note] As the MJD field is 16-bit, present date can be indicated up to April 22, 2038.

5.2.9 Time Offset Table (TOT)

The TOT (see table 5-9) carries the JST-time and date information and local time offset. The TOT shall consist of a single section using the syntax of table 5-9. This TOT section shall be transmitted in TS packets with a PID value of 0x0014, and the table_id shall take the value 0x73.

Table 5-9 Time offset section

Syntax	No. of bits	Identifier
time_offset_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
JST_time	40	bslbf
reserved	4	bslbf
descriptors_loop_length	12	uimsbf
for(i=0;i<N;i++){		
descriptor()		
}		
CRC_32	32	rpchof
}		

Semantics for the time offset section:

table_id: See table 5-2.

section_syntax_ indicator: This is a 1-bit indicator which shall be set to "0".

section_length: This is a 12-bit field, the first two bits of which shall be set to "00". It specifies the number of bytes of the section, starting immediately following the section_length field and up to the end of the section.

JST_time: (Current time and date): This 40-bit field contains the current time and date in Japan Standard Time (JST) and MJD (see Annex C). This field is coded as 16 bits giving the 16 LSBs of MJD followed by 24 bits coded as 6 digits in 4-bit BCD.

descriptors_loop_length: This 12-bit field gives the total length in bytes of the following descriptors.

CRC_32: This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in Annex B after processing the entire private section.

5.2.10 Running Status Table (RST)

The RST (see table 5-10) allows accurate and rapid updating of the timing status of one or more events. This may be necessary when an event starts early or late due to scheduling changes. The use of a separate table enables a fast updating mechanism to be achieved.

The RST shall be segmented into running_status_sections using the syntax of table 5-10. Any sections forming part of an RST shall be transmitted in TS packets with a PID value of 0x0013, and the table_id shall take the value 0x71.

Table 5-10 Running status section

Syntax	No. of bits	Identifier
running_status_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
for(i=0;i<N;i++){		
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
service_id	16	uimsbf
event_id	16	uimsbf
reserved_future_use	5	bslbf
running_status	3	uimsbf
}		
}		

Semantics for the running status section:

table_id: See table 5-2.

section_syntax_ indicator: This is a 1-bit indicator which shall be set to "0".

section_length: This is a 12-bit field, the first two bits of which shall be set to "00". It specifies the number of bytes of the section, starting immediately following the section_length field and up to the end of the section. The section_length shall not exceed 1021 so that the entire section has a maximum length of 1024 bytes.

transport_stream_id: This is a 16-bit field which serves as a label for identification of the Transport Stream, about which the RST informs, from any other multiplex within the delivery system.

original_network_id: This 16-bit field gives the label identifying the network_id of the originating delivery system.

service_id: This is a 16-bit field which serves as a label to identify this service from any other service within a Transport Stream. The service_id is the same as the program_number in the corresponding program_map_section.

event_id: This 16-bit field contains the identification number of the related event.

running_status: This is a 3-bit field indicating the status of the event, as defined in table 5-6.

5.2.11 Stuffing Table (ST)

The purpose of this section (see table 5-11) is to invalidate existing sections at a delivery system boundary, e.g., at a cable head-end. When one section of a sub_table is overwritten, then all the sections of that sub_table shall also be overwritten (stuffed) in order to retain the integrity of the section_number field.

Table 5-11: Stuffing section

Syntax	No. of bits	Identifier
stuffing_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
for(i=0;i<N;i++){		
data_byte	8	uimsbf
}		
}		

Semantics for the stuffing section:

table_id: This field shall take the value "0x72" according to table 5-2.

section_syntax_indicator: This 1-bit field may take either the value "1" or "0".

section_length: This is a 12-bit field. It specifies the number of bytes of the section, starting immediately following the section_length field and up to the end of the section. The section_length shall not exceed 4093 so that the entire section has a maximum length of 4096 bytes.

data_byte: This 8-bit field may take any value and has no meaning.

5.2.12 Partial Content Announcement Table (PCAT)

Partial content announcement table (see table 5-12) is the information of transmission schedule of partial content data in accumulating type data broadcasting.

Table 5-12 Partial content announcement table

Syntax	No. of bits	Identifier
partial_content_announcement_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
service_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
content_id	32	uimsbf
num_of_content_version	8	uimsbf
for(i=0;i<num_of_content_version;i++){		
content_version	16	uimsbf
content_minor_version	16	uimsbf
version_indicator	2	bslbf
reserved_future_use	2	bslbf
content_descriptor_length	12	uimsbf
reserved_future_use	4	bslbf
schedule_description_length	12	uimsbf
for(j=0;j<N;j++){		
start_time	40	bslbf
duration	24	uimsbf
}		
for(j=0;j<N2;j++){		
descriptors()		
}		
}		
CRC_32	32	rpchof
}		

Semantics for the partial content announcement section:

table_id: See table 5-2

section_syntax_indicator: The section_syntax_indicator is a 1-bit field which shall be set to "1".

section_length: This is a 12-bit field. It specifies the number of bytes of the section, starting immediately following the section_length field and including the CRC. The section_length shall not exceed 4093 so that the entire section has a maximum length of 4096 bytes.

service_id: This is a 16-bit field which indicates service_id that announces partial original data broadcasting program and partial data. The service_id is the same as the program_number in the corresponding program_map_section.

version_number: This 5-bit field is the version number of the sub_table. The version_number shall be incremented by 1 when a change in the information carried within the sub_table occurs. When it reaches value 31, it wraps around to 0.

current_next_indicator: This 1-bit indicator, when set to "1" indicates that the sub_table is the currently applicable sub_table.

section_number: This 8-bit field gives the number of the section.

last_section_number: This 8-bit field specifies the number of the last section of the sub_table of which this section is part.

transport_stream_id: This is a 16-bit field which serves as a label to identify the TS, from any other multiplex within the delivery system.

original_network_id: This 16-bit field gives the label identifying the network_id of the originating delivery system.

content_id: This is a 32-bit field which serves as a label to identify in which partial contents the partial data belongs. The content_id is given to the original data broadcasting contents of the partial contents so that it serves as a label to identify the contents in the service uniformly.

num_of_content_version: This 8-bit field indicates the number of contents version announced in the table.

content_version: This 16-bit field indicates the total contents version of the partial contents announced in the table.

content_minor_version: This 16-bit field indicates partial contents version announced in the table.

version_indicator: This 2-bit field indicates the meaning related to contents version and contents minor version.

00: Whole version is target (designation of contents version is invalid.)

01: Target is the version after the designated version

02: Target is the version before the designated version

03: Target is only the designated version

content_descriptor_length: This 12-bit field gives the total length in bytes of the following schedule loop and descriptor loop.

schedule_description_length (Schedule description length): This 12-bit field gives the total length in bytes of the following schedule loop.

start_time (Start time): This 40-bit field indicates the start time of partial contents announcement by JST and MJD.

duration: A 24-bit field indicates the duration of the partial contents announcement by hours, minutes, and seconds.

descriptor0: Stores data contents descriptor in case of partial contents.

5.2.13 Broadcaster Information Table (BIT)

The BIT (see table 5-13) is used to submit broadcaster information on network.

Table 5-13 Broadcaster information section

Syntax	No. of bits	Identifier
broadcaster_information_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
original_network_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
reserved_future_use	3	bslbf
broadcast_view_propriety	1	bslbf
first_descriptors_length	12	uimsbf
for(i = 0;i< N1;i++){		
descriptor()		
}		
for(j = 0;j< N2;j++){		
broadcaster_id	8	uimsbf
reserved_future_use	4	bslbf
broadcaster_descriptors_length	12	uimsbf
for(k=0;k<N3;k++){		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

Semantics for the broadcaster information section

table_id: See table 5-2.

section_syntax_indicator: The section_syntax_indicator is a 1-bit field which shall be set to "1".

section_length: This is a 12-bit field. It specifies the number of bytes of the section, starting immediately following the section_length field and including the CRC. The section_length shall not exceed 4093 so that the entire section has a maximum length of 4096 bytes.

original_network_id: This 16-bit field gives the label identifying the network_id of the originating delivery system.

version_number: This 5-bit field is the version number of the sub_table. The version_number shall be incremented by 1 when a change in the information carried within the sub_table occurs. When it reaches value 31, it wraps around to 0.

current_next_indicator: This 1-bit indicator, when set to "1" indicates that the sub_table is the currently applicable sub_table.

section_number: This 8-bit field gives the number of the section.

last_section_number: This 8-bit field specifies the number of the last section (that is, the section with the highest section_number) of the sub_table to which this section in part.

This serves as a label to specify the network id of the originating delivery system.

broadcast_view_propriety: This 1-bit indicator, when set to "1" indicates that the user indication with a unit of broadcaster name is appropriate and when set to "0" indicates that the user indication with a unit of broadcaster name is not appropriate. (Each setting according to the broadcaster_id in transmission is valid.)

first_descriptors_length: This 12-bit field gives the total length in bytes of the following descriptor.

broadcaster_id: This 8-bit field identifies the broadcaster denoted with this loop.

broadcaster_descriptors_length: This 12-bit field gives the total length in bytes of the following descriptor.

CRC_32: This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in Annex B after processing the entire private section.

5.2.14 Network Board Information Table (NBIT)

The NBIT (see table 5-14) transmits board information on network, e.g. guide. There are two types of NBIT according to purpose and discriminated in table_id (see table 5-2).

1) Table describing the content of board information itself = table_id = "0xC5"

2) Table describing necessary information to obtain the content of board information = table_id = "0xC6"

Table 5-14 Network board information section

Syntax	No. of bits	Identifier
network_board_information_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
original_network_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
for(i=0;i<n;i++){		
information_id	16	uimsbf
information_type	4	uimsbf
description_body_location	2	uimsbf
reserved_future_use	2	bslbf
user_defined	8	bslbf
number_of_keys	8	uimsbf
for(j=0;j<number_of_keys;j++){		
key_id	16	uimsbf
}		
reserved_future_use	4	bslbf
descriptors_loop_length	12	uimsbf
for(j=0;j<m;j++){		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

Semantics for the network board information section:

table_id: See table 5-2.

section_syntax_indicator: The section_syntax_indicator is a 1-bit field which shall be set to "1".

section_length: This is a 12-bit field. It specifies the number of bytes of the section, starting immediately following the section_length field and including the CRC. The section_length shall not exceed 4093 so that the entire section has a maximum length of 4096 bytes.

original_network_id: This 16-bit field gives the label identifying the network_id of the originating delivery system.

version_number: This 5-bit field is the version number of the sub_table. The version_number shall be incremented by 1 when a change in the information carried within the sub_table occurs. When it reaches value 31, it wraps around to 0. When the current_next_indicator is set to "1", then the version_number shall be that of the currently applicable sub_table defined by table_id and network_id. When the current_next_indicator is set to "0", then the version_number shall be that of the next applicable sub_table defined by table_id and network_id.

current_next_indicator: This 1-bit indicator, when set to "1" indicates that the sub_table is the currently applicable sub_table. When the bit is set to "0", it indicates that the sub_table sent is not yet applicable and shall be the next sub_table to be valid.

section_number: This 8-bit field gives the number of the section. The section_number of the first section in the sub_table shall be "0x00". The section_number shall be incremented by 1 with each additional section with the same table_id and network_id.

last_section_number: This 8-bit field specifies the number of the last section (that is, the section with the highest section_number) of the sub_table of which this section is part.

information_id: This is a 16-bit field indicating ID number (allocated uniformly in the network) of the submitted information.

information_type: This 4-bit field indicates the submitted information type according to table 5-15.

Table 5-15 Information type

Value	Semantics	key_id
0x0	Undefined	—
0x1	Information	None
0x2	Information with service_id	service_id
0x3	Information with genre	content_nibble, user_nibble
0x4 – 0xF	Reserved for future use	—

description_body_location: This 2-bit field indicates the location of the table where contents of the information are described according to table 5-16.

Table 5-16 Description body location

Value	Semantics
00	Undefined
01	Detail information is described in the actual TS table
10	Detail information is described in SI prime TS table
11	Reserved for future use

user_defined: Each broadcaster can define this 8-bit field individually.

number_of_keys: This 8-bit field indicates the number of the following key_id.

key_id: This 16-bit field describes key_id according to table 5-15.

descriptors_loop_length: This 12-bit field gives total length in bytes of the following descriptors.

CRC_32: This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in Annex B after processing the entire private section.

5.2.15 Linked Description Table (LDT)

The LDT (see table 5-17) is used to link various descriptions to refer from other tables.

Table 5-17 Link description section

Syntax	No. of bits	Identifier
linked_description_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
original_service_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
for(i=0;i<n;i++){		
description_id	16	uimsbf
reserved_future_use	12	bslbf
descriptors_loop_length	12	uimsbf
for(j=0;j<m;j++){		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

Semantics for the link description section:

table_id: See table 5-2.

section_syntax_indicator: The section_syntax_indicator is a 1-bit field which shall be set to "1".

section_length: This is a 12-bit field. It specifies the number of bytes of the section, starting immediately following the section_length field and including the CRC. The section_length shall not exceed 4093 so that the entire section has a maximum length of 4096 bytes.

original_service_id: This 16-bit field indicates group identification which links descriptions in this sub_table using service_id of the representing service. It is allocated uniformly within the network.

version_number: This 5-bit field is the version number of the sub_table. The version_number shall be incremented by 1 when a change in the information carried within the sub_table occurs. When it reaches value 31, it wraps around to 0. When the current_next_indicator is set to "1", then the version_number shall be that of the currently applicable sub_table defined by table_id and network_id. When the current_next_indicator is set to "0", then the version_number shall be that of the next applicable sub_table defined by table_id and network_id.

current_next_indicator: This 1-bit indicator, when set to "1" indicates that the sub_table is the currently applicable sub_table. When the bit is set to "0", it indicates that the sub_table sent is not yet applicable and shall be the next sub_table to be valid.

section_number: This 8-bit field gives the number of the section. The section_number of the first section in the sub_table shall be "0x00". The section_number shall be incremented by 1 with each additional section with the same table_id and network_id.

last_section_number: This 8-bit field specifies the number of the last section (that is, the section with the highest section_number) of the sub_table of which this section is part.

transport_stream_id: This is a 16-bit field which serves as a label to identify the TS, from any other multiplex within the delivery system.

original_service_id: This 16-bit field gives the label identifying the service_id of the originating delivery system.

description_id: This 16-bit field indicates id_number of collected description (allocated uniformly within the representing service).

descriptors_loop_length: This 12-bit field gives total length in bytes of the following descriptors.

CRC_32: This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in Annex B after processing the entire private section.

5.2.16 Address Map Table (AMT)

[Note] This item is also explained in the Notification.

The AMT (see table 5-18) provides the list of IP packet multicast group which compose several services in the network.

Table 5-18 Address map table

Syntax	No. of bits	Identifier
address_map_table() {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
'1'	1	bslbf
'11'	2	bslbf
section_length	12	uimsbf
table_id_extension	16	uimsbf
'11'	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
num_of_service_id	10	uimsbf
reserved_future_use	6	bslbf
for(i=0; i< num_of_service_id;i++){		
service_id	16	uimsbf
ip_version	1	bslbf
reserved_future_use	5	bslbf
service_loop_length	10	uimsbf
if (ip_version=='0'){ /*IPv4*/		
src_address_32	32	bslbf
src_address_mask_32	8	uimsbf
dst_address_32	32	bslbf
dst_address_mask_32	8	uimsbf
}		
else if (ip_version=='1') { /*IPv6*/		
src_address_128	128	bslbf
src_address_mask_128	8	uimsbf
dst_address_128	128	bslbf
dst_address_mask_128	8	uimsbf
}		
for (j=0; i<N; j++) {		
private_data_byte	8	bslbf
}		
}		
CRC_32	32	rpchof
}		

Semantics for the address map table:

table_id: This is set to 0xFE which indicates to be identified by table id extension (See table 5-2).

section_syntax_indicator: This is set to ‘1’ which indicates extension type.

section_length: This specifies the number of bytes of the section, starting immediately following the section_length field and including the CRC_32.

table_id_extension: This is set to 0x0000 which indicates AMT (Address Map Table) (See table 5-2).

version_number: This field is the version number of the table. The version_number shall be incremented by 1 when a change in the information carried within the table occurs. When it reaches value 31, it wraps around to 0.

current_next_indicator: This is a 1-bit indicator, when set to "1"", it indicates that the table is the currently applicable table. When set to "0", it indicates that the table sent is not yet applicable and shall be the next table to be valid.

section_number: This field gives the number of the section. The section_number of the first section shall be "0x00". The section_number shall be incremented by 1 with each additional section with the same table_id and table_id_extension.

last_section_number: This field specifies the number of the last section (that is, the section with the highest section_number) of the table.

num_of_service_id: This indicates the number of service_id in this address map table.

service_id: This field gives the label identifying the service and provides the same function as service_id described in service_list_descriptor.

ip_version:This indicates the version of IP packet described in a list and is encoded according to table 5-19.

Table 5-19 IP version

ip_version	Version of IP packet
0	IPv4
1	IPv6

service_loop_length: This specifies the number of bytes of the service, starting immediately following the service_loop_length field. and until the previous byte of the next service_id_field.

src_address_32: This describes a source address of an IPv4 packet composing services.

src_address_mask_32: This indicates the number of active digits from MSB in a source IPv4 address, and is not more than 32.

dst_address_32: This describes a destination address of an IPv4 packet composing services.

dst_address_mask_32: This indicates the number of active digits from MSB in a destination IPv4 address, and is not more than 32. For a multicast group composing a service, a source IPv4 address to be activated by src_address_mask_32 coincides with a destination IPv4 address to be activated by dst_address_mask_32.

src_address_128: This describes a source address of an IPv6 packet composing services.

src_address_mask_128: This indicates the number of active digits from MSB in a source IPv6 address, and is not more than 128.

dst_address_128: This describes a destination address of an IPv6 packet composing services.

dst_address_mask_128: This indicates the number of active digits from MSB in a destination IPv6 address, and is not more than 128. For a multicast group composing a service, a source IPv6 address to be activated by src_address_mask_128 coincides with a destination IPv6 address to be activated by dst_address_mask_128.

private_data_byte: This stores data defined individually.

CRC_32: This is a 32-bit field that contains the CRC value that gives a zero output of the registers after processing the entire private section in the decoder defined in Annex B.

5.2.17 IP/MAC Notification Table (INT)

[Note] This item is also explained in the Notification.

The INT (See table 5-20) provides information relating IP/MAC streams composing broadcast programs.

Table 5-20 IP/MAC notification table

Syntax	No. of bits	Identifier
IP/MAC_notification_table(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
action_type	8	uimsbf
platform_id_hash	8	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
platform_id	24	uimsbf
processing_order	8	uimsbf
reserved	4	bslbf
platform_descriptor_loop_length	12	uimsbf
for (i=0; i<N1; i++) {		
platform_descriptor()		
}		
for (i=0; i<N; i++) {		
reserved	4	bslbf
target_descriptor_loop_length	12	uimsbf
for(j=0; j<N2; j++){		
target_descriptor()		
}		
reserved	4	bslbf
operational_descriptor_loop_length	12	uimsbf
for(j=0; j<N3; j++){		
operational_descriptor()		
}		
}		
CRC_32	32	rpchof
}		

Semantics for the IP/MAC notification table:

table_id: See table 5-2.

section_syntax_indicator: This 1-bit field is set to 1.

section_length: The first 2-bit of this 12-bit field is set to '00'. It specifies the number of bytes of the section, starting immediately following the section_length field and including the CRC. The section_length shall not exceed 4093 so that the entire section has a maximum length of 4096 bytes.

action_type: This 8-bit field indicates location of an IP/MAC stream. The allocation of the filed values are shown in table 5-21.

Table 5-21 Allocation of action_type

Value	Semantics
0x00	Undefined
0x01	Location of IP/MAC stream in multimedia broadcasting
0x02 – 0xFF	Undefined

platform_id_hash: This 8-bit field is set to the value which is provided by XOR of platform_id [23..16], platform_id [15..8], platform_id [7..0] every bit.

version_number: This 5-bit field is the version number of the sub_table. The version_number shall be incremented by 1 when a change in the information carried within the sub_table occurs. When it reaches value 31, it wraps around to 0. When the current_next_indicator is set to "1", then the version_number shall be that of the currently applicable sub_table defined by table_id, platform_id and action_type. When the current_next_indicator is set to "0", then the version_number shall be that of the next applicable sub_table defined by table_id, platform_id and action_type.

current_next_indicator: This 1-bit indicator, when set to "1" indicates that the sub_table is the currently applicable sub_table. When the bit is set to "0", it indicates that the sub_table sent is not yet applicable and shall be the next sub_table to be valid.

section_number: This 8-bit field gives the number of the section. The section_number of the first section in the sub_table shall be "0x00". The section_number shall be incremented by 1 with each additional section with the same table_id, platform_id and action_type.

last_section_number: This 8-bit field specifies the number of the last section (that is, the section with the highest section_number) of the sub_table of which this section belongs.

platform_id: This 24-bit field identifies platform of IP services. The allocation of field values is specified by standardization organization. (see Annex P)

processing order: This 8-bit field indicates the order of processing shown in table 5-22.

able 5-22 Processing order

Value	Semantics
0x00	Initial processing
0x01 – 0xFE	Continuous processing
0xFF	No meaning in processing order

platform_descriptor_loop_length: This 12-bit field indicates the total byte number of succeeding platform descriptor.

target_descriptor_loop_length: This 12-bit field indicates the total byte number of succeeding target descriptor.

operational_descriptor_loop_length: This 12-bit field indicates the total number of bytes of succeeding operational descriptor.

CRC_32: This is a 32-bit field that contains the CRC value that gives a zero output of the registers after processing the entire private section in the decoder defined in Annex B.

6. Descriptors

This clause describes the various descriptors that can be used within the SI.

6.1 Descriptor identification and location

Table 6-1 lists the descriptors defined within this standard, giving the intended placement within the SI tables. This does not imply that their use in other tables is restricted.

Table 6-1 Location and requirements of SI descriptors

Descriptor	Transmission level	CAT	PMT	NIT	BAT	SDT	EIT	TOT	BIT	NBIT	LDT
conditional_access_ descriptor *1	This descriptor or access control descriptor is mandatory for access control	O	O								
copyright_descriptor *1	*3		O								
network_name_descriptor *2	Mandatory			O							
service_list_descriptor *1	Mandatory in NIT/TLV-NIT (actual network) Optional in NIT (other network) Mandatory in BAT, Optional in BIT			O	O				O		
stuffing_descriptor	Optional			O	O	O	O			O	O
satellite_delivery_system _descriptor *1	Mandatory in digital satellite broadcasting			O							
bouquet_name_descriptor	Mandatory in BAT				O	O					
service_descriptor *2	Mandatory in SDT (actual stream) Optional in SDT (other stream)					O					
country_availability _descriptor	Optional		O		O	O					
linkage_descriptor	Optional		O	O	O	O	O				
NVOD_reference_service _descriptor	Mandatory for NVOD					O					
time_shifted_service _descriptor *2	Mandatory for time shift service					O					
short_event_descriptor *2	Mandatory in EIT						O				O
extended_event _descriptor	Optional						O				O
time_shifted_event_ descriptor *2	Mandatory in time shift event						O				
component_descriptor	Optional		O				O				
mosaic_descriptor	Optional		O			O					

Descriptor	Transmission level	CAT	PMT	NIT	BAT	SDT	EIT	TOT	BIT	NBIT	LDT
stream_identifier_ descriptor	Optional		o								
CA_identifier_descriptor	Optional				o	o	o				
content_descriptor	Optional						o				
parental_rating_descriptor	Optional		o				o				
hierarchical_transmission _descriptor	Mandatory for hierarchical transmission		o								
digital_copy_control _descriptor	Optional		o			o	o				
emergency_information _descriptor [*1]	Mandatory for emergency alarm broadcasting		o	o							
data_component _descriptor [*1]	Mandatory for data broadcasting		o								
system_control _descriptor [*1]	Mandatory either in PMT or NIT/TLV-NIT		o	o							
local_time_offset _descriptor	Mandatory for local time execution							o			
audio_component _descriptor	Optional						o				
hyper_link_descriptor	Optional						o		o		
target_area_descriptor	Optional		o								
data_contents_descriptor	Optional						o				
video_decode_control _descriptor	Optional		o								
terrestrial_delivery _system_descriptor [*1]	Mandatory for digital terrestrial broadcasting and multimedia broadcasting			o							
partial_reception _descriptor [*1]	Mandatory for partial reception service			o							
series_descriptor	Optional						o				
event_group_descriptor	Optional						o				
SI_transmission _parameter_descriptor	Optional								o		
broadcaster_name _descriptor	Optional								o		
component_group _descriptor	Optional						o				
SI_prime_TS_descriptor	Optional								o		
board_information _descriptor	Optional									o	
LDT_link_descriptor	Optional						o				
connected_transmission_ descriptor	Mandatory for connected transmission.			o							
TS information descriptor	Optional			o							
Extension broadcaster descriptor	Optional								o		

Descriptor	Transmission level	CAT	PMT	NIT	BAT	SDT	EIT	TOT	BIT	NBIT	LDT
Logo transmission descriptor	Optional					O					
Content availability descriptor	Optional		O			O	O				
Carousel compatible composite descriptor[*1]	Optional		O				O				
Conditional playback descriptor[*1,*5]	Mandatory in case of conditional playback[*4]	O	O								
AVC video descriptor	Optional		O								
AVC timing HRD descriptor	Optional		O								
Service group descriptor	Optional			O							
MPEG-4 audio descriptor	Mandatory in using ALS coding		O								
MPEG-4 audio extension descriptor	Mandatory in using ALS coding		O								
Registration descriptor	Mandatory in using a stream not uniquely identified by stream type.		O								
Data broadcast identification descriptor	Optional		O								
Access control descriptor[*1]	This descriptor or access control descriptor is mandatory for access control	O	O								
Area broadcasting information descriptor	Mandatory in area broadcasting			O							
HEVC video descriptor	Mandatory in case of using HEVC coding method		O								
Hierarchy descriptor[*1]	Mandatory in case of using HEVC temporal hierarchy coding method		O								
Hybrid information descriptor	Optional		O								
Scrambler descriptor[*1]	Mandatory for advanced wideband satellite digital broadcasting	O	O								

[*1]: In accordance with the Notification
[*2]: Can be substituted with the descriptor defined by service provider, if it has at least the same function.
[*3]: Locations and requirements of descriptors shall be obeyed the future international standard.
[*4]: This is not applicable when using the function with conditional access descriptor or access control descriptor.
[*5]: Specified in ARIB STD-B25

6.2 Descriptor coding

When the construct "descriptor()" appears in the sections of subclause 5.2, this indicates that zero or more of the descriptors defined within this subclause shall occur.

The following semantics apply to all the descriptors defined in this subclause.

descriptor_tag: The descriptor tag is an 8-bit field which identifies each descriptor. Those values with MPEG-2 normative meaning are described in ISO/IEC 13818-1 [21]. The values of descriptor_tag are defined in tables 5-3 and 5-4 in Part 1.

descriptor_length: The descriptor length is an 8-bit field specifying the total number of bytes of the data portion of the descriptor following the byte defining the value of this field.

6.2.1 Bouquet name descriptor

The bouquet name descriptor provides the bouquet name in text form, see table 6-2.

Table 6-2 Bouquet name descriptor

Syntax	No. of bits	Identifier
bouquet_name_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0;i<N;i++){		
char	8	uimsbf
}		
}		

Semantics for the bouquet name descriptor:
char: This is an 8-bit field, a sequence of which conveys the name of the bouquet about which the BAT sub_table informs. Text information is coded using the character sets and methods described in Annex A.

6.2.2 CA identifier descriptor

The CA identifier descriptor (see table 6-3) indicates whether a particular bouquet, service or event is associated with an access control system and identifies the access control type by means of the CA_system_id.

Table 6-3 CA identifier descriptor

Syntax	No. of bits	Identifier
CA_identifier_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0;i<N;i++){		
CA_system_id	16	uimsbf
}		
}		

Semantics for the CA identifier descriptor:

CA_system_id: This 16-bit field identifies the CA system. The standardization organization shall specify allocation of the value of this field (see Annex M).

6.2.3 Component descriptor

The component descriptor identifies the type of component stream and may be used to provide a text description of the elementary stream (see table 6-4).

Table 6-4 Component descriptor

Syntax	No. of bits	Identifier
component_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
reserved_future_use	4	bslbf
stream_content	4	uimsbf
component_type	8	uimsbf
component_tag	8	uimsbf
ISO_639_language_code	24	bslbf
for(i=0;i<N;i++){		
text_char	8	uimsbf
}		
}		

Semantics for the component descriptor:

stream_content: This 4-bit field specifies the type (video, audio, or data) of stream. The coding of this field is specified in table 6-5.

component_type: This 8-bit field specifies the type of the video, audio or data component. The coding of this field is specified in table 6-5.

component_tag: This 8-bit field has the same value as the component_tag field in the stream identifier descriptor (see sub-clause 6.2.16) (if present in the PSI program map section) for the component stream.

ISO_639_language_code: This 24-bit field identifies the language of the component (in the case of audio or data) and of the text description which may be contained in this descriptor. The ISO 639_language_code contains a 3-character code as specified by ISO 639-2[22]. Each character is coded into 8 bits according to ISO/IEC 8859-1[24] and inserted in order into the 24-bit field.

EXAMPLE: Japan has 3-character code "jpn", which is coded as:

"0110 1010 0111 0000 0110 1110"

text_char: This is an 8-bit field. A string of "text_char" fields specifies a text description of the component stream. Text information is coded using the character sets and methods described in Annex A.

Table 6-5 stream_content and component_type

Stream_content	Component_type	Description
0x0	0x00 – 0xFF	Reserved for future use
0x1	0x00	Reserved for future use
0x1	0x01	Video 480i(525i), 4:3 aspect ratio
0x1	0x02	Video 480i(525i), 16:9 aspect ratio, with pan vectors
0x1	0x03	Video 480i(525i), 16:9 aspect ratio, without pan vectors
0x1	0x04	Video 480i(525i), >16:9 aspect ratio
0x1	0x05 – 0x82	Reserved for future use
0x1	0x83	Video 4320p, 16:9 aspect ratio
0x1	0x84 – 0x90	Reserved for future use
0x1	0x91	Video 2160p, 4:3 aspect ratio
0x1	0x92	Video 2160p, 16:9 aspect ratio, with pan vectors
0x1	0x93	Video 2160p, 16:9 aspect ratio, without pan vectors
0x1	0x94	Video 2160p, >16:9 aspect ratio
0x1	0x95 – 0xA0	Reserved for future use
0x1	0xA1	Video 480p(525p), 4:3 aspect ratio
0x1	0xA2	Video 480p(525p), 16:9 aspect ratio, with pan vectors
0x1	0xA3	Video 480p(525p), 16:9 aspect ratio, without pan vectors
0x1	0xA4	Video 480p(525p), >16:9 aspect ratio
0x1	0xA5 – 0xB0	Reserved for future use
0x1	0xB1	Video 1080i(1125i), 4:3 aspect ratio
0x1	0xB2	Video 1080i(1125i), 16:9 aspect ratio, with pan vectors
0x1	0xB3	Video 1080i(1125i), 16:9 aspect ratio, without pan vectors

0x1	0xB4	Video 1080i(1125i), >16:9 aspect ratio
0x1	0xB5 – 0xC0	Reserved for future use
0x1	0xC1	Video 720p(750p), 4:3 aspect ratio
0x1	0xC2	Video 720p(750p), 16:9 aspect ratio, with pan vectors
0x1	0xC3	Video 720p(750p), 16:9 aspect ratio, without pan vectors
0x1	0xC4	Video 720p(750p), >16:9 aspect ratio
0x1	0xC5 – 0xD0	Reserved for future use
0x1	0xD1	Video 240p, 4:3 aspect ratio
0x1	0xD2	Video 240p, 4:3 aspect ratio, with pan vectors
0x1	0xD3	Video 240p, 4:3 aspect ratio, without pan vector
0x1	0xD4	Video 240p, 4:3 aspect ratio > 16:9
0x1	0xD5 – 0xE0	Reserved for future use
0x1	0xE1	Video 1080p(1125p), 4:3 aspect ratio
0x1	0xE2	Video 1080p(1125p), 16:9 aspect ratio, with pan vectors
0x1	0xE3	Video 1080p(1125p), 16:9 aspect ratio, without pan vectors
0x1	0xE4	Video 1080p(1125p), >16:9 aspect ratio
0x1	0xE5 – 0xF0	Reserved for future use
0x1	0xF1	Video 180p, 4:3 aspect ratio
0x1	0xF2	Video 180p, 16:9 aspect ratio, with pan vectors
0x1	0xF3	Video 180p, 16:9 aspect ratio, without pan vector
0x1	0xF4	Video 180p, aspect ratio > 16:9
0x1	0xF5 – 0xFF	Reserved for future use
0x2 – 0x4	0x00 – 0xFF	Refer to audio component descriptor (6.2.26)
0x5	0x00	Reserved for future use
0x5	0x01	H.264|MPEG-4 AVC, Video 480i(525i), 4:3 aspect ratio
0x5	0x02	H.264|MPEG-4 AVC, Video 480i(525i), 16:9 aspect ratio with pan vectors
0x5	0x03	H.264|MPEG-4 AVC, Video 480i(525i), 16:9 aspect ratio without pan vectors
0x5	0x04	H.264|MPEG-4 AVC, Video 480i(525i), aspect ratio > 16:9
0x5	0x05 – 0x90	Reserved for future use
0x5	0x91	H.264|MPEG-4 AVC, Video 2160p, 4:3 aspect ratio
0x5	0x92	H.264|MPEG-4 AVC, Video 2160p, 16:9 aspect ratio with pan vectors
0x5	0x93	H.264|MPEG-4 AVC, Video 2160p, 16:9 aspect ratio without pan vectors
0x5	0x94	H.264|MPEG-4 AVC, Video 2160p, aspect ratio > 16:9
0x5	0x95 – 0xA0	Reserved for future use
0x5	0xA1	H.264|MPEG-4 AVC, Video 480p(525p), 4:3 aspect ratio
0x5	0xA2	H.264|MPEG-4 AVC, Video 480p(525p), 16:9 aspect ratio with pan vectors

0x5	0xA3	H.264|MPEG-4 AVC, Video 480p(525p), 16:9 aspect ratio without pan vectors
0x5	0xA4	H.264|MPEG-4 AVC, Video 480p(525p), aspect ratio > 16:9
0x5	0xA5 – 0xB0	Reserved for future use
0x5	0xB1	H.264|MPEG-4 AVC, Video 1080i(1125i), 4:3 aspect ratio
0x5	0xB2	H.264|MPEG-4 AVC, Video 1080i(1125i), 16:9 aspect ratio with pan vectors
0x5	0xB3	H.264|MPEG-4 AVC, Video 1080i(1125i), 16:9 aspect ratio without pan vectors
0x5	0xB4	H.264|MPEG-4 AVC, Video 1080i(1125i), aspect ratio > 16:9
0x5	0xB5 – 0xC0	Reserved for future use
0x5	0xC1	H.264|MPEG-4 AVC, Video 720p(750p), 4:3 aspect ratio
0x5	0xC2	H.264|MPEG-4 AVC, Video 720p(750p), 16:9 aspect ratio with pan vectors
0x5	0xC3	H.264|MPEG-4 AVC, Video 720p(750p), 16:9 aspect ratio without pan vectors
0x5	0xC4	H.264|MPEG-4 AVC, Video 720p(750p), aspect ratio > 16:9
0x5	0xC5 – 0xD0	Reserved for future use
0x5	0xD1	H.264|MPEG-4 AVC, Video 240p, 4:3 aspect ratio
0x5	0xD2	H.264|MPEG-4 AVC, Video 240p, 16:9 aspect ratio with pan vectors
0x5	0xD3	H.264|MPEG-4 AVC, Video 240p, 16:9 aspect ratio without pan vectors
0x5	0xD4	H.264|MPEG-4 AVC, Video 240p, aspect ratio > 16:9
0x5	0xD5 – 0xE0	Reserved for future use
0x5	0xE1	H.264|MPEG-4 AVC, Video 1080p(1125p), 4:3 aspect ratio
0x5	0xE2	H.264|MPEG-4 AVC, Video 1080p(1125p), 16:9 aspect ratio with pan vectors
0x5	0xE3	H.264|MPEG-4 AVC, Video 1080p(1125p), 16:9 aspect ratio without pan vectors
0x5	0xE4	H.264|MPEG-4 AVC, Video 1080p(1125p), aspect ratio > 16:9
0x5	0xE5 – 0xF0	Reserved for future use
0x5	0xF1	H.264|MPEG-4 AVC, Video 180p, 4:3 aspect ratio
0x5	0xF2	H.264|MPEG-4 AVC, Video 180p, 16:9 aspect ratio with pan vectors
0x5	0xF3	H.264|MPEG-4 AVC, Video 180p, 16:9 aspect ratio without pan vectors
0x5	0xF4	H.264|MPEG-4 AVC, Video 180p, aspect ratio > 16:9
0x5	0xF5 – 0xFF	Reserved for future use
0x6 – 0x8	0x00 – 0xFF	Reserved for future use

0x9	0x00- 0x82	Reserved for future use
0x9	0x83	H.265|MPEG-H HEVC, Video 4320p, 16:9 aspect ratio
0x9	0x84 – 0x92	Reserved for future use
0x9	0x93	H.265|MPEG-H HEVC, Video 2160p, 16:9 aspect ratio
0x9	0x94 – 0xB2	Reserved for future use
0x9	0xB3	H.265|MPEG-H HEVC, Video 1080i(1125i), 16:9 aspect ratio
0x9	0xB4 – 0xE2	Reserved for future use
0x9	0xE3	H.265|MPEG-H HEVC, Video 1080p(1125p), 16:9 aspect ratio
0x9	0xE4 – 0xFF	Reserved for future use
0xA – 0xB	0x00 – 0xFF	Reserved for future use
0xC – 0xF	0x00 – 0xFF	User-defined

6.2.4 Content descriptor

The intention of the content descriptor (see table 6-6) is to provide classification information for an event.

Table 6-6 Content descriptor

Syntax	No. of bits	Identifier
content_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0;i<N;i++){		
content_nibble_level_1	4	uimsbf
content_nibble_level_2	4	uimsbf
user_nibble	4	uimsbf
user_nibble	4	uimsbf
}		
}		

Semantics of the content descriptor:

content_nibble_level_1: This 4-bit field represents the first level of a content identifier. Coding of this field shall be specified otherwise (see Annex H).

content_nibble_level_2: This 4-bit field represents the second level of a content identifier. Coding of this field shall be specified otherwise (see Annex H).

user_nibble: This 4-bit field is defined by the broadcaster.

6.2.5 Country availability descriptor

In order to identify various combinations of countries efficiently, the descriptor may appear twice for each service, once giving a list of countries and/or groups of countries where the service is intended to be available, and the second giving a list of countries and/or groups where it is not. The latter list overrides the former list. If only one descriptor is used, which lists countries where the service is intended to be available, it indicates that the service is not intended to be available in any other country. If only one descriptor is used, which lists countries where the service is not intended to be available, it indicates that the service is intended to be available in every other country. If no descriptor is used, then it is not defined for which countries the service is intended to be available (see table 6-7).

Table 6-7 Country availability descriptor

Syntax	No. of bits	Identifier
country_availability_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
country_availability_flag	1	bslbf
reserved_future_use	7	bslbf
for(i=0;i<N;i++){		
country_code	24	bslbf
}		
}		

Semantics for the country availability descriptor:

country_availability_flag: This 1-bit indicates whether the following country codes represent the countries in which the reception of the service is intended or not. If country_availability_flag is set to "1" the following country codes specify the countries in which the reception of the service is intended. If set to "0", the following country codes specify the countries in which the reception of the service is not intended.

country_code: This 24-bit field identifies a country using the 3-character code as specified in ISO 3166 [23]. Each character is coded into 8-bits according to ISO/IEC 8859-1 [24] and inserted in order into the 24-bit field.

EXAMPLE: Japan has 3-character code "JPN", which is coded as:

"0100 1010 0101 0000 0100 1110"

6.2.6 Satellite delivery system descriptor

The satellite delivery system descriptor indicates the physical conditions of the satellite transmission path. See table 6-8.

Table 6-8 Satellite delivery system descriptor

Syntax	No. of bits	Identifier
Satellite_delivery_system_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
Frequency	32	bslbf
orbital_position	16	bslbf
west_east_flag	1	bslbf
polarization	2	bslbf
modulation	5	bslbf
system_rate	28	bslbf
FEC_inner	4	bslbf
}		

Semantics for satellite delivery system descriptor:

frequency: The frequency is a 32-bit field giving the 4-bit BCD values specifying 8 characters of the frequency value. For the satellite_delivery_system_descriptor, the frequency is coded in GHz, where the decimal point occurs after the third character (e.g. 012.73300GHz)

orbital_position: The orbital_position is a 16-bit field giving the 4-bit BCD values specifying 4 characters of the orbital position in degrees where the decimal point occurs after the third character (e.g. 144.0 degrees).

west_east_flag: The west_east_flag is a 1-bit field indicating if the satellite position is in the western or eastern part of the orbit. A value "0" indicates the western position and a value "1" indicates the eastern position.

polarization: The polarization is a 2-bit field specifying the polarization of the transmitted signal. The first bit defines whether the polarization is linear or circular (see table 6-9).

Table 6-9 Polarization

Polarization	Description
00	linear - horizontal
01	linear - vertical
10	circular - left
11	circular - right

modulation: This is a 5-bit field. It specifies the modulation scheme used on a satellite delivery system according to table 6-10.

Table 6-10 Modulation scheme for satellite

Modulation bit 43210	Description
0 0000	Not defined
0 0001	QPSK
0 1000	Wide broad-band satellite digital broadcasting system (refer to TMCC signal)
0 1001	Not defined
0 1010	Advanced narrow-band CS digital broadcasting system (refer to PLHEADER and BBHEADER)
0 1011	Advanced wide band satellite digital broadcasting (refer to TMCC signal)
0 0010 – 0 0111 0 1100 – 1 1111	Reserved for future use

symbol_rate: The symbol_rate is a 28-bit field giving the 4-bit BCD values specifying 7 characters of the symbol_rate in Msymbol/s where the decimal point occurs after the third character (e.g. 027.4500).

FEC_inner: The FEC_inner is a 4-bit field specifying the inner FEC scheme used according to table 6-11.

Table 6-11 Inner FEC scheme

FEC_inner bit 3210	Description
0000	Not defined
0001	1/2 conv. code rate
0010	2/3 conv. code rate
0011	3/4 conv. code rate
0100	5/6 conv. code rate
0101	7/8 conv. code rate
1000	wide band satellite digital broad-casting system (refer to TMCC signal)
1001	Not defined
1010	Advanced narrow-band CS digital broadcasting system (refer to PLHEADER)
1011	Advanced wide broad-band satellite digital broadcasting (refer to TMCC signal)
1111	No conv. coding
0110 – 0111 1100 – 1110	Reserved for future use

6.2.7 Extended event descriptor

The extended event descriptor provides (see table 6-12) a detailed text description of an event, which may be used in addition to the short event descriptor. More than one extended event descriptor can be associated to allow information about one event greater in length than 256 bytes to be conveyed. Text information can be structured into two columns, one giving an item description field and the other the item text. A typical application for this structure is to give a cast list, where for example the item description field might be "Producer" and the item field would give the name of the producer.

Table 6-12 Extended event descriptor

Syntax	No. of bits	Identifier
Extended_even_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
descriptor_number	4	uimsbf
last_descriptor_number	4	uimsbf
ISO_639_language_code	24	bslbf
length_of_items	8	uimsbf
for(i=0;i<N;i++){		
item_description_length	8	uimsbf
for(j=0;j<N;j++){		
item_descriptor_char	8	uimsbf
}		
item_length	8	uimsbf
for(j=0;j<N;j++){		
item_char	8	uimsbf
}		
}		
text_length	8	uimsbf
for(i=0;i<N;i++){		
text_char	8	uimsbf
}		
}		

Semantics for the extended event descriptor:

descriptor_number: This 4-bit field gives the number of the descriptor. It is used to associate information which cannot be fitted into a single descriptor. The descriptor_number of the first extended_event_descriptor of an associated set of extended_event_descriptors shall be "0x0". The descriptor_number shall be incremented by 1 with each additional extended_event_descriptor in this section.

last_descriptor_number: This 4-bit field specifies the number of the last extended_event_descriptor (that is, the descriptor with the highest value of descriptor_number) of the associated set of descriptors of which this descriptor is part.

ISO_639_language_code: This 24-bit field identifies the language of the following text fields. The ISO 639_language_code contains a 3-character code as specified by ISO 639-2 [22]. Each character is coded into 8 bits according to ISO 8859-1 [24] and inserted in order into the 24-bit field.

EXAMPLE: Japan has 3-character code "jpn", which is coded as:

"0110 1010 0111 0000 0110 1110"

length_of_items: This is an 8-bit field specifying the length in bytes of the following items.

item_description_length: This 8-bit field specifies the length in bytes of the item description.

item_description_char: This is an 8-bit field. A string of "item_description_char" fields specifies the item description. Text information is coded using the character sets and methods described in Annex A.

item_length: This 8-bit field specifies the length in bytes of the item text.

item_char: This is an 8-bit field. A string of "item_char" fields specify the item text. Text information is coded using the character sets and methods described in Annex A.

text_length: This 8-bit field specifies the length in bytes of the non itemized extended text.

text_char: This is an 8-bit field. A string of "text_char" fields specify the non itemized extended text. Text information is coded using the character sets and methods described in Annex A.

6.2.8 Linkage descriptor

The linkage descriptor (see table 6-13) identifies a service that can be presented if the consumer requests additional information related to a specific entity described by the SI system. The location of the linkage descriptor in the syntax indicates the entity for which additional information is available. For example a linkage descriptor located within the NIT shall point to a service providing additional information on the network, a linkage descriptor in the BAT shall provide a link to a service informing about the bouquet, etc.

A CA replacement service can be identified using the linkage descriptor. This service may be selected automatically by the IRD if the CA denies access to the specific entity described by the SI system.

Table 6-13 Linkage descriptor

Syntax	No. of bits	Identifier
linkage_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
service_id	16	bslbf
linkage_type	8	uimsbf
for(i=0;i<N;i++){		
private_data_byte	8	bslbf
}		
}		

Semantics for the linkage descriptor:

transport_stream_id: This is a 16-bit field which identifies the Transport Stream containing the information service indicated.

original_network_id: This 16-bit field gives the label identifying the network_id of the originating delivery system of the information service indicated.

service_id: This is a 16-bit field which uniquely identifies an information service within a Transport Stream. The service_id is the same as the program_number in the corresponding program_map_section. If the linkage_type field has the value 0x04, then the service_id field is not relevant, and shall be set to 0x0000.

linkage_type: This is an 8-bit field specifying the type of linkage e.g. to information (see table 6-14).

Table 6-14 Linkage type coding

Linkage_type	Description
0x00	Reserved for future use
0x01	Information service
0x02	EPG service
0x03	CA replacement service
0x04	TS containing complete Network/Bouquet SI
0x05	Service replacement service
0x06	Data broadcast service
0x07 – 0x0A	Reserved for future use
0x0B	INT
0x0C – 0x7F	Reserved for future use
0x80 – 0xBF	User defined
0xC0 – 0xFD	Reserved for future use (Standardization organization defined area)
0xFE	Reserved for re-transmission
0xFF	Reserved for future use

private_data_byte: This is an 8-bit field, the value of which is privately defined.

6.2.8.1 linkage_type and private_data_byte

When linkage_type is set to 0x0B, the private_data_type is specified as shown in table 6-14-1.

Table 6-14-1 private_data_byte in case of linkage_type set to 0x0B

Syntax	No. of bits	Identifier
platform_id_data_length	8	uimsbf
for(i=0; i<N; i++){		
platform_id	24	uimsbf
platform_name_loop_length	8	uimsbf
for(j=0; j<M; j++){		
ISO_639_language_code	24	bslbf
plarform_name_length	8	uimsbf
for(k=0; k<L; k++){		
text_char	8	uimsbf
}		
}		
}		
for (i=0;i<N;i++){		
private_data_byte	8	uimsbf
}		

platform_id_data_length: This 8-bit field indicates the total byte length of the succeeding platform identification loop.

platform_id: This 24-bit field identifies platform of IP service. The allocation of this field is specified by standard organization. (see Annex P)

platform_name_loop_length: This 8-bit field indicates the total byte length of the succeeding platform name loop.

ISO 639_language_code: This 24-bit field contains the ISO 639-2 [27] three character language code of the language of the following text fields. Each character is coded into 8 bits according to ISO8859-1 [29] and inserted in order into the 24-bit field.

EXAMPLE: Japan has 3-character code "jpn", which is coded as:

"0110 1010 0111 0000 0110 1110"

platform_name_length: This 8-bit field indicates byte length of following platform name.

text_char: This is an 8-bit field. A string of "char" fields specifies the text description for the event. Text information is coded using the character sets and methods described in Annex A.

6.2.9 Mosaic descriptor

A mosaic component is a collection of different video images to form a coded video component. The information is organized so that each specific information, when displayed, appears on a small area of a screen.

The mosaic descriptor gives a partitioning of a digital video component into elementary cells, the allocation of elementary cells to logical cells, and gives a link between the content of the logical cell and the corresponding information (e.g. bouquet, service, event etc.); see table 6-15.

Table 6-15 Mosaic descriptor

Syntax	No. of bits	Identifier
mosaic_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
mosaic_entry_point	1	bslbf
number_of_horizontal_elementary_cells	3	uimsbf
reserved_future_use	1	bslbf
number_of_vertical_elementary_cells	3	uimsbf
for(i=0,i<N; i++){		
logical_cell_id	6	uimsbf
reserved_future_use	7	bslbf
logical_cell_presentation_info	3	uimsbf
elementary_cell_field_length	8	uimsbf
for(j=0,j<elementary_cell_field_length;j++){		
reserved_future_use	2	bslbf
elementary_cell_id	6	uimsbf
}		
cell_linkage_info	8	uimsbf
if(cell_linkage_info ==0x01){		
bouquet_id	16	uimsbf
}		
if(cell_linkage_info ==0x02){		
original_network_id	16	uimsbf
transport_stream_id	16	uimsbf
service_id	16	uimsbf
}		
if(cell_linkage_info ==0x03){		
original_network_id	16	uimsbf
transport_stream_id	16	uimsbf
service_id	16	uimsbf
}		
if(cell_linkage_info ==0x04){		
original_network_id	16	uimsbf
transport_stream_id	16	uimsbf
service_id	16	uimsbf
event_id	16	uimsbf
}		
}		
}		

Semantics for mosaic descriptor:

mosaic_entry_point: This is a 1-bit field which when set to a value of "1" indicates that the mosaic is the highest mosaic in a hierarchy. A complete mosaic system could be organized in a tree structure, the flag being set to identify the entry point in the tree.

number_of_horizontal_elementary_cells: This 3-bit field indicates the number of cells of horizontal screen display, see table 6-16 for coding.

Table 6-16 Coding of horizontal_elementary_cells

Value	Meaning
0x00	one cell
0x01	two cells
0x02	three cells
0x03	four cells
0x04	five cells
0x05	six cells
0x06	seven cells
0x07	eight cells

number_of_vertical_elementary_cells: This 3-bit field indicates the number of cells of vertical screen display, see table 6-17 for coding.

Table 6-17 Coding of vertical_elementary_cells

Value	Meaning
0x00	one cell
0x01	two cells
0x02	three cells
0x03	four cells
0x04	five cells
0x05	six cells
0x06	seven cells
0x07	eight cells

logical_cell_id: This 6-bit field is coded in binary form.

Different adjacent (see figure 6-1) elementary cells may be grouped together to form a logical cell. A logical_cell_number is associated to such a group of adjacent elementary_cell_ids. The total number of logical cells shall not exceed the number of elementary cells (maximum = 64). Each elementary cell shall be allocated to one logical cell. More than one elementary cell may belong to one logical cell.

A	B	C
D	E	F
G	H	I

Cells B, D, H, F are adjacent to cell E; C is not adjacent to A or D; D is not adjacent to H.

Figure 6-1 Adjacent cells

logical_cell_presentation_info: This 3-bit field identifies the type of presentation for a logical cell.

The logical_cell_presentation information allows an identification of presentation styles, which are defined in table 6-18.

Table 6-18 Coding of logical_cell_presentation_info

Value	Meaning
0x00	undefined
0x01	video
0x02	still picture (Note 1)
0x03	graphics/text
0x04 to 0x07	reserved for future use

[Note 1]: Still picture: A coded still picture consists of a video sequence containing exactly one coded picture which is intra-coded.

elementary_cell_field_length: The elementary_cell_field_length is an 8-bit field specifying the number of bytes following this field up to and including the last elementary_cell_id in this logical_cell_id loop.

elementary_cell_id: This 6-bit field indicates in binary form the number of the cell. The value of this field is in the range 0 to N.

[Note 2]: The elementary cells are implicitly numbered from 0 to N. The value 0 is allocated to the cell of the first row (top left corner). This number is incremented from left to right and from top to bottom in such a way that the number N is allocated to the cell of the last position of the last row (bottom right corner).

cell_linkage_info: This 8-bit field identifies the type of information carried in a logical cell, see table 6-19 for coding.

Table 6-19 Coding of cell_linkage_info

Value	Meaning
0x00	undefined
0x01	bouquet related
0x02	service related
0x03	other mosaic related
0x04	event related
0x05 to 0xFF	reserved for future use

bouquet_id: This is a 16-bit field which serves as a label to identify the bouquet described by the cell.

original_network_id: This 16-bit field is a label (see subclause 5.2) which in conjunction with the following fields uniquely identifies a service, event or mosaic.

transport_stream_id: This is a 16-bit field which serves as a label identifying the transport stream which contains the service, event or mosaic described by the cell.

service_id: This is a 16-bit field which identifies a service within a transport stream. The service_id is the same as the program_number in the corresponding program_map_section.

The interpretation of this field is context sensitive, dependent on the value of cell_linkage_info:

- when cell_linkage_info = "0x02", this is the service_id of the service described by the cell.
- when cell_linkage_info = "0x03", this is the service_id of the mosaic service described by the cell.
- when cell_linkage_info = "0x04", this is the service_id of the service to which the event described by the cell belongs.

event_id: This is a 16-bit field containing the identification number of the described event.

6.2.10 Near Video On Demand (NVOD) reference descriptor

This descriptor, in conjunction with the time shifted service and time shifted event descriptors, provides a mechanism for efficiently describing a number of services which carry the same sequence of events, but with the start times offset from one another. Such a group of time-shifted services is referred to as Near Video On Demand, since a user can at any time access near to the start of an event by selecting the appropriate service of the group.

The NVOD reference descriptor (see table 6-20) gives a list of the services which together form a NVOD service. Each service is also described in the appropriate SDT sub_table by a time shifted service descriptor, see sub-clause 6.2.19.

The time shifted service descriptor associates a time shifted service with a reference_service_id. The reference_service_id is the label under which a full description of the NVOD service is given, but the reference_service_id does not itself correspond to any program_number in the program_map_section.

The time shifted event descriptor is used in the event information for each time shifted service. Instead of duplicating the full information for each event, the time shifted event descriptor points to a reference_event_id in the reference service. The full event information is provided in the event information for the reference service.

The services which make up an NVOD service need not all be carried in the same TS. However, a reference service shall be described in the SI in each TS which carries any services of the NVOD service.

Table 6-20 NVOD reference descriptor

Syntax	No. of bits	Identifier
NVOD_reference_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0;i<N;i++){		
transport_stream_id	16	uimsbf
original_netwrok_id	16	
service_id	16	uimsbf
}		
}		

Semantics for the NVOD reference descriptor:

transport_stream_id: This is a 16-bit field which identifies the Transport Stream.

original_network_id: This 16-bit field gives the label identifying the network_id of the original delivery system.

service_id: This is a 16-bit field which uniquely identifies a service within a Transport Stream. The service_id is the same as the program_number in the corresponding program_map_section.

6.2.11 Network name descriptor

The network name descriptor provides the network name in text form (see table 6-21).

Table 6-21 Network name descriptor

Syntax	No. of bits	Identifier
network_name_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0;i<N;i++){		
char	8	uimsbf
}		
}		

Semantics for the network name descriptor

char: This is an 8-bit field. A string of char fields specifies the name of the delivery system about which the NIT informs. Text information is coded using the character sets and methods described in Annex A.

6.2.12 Parental rating descriptor

This descriptor (see table 6-22) gives a rating based on age and allows for extensions based on other rating criteria.

Table 6-22 Parental rating descriptor

Syntax	No. of bits	Identifier
parental_rating_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0;i<N;i++){		
country_code	24	bslbf
Rating	8	uimsbf
}		
}		

Semantics for the parental rating descriptor:

country_code: This 24-bit field identifies a country using the 3-character code as specified in ISO 3166 [23]. Each character is coded into 8-bits according to ISO 8859-1 [24] and inserted in order into the 24-bit field.

EXAMPLE: Japan has 3-character code "JPN" which is coded as:

"0100 1010 0101 0000 0100 1110"

rating: This 8-bit field is coded according to table 6-23, giving the recommended minimum age in years of the end user.

Table 6-23 Parental rating descriptor, rating

Rating	Description
0x00	undefined
0x01 – 0x0F	minimum age = rating + 3 years
0x10 – 0xFF	defined by the broadcaster

EXAMPLE: 0x04 implies that end users should be at least 7 years old.

6.2.13 Service descriptor

The service descriptor (see table 6-24) provides the names of the service provider and the service in text form together with the service_type.

Table 6-24 Service descriptor

Syntax	No. of bits	Identifier
service_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
service_type	8	uimsbf
service_provider_name_length	8	uimsbf
for(i=0;i<N;i++){		
char	8	uimsbf
}		
service_name_length	8	uimsbf
for(i=0;i<N;i++){		
char	8	uimsbf
}		
}		

Semantics for the service descriptor:

service_type: This is an 8-bit field specifying the type of the service. It shall be coded according to table 6-25.

[Note]: This field is expressed as "service_type_id" in the Notification.

Table 6-25 Service type coding

Service_type	Description
0x00	Not defined
0x01	Digital television service
0x02	Digital audio service
0x03 – 0x7F	Not defined
0x80 – 0xA0	Service provider defined
0xA1	Special video service
0xA2	Special audio service
0xA3	Special data service
0xA4	Engineering service
0xA5	Promotion video service
0xA6	Promotion audio service
0xA7	Promotion data service
0xA8	Data service for accumulation in advance
0xA9	Data service exclusive for accumulation
0xAA	Book mark list service
0xAB	Server-type simultaneous service
0xAC	Independent file service
0xAD	Ultra HD 4K TV service
0xAE – 0xBF	Not defined (to be defined by standardization organization)
0xC0	Data service
0xC1	Storage service using TLV
0xC2	Multimedia service
0xC3 – 0xFF	Not defined

service_provider_name_length: This 8-bit field specifies the number of bytes that follow the service_provider_name_length field for describing characters of the name of the service provider.

char: This is an 8-bit field. A string of char fields specifies the name of the service provider or service. Text information is coded using the character sets and methods described in Annex A.

service_name_length: This 8-bit field specifies the number of bytes that follow the service_name_length field for describing characters of the name of the service.

6.2.14 Service list descriptor

The service list descriptor (see table 6-26) provides a means of listing the services by service_id and service type.

Table 6-26 Service list descriptor

Syntax	No. of bits	Identifier
service_list_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0;i<N;i++){		
service_id	16	uimsbf
service_type	8	uimsbf
}		
}		

Semantics for the service list descriptor:

service_id: This is a 16-bit field which uniquely identifies a service within a Transport Stream. The service_id is the same as the program_number in the corresponding program_map_section.

service_type: This is an 8-bit field specifying the type of the service. It shall be coded according to table 6-25.

6.2.15 Short event descriptor

The short event descriptor provides the name of the event and a short description of the event in the text form (table 6-27).

Table 6-27 Short event descriptor

Syntax	No. of bits	Identifier
short_event_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
ISO_639_language_code	24	uimsbf
event_name_length	8	uimsbf
for(i=0;i<event_name_langth;i++){		
event_name_char	8	uimsbf
}		
text_length	8	uimsbf
for(i=0;i<text_length;i++){		
text_char	8	uimsbf
}		
}		

Semantics for the short event descriptor:

ISO 639_language_code: This 24-bit field contains the ISO 639-2 [27] three character language code of the language of the following text fields. Each character is coded into 8 bits according to ISO8859-1 [29] and inserted in order into the 24-bit field.

EXAMPLE: Japan has 3-character code "jpn", which is coded as:
"0110 1010 0111 0000 0110 1110"

event_name_length: An 8-bit field specifying the length in bytes of the event name.

event_name_char: This is an 8-bit field. A string of "char" fields specifies the event name. Text information is coded using the character sets and methods described in Annex.

text_length: This 8-bit field specifies the length in bytes of the following text describing the event.

text_char: This is an 8-bit field. A string of "char" fields specifies the text description for the event. Txt information is coded using the character sets and methods described in Annex A.

6.2.16 Stream identifier descriptor

The stream identifier descriptor (see table 6-28) may be used in the PMT to label component streams of a service (table 6-5) given in component descriptors in the EIT if present so that they can be differentiated (e.g. a component stream of a certain service is "video, 16:9 aspect ratio, with pan vector") . The stream identifier descriptor shall be located following the relevant ES_info_length_field.

Table 6-28 Stream identifier descriptor

Syntax	No. of bits	Identifier
service_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
component_tag	8	uimsbf
}		

Semantics for the stream identifier descriptor:

component_tag: This 8-bit field identifies the component stream for associating it with a description given in a component descriptor. Within a program map section each stream identifier descriptor shall have a different value for this field.

6.2.17 Stuffing descriptor

The stuffing descriptor provides a means of invalidating previously coded descriptors or inserting dummy descriptors for table stuffing (see table 6-29).

Table 6-29 Stuffing descriptor

Syntax	No. of bits	Identifier
stuffing_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0;i<N;i++){		
stuffing_byte	8	bslbf
}		
}		

Semantics for the stuffing descriptor:

stuffing_byte: This is an 8-bit field. Each occurrence of the field may be set to any value. The IRDs may discard the stuffing bytes.

6.2.18 Time shifted event descriptor

The time shifted event descriptor (see table 6-30) is used in place of the short_event_descriptor to indicate an event which is a time shifted copy of another event.

Table 6-30 Time shifted event descriptor

Syntax	No. of bits	Identifier
time_shifted_event_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
reference_service_id	16	uimsbf
reference_event_id	16	uimsbf
}		

Semantics for the time shifted event descriptor:

reference_service_id: This 16-bit field identifies the reference service of a NVOD collection of services. The reference service can always be found in this Transport Stream. The service_id here does not have a corresponding program_number in the program_map_section.

reference_event_id: This 16-bit field identifies the reference event of which the event described by this descriptor is a time shifted-copy.

6.2.19 Time shifted service descriptor

This descriptor is used in place of the service descriptor to indicate services which are time shifted copies of other services (see table 6-31).

Table 6-31 Time shifted service descriptor

Syntax	No. of bits	Identifier
time_shifted_service_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
reference_service_id	16	uimsbf
}		

Semantics for the time shifted service descriptor:

reference_service_id: This 16-bit field identifies the reference service of a NVOD collection of services. The reference service can always be found in this Transport Stream. The service_id here does not have a corresponding program_number in the program_map_section.

6.2.20 Data component descriptor

[Note] This item is specified in Notification.

The data component descriptor (see table 6-32) is used to identify data components.

Table 6-32 Data component descriptor

Syntax	No. of bits	Identifier
data_component_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
data_component_id	16	uimsbf
for(i=0;i<N;i++){		
additional_data_component_info	8	uimsbf
}		
}		

Semantics for the data component descriptor:

data_component_id: This 16-bit field is used to identify data coding method. The standardization organization shall allocate this field value.

additional_data_component_info: This is an 8-bit field and used to extend identifier number or storage of supplement information specified in each coding method. Syntax of information described in this area is specified otherwise for each data coding method. (see Annex J)

6.2.21 System management descriptor

[Note] This item is specified in Notification.

The system management descriptor (see table 6-33) is used to identify broadcasting and non-broadcasting.

Table 6-33 System management descriptor

Syntax	No. of bits	Identifier
system_management_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
system_management_id	16	uimsbf
for(i=0;i <N;i++){		
additional_identification_info	8	uimsbf
}		
}		

Semantics for the system management descriptor:

system_management_id: This is a 16-bit field and composed as shown in table 6-34.

Table 6-34 Structure of system management identifier

Syntax	No. of bits	Identifier
system_management_id(){		
broadcasting_flag	2	uimsbf
broadcasting_identifier	6	uimsbf
additional_broadcasting_identification	8	uimsbf
}		

broadcasting_flag: This is a 2-bit field and indicates type of broadcasting/non-broadcasting in accordance with table 6-35.

Table 6-35 Broadcasting/non-broadcasting type

Value	Semantics
00	Broadcasting
01,10	Non-broadcasting
11	Undefined

broadcasting_identifier: This is a 6-bit field and indicates standard broadcasting method in accordance with table 6-36.

Table 6-36 Types of standard broadcasting system

Value	Semantics
000000	Undefined
000001	Standard system specified as narrow-band digital satellite broadcasting using 27 MHz bandwidth in 12.2 to 12.75 GHz frequency band
000010	Standard system specified as wide band digital satellite broadcasting using 34.5 MHz bandwidth in 11.7 to 12.2 GHz frequency band
000011	Standard system specified as digital terrestrial television broadcasting.
000100	Standard system specified as wide band digital satellite broadcasting using 34.5 MHz bandwidth in 12.2 to 12.75 GHz frequency band
000101	Standard system specified as digital terrestrial sound broadcasting.
000110	Undefined
000111	Standard system specified as digital satellite broadcasting based on advanced narrow-band transmission system using 27 MHz bandwidth in 12.2 to 12.75 GHz frequency band
001000	Standard system specified as advanced wide broad-band digital satellite broadcasting using 34.5 MHz bandwidth in 11.7 to 12.2 GHz frequency band
001001	Standard system specified as advanced wide band digital satellite broadcasting using 34.5 MHz bandwidth in 12.2 to 12.75 GHz frequency band
001010	Standard system specified as television broadcasting and multimedia broadcasting using segment connection transmission method in 207.5 to 222 MHz frequency band.
001011	Standard system specified as multimedia broadcasting using segment connection transmission method in 99 to 108 MHz frequency band.
001100 – 111111	Undefined

additional_broadcasting_identification: This is an 8-bit field and is specified by the operation standard of service providers.

additional_identification_info: This is an 8-bit field and used to extend system management indication number.

6.2.22 Hierarchical transmission descriptor

The hierarchical transmission descriptor (see table 6-37) is used to indicate relation between hierarchical streams when transmitting events hierarchically.

Table 6-37 Hierarchical transmission descriptor

Syntax	No. of bits	Identifier
hierarchical_transmission_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
reserved_future_use	7	bslbf
quality_level	1	bslbf
reserved_future_use	3	bslbf
reference_PID	13	uimsbf
}		

Semantics for the hierarchical transmission descriptor:

quality_level: This 1-bit information indicates hierarchy level. Hierarchical structure is in two levels, HQ and LQ, and when the hierarchy level is "1", the stream is in high quality. When the level is "0", the stream is in low quality.

reference_PID: This 3-bit information indicates PID of elementary stream to be referred, for all the stream having hierarchical structure.

6.2.23 Digital copy control descriptor

The digital copy control descriptor (see table 6-38) indicates information to control copied generation in digital recording equipments, and when digital recording is assumed, broadcasting service provider (holder of copyrights) use it to inform about event recording and copyright information for the digital recording equipment. This descriptor is also used to identify maximum transmission rate to each event.

Table 6-38 Digital copy control descriptor

Syntax	No. of bits	Identifier
digital_copy_control_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
digital_recording_control_data	2	bslbf
maximum_bitrate_flag	1	bslbf
component_control_flag	1	bslbf
user_defined	4	bslbf
if(maximum_bitrate_flag == 1){		
maximum_bitrate	8	uimsbf
}		
if(component_control_flag == 1){		
component_control_length	8	uimsbf
for(i=0;i<N;i++){		
component_tag	8	uimsbf
digital_recording_control_data	2	bslbf
maximum_bitrate_flag	1	bslbf
reserved_future_use	1	bslbf
user_defined	4	bslbf
if(maximum_bitrate_flag == 1){		
maximum_bitrate	8	uimsbf
}		
}		
}		
}		

Semantics for the digital copy control descriptor:

digital_recording_control_data: This 2-bit field indicates information to control copy generation and coded in accordance with table 6-39.

Table 6-39 Digital copy control information

Digital copy control information	Description
00	Copy can be made without control condition
01	Defined by service provider *1
10	Copy can be made for only one generation *2
11	Copy is forbidden

*1: Broadcasting service provider can define independently.

*2: Received broadcasting signals can be recorded (first-generation copy) but the recorded signals cannot be reproduced furthermore.

maximum_bit_rate_flag: When this 1-bit flag is "1", it means that following maximum transmitting rate field is effective. When it is "0", following maximum transmitting rate field does not exist.

component_control_flag: This 1-bit flag indicates whether to specify digital copy control information in each component consisting event. When this flag is "1," field after component control length is effective and digital copy control information is specified in each component consisting event. When it is "0", digital copy control information is specified for the whole event and field after component control length does not exist. When this descriptor is transmitted by PMT, component control flag should always be "0".

user_defined: This is a 4-bit field, and broadcasting service provider can define it originally.

maximum_bit_rate: This 8-bit field describes transmission rate of TS packet of each event or elementary stream by rolling up in each 1/4Mbps. In case of variable transmission rate, maximum value is described.

component_control_length: This 8-bit field indicates byte length of the following component control loop.

component_tag: This is an 8-bit field. Component tag is a label to identify elementary stream of component, which composing events and is the same value as the component tag in the stream identifier descriptor and the component descriptor.

6.2.24 Emergency information descriptor

[Note] This item is specified in Notification.

The emergency information descriptor (see table 6-40) is a signal in accordance with emergency alarm signal specified in No.5 of clause 9-3 of Radio Equipment Regulation and used in case of emergency alarm broadcasting.

Table 6-40 Emergency information descriptor

Syntax	No. of bits	Identifier
emergency_information_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0;i<N;i++){		
service_id	16	uimsbf
start_end_flag	1	bslbf
signal_level	1	bslbf
reserved_future_use	6	bslbf
area_code_length	8	uimsbf
for(j=0;j<N;j++){		
area_code	12	bslbf
reserved_future_use	4	bslbf
}		
}		
}		

Semantics for the emergency information descriptor:

service_id: This 16-bit field indicates broadcasting event number. This is the same as program_number.

start_end_flag: This 1-bit flag corresponds to start signal and end signal in the emergency alarm signal specified in Notation No. 405 of the Ministry of Posts and Telecommunications in 1985. When this bit is 1, it means that emergency alarm signal has started or is being broadcast. When this bit is 0, it means that the emergency alarm signal is ended.

signal_level: This 1-bit field corresponds to emergency alarm signal specified in article 138-2 of Radio Station Operation Rule. When this bit is 0, it means that broadcast emergency alarm signal is the 1st type of start signal. When this bit is 1, it means that broadcast emergency alarm signal is the

2nd type of start signal (see Annex D).

area_code_length: This is an 8-bit field, which indicates following area code byte length.

area_code: This is a 12-bit field and corresponds to area code specified in clause 138-3 of Radio Station Operation Rule. For allocation of area code, specification in the Notation No. 405 of the Ministry of Posts and Telecommunications in 1985 is used (see Annex D).

6.2.25 Local time offset descriptor

The local time offset descriptor (see table 6-41) is used to give fixed offset value to present time (UTC + 9 hours) and indicating time for human in local time.

Table 6-41 Local time offset descriptor

Syntax	No. of bits	Identifier
local_time_offset_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0;i<N;i++){		
country_code	24	bslbf
country_region_id	6	bslbf
reserved	1	bslbf
local_time_offset_polarity	1	bslbf
local_time_offset	16	bslbf
time_of_change	40	bslbf
next_time_offset	16	bslbf
}		
}		

Semantics for the local time offset descriptor:

country_code: This 24-bit field identifies a country using the 3-character code as specified in ISO 3166. Each character is coded into 8-bits according to ISO 8859-1 and inserted in order into the 24-bit field.

EXAMPLE: Japan has 3-character code "JPN" which is coded as:
"0100 1010 0101 0000 0100 1110"

country_region_id: This 6-bit field identifies a zone in the country. Use "000000" if regions are not distinguished.

local_time_offset_polarity: This 1-bit information indicates the polarity of the value of following local_time_offset and next_time_offset. If this bit is set to "0", the local time is in advance of JST_time. If this bit is set to "1", the local time is behind JST_time.

local_time_offset: This 16-bit field contains the current offset time from JST (UTC+9 hours) in the range between -12 hours and +12 hours at the area which is indicated by the combination of country_code and country_region_id in advance. These 16 bits are coded as 4 digits in 4-bit BCD in the order hour tens, hour, minute tens, and minutes.

time_of_change: This is a 40-bit field which specifies the date and time in MJD and JST (see Annex C), when the time change takes place. This 40-bit field is coded as 16 bits giving the 16 LSBs of MJD followed by 24 bits coded as 6 digits in the 4-bit BCD.

next_time_offset: This 16-bit field contains the next offset time after the change from JST in the range between -12hours and +12hours at the area which is indicated by the combination of country_code and country_region_id in advance. These 16-bits are coded as 4-digits in 4-bit BCD in the order hour tens, hour, minute tens and minutes.

6.2.26 Audio component descriptor

The audio component descriptor is used to indicate each parameter of audio elementary stream and to express the elementary stream in character form. (see table 6-42.)

Table 6-42 Audio component descriptor

Syntax	No. of bits	Identifier
audio_component_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
reserved_future_use	4	bslbf
stream_content	4	uimsbf
component_type	8	uimsbf
component_tag	8	uimsbf
stream_type	8	uimsbf
simulcast_group_tag	8	bslbf
ES_multi_lingual_flag	1	bslbf
main_component_flag	1	bslbf
quality_indicator	2	bslbf
sampling_rate	3	uimsbf
reserved_future_use	1	bslbf
ISO_639_language_code	24	bslbf
if(ES_multi_lingual_flag == 1){		
ISO_639_language_code_2	24	bslbf
}		
for(i=0;i<N;i++){		
text_char	8	uimsbf
}		
}		

Semantics for the audio component descriptor:

stream_content: This 4-bit field indicates a stream type. For an audio stream whose coding method isn't specified, it is set to "0x2". For an MPEG-4 AAC stream, it is set to “0x3”. For an MPEG-4 ALS, it is set to “0x4”.

component_type: This 8-bit field indicates audio component type and coded in accordance with table 6-43. In 8 bits (b7- b0), the most significant 1 bit (b7) indicates whether or not the dialogue control information is included, the following 2 bits (b6 – b5) whether or not the audio for a person with disabilities is available, and, the least significant 5 bits (b4 – b0) indicates audio mode.

Table 6-43 Audio component descriptor

Dialog control (b7)	Description
0	Audio stream does not include dialog control information
1	Audio stream includes dialog control information

Audio for a person with disabilities (b6 – b5)	Description
00	Not specified audio for a person with disabilities
01	Audio commentary for a person with visually impaired
10	Audio for a person with hearing impaired
11	Reserved for future use

Audio mode (b4 – b0)	Description
00000	Reserved for future use
00001	1/0 mode (single monaural channel)
00010	1/0 + 1/0 mode (dual monaural channel)
00011	2/0 mode (stereo)
00100	2/1 mode
00101	3/0 mode
00110	2/2 mode
00111	3/1 mode
01000	3/2 mode
01001	3/2 + LFE mode (3/2.1 mode*1)
01010	3/3.1 mode*1
01011	2/0/0-2/0/2-0.1 mode*1
01100	5/2.1 mode*1
01101	3/2/2.1 mode*1
01110	2/0/0-3/0/2-0.1 mode*1
01111	0/2/0-3/0/2-0.1 mode*1
10000	2/0/0-3/2/3-0.2 mode*1
10001	3/3/3-5/2/3-3/0/0.2 mode*1
10010 - 11111	Reserved for future use

*1 Notation of audio mode in multi-channel stereo: The channel number is shown as 'upper layer (front/side/rear) – middle layer (front/side/rear) – lower layer (front/side/rear) . LFE'. The layer without allocated channel is shown as 0. In case of the audio mode only with middle layer, it is simplified as 'middle layer (front/side/rear) . LFE'. In case of the audio mode only with middle layer and without side channel, it is simplified as 'middle layer (front/rear) . LFE'.	

component_tag: This 8-bit field is a label to identify the component stream and has the same value as the component_tag field in the stream identifier descriptor (see subclause 6.2.16) (if present in the PSI program map section) for the component stream.

stream_type: This 8-bit field indicates audio stream type (MPEG2 BC Audio, MPEG-2 AAC Audio, MPEG-4 Audio) (see Annex E).

simulcast_group_tag: This 8-bit field gives the same number to the component operating simulcast (transmit the same contents by different coding method). For component, which does not operate simulcast, it is set to "0xFF".

ES_multi_lingual_flag: This 1-bit flag is set to "1" when 2-language multilingual (ES multilingual mode) is made in ES at 1/0 + 1/0 mode. In case of other mode, this bit is reserved.

main_component_flag: This 1-bit flag is set to "1" when the audio component is the main audio. In case of 1/0 + 1/0 mode, it is set to "1" when the 1st audio component is the main audio.

quality_indicator: This 2-bit field indicates tone quality mode and coded in accordance with table 6-44.

Table 6-44 Quality indicator

Quality indicator	Description
00	Reserved for future use
01	Mode 1*
10	Mode 2*
11	Mode 3*

*: For detail, refer to ARIB STD-B32 Part 2 Appendix 2.

sampling_rate: This 3-bit field indicates sampling frequency and is coded in accordance with table 6-45.

Table 6-45 Sampling frequency

Sampling frequency	Description
000	Reserved for future use
001	16kHz
010	22.05kHz
011	24kHz
100	Reserved
101	32kHz
110	44.1kHz
111	48kHz

ISO_639_language_code: This 24-bit field identifies the language of the audio component. In the case of ES multilingual mode, it indicates the first audio component language. This field contains a 3-character code as specified by ISO 639-2 (21). Each character is coded into 8 bits according to ISO 8859-1(23) and inserted in order into the 24-bit field.

EXAMPLE: Japan has 3-character code "jpn", which is coded as:
"0100 1010 0101 0000 0100 1110"

ISO_639_language_code_2: This 24-bit field identifies the second audio component language in ES multilingual mode.

text_char: This is an 8-bit field. A string of "text_char" fields specifies a text description of the component stream. Text information is coded using the character sets and methods described in Annex A.

6.2.27 Target region descriptor

Target region descriptor (see table 6-46) is used to describe target region of the program or a part of the stream composing a program.

Table 6-46 Target region descriptor

Syntax	No. of bits	Identifier
target_region_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
region_spec_type	8	uimsbf
target_region_spec()		
}		

Semantics for the target region descriptor:

region_spec_type: This 8-bit field designates region description method in the following target_region_spec() structure and coded in accordance with table 6-47.

Table 6-47 Region description method designation

Value of region_spec_type	Semantics
0x00	Reservation
0x01	Region designation of prefecture for BS digital
0x02 – 0xFF	Reservation

target_region_spec(): This field indicates syntax for the target region specified by eachregion_sepc_type (see Annex G).

6.2.28 Data content descriptor

The data content descriptor (see table 6-48) is used to describe detail information relating to individual contents of data broadcasting event.

Table 6-48 Data contents descriptor

Syntax	No. of bits	Identifier
data_content_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
data_component_id	16	uimsbf
entry_component	8	uimsbf
selector_length	8	uimsbf
for(i=0;i<N;i++){		
selector_byte	8	uimsbf
}		
num_of_component_ref	8	uimsbf
for(i=0;i<num_of_component_ref;i++){		
component_ref	8	uimsbf
}		
ISO_639_language_code	24	bslbf
text_length	8	uimsbf
for(i=0;i<N;i++){		
text_char	8	uimsbf
}		
}		

Semantics for the data contents descriptor:

data_component_id: This is a 16-bit field and describes the same value as the data component identifier of data component descriptor.

entry_component: This 8-bit field designates the component stream including data to be referred first, among multiple component streams composing data broadcasting contents, using component tag.

selector_length: This 8-bit field specifies byte length of the following selector area.

selector_byte: This is an 8-bit field. Series of selector area describes necessary information to get data. Syntax described in this area is specified otherwise in each data component. (See Annex J)

num_of_component_ref: This 8-bit field indicates number of all component stream in the event it is necessary to playback and record contents indicated by this descriptor (however, component stream designated by entry component is excluded). This number corresponds to byte length of the following component reference loop.

component_ref: This 8-bit field describes component tag of the component stream in the event it is necessary to watch or record the contents (however, component stream designated by the entry component is excluded).

ISO_639_language_code: This 24-bit field identifies the language of the character description used in the following service descriptor containing a 3-character code as specified by ISO 639-2.

text_length: This 8-bit field indicates byte length of following contents descriptor.

text_char: This is an 8-bit field. A string of "text_char" fields describes the explanation related to transmitted contents.

6.2.29 Hyperlink descriptor

The hyperlink descriptor (see table 6-49) is used to describe linkage to other event, event contents, and events relating to information.

Table 6-49 Hyperlink descriptor

Syntax	No. of bits	Identifier
hyperlink_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
hyper_linkage_type	8	uimsbf
link_destination_type	8	uimsbf
selector_length	8	uimsbf
for(i=0; i<selector_length; i++){		
selector_byte	8	uimsbf
}		
for(i=0; i<N; i++){		
private_data	8	uimsbf
}		
}		

Semantics for the hyperlink descriptor:

hyper_linkage_type: This 8-bit field indicates linkage type and coded in accordance with table 6-50.

Table 6-50 Hyperlink descriptor

hyper_linkage_type	Semantics
reserved(0x00)	Reservation
combined_data(0x01)	Used to indicate data broadcast event that transport SI related to TV event, when the SI is transported in other time frame or other event. In order to make reservation or playback of the linkage destination data-broadcast event when reserving or playing back TV event, it is described as detail SI of the TV event.
combined_stream(0x02)	Used to indicate TV event that relates to SI transported in data broadcast event, when the SI is transported in other time frame or other event. In order to make reservation or playback of the linkage destination data broadcast event when reserving or playing back TV event, it is described as detail SI of the data broad cast event.
content_to_index(0x03)	Used to indicate event that transport internal index information related to TV event, when the internal index information is transported in other time frame or other event. In order to accumulate or utilize the related index information when accumulating or playing back TV event, it is described as detail internal index information of the TV event.
index_to_content(0x04)	Used to indicate TV event that relates to internal index information transported in event, when the internal index information is transported in other time frame or other event. In order to accumulate or playback the related TV event when accumulating or utilizing index information, it is described as detail information of the index information event.
guide_data(0x05)	Used to indicate data broadcast event that transport the event guide information related to this event, when the event guide information is transported in other data broadcast event. In order to get detail information on event guide application as requested, it indicates the designated data.
(0x06)	Undefined
content_to_metadata(0x07)	Used to indicate event and service that transport metadata related to TV event or data broadcast event, when metadata is broadcasted in time frame or service different from TV event or data broadcast event. In order to accumulate or utilize the related metadata when accumulating or playing back TV event or data broadcast event, it is described as detail information of TV event or data broadcast event.
metadata_to_content(0x08)	Used to indicate TV event or data broadcast event that relate to metadata transported in event or service, when metadata is broadcasted in time frame or service different from TV event or data broadcast event. In order to accumulate or playback the related TV event or data broadcast event when storing or utilizing metadata, it is described as detail information of metadata event.

portal_URI (0x09)	Used to indicate the URI of the portal link destination in server-type broadcasting. The URI of the portal link destination corresponds to the URI of the BML document provided by the broadcaster for the contract between the broadcaster and the audience.
authority_URI (0x0A)	Used to indicate the URI of the authority in server-type broadcasting. The authority is the character string used as the name space for each broadcaster when accumulating server-type contents in server-type broadcasting receivers.
(0x0B – 0x3F)	Undefined
index_module(0x40)	Used only for LIT used as internal index information in data broadcast event, to indicate correspondence of local event identifier and data broadcasting event module. Operation depends on receiver application using the internal index information.
(0x41 – 0x7F)	Undefined
user_private(0x80 – 0xFF)	Linkage type defined by the users.

link_destination_type: This 8-bit field indicates link designation type and coded in accordance with table 6-51.

Table 6-51 Link destination type

link_destination_type	selector_length	Target of link
reserved(0x00)	-	
link_to_service(0x01)	6	Service
link_to_event(0x02)	8	Event
link_to_module(0x03)	11	Specific module of event
link_to_content(0x04)	10	Content
link_to_content_module(0x05)	13	Specific module of content
link_to_ert_node(0x06)	6	Node of event related table
link_to_stored_content(0x07)	Variable length	Accumulated content
reserved_future_use(0x08 – 0x7F)		Reserved for future use
user_private(0x80 – 0xFE)	-	Link destination type of user definition
reserved(0xFF)	-	

selector_length: This 8-bit field indicates byte length of the following selector area.

selector_byte: This is an 8-bit field. Series of selector area describes link destination by the following type specified in each link destination type.

Table 6-52 Description of selector area (link_destination_type: 0x01)

Syntax (link_destination_type:0x01)	No. of bits	Identifier
link_service_info(){		
original_network_id	16	uimsbf
transport_stream_id	16	uimsbf
service_id	16	uimsbf
}		

original_network_id: This 16-bit field gives the label identifying the network_id of the originating delivery system where the linked service belongs.

transport_stream_id: This 16-bit field gives the label identifying the Transport Stream where the linked service belongs.

service_id: This 16-bit field gives the label identifying the service in the linked Transport Stream and describes the same service_id as the program_number in the corresponding program map section.

Table 6-53 Description of selector area (link_destination_type: 0x02)

Syntax (link_destination_type:0x02)	No. of bits	Identifier
link_event_info(){		
original_network_id	16	uimsbf
transport_stream_id	16	uimsbf
service_id	16	uimsbf
event_id	16	uimsbf
}		

original_network_id: This 16-bit field gives the label identifying the network_id of the originating delivery system where the linked event belongs.

transport_stream_id: This 16-bit field gives the label identifying the Transport Stream where the linked event belongs.

service_id: This 16-bit field gives the label identifying the service in the Transport Stream where the linked event belongs and describes the same service_id as the program_number in the corresponding program map section.

event_id: This 16-bit field describes the identifier number of the linked event.

Table 6-54 Description of selector area (link_destination_type: 0x03)

Syntax (link_destination_type:0x03)	No. of bits	Identifier
link_module_info(){		
original_network_id	16	uimsbf
transport_stream_id	16	uimsbf
service_id	16	uimsbf
event_id	16	uimsbf
component_tag	8	uimsbf
moduleId	16	uimsbf
}		

original_network_id: This 16-bit field describes the label identifying the network_id of the originating delivery system where the linked carousel module belongs.

transport_stream_id: This 16-bit field describes the label identifying the Transport Stream where the linked carousel module belongs.

service_id: This 16-bit field gives the label identifying the service in the Transport Stream where the linked carousel module belongs and describes the same service_id as the program_number in the corresponding program map section.

event_id: This 16-bit field describes the identifier number of the event where the linked carousel module belongs.

component_tag: This 8-bit field describes the label identifying the component stream transmitting the linked carousel module.

moduleId: This 16-bit field describes the identifier number of the linked carousel module.

Table 6-55 Description of selector area (link_destination_type: 0x04)

Syntax (link_destination_type:0x04)	No. of bits	Identifier
link_content_info(){		
original_network_id	16	uimsbf
transport_stream_id	16	uimsbf
service_id	16	uimsbf
content_id	32	uimsbf
}		

original_network_id: This 16-bit field gives the label identifying the network_id of the originating delivery system where the linked content belongs.

transport_stream_id: This 16-bit field gives the label identifying the Transport Stream where the linked contents belong.

service_id: This 16-bit field gives the label identifying the service in the Transport Stream where the linked content belongs and describes the same service_id as the program_number in the corresponding program map section.

content_id: This 32-bit field describes identifier number to identify linked contents in the service uniformly.

Table 6-56 Description of selector area (link_destination_type: 0x05)

Syntax (link_destination_type:0x05)	No. of bits	Identifier
link_content_module_info(){		
original_network_id	16	uimsbf
transport_stream_id	16	uimsbf
service_id	16	uimsbf
content_id	32	uimsbf
component_tag	8	uimsbf
moduleId	16	uimsbf
}		

original_network_id: This 16-bit field describes the label identifying the network_id of the originating delivery system where the linked contents module belongs.

transport_stream_id: This 16-bit field describes the label identifying the Transport Stream where the linked contents module belongs.

service_id: This 16-bit field gives the label identifying the service in the Transport Stream where the linked contents module belongs and describes the same service_id as the program_number in the corresponding program map section.

content_id: This 32-bit field describes identifier number to identify content where the linked module belongs in the service uniformly.

component_tag: This 8-bit field describes the label identifying the component stream transmitting the linked carousel module.

moduleId: This 16-bit field describes the identifier number of the linked module.

Table 6-57 Description of selector area (link_destination_type: 0x06)

Syntax (link_destination_type:0x06)	No. of bits	Identifier
link_ert_node_info(){		
information_provider_id	16	uimsbf
event_relation_id	16	uimsbf
node_id	16	uimsbf
}		

information_provider_id: This 16-bit field designates information provider identifier of event relation sub_table to which the linked node belongs.
event_relation_id: This 16-bit field designates event relation identifier of event relation sub_table to which the linked destination belongs.

node_id: This 16-bit field designates node identifier of linked destination node.

Table 6-58 Selector area description (link_destination_type: 0x07)

Syntax (link_destination_type:0x07)	No. of bits	Identifier
link_stored_content_info(){		
for(i=0; i<N; i++){		
uri_char	8	uimsbf
}		
}		

uri_char: The series of fields describes URI of the contents of the accumulated data service. Describing method of URI is specified in ARIB STD-B24 Part 2 Section 9.

6.2.30 Video decode control descriptor

The video decode control descriptor (see table 6-59) is used to control video decoding to receive still picture composed of MPEG-I pictures transmitted at low transmission speed and to get smooth displaying at video splice point where video format is changed.

Table 6-59 Video decode control descriptor

Syntax	No. of bits	Identifier
video_decode_control_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
still_picture_flag	1	bslbf
sequence_end_code_flag	1	bslbf
video_encode_format	4	bslbf
transfer_characteristics	2	bslbf
}		

Semantics for the video decode control descriptor:

still_picture_flag: This is a 1-bit field and when it is "1" it means that this component is still picture (MPEG-I picture). When it is "0", it means that this component is moving picture.

sequence_end_code_flag: This is a 1-bit field and it indicates whether or not this video component has a sequence end code (in case of MPEG-2 Video, an end-of-sequence NAL unit in case of MPEG-4 AVC and HEVC, the same hereinafter) at the end of the sequence that is defined by the video format. When it is "1", it means that the video stream has a sequence end code at the end of the sequence and when it is "0", it means that the video stream does not have a sequence end code.

video_encode_format: This is a 4-bit field, and shows the encode format of the component in accordance with table 6-60.

Table 6-60 Video encode format

Video encode format	Description
0000	1080/P
0001	1080/I
0010	720/P
0011	480/P
0100	480/I
0101	240/P
0110	120/P
0111	2160/60/P
1000	180/p
1001	2160/120/P
1010	4320/60/P
1011	4320/120/P
1100 – 1111	For extension of video encode format

transfer_characteristics (Transfer characteristics): This is an 8-bit field which identifies transfer characteristics of video signals shown in table 6-60-1.

Table 6-60-1 Transfer characteristics

Value of transfer characteristics	Semantics
00	transfer_characteristics = 1, 11 or 14（Rec. ITU-R BT.709-5, BT.2020 or IEC 61966-2-4） in VUI
01	transfer_characteristics = 16（Rec. ITU-R BT.2100 PQ） in VUI
10	transfer_characteristics = 18（Rec. ITU-R BT.2100 HLG） in VUI
11	Undefined

6.2.31 Terrestrial delivery system descriptor

The terrestrial delivery system descriptor indicates the physical condition of terrestrial transmission path. See table 6-61.

Table 6-61 Terrestrial delivery system descriptor

Syntax	No. of bits	Identifier
terrestrial_delivery_system_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
area_code	12	bslbf
guard_interval	2	bslbf
transmission_mode	2	bslbf
for(i=0; i<N; i++){		
frequency	16	uimsbf
}		
}		

Semantics for the terrestrial delivery system descriptor:

area_code: This is a 12-bit field and corresponds to area code specified in clause 138-3 of Radio Station Operation Rule. The allocation of area code is specified in accordance with the Notation No. 405 of the Ministry of Posts and Telecommunications in 1985 is used (see Annex D).

guard_interval: This is a 2-bit field and indicates guard interval in accordance with table 6-62.

Table 6-62 Guard interval

Guard interval	Description
00	1/32
01	1/16
10	1/8
11	1/4

transmission_mode: This is a 2-bit field and indicates mode information in accordance with table 6-63.

Table 6-63 Mode information

Mode information	Description
00	Mode 1
01	Mode 2
10	Mode 3
11	Undefined

frequency: This 16-bit field indicates center frequency. Frequency unit shall be 1/7MHz, which is the same as tuning step of digital terrestrial broadcasting system. In case of MFN, list multiple frequencies that are used.

6.2.32 Partial reception descriptor

The partial reception descriptor describes service_id transmitted by the partial reception hierarchy of the terrestrial transmission path. See table 6-64.

Table 6-64 Partial reception descriptor

Syntax	No. of bits	Identifier
partial_reception_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0; i<N; i++){		
service_id	16	uimsbf
}		
}		

Semantics for the partial reception descriptor:

service_id: This 16-bit field indicates service_id of the information service transmitted in the partial reception hierarchy. The service_id is the same as the program_number in the corresponding program map section.

6.2.33 Series descriptor

The series descriptor is used to identify series event. See table 6-65.

Table 6-65 Series descriptor

Syntax	No. of bits	Identifier
series_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
series_id	16	uimsbf
repeat_label	4	uimsbf
program_pattern	3	uimsbf
expire_date_valid_flag	1	uimsbf
expire_date	16	uimsbf
episode_number	12	uimsbf
last_episode_number	12	uimsbf
for(i=0; i<N; i++){		
series_name_char	8	uimsbf
}		
}		

Semantics for the series descriptor:

series_id: This is a 16-bit field and identifies series uniquely.

repeat_label: This 4-bit field gives the label identifying program when the broadcasting duration of the series and that of repeating the series of broadcasting. Original series broadcasting is given with "0x0".

program_pattern: This 3-bit field indicates program pattern of the series content according to table 6-66. This may show when the event belonging the series appears next time.

Table 6-66 Program pattern

Program pattern	Description
0x0	Nonscheduled (other than defined as 0x1 to 0x7)
0x1	Regular program (every day, every day except week-end, only weekends, etc.), programmed several days a week
0x2	Programmed about once a week
0x3	Programmed about once a month
0x4	Programmed several events in a day
0x5	Division of long hour program
0x6	Program for regular or irregular accumulation
0x7	Undefined

expire_date_valid_flag: This 1-bit flag indicates that the following expire_date value is valid. When the value of the scheduled series end date is valid, set this value to "1".

expire_date: This 16-bit field indicates the date of the effective limit of the series in lower 16 bits of MJD. Even when the last event could not be recognized for some reason, the IRD recognizes that the series is ended when the date is passed.

episode_number: This 12-bit field indicates the episode number in the series in the event which this descriptor indicates. It can be indicated from No. 1 to No. 4095. When the episode number exceeds this value, define the series separately. When the event number cannot be defined due to a series event, set to "0x000".

last_episode_number: This 12-bit field indicates the total number of the corresponding series. It can be indicated from No. 1 to No. 4095. When the episode number exceeds this value, define the series separately. When the last time is not yet decided, set to "0x000".

series_name_char: In this character code field, series name is transmitted. For coding character information, see Annex A.

6.2.34 Event group descriptor

When there is a relation between multiple events, the event group descriptor is given to indicate that those events are in a group. See table 6-67.

Table 6-67 Event group descriptor

Syntax	No. of bits	Identifier
event_group_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
group_type	4	uimsbf
event_count	4	uimsbf
for(i=0; i< event_count; i++){		
service_id	16	uimsbf
event_id	16	uimsbf
}		
if(group_type == 4 || group_type ==5){		
for(i=0 ;i< N ; i++){		
original_network_id	16	uimsbf
transport_stream_id	16	uimsbf
service_id	16	uimsbf
event_id	16	uimsbf
}		
}		
else{		
for(i=0; i< N; i++){		
private_data_byte	8	uimsbf
}		
}		
}		

Semantics for the event group descriptor:

group_type: This is a 4-bit field and indicates group type of the event in accordance with table 6-68.

Table 6-68 Group type

Group type	Description
0x1	Event common
0x2	Event relay
0x3	Event movement
0x4	Event relay to other networks
0x5	Event movement from other networks
0x0, 0x06 – 0xF	Undefined

event_count: This is a 4-bit field and indicates the following event_id loop number.

service_id: This is a 16-bit field and indicates the service_id of the related information service. The service_id is the same as the program_number in the corresponding program map section.

event_id: This is a 16-bit field and indicates the event_id of the related event.

original_network_id: This is a 16-bit field and indicates the original_network_id of the related event transmitted at the time of event relay or event move across networks.

transport_stream_id: This is a 16-bit field and indicates the transport_stream_id of the related event transmitted at the time of event relay or event move across networks.

6.2.35 SI parameter descriptor

The SI parameter descriptor is used to indicate the SI parameter. See table 6-69.

Table 6-69 SI parameter descriptor

Syntax	No. of bits	Identifier
SI_parameter_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
parameter_version	8	uimsbf
update_time	16	uimsbf
for(i=0; i<N; i++){		
table_id	8	uimsbf
table_description_length	8	uimsbf
for(j=0; j<N; j++){		
table_description_byte	8	uimsbf
}		
}		
}		

Semantics for the SI parameter descriptor:

parameter_version: This is an 8-bit field and indicates SI parameter version. It denotes value incremented by 1 when a parameter is updated.

update_time: This is a 16-bit field which is denoted in lower 16 bits of MJD when the denoted parameter becomes valid.

table_id: This 8-bit field indicates the table_id described in the following table_description_byte field.

table_description_length: This 8-bit field indicates the byte length of the following table_description_byte.

table_description_byte: This is an 8-bit field. A series of table description area describes parameter in each table specified in the operational guidelines of service providers.

6.2.36 Broadcaster name descriptor

The broadcaster name descriptor describes the name of the broadcaster. See table 6-70.

Table 6-70 Broadcaster name descriptor

Syntax	No. of bits	Identifier
broadcaster_name_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0; i<N; i++){		
char	8	uimsbf
}		
}		

Semantics for the broadcaster name descriptor:

char: This is an 8-bit field. A string of character information field indicates the broadcaster name. For character information coding, see Annex A.

6.2.37 Component group descriptor

The component group descriptor defines and identifies component grouping in the event. See table 6-71.

Table 6-71 Component group descriptor

Syntax	No. of bits	Identifier
component_group_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
component_group_type	3	uimsbf
total_bit_rate_flag	1	uimsbf
num_of_group	4	uimsbf
for(i=0; i< num_of_group; i++){		
component_group_id	4	uimsbf
num_of_CA_unit	4	uimsbf
for(i=0; i< num_of_CA_unit; i++){		
CA_unit_id	4	uimsbf
num_of_component	4	uimsbf
for(i=0; i< num_of_component; i++){		
component_tag	8	uimsbf
}		
}		
if(total_bit_rate_flag==1){		
total_bit_rate	8	uimsbf
}		
text_length	8	uimsbf
for(i=0; i< text_length; i++){		
text_char	8	uimsbf
}		
}		
}		

Semantics for the component group descriptor:

component_group_type: This is a 3-bit field and indicates group type of the component in accordance with table 6-72.

Table 6-72 Component group type

Component group type	Description
000	Multi-view TV service
001 – 111	Undefined

total_bit_rate_flag: This is a 1-bit flag and indicates the description status of the total bit rate in the component group in the event. When this bit is "0", the total bit rate field in the component group does not exist in the corresponding descriptor. When this bit is "1", the total bit rate field in the component group exists in the corresponding descriptor.

num_of_group: This is a 4-bit field indicating number of component groups in the event.

component_group_id: This is a 4-bit field and describes the component group identifier in accordance with table 6-73.

Table 6-73 Component group identifier

Component group identifier	Description
0x0	Main group
0x1 – 0xF	Sub group

num_of_CA_unit: This is a 4-bit field and indicates CA/non-CA unit within the component group.

CA_unit_id: This is a 4-bit field and describes the CA_unit_id, to which the component belongs in accordance with table 6-74.

Table 6-74 CA_unit_id

CA_unit_id	Description
0x0	Non-CA unit group
0x1	CA unit group including default ES group
0x2 – 0xF	CA unit group other than above

num_of_component: This is a 4-bit field indicating number of components which belong to the corresponding component group and the CA CA/non-CA unit indicated in the CA_unit_id immediately before.

component_tag: This is an 8-bit field and indicates the component tag value belonging to the component group.

total_bit_rate: This is an 8-bit field and describes the total bit rate of the component in the component group by rounding up the TS packet transmission rate in each 1/4Mbps.

text_length: This is an 8-bit field and indicates the byte length of the following component group description.

text_char: This is an 8-bit field. A series of character information field describes explanation of component group. For character information coding, see Annex A.

6.2.38 SI prime_ts descriptor

The SI prime_ts descriptor is used to indicate the identifier information of the SI prime_ts (Transport Stream having special transmission format for SI) and its transmission parameter. See table 6-75.

Table 6-75 SI prime_ts descriptor

Syntax	No. of bits	Identifier
SI_prime_ts_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
parameter_version	8	uimsbf
update_time	16	uimsbf
SI_prime_ts_network_id	16	uimsbf
SI_prime_transport_stream_id	16	uimsbf
for(i=0; i<N; i++){		
table_id	8	uimsbf
table_description_length	8	uimsbf
for(j=0; j<N; j++){		
table_description_byte	8	uimsbf
}		
}		
}		

Semantics for the SI prime_ts descriptor

parameter_version: This is an 8-bit field and indicates the version of SI parameter. The version_number shall be incremented by 1 when the parameter is updated.

update_time: This is a 16-bit field which is denoted in the lower 16 bits of MJD when the denoted parameter becomes valid.

SI_prime_ts_network_id: This 16-bit field indicates the SI_prime_ts_network_id.

SI_prime_transport_stream_id: This 16-bit field indicates the SI_prime_transport_stream_id.

table_id: This 8-bit field indicates the table_id described in the following table_description_byte field.

table_description_length: This 8-bit field indicates the byte length of the following table_description_byte.

table_description_byte: This is an 8-bit field. A series of table description area describes parameter in each table specified in the operational guidelines of service providers.

6.2.39 Board information descriptor

The board information descriptor indicates title and content of the board information in text format. See table 6-76.

Table 6-76 Board information descriptor

Syntax	No. of bits	Identifier
board_information_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
title_length	8	uimsbf
for(i=0;i<title_length;i++){		
title_char	8	uimsbf
}		
text_length	8	uimsbf
for(i=0;i<text_length;i++){		
text_char	8	uimsbf
}		
}		

Semantics for the board information descriptor:

title_length: This 8-bit field indicates the byte length of the following title.

title_char: This is an 8-bit field. A series of character information indicates the board information title. For character information coding see Annex A.

text_length: This 8-bit field indicates the byte length of the following content description.

text_char: This is an 8-bit field. A series of character information field describes the content of board information. For character information coding see Annex A.

6.2.40 LDT linkage descriptor

The LDT linkage descriptor is used to describe linkage of the information collected in LDT. See table 6-77.

Table 6-77 LDT linkage descriptor

Syntax	No. of bits	Identifier
LDT_linkage_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
original_service_id	16	uimsbf
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
for(i=0;i<N;i++){		
description_id	16	uimsbf
reserved_future_use	4	uimsbf
description_type	4	uimsbf
user_defined	8	bslbf
}		
}		

Semantics for the LDT linkage descriptor:

original_service_id: This 16-bit field indicates the original_service_id of the linked LDT sub_table.

transport_stream_id: This 16_bit field indicates the ts_id of the LDT sub_table which the linked LDT sub_table is included.

original_network_id: This 16-bit field indicates the network_id of the originating delivery system in which the linked LDT sub_table is included.

description_id: This 16-bit field indicates the id_number of the linked descriptor.

description_type: This 8-bit field indicates the linked description type in accordance with table 6-78.

Table 6-78 Description type

Value	Semantics
0x0	Undefined
0x1	Described with short_event_descriptor
0x2	Described with extended_event_descriptor (Independent type without describing item_description is used)
0x3	Described with extended_event_descriptor
0x4 – 0xE	Reserved for future use
0xF	Others (Including not specified descriptor and mixed)

user_defined: The service provider can define this 8-bit field independently.

6.2.41 Connected transmission descriptor

The connected transmission descriptor indicates physical condition in connected transmission in terrestrial transmission path. See table 6-79.

Table 6-79 Connected transmission descriptor

Syntax	No. of bits	Identifier
connected_transmission_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
connected_transmission_group_id	16	uimsbf
segment _type	2	bslbf
modulation_type_A	2	bslbf
modulation_type_B	2	bslbf
modulation_type_C	2	bslbf
for(i= 0;i<N;i++){		
addtional_connected_transmission_info	8	uimsbf
}		
}		

Semantics for the connected transmission descriptor:

connected_transmission_group_id: This 16-bit field gives the label identifying the connected transmission group.

segment_type: This is a 2-bit field and indicates segment type in accordance with table 6-80. The number of layers and the layer of each segment type are shown in the table.

Table 6-80 Segment type

Segment type	Description	Number of layers, and layer
00	1 segment	1 (A layer)
01	3 segments	2 (A layer, B layer)
10	13 segments	1 (A layer) 2 (A layer, B layer) or 3 (A layer, B layer, C layer)
11	Refer to TMCC signal	

modulation_type_A: This is a 2-bit field which indicates modulation_type_A in accordance with table 6-81.

modulation_type_B: This is a 2-bit field which indicates modulation_type_B in accordance with table 6-81.

modulation_type_C: This is a 2-bit field which indicates modulation_type_C in accordance with table 6-81.

Table 6-81 Modulation type

Modulation type	Description
00	Differential modulation
01	Synchronous modulation
10	Reserved for future use
11	Refer to TMCC signal

additional_connected_transmission_info: This is an 8-bit field and used to store the additional information specified in the operational guidelines of service providers.

6.2.42 TS information descriptor

The TS information descriptor specifies the remote control key identifier assigned to the applicable TS and indicates the relationship between the service identifier and the transmission layer during hierarchical transmission (see table 6-82).

Table 6-82 TS information descriptor

Syntax	No. of bits	Identifier
ts_information_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
remote_control_key_id	8	uimsbf
length_of_ts_name	6	uimsbf
transmission_type_count	2	uimsbf
for(i = 0;i< length_of_ts_name;i++){		
ts_name_char	8	uimsbf
}		
for(j = 0;j< transmission_type_count;j++){		
transmission_type_info	8	bslbf
num_of_service	8	uimsbf
for(k = 0;k< num_of_service;k++){		
service_id	16	uimsbf
}		
}		
for(l = 0;l< N;l++){		
reserved_future_use	8	bslbf
}		
}		

Semantics for the TS information descriptor:

remote_control_key_id: This 8-bit field indicates the recommended remote control key number to which the applicable TS shall be assigned.

length_of_ts_name: This 6-bit field indicates the byte length of TS name description.

transmission_type_count: This 2-bit field indicates the number of loops for the subsequent information on the number of transmission types.

ts_name_char: This is a 8-bit field. A series of TS name description fields describes the applicable TS name. See Annex A for character information coding.

transmission_type_info: This 8-bit field, which is used for discriminating hierarchical layers etc., is defined by the operational guidelines of service providers.

num_of_service: This 8-bit field indicates the number of loops for the subsequent service identifier.

service_id: This 16-bit field indicates the service identifier transmitted at each hierarchy of transmission type.

6.2.43 Extended broadcaster descriptor

The extended broadcaster descriptor specifies the extended broadcaster identification information such as terrestrial broadcaster identifier and defines the relationships with other extended broadcasters and broadcasters of other networks (see table 6-83).

Table 6-83 Extended broadcaster descriptor

Syntax	No. of bits	Identifier
extended_broadcaster_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
broadcaster_type	4	uimsbf
reserved_future_use	4	bslbf
if(broadcaster_type == 0x1){		
terrestrial_broadcaster_id	16	uimsbf
number_of_affiliation_id_loop	4	uimsbf
number_of_broadcaster_id_loop	4	uimsbf
for(i=0; i<N1; i++){		
affiliation_id	8	uimsbf
}		
for(j=0; j<N2; j++){		
original_network_id	16	uimsbf
broadcaster_id	8	uimsbf
}		
for(k=0; k<N3; k++){		
private_data_byte	8	bslbf
}		
}		
Else if(broadcaster_type == 0x2){		
terrestrial_sound_broadcaster_id	16	uimsbf
number_of_sound_broadcast_affiliation_id_loop	4	uimsbf
number_of_broadcaster_id_loop	4	uimsbf
for(i=0; i<N1; i++){		
sound_broadcast_affiliation_id	8	uimsbf
}		
for(j=0; j<N2; j++){		
original_network_id	16	uimsbf
broadcaster_id	8	uimsbf
}		
for(k=0; k<N3; k++){		
private_data_byte	8	bslbf
}		
}		
else{		
for(i=0;i<N;i++){		
reserved_future_use	8	bslbf
}		
}		
}		

Semantics for the extended broadcaster descriptor:

broadcaster_type(broadcaster type): This is a 4-bit field whose coding is specified in table 6-84.

Table 6-84 broadcaster type

value	type
0x1	Digital terrestrial television broadcast
0x2	Digital terrestrial sound broadcast
Except above	Not defined

terrestrial_broadcaster_id: This 16-bit filed identifies the terrestrial broadcaster described in this field.

number_of_affiliation_id_loop: This 4-bit field indicates the number of loops for the subsequent affiliation identifier.

number_of_broadcaster_id_loop: This 4-bit field indicates the number of loops for the subsequent broadcaster identifier.

affiliation_id: This 8-bit field is used for identifying the affiliation of the applicable terrestrial broadcaster identifier.

original_network_id: This 16-bit field works as a label to specify the network identifier of the original distribution system.

broadcaster_id: This 8-bit field identifies the broadcaster in the original network.

terrestrial_sound_broadcaster_id: This 16-bit field identifies the terrestrial sound broadcaster described in this field.

number_of_sound_broadcast_affiliation_id_loop: This 4-bit field indicates the number of loops for the subsequent sound broadcasting affiliation identifier. .

number_of_sound_broadcaster_id_loop: This 4-bit field indicates the number of loops for the subsequent terrestrial sound broadcaster identifier.

sound_broadcast_affiliation_id: This 8-bit field is used for identifying the sound broadcasting affiliation of the applicable terrestrial sound broadcaster identifier.

6.2.44 Logo transmission descriptor

The logo transmission descriptor is used for describing such information as the character string for simple logo and pointing to CDT-format logo data (see table 6-85).

Table 6-85 Logo transmission descriptor

Syntax	No. of bits	Identifier
logo_transmission_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
logo_transmission_type	8	uimsbf
if(logo_transmission_type == 0x01){		
reserved_future_use	7	bslbf
logo_id	9	uimsbf
reserved_future_use	4	bslbf
logo_version	12	uimsbf
download_data_id	16	uimsbf
}		
else if(logo_transmission_type == 0x02){		
reserved_future_use	7	bslbf
logo_id	9	uimsbf
}		
else if(logo_transmission_type == 0x03){		
for(i=0;i<N;i++){		
logo_char	8	uimsbf
}		
}		
else{		
for(j=0;j<M;j++){		
reserved_future_use	8	bslbf
}		
}		
}		

Semantics for the logo transmission descriptor:

logo_transmission_type : This 8-bit field indicates the logo transmission scheme shown in table 6-86 (see ARIB STD-B21).

Table 6-86 Logo transmission scheme

logo_transmission_type value	explanation
0x01	CDT transmission scheme 1 : when referring to CDT directly with download data identification
0x02	CDT transmission scheme 2 : when referring to CDT using logo identification indirectly with download data identification
0x03	Simple logo system
Except above	Reserved for future use

logo_id: This 9-bit data denotes the ID value of the logo data defined in the applicable service (see ARIB STD-B21).

download_data_id : This 16-bit field identifies data to be downloaded. Its value should be the same as the table_id_extension value of the CDT where logo data is located (see ARIB STD-B21).

logo_version: This 12-bit field denotes the version number of the applicable logo_id (see ARIB STD-B21).

logo_char: This 8-bit field describes the 8-unit code character string for simple logo.

6.2.45 Content availability descriptor

The content availability descriptor (see table 6-87), which describes information to control record and output, is used in combination with the digital copy control descriptor by the broadcasting service provider (copyright holder) to control the record and output of programs.

Table 6-87 Content availability descriptor

Syntax	No. of bits	Identifier
content_availability_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
reserved_future_use	1	bslbf
copy_restriction_mode	1	bslbf
image_constraint_token	1	bslbf
retention_mode	1	bslbf
retention_state	3	bslbf
encryption_mode	1	bslbf
for(i=0;i<N;i++){		
reserved_future_use	8	uimsbf
}		
}		

Semantics for the content availability descriptor:

copy_restriction_mode: This 1-bit field indicates the restriction mode of the number of copies allowed. The usage is specified by the operational guidelines of service providers.

image_constraint_token: This 1-bit field indicates whether the image quality of video signal output is constrained. The resolution of video signal output shall be constrained when this field is "0". When the field is "1", the constraint is not required.

retention_mode: When this 1-bit field is "0", temporal accumulation is possible even if copy is prohibited by the digital_recording_control_data of the digital copy control descriptor. When this field is "1", temporal accumulation is not possible.

retention_state: This 3-bit field, whose coding is shown in table 6-88, indicates the allowable time of temporal accumulation after the reception of contents.

Table 6-88 Allowable time of temporal accumulation

Allowable time of temporal accumulation	description
111	1 hour and half
110	3 hours
101	6 hours
100	12 hours
011	1 day
010	2 days
001	1 week
000	No limit

encryption_mode (output protection bit): This 1-bit field indicates whether the output of high-speed digital interface is protected. When this field is "0", the output of high-speed digital interface shall be protected. When the field is "1", the protection is not required.

6.2.46 Carousel compatible composite descriptor

The carousel compatible composite descriptor uses descriptors defined in the data carousel transmission scheme (Chapter 6 of ARIB STD-B24 Part 3) as subdescriptors, and describes accumulation control of stream-type contents etc. by applying the descriptive functions of the subdescriptors (see table 6-89).

Table 6-89 Carousel compatible composite descriptor

Syntax	No. of bits	Identifier
carousel_compatible_composite_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0;i<N;i++){		
sub_descriptor()		
}		
}		

Semantics for the carousel compatible composite descriptor:

sub_descriptor( ): A subdescriptor is placed in this area. The descriptors in the module information area and the private area defined in the data carousel transmission scheme (Chapter 6 of ARIB STD-B24 Part 3) are used as subdescriptors, and the descriptive function of each descriptor is inherited. Refer to Annex K for the functions of the subdescriptors.

6.2.47 AVC video descriptor

The AVC video descriptor (see table 6-90) is used for describing the basic coding parameters of the AVC video stream in ITU-T Recommendation H.264 and ISO/IEC 14496-10. When this descriptor is not described in the PMT, the AVC stream should not contain AVC still images or AVC 24-hour pictures. For more information, see ITU-T Recommendation H.222.0 and ISO/IEC 13818-1.

Table 6-90 AVC video descriptor

Syntax	No. of bits	Identifier
AVC_video_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
profile_idc	8	uimsbf
constraint_set0_flag	1	bslbf
constraint_set1_flag	1	bslbf
constraint_set2_flag	1	bslbf
constraint_set3_flag	1	bslbf
constraint_set4_flag	1	bslbf
constraint_set5_flag	1	bslbf
AVC_compatible_flags	2	bslbf
level_idc	8	uimsbf
AVC_still_present	1	bslbf
AVC_24_hour_picture_flag	1	bslbf
Frame_Packing_SEI_not_present_flag	1	bslbf
reserved	5	bslbf
}		

Semantics for the AVC video descriptor:

profile_idc: Shows the profile of the AVC video stream. See Section 7.4.2.1 of ITU-T Recommendation H.264 and ISO/IEC 14496-10.

constraint_set0_flag: See Section 7.4.2.1 of ITU-T Recommendation H.264 and ISO/IEC 14496-10.

constraint_set1_flag: See Section 7.4.2.1 of ITU-T Recommendation H.264 and ISO/IEC 14496-10.

constraint_set2_flag: See Section 7.4.2.1 of ITU-T Recommendation H.264 and ISO/IEC 14496-10.

constraint_set3_flag: See Section 7.4.2.1 of ITU-T Recommendation H.264 and ISO/IEC 14496-10.

constraint_set4_flag: See Section 7.4.2.1 of ITU-T Recommendation H.264 and ISO/IEC 14496-10.

constraint_set5_flag: See Section 7.4.2.1 of ITU-T Recommendation H.264 and ISO/IEC 14496-10.AVC_compatible_flags: The same value as reserved_zero_2bits in the sequence parameter set specified in ITU-T Recommendation H.264 and ISO/IEC 14496-10.

level_idc: Shows the level of the AVC video stream. See Section 7.4.2.1 of ITU-T Recommendation H.264 and ISO/IEC 14496-10.

AVC_still_present: When this field is "1", the AVC video stream contains AVC still images. When this field is "0", the AVC video stream should not contain AVC still images.

AVC_24_hour_picture_flag: When this field is "1", the AVC video stream contains 24-hour pictures, which are AVC access units having presentation times exceeding 24 hours. When this field is "0", the AVC video stream should not contain AVC 24-hour pictures.

Frame_Packing_SEI_not_present_flag: When this field is "0", an AVC video stream shall contain a frame packing arrangement SEI message or stereo a video information SEI message. When this field is set to ‘1’ by AVC video descriptor, it is not indicated whether these SEI messages are used, or not.

6.2.48 AVC timing and HRD descriptor

The AVC timing and HRD descriptor (see table 6-91) is used to describe the video stream time information and the hypothetical reference decoder (HRD) information of ITU-T Recommendation H.264 and ISO/IEC 14496-10. When the AVC video stream does not transmit the video usability information (VUI) parameter, this descriptor shall be described in the PMT. For more information, see ITU-T Recommendation H.222.0 and ISO/IEC 13818-1[(24)].

Table 6-91 AVC timing and HRD descriptor

Syntax	No. of bits	Identifier
AVC_timing_and_HRD_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
hrd_management_valid_flag	1	bslbf
reserved	6	bslbf
picture_and_timing_info_present	1	bslbf
if(picture_and_timing_info_present = = 1){		
90kHz_flag	1	bslbf
reserved	7	bslbf
if(90kHz_flag = = 0){		
N	32	uimsbf
K	32	uimsbf
}		
num_units_in_tick	32	uimsbf
}		
fixed_frame_rate_flag	1	bslbf
temporal_poc_flag	1	bslbf
picture_to_display_conversion_flag	1	bslbf
reserved	5	bslbf
}		

Semantics for the AVC timing and HRD descriptor:

hrd_management_valid_flag: When this 1-bit field is "1", the buffering period SEI defined in Annex C of ITU-T Recommendation H.264 and ISO/IEC 14496-10 needs to be contained in the AVC video stream, and bytes shall be transferred from MBn to EBn according to the schedule of transfer to the coded picture buffer (CPB) in the network abstraction layer hypothetical reference decoder (NAL HRD). When this field is "0", the leak method defined in Section 2.14.3.1 of ITU-T Recommendation H.222.0 and ISO/IEC 13818-1 is used for transfer from MBn to EBn

picture_and_timing_info_present: When this field is "1", the descriptor contains 90kHz_flag and parameters for precise mapping to the system clock.

90kHz_flag: When this field is "1", the AVC time base is 90 kHz. The AVC time base period is specified by AVC's time_scale defined in Annex E of ITU-T Recommendation H.264 and ISO/IEC 14496-10.

N, K: Parameters to describe the relationship between AVC's time_scale and system_clock_reference with the following equation (K is equal to or greater than N):

$$time\_scale = \frac{(N \times system\_clock\_frequency)}{K}$$

num_units_in_tick: See Annex E of ITU-T Recommendation H.264|ISO/IEC 14496-10.

fixed_frame_rate_flag: See Annex E of ITU-T Recommendation H.264|ISO/IEC 14496-10. When this flag is "1", the coded frame rate is constant within the AVC video elementary stream. When this flag is "0", there is no information on the frame rate of the AVC video stream in the descriptor.

temporal_poc_flag: When this field is "1" and fixed_frame_rate_flag is "1", the AVC video stream shall transmit the picture order count (POC) information. See Annex E of ITU-T Recommendation H.264|ISO/IEC 14496-10. When this field is "0", information on the relationship between the POC information of the AVC video stream and time is not transmitted.

picture_to_display_conversion_flag: When this field is "1", the AVC video stream transmits information on displaying coded pictures. When this field is "0", pic_struct_present_flag, which is a VUI parameter of the AVC video stream, shall be set to "0".

6.2.49 Service group descriptor

The service group descriptor (see table 6-92) is used to indicate that multiple services are grouped together when they are related to each other.

Table 6-92 Service group descriptor

Syntax	No. of bits	Identifier
service_group_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
service_group_type	4	uimsbf
reserved_future_use	4	uimsbf
if(service_group_type==1){		
for(i=0; i<N; i++){		
primary_service_id	16	uimsbf
secondary_service_id	16	uimsbf
}		
}		
else{		
for(i=0; i<N; i++){		
private_data_byte	8	uimsbf
}		
}		
}		

Semantics for service group descriptor:

service_group_type (service group type): This 4-bit field indicates, according to table 6-93, the type of service that constitute the group.

Table 6-93 Service group type

Service group type	Description
0x1	Server-type simultaneous service
0x0, 0x2 – 0xF	Undefined

primary_service_id (primary service identifier): This 16-bit field indicates the service identifier of the primary service for grouping.

secondary_service_id (secondary service identifier): This 16-bit field indicates the service identifier of the secondary service for grouping.

6.2.50 MPEG-4 audio descriptor

The MPEG-4 audio descriptor (see table 6-94) is used for describing the basic coding parameters of the audio stream in ISO/IEC 14496-3 (MPEG-4 audio).

Table 6-94 MPEG-4 audio descriptor

Syntax	No. of bits	Identifier
MPEG-4_audio_descriptor (){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
MPEG-4_audio_profile_and_level	8	uimsbf
}		

Semantics for the MPEG-4 audio descriptor:

MPEG-4_audio_profile_and_level: This 8-bit field indicates profile and level of an MPEG-4 audio stream. The allocation of this field value is giving in Table 2-71 of ITU-T H.222.0|ISO/IEC 13818-1 (2006)/Amd.1 (01/2007). The value of 0x0F indicates that profile and level are not specified. The value of 0xFF indicates that profile and level are not specified by this field and are described in an MPEG-4 audio extension descriptor of 6.2.51.

6.2.51 MPEG-4 audio extension descriptor

The MPEG-4 audio extension descriptor (see table 6-95) is used for describing profile, level and individual coding parameters of the audio stream.

Table 6-95 MPEG-4 audio extension descriptor

Syntax	No. of bits	Identifier
MPEG-4_audio_extension_ descriptor (){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
ASC_flag	1	bslbf
reserved	3	bslbf
num_of_loops	4	uimsbf
for(i=0 ;i< N ; i++){		
audioProfileLevelIndication	8	uimsbf
}		
if (ASC_flag == ‘1’){		
ASC_size	8	
audioSpecificConfig()		uimsbf
}		
}		

Semantics for the MPEG-4 audio extension descriptor:

ASC_flag: In case that ASC_size is included in this descriptor, this field is set to ‘1’.

num_of_loops: This 4-bit field indicates the number of succeeding audioProfileLevelIndication.

audioProfileLevelIndication: This field indicates the profile and level of MPEG-4 audio stream in accordance with 1.5.2.4 of ISO/IEC 14496-3. As it is possible that MPEG-4 audio profile is in accordance with multiple profiles and levels, this descriptor can describe 15 audio profiles and levels at most.

ASC_size: This 8-bit field indicates the number of bytes of succeeding audioSpecificConfig().

audioSpecificConfig(): This indicates individual parameters of an MPEG-4 audio steam. Refer to 1.6.2.1 of ISO/IEC 14496-3.

6.2.52 Registration descriptor

The registration descriptor (see table 6-96) is used to uniquely identify the private data which does not specified in ISO/IEC 13818-1[(24)].

Table 6-96 Registration descriptor

Syntax	No. of bits	Identifier
registration_descriptor (){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
format_identifier	32	uimsbf
for(i=0 ;i< N ; i++){		
additional_identification_info	8	bslbf
}		
}		

Semantics for the registration descriptor:

format identifier: This 32-bit field is format identifier provided from the registration organization (SMPTE) designated by ISO/IEC JTC 1/SC 29.

additional_identification_info: This field is information defined by the user who takes the format identifier, and is not changed after it is defined.

6.2.53 Data broadcast id descriptor

The data broadcast id descriptor (see table 6-97) is used for describing data broadcast identification.

Table 6-97 Data broadcast id descriptor

Syntax	No. of bits	Identifier
data_broadcast_id descriptor (){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
data_broadcast_id	16	uimsbf
for(i=0 ;i< N ; i++){		
id_selector_byte	8	uimsbf
}		
}		

Semantics for the data broadcast id descriptor:

data_broadcast_id: This 16-bit field is used identifying specifications of data broadcast. The field values are shown in table 6-98.

Table 6-98 Allocation of data_broadcast_id

data_broadcast_id	Description
0x0000 – 0x000A	Undefined
0x000B	INT
0x000C – 0xFFFF	Undefined

id_selector_byte: This 8-bit field depends on data_broadcast_id. (see 6.2.53.1)

6.2.53.1 data_broadcast_id and ID selector information

When data_broadcast_id is set to 0x000B, ID selector information has data structure for INT as shown in table 6-99.

Table 6-99 ID selection information for INT

Syntax	No. of bits	Identifier
IP/MAC_notification_info(){		
platform_id_data_length	8	uimsbf
for(i=0;i<N;i++){		
platform_id	24	uimsbf
action_type	8	uimsbf
reserved	2	bslbf
INT_versioning_flag	1	bslbf
INT_version	5	uimsbf
}		
for(i=0;i<N;i++){		
private_data_byte	8	uimsbf
}		
}		

platform_id_data_length: This 8-bit field indicates the total byte length of the succeeding platform identification loop.

platform_id: This 24-bit field identifies platform of IP service. The allocation of this field is specified by standard organization. (see Annex P)

action_type: This 8-bit field indicates the location of IP/MAC stream. The allocation of this field value is shown in table 5-22.

INT_versioning_flag: In case of '1', the change of INT is reflected in version field. In case of '0', the value of version field is invalid.

INT_version: This 5-bit field is valid in case of '1' of INT version flag and is incremented by 1 every change of INT.

private_data_byte: This field is defined individually.

6.2.54 Access control descriptor

The Access control descriptor (see table 6-100) is used to describe describing an access control method, the PID of TS packet to transmit the related information (ECM, EMM) and transmission information.

Table 6-100 Access control descriptor

Syntax	No. of bits	Identifier
access control descriptor (){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
CA_system id	16	uimsbf
transmission_type	3	bslbf
PID	13	uimsbf
for(i=0; i<N; i++){		
private_data_byte	8	uimsbf
}		
}		

Semantics for the access control descriptor:

CA_system_id: This 16-bit field identifies a CA system. The allocation of field value is specified by standardization organization. (see Annex M)

transmission_type: This 3-bit field is used for identifying type of transmission path for related information. The allocation of this field is shown in table 6-101.

Table 6-101 Allocation of transmission type

Transmission type	Description
000 – 110	Undefined
111	Broadcasting path

PID: This field describes PID of TS packet transmitting related information.

6.2.55 Area broadcasting information descriptor

The Area broadcasting information descriptor (see table 6-102) is used for describing the identification, signal format and related information of the stations in an limited area.

Table 6-102 Area broadcasting information descriptor

Syntax	No. of bits	Identifier
area_broadcasting_information_descriptor (){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
num_of_station_point	8	uimsbf
for(i=0; i<N; i++){		
station_id	24	uimsbf
location_code	16	uimsbf
broadcast_signal_format	8	uimsbf
additional_station_info_length	8	uimsbf
additional_station_info		
}		
}		

Semantics for the area broadcasting information descriptor:

num_of_station_point: This 8-bit field indicates the number of succeeding station information loop.

station_id: This 24-bit field is used as label identifying stations. An allocation of this field is shown in table 6-103. The allocation in detail is defined by operational guidelines of service providers.

Table 6-103 Allocation of station identification

Station identification	Description
0x000000 – 0x3FFFFF	Area broadcasting
0x400000 – 0xFFFFFF	Undefined

location_code: This 16-bit field indicates the location of a broadcasting station and uses the upper 5 digits of the Japan local government code.

broadcast_signal_format: This 8-bit field indicates the format of a broadcasting signal. The allocation of this field is shown in table 6-104.

Table 6-104 Allocation of broadcasting signal format

Broadcasting signal type	Description
0x00	Undefined
0x01	Full-seg type area broadcasting*
0x02	One-seg type area broadcasting*
0x03	One-seg type area broadcasting with null
0x04	Undefined
0x05	Full-seg type area broadcasting for multi-media transmission
0x06	One-seg type area broadcasting for multi-media transmission
0x07 - -0xFF	Undefined

*: Includes multimedia transmission

additional_station_info_length: This 8-bit field indicates the length (byte) of the succeeding additional_station_info.

additional_station_info: This field indicates the additional staion information. This is defined by operational guidelines of service providers.

6.2.56 HEVC video descriptor

The HEVC video descriptor (see table 6-105) is used for describing the basic coding parameters of the HEVC video stream in ITU-T Recommendation H.265 and ISO/IEC 23008-2. For the sub-bit-stream or subset of HEVC temporal hierarchy coding, this describes information on the highest sub-layer included in an elementary stream.

The ITU-T Recommendation H.222.0 and ISO/IEC 13818-1(24) is referred to in detail.

Table 6-105 HEVC video descriptor

Syntax	No. of bits	Identifier
HEVC_video descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
profile_space	2	uimsbf
tier_flag	1	bslbf
profile_idc	5	uimsbf
profile_compatibility indication	32	bslbf
progressive_source_flag	1	bslbf
interlaced_source_flag	1	bslbf
non_packed_constraint_flag	1	bslbf
frame_only_constraint_flag	1	bslbf
copied_44bits	44	bslbf
level_idc	8	uimsbf
temporal_layer_subset_flag	1	bslbf
HEVC_still_present_flag	1	bslbf
HEVC_24hr_picture_present_flg	1	bslbf
sub_pic_hrd_params_not_present_flag	1	bslbf
reserved	2	bslbf
HDR_WCG_idc	2	uimsbf
if (temporal_layer_subset_flg == '1'){		
temporal_id_min	3	uimsbf
reserved	5	bslbf
temporal_id_max	3	uimsbf
reserved	5	bslbf
}		
}		

Semantics for the HEVC video descriptor:

profile_space: The value of general_profile_space or sub_layer_profle_space specified by ITU-T Recommendation H.265 and ISO/IEC 23008-2 is set to this field.

tier_flag: The value of general_tier_flag or sub_layer_tier_flag specified by ITU-T Recommendation H.265 and ISO/IEC 23008-2 is set to this field.

profile_idc: The value of general_profile_idc or sub_layer_profile_idc specified by ITU-T Recommendation H.265 and ISO/IEC 23008-2 is set to this field.

profile_compatibility_indication: The value of general_profile_compatibility_flag[i] or sub_layer_profile_compatibility_flag[i] specified by ITU-T Recommendation H.265 and ISO/IEC 23008-2 is set to this field.

progressive_source_flag: The value of general_progressive_source_flag or sub_layer_progressive_source_flag specified by ITU-T Recommendation H.265 and ISO/IEC 23008-2 is set to this field.

interlaced_source_flag: The value of general_interlaced_source_flag or sub_layer_interlaced_source_flag specified by ITU-T Recommendation H.265 and ISO/IEC 23008-2 is set to this field.

non_packed_constraint_flag: The value of general_non_packed_constraint_flag or sub_layer_non_packed_constraint_flag specified by ITU-T Recommendation H.265 and ISO/IEC 23008-2 is set to this field.

frame_only_constraint_flag: The value of general_frame_only_constraint_flag or sub_layer_frame_only_constraint_flag specified by ITU-T Recommendation H.265 and ISO/IEC 23008-2 is set to this field.

copied_44bits: The value of the 44 bits found in the profile_tier_level() syntax element between sub_layer_frame_only_constraint_flag and sub_layer_level_idc specified by ITU-T Recommendation H.265 and ISO/IEC 23008-2 is set to this field.

level_idc: The value of general_level_idc or sub_layer_level_idc specified by ITU-T Recommendation H.265 and ISO/IEC 23008-2 is set to this field.

temporal_layer_subset_flag: The information of temporal layer sub-set is included in this descriptor in case that this field is set to '1'. This field is set to '1' for sub-set or sub-bit stream of temporal hierarchy coding.

HEVC_still_present_flag: A still picture is included in an HEVC stream in case that this field is set to '1'. An HEVC still picture is not included in an HEVC stream in case that this field is set to '0'.

HEVC_24hr_picture_present_flg: An HEVC 24 hour picture is included in an HEVC stream in case that this field is set to '1'. An HEVC 24 hour picture is an HEVC access unit with representing time of more than 24 hours. An HEVC 24 hour picture is not included in an HEVC stream in case that this field is set to '0'.

temporal_id_min: This indicates the minimum value of temporal ID specified by ITU-T Recommendation H.265 and ISO/IEC 23008-2 in all access units included in this elementary stream.

temporal_id_max: This indicates the maximum value of temporal ID specified by ITU-T Recommendation H.265 and ISO/IEC 23008-2 in all access units included in this elementary stream.

sub_pic_hrd_params_not_present_flag: This 1-bit field, when set to '0', indicates that the VUI in the HEVC video stream shall have the syntax element sub_pic_hrd_params_present_flag set to '1'. When the sub_pic_hrd_params_not_present_flag is equal to '1', the associated HEVC video stream may not contain sub_pic_hrd_params_present_flag in the VUI or the sub_pic_hrd_params_present_flag may be set to '0'.

HDR_WCG_idc (HDR/WCG): The value of this syntax element indicates the presence or absence of high dynamic range (HDR) and/or wide color gamut (WCG) video components in the associated PID according to Table 6-105-1.

Table 6-105-1　HDR_WCG_idc

HDR_WCG_idc	Description
0	SDR(Rec. ITU-R BT.1886 reference EOTF with a color gamut that is contained within Rec. ITU-R BT.709)
1	WCG（video color gamut in Rec ITU-R BT.2020）
2	HDR and WCG
3	No indication

6.2.57 Hierarchy descriptor

The hierarchy descriptor (see table 6-106) is used for identifying program components including video stream components coded by hierarchy coding.

Table 6-106 Hierarchy descriptor

Syntax	No. of bits	Identifier
hierarchy_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
reserved	1	bslbf
temporal_scalability_flag	1	bslbf
spatial_scalability_flag	1	bslbf
quality_scalability_flag	1	bslbf
hierarchy_type	4	uimsbf
reserved	2	bslbf
hierarchy_layer_index	6	uimsbf
tref_present_flag	1	bslbf
reserved	1	bslbf

hierarchy_embedded_layer_index	6	uimsbf
reserved	2	bslbf
hierarchy_channel	6	uimsbf
}		

Semantics for the Hierarchy descriptor:

temporal_scalability_flag: This is a 1-bit flag. When this flag is set to '0', the frame rate of a bit-stream of a program component referred by hierarchy_embedded_layer_index is increased. The flag of '1' is undefined.

spatial_scalability_flag: This is a 1-bit flag. When this flag is set to '0', the spatial resolution of a bit-stream of a program component referred by hierarchy_embedded_layer_index is increased. The flag of '1' is undefined.

quality_scalability_flag: This is a 1-bit flag. When this flag is set to '0', the SNR or fidelity of a bit-stream of a program component referred by hierarchy_embedded_layer_index is increased. The flag of '1' is undefined.

hierarchy_type: The relation between this layer and a base layer is shown in table 6-107. When more than 2 types of hierarchy coding are applied, this field is set to '8' to indicate to a complex hierarchy coding. The flags of temporal_scalability_flag, spatial_scalability_flag and quality_scalability_flag are appropriately set.

Table 6-107 Hierarchy coding type

Hierarchy coding type	Description
0	Undefined
1	Spatial scalability
2	Quality scalability
3	Temporal scalability
4	Data partitioning
5	Extended bit- stream
6	Private stream
7	Multi-view-profile
8	Complex hierarchy coding
9	MVC video sub-bit stream
10 - 14	Undefined
15	Base layer or MVC base view sub-bit stream or AVC video sub-bit stream of MVC or HEVC temporal sub-bit stream

hierarchy_layer_index: This is a 6–bit field. The value of this program component is specified in the table for coding layer. This value is unique in a program.

tref_present_flag: This is a 1–bit field. The flag of '0' indicates the possibility of a TREF (timestamp reference) field in a PES packet header of this elementary stream. The flag of '1' is undefined.

hierarchy_embedded_layer_index: This is 6-bit field which is set to the value of hierarchy_layer_index of a base program component. This index is accessed before decoding the elementary stream related by this hierarchy descriptor, and represented in decoded order. This flag is undefined in case that hierarchy_type is set to '15'.

hierarchy_channel: This is a 6–bit field. This field indicates the channel number of this program component among transmission channels with order. The minimum value of this field indicates the most robust transmission channel in the defined transmission layers.

[Note] hierarchy_channel may be applied to plural program components with the same value at the same time.

6.2.58 Hybrid information descriptor

The hybrid information_descriptor (see table 6-108) is used for describing information on acquiring communication components and meta-files.

Table 6-108 Hybrid information descriptor

Syntax	No. of bits	Identifier
hybrid_information_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
has_location	1	bslbf
location_type	1	bslbf
format	4	uimsbf
reserved	2	bslbf
if (has_location){		
if (location_type == 0){		
component_tag	8	uimsbf
module_id	16	uimsbf
}		
else{		
URL_length	8	uimsbf

```
                  for(i=0;i<N;i++){
                          URL_byte                          8       char
                  }
          }
     }
}
```

Semantics for the Hybrid information descriptor:

has_location: This field indicates the location where information for acquiring communication components and meta-files is described. This information is described in this descriptor in case that this field is set to '0'. The flag of '1' is undefined.

location_type: This field indicates a transmission path for delivering communication components and meta-files. Meta-files are transmitted via broadcast channel in case that this flag is set to '0'. Communication components and meta-files are transmitted via communication channel in case that this flag is set to '1'.

format: This field indicates formats of communication components and meta-files, and is coded according to table 6-109.

Table 6-109 Format

Format	Description
0	MPD (Media Presentation Descriptor) in MPEG-DASH (ISO/IEC 23009-1)
1	Playback control meta-file in VOD standard (IPTVFJ STD-0002) specified by IPTV
2	Timestamped TS in VOD standard (IPTVFJ STD-0002) specified by IPTV
3 - 7	Undefined

component_tag: This field indicates a label for identifying a component stream conveying a carousel module which transmits a meta-file in case that the meta-file is transmitted via broadcast channel.

module_id: This field indicates an identification number of a carousel module which transmits a meta-file in case that the meta-file is transmitted via broadcast channel.

URL_length: This field indicates the length of URL_byte field in case that communication components and meta-files are transmitted via communication channel.

URL_byte: This field indicates the URL in case that communication components and meta-files are transmitted via communication channel.

6.3 Descriptor used in INT

6.3.1 Identification and allocation of descriptor

Table 6-110 shows the guidelines for delivering descriptors used in INT.

Table 6-110 Guidelines for delivering descriptors used in INT

Descriptor	Descriptor location		
	platform	target	operational
Target smartcard descriptor	×	○	×
Target IP address descriptor	×	○	×
Target IPv6 address descriptor	×	○	×
IP/MAC platform name descriptor	◎	×	×
IP/MAC platform provide name descriptor	◎	×	×
IP/MAC stream location descriptor	◎	×	◎

◎: Indicates mandatory insertion into the descriptor.

○: Indicates optional insertion into the descriptor.

×: Indicates no insertion into the descriptor.

6.3.2 Coding of descriptor

6.3.2.1 Target smartcard descriptor

Target smartcard descriptor (see table 6-111) indicates receivers with smartcard ID. Smartcard ID is indicated by private_data_byte.

Table 6-111 Data structure of target smartcard descriptor

Syntax	No. of bits	Identifier
target_smartcard_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
super_CA_system_id	32	uimsbf
for (i=0;i<N;i++){		
private_data_byte	8	uimsbf
}		
}		

Semantics for the target smartcard descriptor:

super_CA_system_id: This 32-bit field is the linkage of CA_system_id of 16 bits with CA_subsystem_id of 16 bits and identifies conditional access. The allocation of CA_system_id is shown in Annex M. The allocation of CA_subsystem_id is specified by service providers.

private_data_byte: This 8-bit field describes the value of smartcard ID.

6.3.2.2 Target IP address descriptor

Target IP address descriptor (see table 6-112) indicates IPv4 addresses (plural) of targets

Table 6-112 Target IP address descriptor

Syntax	No. of bits	Identifier
target_IP_address_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
IPv4_addr_mask	32	uimsbf
for (i=0;i<N;i++){		
IPv4_addr	32	uimsbf
}		
}		

Semantics for the target IP address descriptor:

IPv4_addr_mask: This 32-bit field indicates mask of IPv4.

IPv4_addr: This 32-bit field describes IPv4 address of unicast, multicast and broadcast.

6.3.2.3 Target IPv6 address descriptor

Target IPv6 address descriptor (see table 6-113) indicates IPv6 addresses (plural) of targets

Table 6-113 Target IPv6 address descriptor

Syntax	No. of bits	Identifier
target_IP_address_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
IPv6_addr_mask	128	uimsbf
for (i=0;i<N;i++){		
IPv6_addr	128	uimsbf
}		
}		

Semantics for the target IPv6 address descriptor:

IPv6_addr_mask: This 128-bit field indicates mask of IPv6.

IPv6_addr: This 128-bit field describes IPv6 address of unicast, multicast and broadcast.

6.3.2.4 IP/MAC platform name descriptor

IP/MAC platform name descriptor (see table 6-114) shows names of IP platforms.

Table 6-114 IP/MAC platform name descriptor

Syntax	No. of bits	Identifier
IP/MAC_platform_name_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
ISO_639_language_code	24	bslbf
for (i=0;i<N;i++){		
text_char	8	uimsbf
}		
}		

Semantics for the IP/MAC platform name descriptor:

ISO 639_language_code: This 24-bit field contains the ISO 639-2 [27] three character language code of the language of the following text fields. Each character is coded into 8 bits according to ISO8859-1 [29] and inserted in order into the 24-bit field.

EXAMPLE: Japanese has 3-character code "jpn", which is coded as:
"0110 1010 0111 0000 0110 1110"

text_char: This is an 8-bit field. A string of "char" fields specify the text description for the event. Text information is coded using the character sets and methods described in Annex A.

6.3.2.5 IP/MAC platform provider name descriptor

IP/MAC platform provider name descriptor (see table 6-115) shows names of IP platform providers.

Table 6-115 IP/MAC platform provider name descriptor

Syntax	No. of bits	Identifier
IP/MAC_platform_provider_name_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
ISO_639_language_code	24	bslbf
for (i=0;i<N;i++){		
text_char	8	uimsbf
}		
}		

Semantics for the IP/MAC platform provider name descriptor:

ISO 639_language_code: This 24-bit field contains the ISO 639-2 [27] three character language code of the language of the following text fields. Each character is coded into 8 bits according to ISO8859-1 [29] and inserted in order into the 24-bit field.

EXAMPLE: Japanese has 3-character code "jpn", which is coded as:

"0110 1010 0111 0000 0110 1110"

text_char: This is an 8-bit field. A string of "char" fields specify the text description for the event. Text information is coded using the character sets and methods described in Annex A.

6.3.2.6 IP/MAC stream location descriptor

IP/MAC stream location descriptor (see table 6-116) shows locations (network_id, original_network_id, transport_stream_id, service_id, component tag) of IP streams.

Table 6-116 IP/MAC stream location descriptor

Syntax	No. of bits	Identifier
IP/MAC_stream_location _descriptor (){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
network_id	16	uimsbf
original_network_id	16	uimsbf
transport_stream_id	16	uimsbf
service_id	16	uimsbf
component_tag	8	uimsbf
}		

Semantics for the IP/MAC stream location descriptor:

network_id: This 16-bit field indicates the identification label discriminating the delivery system designated by NIT from other systems. The allocation of field values is specified by standard organization. (see Annex N)

original_network_id: This 16-bit field indicates the network identification label of the original delivery system for designated information service.

transport_stream_id: This 16-bit field identifies a transport stream including the designated information service.

service_id: This 16-bit field uniquely identifies the information service in the transport stream. The service_id is equal to program_number in the corresponding program map section.

component_tag: This 8-bit field identifies component stream indicated by component descriptor. The value of component tag for each stream is set to different value in a program map section.

Annex A (Normative)

Coding of character

Characters and control codes used in SI are in accordance with the 8-unit character code specified in sub-clause 7.1, section 7, part 2 of Vol. 1 of the ARIB STD-B24 "Data Coding and Transmission Specification for Digital Broadcasting".

-

However, the details of the character set shall be specified in the operational guidelines of service providers.

Annex B (Normative)

CRC decoder model

32-bit CRC decoder is shown in figure B-1.

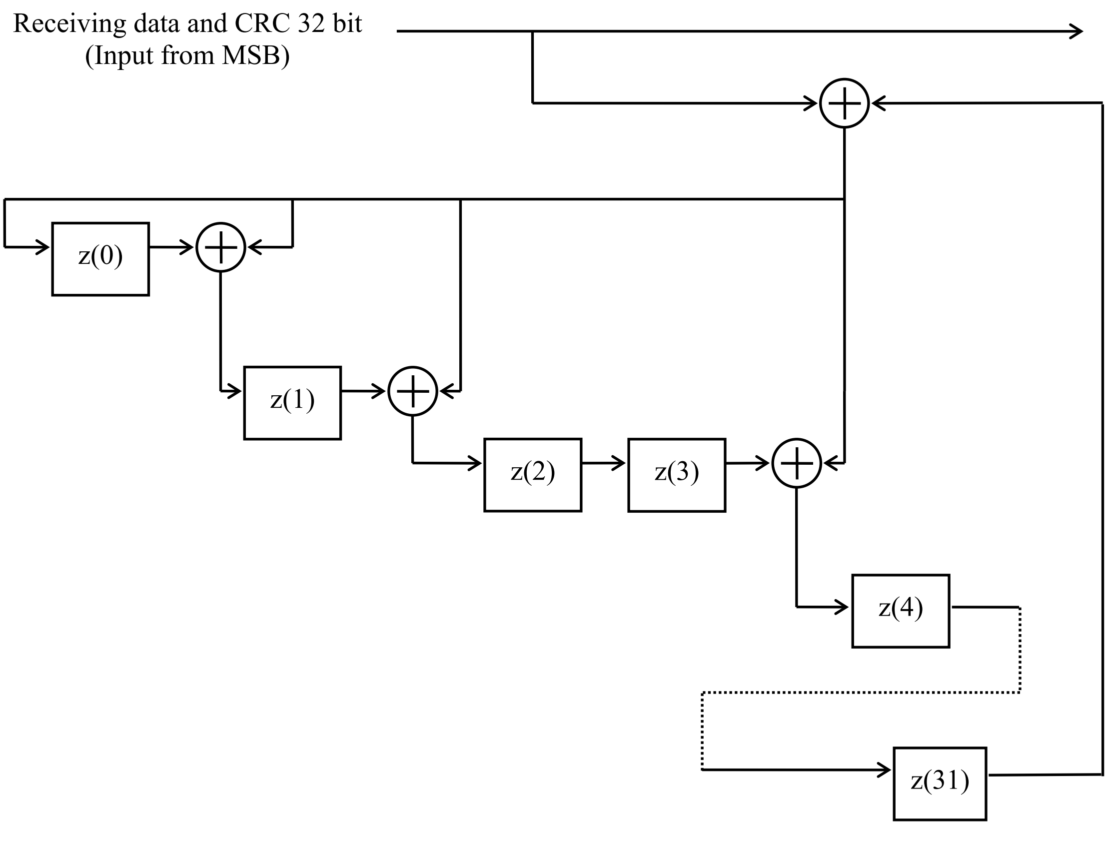


Figure B-1 32 bit CRC decoder model

The 32-bit CRC decoder is operated in bit level and is constituted of 14 adders (+) and 32 delay elements z(i). Input of the CRC decoder is added to the output of z(31), and the result is divided into the input of z(0) and input of one side of the rest of each of the adders. Input of the other side of the rest of the adders are output of z(i), and output of the rest of each adders is connected to the input of z(i+1), with i= 0, 1, 3, 4, 6, 7, 9, 10, 11, 15, 21, 22, 25. Refer to the figure above.

CRC is calculated by the following polynomial:

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$$

Reception at the input of the CRC decoder is made in byte unit. Each byte is shifted to CRC decoder in 1 bit each, in the order of MSB. For example, where byte 0x01 (last byte of start code prefix), first 7 "0"s are input to the CRC decoder and then 1 "1" is input. Output of each delay element z(i) is set to initial value "1", before data of 1 section is processed by CRC. After initialized, each byte of section including 4 CRC_32 byte is provided to input of the CRC decoder. After the last bit of the last CRC_32 byte is shifted to the decoder, which means that when added to output z(31) and then input to z(0), output of all delay element z(i) is read out. When there is no error, output of each z(i) is zero. In the CRC encoder, CRC_32 field is encoded in such value that it is assured.

Annex C (Informative)

Conversion of hours and dates

Conversion of Modified Julian Date (Japan time) and Japan standard time is as shown in figure C-1.

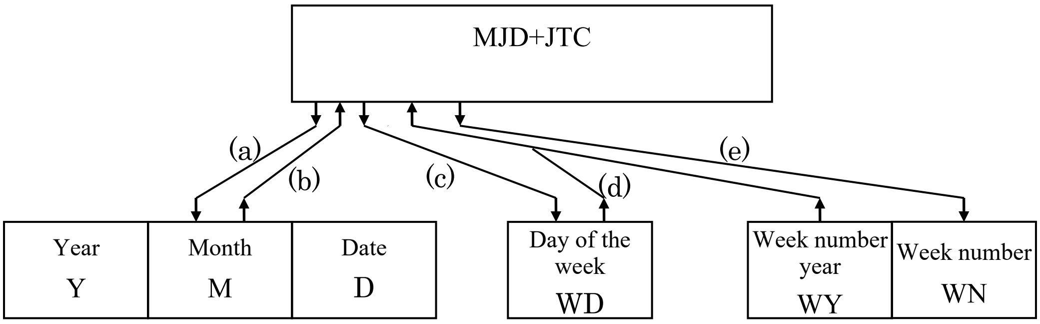


Figure C-1 Conversion of MJD (Japan time) and Japan standard time (JTC)

Relation of year-month-date and MJD (Japan time) is as shown below.

Used symbol:

MJD: Modified Julian Date (Japan time)

JTC: Japan Time Code

Y: Year from 1900 (For example, 2003 is Y=103)

M: Month (January = 1 to December = 12)

D: Date (1 to 31)

WY: Week number year since 1900

WN: Week number in accordance with ISO 2015

WD: Week day (Monday = 1 to Sunday = 7)

K, L, M', W, Y': Intermediate variables

x: Multiplication symbol

int: Integer part, ignoring remainder

mod 7: Remainder number after dividing integer by 7

a) Method to find year, month and date (Y, M, D) from MJD

$Y' = \mathrm{int}[(MJD - 15078.2) / 365.25]$

$M' = \mathrm{int}\{[MJD - 14956.1 - \mathrm{int}(Y' \times 365.25)] / 30.6001\}$

$D = MJD - 14956 - \mathrm{int}(Y' \times 365.25) - \mathrm{int}(M' \times 30.6001)$

Where $M' = 14$ or $M' = 15$: $K = 1$

In other cases,: $K = 0$

$Y = Y' + K$

$M = M' - 1 - K \times 12$

b) Method to find MJD from year, month and date (Y, M, D)

Where in case of $M = 1$ or $M = 2$: $L = 1$

In other cases: $L = 0$

$MJD = 14956 + D + \mathrm{int}[(Y - L) \times 365.25] + \mathrm{int}[(M + 1 + L \times 12) \times 30.6001]$

c) Method to find week day (WD) from MJD

$WD = [(MJD + 2) \bmod 7] + 1$

d) Method to find MJD from WY, WN and WD

$MJD = 15012 + WD + 7 \times \{WN + \mathrm{int}[(WY \times 1461 / 28) + 0.41]\}$

e) Method to find WY and WN from MJD

$W = \mathrm{int}[(MJD / 7) - 2144.64]$

$WY = \mathrm{int}[(W \times 28 / 1461) - 0.0079]$

$WN = W - \mathrm{int}[(WY \times 1461 / 28) + 0.41]$

Example:	MJD	=	45218	W	=	4315
	Y	=	(19)82	WY	=	(19)82
	M	=	9 (Sept.)	WN	=	36
	D	=	6	WD	=	1 (Monday)

[Note]: These formulas are effective from March 1, 1900 to February 28, 2100.

Annex D (Informative)

Specification of emergency alarm signal

Emergency alarm signal is specified in No.5 of clause 9-3 of Radio Equipment Regulation, article 138 of Radio Station Operation Rule, and Notation No. 405 of the Ministry of Posts and Telecommunications, 1985.

Signal type and local code specified in these rules are shown as follows.

Table D-1 Signal type

Signal type	Description	Classification of usage
0	1st type start signal	• When broadcasting that alarm declaration is issued by the specification of article 9, clause 1 of "Large scale earthquake countermeasure exceptional action law" (Law No. 73 in 1978). • When broadcasting in accordance with the specification of article 57 of "Disaster countermeasure basic law" (Law No. 223 in 1961) (including when applying article 20 of "Large scale earthquake countermeasure exceptional action law".)
1	2nd type start signal	• When broadcasting that tidal wave alarm has been issued by the specification of article 13 clause 1 of "Weather business law" (Law No. 165 in 1952.)

Table D-2 Local code

Local code	Description		Local code	Description	
0011 0100 1101	Local common code		1101 0100 1010	Prefecture code	Yamanashi
0101 1010 0101	Wide area code	Wide area of Kanto	1001 1101 0010		Nagano
			1010 0110 0101		Gifu
0111 0010 1010		Wide area of Chukyo	1010 0101 1010		Shizuoka
			1001 0110 0110		Aichi
1000 1101 0101		Wide area of Kinki	0010 1101 1100		Mie
			1100 1110 0100		Shiga
0110 1001 1001		Tottori, Shimane area	0101 1001 1010		Kyoto
			1100 1011 0010		Osaka
0101 0101 0011		Okayama, Kagawa area	0110 0111 0100		Hyogo
			1010 1001 0011		Nara
0001 0110 1011	Prefecture code	Hokkaido	0011 1001 0110		Wakayama
0100 0110 0111		Aomori	1101 0010 0011		Tottori
0101 1101 0100		Iwate	0011 0001 1011		Shimane
0111 0101 1000		Miyagi	0010 1011 0101		Okayama
1010 1100 0110		Akita	1011 0011 0001		Hiroshima
1110 0100 1100		Yamagata	1011 1001 1000		Yamaguchi
0001 1010 1110		Fukushima	1110 0110 0010		Tokushima
1100 0110 1001		Ibaraki	1001 1011 0100		Kagawa
1110 0011 1000		Tochigi	0001 1001 1101		Ehime
1001 1000 1011		Gunma	0010 1110 0011		Kochi
0110 0100 1011		Saitama	0110 0010 1101		Fukuoka
0001 1100 0111		Chiba	1001 0101 1001		Saga
1010 1010 1100		Tokyo	1010 0010 1011		Nagasaki
0101 0110 1100		Kanagawa	1000 1010 0111		Kumamoto
0100 1100 1110		Niigata	1100 1000 1101		Oita
0101 0011 1001		Toyama	1101 0001 1100		Miyazaki
0110 1010 0110		Ishikawa	1101 0100 0101		Kagoshima
1001 0010 1101		Fukui	0011 0111 0010		Okinawa

Annex E (Informative)

Table specified in ISO/IEC 13818-1

PAT, CAT, PMT are defined in ISO/IEC 13818-1. Data structure and parameter of each table specified in this standard are as follows.

Table E-1 Syntax of PAT

Syntax	No. of bits	Identifier
program_association_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
‘0’	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
transport_stream_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
for(i=0;i<N;i++){		
program_number	16	uimsbf
reserved	3	bslbf
if(program_number = = ‘0’){		
network_PID	13	uimsbf
}		
else{		
program_map_PID	13	uimsbf
}		
}		
CRC_32	32	rpchof
}		

Table E-2 Syntax of CAT

Syntax	No. of bits	Identifier
CA_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
‘0’	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
reserved	18	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
for(i=0;i<N;i++){		
descriptor()		
}		
CRC_32	32	rpchof
}		

Table E-3 Syntax of PMT

Syntax	No. of bits	Identifier
TS_program_map_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
'0'	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
program_number	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
reserved	3	bslbf
PCR_PID	13	uimsbf
reserved	4	bslbf
program_info_length	12	uimsbf
for(i=0;i<N;i++){		
descriptor()		
}		
for(i=0;i<N1;i++){		
stream_type	8	uimsbf
reserved	3	bslbf
elementary_PID	13	uimsnf
reserved	4	bslbf
ES_info_length	12	uimsbf
for(i=0;i<N2;i++){		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

Table E-4 Allocation of stream type

stream_type	Semantics
0x00	ITU-T | ISO/IEC Reserved
0x01	ISO/IEC 11172-2 Video
0x02	ITU-T Rec. H.262|ISO/IEC 13818-2 Video or ISO/IEC 11172-2 constrained parameter video stream
0x03	ISO/IEC 11172-3 Audio
0x04	ISO/IEC 13818-3 Audio
0x05	ITU-T Rec. H.222.0|ISO/IEC 13818-1 private_sections
0x06	ITU-T Rec. H.222.0|ISO/IEC 13818-1 PES packets containing private data
0x07	ISO/IEC 13522 MHEG
0x08	ITU-T Rec. H.222.0|ISO/IEC 13818-1 Annex A DSM-CC
0x09	ITU-T Rec. H.222.1
0x0A	ISO/IEC 13818-6 type A
0x0B	ISO/IEC 13818-6 type B
0x0C	ISO/IEC 13818-6 type C
0x0D	ISO/IEC 13818-6 type D
0x0E	ITU-T Rec. H.222.0|ISO/IEC 13818-1 auxiliary
0x0F	ISO/IEC 13818-7 Audio with ADTS transport syntax
0x10	ISO/IEC 14496-2 Visual
0x11	ISO/IEC 14496-3 Audio with the LATM transport syntax as defined in ISO/IEC 14496-3/AMD 1
0x12	ISO/IEC 14496-1 SL packetized stream or FlexMux stream carried in PES packets
0x13	ISO/IEC 14496-1 SL packetized stream or FlexMux stream carried in ISO/ IEC 14496_sections
0x14	ISO/IEC 13818-6 Synchronized Download Protocol
0x15	Metadata carried in PES packets
0x16	Metadata carried in metadata_sections
0x17	Metadata carried in ISO/IEC 13818-6 Data Carousel
0x18	Metadata carried in ISO/IEC 13818-6 Object Carousel
0x19	Metadata carried in ISO/IEC 13818-6 Synchronized Download Protocol
0x1A	IPMP stream (defined in ISO/IEC 13818-11, MPEG-2 IPMP)
0x1B	AVC video stream as defined in ITU-T Rec. H.264|ISO/IEC 14496-10 Video
0x1C	ISO/IEC 14496-3 Audio, without using any additional transport syntax, such as DST, ALS and SLS
0x1D	ISO/IEC 14496-17 Text
0x1E	Auxiliary video stream as defined in ISO/IEC 23002-3
0x1F	SVC video sub-bit stream of AVC video stream in accordance with more than one profile specified in Annex G of ITU-T Rec. H.264|ISO/IEC 14496-10 Video

0x20	MVC video sub-bit stream of AVC video stream in accordance with more than one profile specified in Annex H of ITU-T Rec. H.264|ISO/IEC 14496-10 Video
0x21	Video stream in accordance with more than one profile specified in ITU-T Rec. T.800|ISO/IEC 15444-1
0x22	ITU-T Rec. H.262| ISO/IEC 13818-2 video stream regarding additional viewpoint video in service-compatible stereoscopic 3D service
0x23	ITU-T Rec. H.264| ISO/IEC 14496-10 video stream regarding additional viewpoint video in service-compatible stereoscopic 3D service in accordance with ITU-T Rec. H.264| ISO/IEC 14496-10 Annex A
0x24	HEVC video stream or HEVC temporal video sub-bit stream
0x25	HEVC temporal video subset of HEVC video stream in accordance with more than one profile specified in Annex A of ITU-T Rec. H.265|ISO/IEC 23008-2
0x26 – 0x7E	Reserved
0x7F	IPMP stream
0x80 – 0xFF	User Private(*1)

*1 Allocation of stream type in User Private is shown in Annex O.

Annex F (Informative)

Example of service provider define bit of digital copy control descriptor

Example of service provider define bit of digital copy control descriptor is shown in table F-1.

Table F-1 Digital copy control descriptor

Syntax	No. of bits	Identifier
digital_copy_control_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
digital_recording_control_data	2	bslbf
maximum_bitrate_flag	1	bslbf
component_control_flag	1	bslbf
copy_control_type	2	bslbf
if(copy_control_type == 01){		
APS_control_data	2	bslbf
}		
else{		
reserved_future_use	2	bslbf
}		
if(maximum_bitrate_flag == 1){		
maximum_bitrate	8	uimsbf
}		
if(component_control_flag == 1){		
component_control_length	8	uimsbf
for(j=0;j<N;j++){		
component_tag	8	uimsbf
digital_recording_control_data	2	bslbf
maximum_bitrate_flag	1	bslbf
reserved_future_use	1	bslbf
copy_control_type	2	bslbf
if(copy_control_type == 01){		
APS_control_data	2	bslbf
}		
else{		
reserved_future_use	2	bslbf
}		
if(maximum_bitrate_flag == 1){		
maximum_bitrate	8	uimsbf
}		
}		
}		
}		

Semantics for the digital copy control descriptor:

copy_control_type: This 2-bit field indicates type information to control copy generation and encoded in accordance with table F-2.

Table F-2 Copy control type information

Copy control type information	Description
00	Undefined
01	Output by encoding to serial interface *1
10	Undefined
11	Output by not encoding to serial interface

*1 : Encoding method specified by service provider is used.

digital_recording_control_data: This 2-bit field indicates information to control copy generation and encoded in accordance with table F-3.

Table F-3 Digital recording control data

Digital recording control data	Description	
	When copy_control_type is 11	When copy_control_type is 01
00	Can be copied without control condition	Can be copied without control condition
01	Not used	Copy forbidden
10	Can be copied only once	Can be copied only once
11	Copy forbidden	Copy forbidden

APS_control_data: This 2-bit field indicates data to control analog output copy when the copy_control_type is 01 and encoded in accordance with table F-4.

Table F-4 Analog output copy control data

Analog output copy control data	Description
00	Can be copied without control condition
01	With pseudo-sync pulse
10	Pseudo-sync pulse + 2-line reversed division burst insertion
11	Pseudo-sync pulse + 4-line reversed division burst insertion

Annex G (Normative)

Region designator for prefecture designation for target region descriptor

When the area description method designation (region_spec_type) in target region descriptor is 0x01, that is when it is prefecture designation for BS digital, syntax of the bs_prefecture_spec(), which is the content of region designator target_region_spec(), is as shown below.

Table G-1 Region designator in prefecture designation

Syntax	No. of bits	Identifier
bs_prefecture_spec(){		
prefecture_bitmap	56	bslbf
}		

Semantics for the region designator in prefecture designation:

prefecture_bitmap: This 56-bit field is a bit map specified in table G-2. Bit designated with value 1 indicates that the region is the target and bit designated with value 0 indicates that the region is out of the target.

EXAMPLE: When the target area is Tokyo (excluding islands) value 1 is designated in the 14th bit from left.
"0000 0000 0000 0100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000"

Table G-2 Prefecture designation bit map

Order of bit	Description	Order of bit	Description	Order of bit	Description
1^{st}	East Hokkaido	20^{th}	Yamanashi	40^{th}	Kochi
2^{nd}	West Hokkaido	21^{st}	Nagano	41^{st}	Fukuoka
3^{rd}	Aomori	22^{nd}	Gifu	42^{nd}	Saga
4^{th}	Iwate	23^{rd}	Shizuoka	43^{rd}	Nagasaki
5^{th}	Miyagi	24^{th}	Aichi	44^{th}	Kumamoto
6^{th}	Akita	25^{th}	Mie	45^{th}	Oita
7^{th}	Yamagata	26^{th}	Shiga	46^{th}	Miyazaki
8^{th}	Fukushima	27^{th}	Kyoto	47^{th}	Kagoshima (excluding south west islands)
9^{th}	Ibaraki	28^{th}	Osaka		
10^{th}	Tochigi	29^{th}	Hyogo	48^{th}	Okinawa
11^{th}	Gunma	30^{th}	Nara	49^{th}	Island part of Tokyo (Izu, Ogasawara islands)
12^{th}	Saitama	31^{st}	Wakayama		
13^{th}	Chiba	32^{nd}	Tottori	50^{th}	Island part of Kagoshima (south west islands)
14^{th}	Tokyo (excluding island area)	33^{rd}	Shimane		
		34^{th}	Okayama	51^{st}	Reserved
15^{th}	Kanagawa	35^{th}	Hiroshima	52^{nd}	Reserved
16^{th}	Niigata	36^{th}	Yamaguchi	53^{rd}	Reserved
17^{th}	Toyama	37^{th}	Tokushima	54^{th}	Reserved
18^{th}	Ishikawa	38^{th}	Kagawa	55^{th}	Reserved
19^{th}	Fukui	39^{th}	Ehime	56^{th}	Reserved

Annex H (Normative)

Genre designation in content descriptor

The event genre of the content descriptor is designated in the following classification.

For events hard to classify genre should select "others".

For the future genre addition area, content_nibble_level1 = "0xC" to "0xD" is reserved.

"0xE" is an extension area and is defined as designation classification enabling to make reference to user_nibble.

[Large genre classification]

Large genre classification	Described content
0x0	News, report
0x1	Sports
0x2	Information/tabloid show
0x3	Drama
0x4	Music
0x5	Variety show
0x6	Movies
0x7	Animation/special effect movies
0x8	Documentary/culture
0x9	Theatre/public performance
0xA	Hobby/education
0xB	Welfare
0xC – 0xD	Reserved
0xE	For extension
0xF	Others

Large classification and medium classification list are shown as follows.

Content_nibble_level_1 Large genre classification	Content_nibble_level_2 Middle genre classification	Description
0x0	*	**News/reports**
0x0	0x0	Regular, general
0x0	0x1	Weather report
0x0	0x2	Special program, documentary
0x0	0x3	Politics, national assembly
0x0	0x4	Economics, market report
0x0	0x5	Overseas, international report
0x0	0x6	News analysis
0x0	0x7	Discussion, conference
0x0	0x8	Special report
0x0	0x9	Local program
0x0	0xA	Traffic report
0x0	0xB	
0x0	0xC	
0x0	0xD	
0x0	0xE	
0x0	0xF	Others
0x1	*	**Sports**
0x1	0x0	Sports news
0x1	0x1	Baseball
0x1	0x2	Soccer
0x1	0x3	Golf
0x1	0x4	Other ball games
0x1	0x5	Sumo, combative sports
0x1	0x6	Olympic, international games
0x1	0x7	Marathon, athletic sports, swimming
0x1	0x8	Motor sports
0x1	0x9	Marine sports, winter sports
0x1	0xA	Horse race, public race
0x1	0xB	
0x1	0xC	
0x1	0xD	
0x1	0xE	
0x1	0xF	Others

Content_nibble_level_1 Large genre classification	Content_nibble_level_2 Middle genre classification	Description
0x2	*	**Information/tabloid show**
0x2	0x0	Gossip/tabloid show
0x2	0x1	Fashion
0x2	0x2	Living, home
0x2	0x3	Health, medical treatment
0x2	0x4	Shopping, mail-order business
0x2	0x5	Gourmet, cocking
0x2	0x6	Events
0x2	0x7	Program guide, information
0x2	0x8	
0x2	0x9	
0x2	0xA	
0x2	0xB	
0x2	0xC	
0x2	0xD	
0x2	0xE	
0x2	0xF	Others
0x3	*	**Dramas**
0x3	0x0	Japanese dramas
0x3	0x1	Overseas dramas
0x3	0x2	Period dramas
0x3	0x3	
0x3	0x4	
0x3	0x5	
0x3	0x6	
0x3	0x7	
0x3	0x8	
0x3	0x9	
0x3	0xA	
0x3	0xB	
0x3	0xC	
0x3	0xD	
0x3	0xE	
0x3	0xF	Others

Content_nibble_level_1 Large genre classification	Content_nibble_level_2 Middle genre classification	Description
0x4	*	**Music**
0x4	0x0	Japanese rock, pop music
0x4	0x1	Overseas rock, pop music
0x4	0x2	Classic, opera
0x4	0x3	Jazz, fusion
0x4	0x4	Popular songs, Japanese popular songs (enka songs)
0x4	0x5	Live concert
0x4	0x6	Ranking, request music
0x4	0x7	Karaoke, amateur singing contests
0x4	0x8	Japanese ballad, Japanese traditional music
0x4	0x9	Children's song
0x4	0xA	Folk music, world music
0x4	0xB	
0x4	0xC	
0x4	0xD	
0x4	0xE	
0x4	0xF	Others
0x5	*	**Variety**
0x5	0x0	Quiz
0x5	0x1	Game
0x5	0x2	Talk variety
0x5	0x3	Comedy program
0x5	0x4	Music variety
0x5	0x5	Tour variety
0x5	0x6	Cocking variety
0x5	0x7	
0x5	0x8	
0x5	0x9	
0x5	0xA	
0x5	0xB	
0x5	0xC	
0x5	0xD	
0x5	0xE	
0x5	0xF	Others

Content_nibble_level_1 Large genre classification	Content_nibble_level_2 Middle genre classification	Description
0x6	*	**Movies**
0x6	0x0	Overseas movies
0x6	0x1	Japanese movies
0x6	0x2	Animation
0x6	0x3	
0x6	0x4	
0x6	0x5	
0x6	0x6	
0x6	0x7	
0x6	0x8	
0x6	0x9	
0x6	0xA	
0x6	0xB	
0x6	0xC	
0x6	0xD	
0x6	0xE	
0x6	0xF	Others
0x7	*	**Animation, special effects**
0x7	0x0	Japanese animation
0x7	0x1	Overseas animation
0x7	0x2	Special effects
0x7	0x3	
0x7	0x4	
0x7	0x5	
0x7	0x6	
0x7	0x7	
0x7	0x8	
0x7	0x9	
0x7	0xA	
0x7	0xB	
0x7	0xC	
0x7	0xD	
0x7	0xE	
0x7	0xF	Others

Content_nibble_level_1 Large genre classification	Content_nibble_level_2 Middle genre classification	Description
0x8	*	**Documentary/culture**
0x8	0x0	Social, current events
0x8	0x1	History, travel record
0x8	0x2	Nature, animal, environment
0x8	0x3	Space, science, medical science
0x8	0x4	Culture, traditional culture
0x8	0x5	Literature, literary art
0x8	0x6	Sports
0x8	0x7	Total documentary
0x8	0x8	Interviews, discussions
0x8	0x9	
0x8	0xA	
0x8	0xB	
0x8	0xC	
0x8	0xD	
0x8	0xE	
0x8	0xF	Others
0x9	*	**Theatre, public performance**
0x9	0x0	Modern drama, Western-style drama
0x9	0x1	Musical
0x9	0x2	Dance, ballet
0x9	0x3	Comic story, entertainment
0x9	0x4	Kabuki, classical drama
0x9	0x5	
0x9	0x6	
0x9	0x7	
0x9	0x8	
0x9	0x9	
0x9	0xA	
0x9	0xB	
0x9	0xC	
0x9	0xD	
0x9	0xE	
0x9	0xF	Others

Content_nibble_level_1 Large genre classification	Content_nibble_level_2 Middle genre classification	Description
0xA	*	**Hobby/education**
0xA	0x0	Trip, fishing, outdoor entertainment
0xA	0x1	Gardening, pet, handicrafts
0xA	0x2	Music, art, industrial art
0xA	0x3	Japanese chess (shogi) and "go"
0xA	0x4	Mah-jong, pinball games
0xA	0x5	Cars, motorbikes
0xA	0x6	Computer, TV games
0xA	0x7	Conversation, languages
0xA	0x8	Little children, schoolchildren
0xA	0x9	Junior high school and high school students
0xA	0xA	University students, examinations
0xA	0xB	Lifelong education, qualifications
0xA	0xC	Educational problem
0xA	0xD	
0xA	0xE	
0xA	0xF	Others
0xB	*	**Welfare**
0xB	0x0	Old aged persons
0xB	0x1	Handicapped persons
0xB	0x2	Social welfare
0xB	0x3	Volunteers
0xB	0x4	Sign language
0xB	0x5	Text (subtitles)
0xB	0x6	Explanation on sound multiplex broadcast
0xB	0x7	
0xB	0x8	
0xB	0x9	
0xB	0xA	
0xB	0xB	
0xB	0xC	
0xB	0xD	
0xB	0xE	
0xB	0xF	Others

Content_nibble_level_1 Large genre classification	Content_nibble_level_2 Middle genre classification	Description
0xC	*	**Reserved**
0xC	0x0	
0xC	0x1	
0xC	0x2	
0xC	0x3	
0xC	0x4	
0xC	0x5	
0xC	0x6	
0xC	0x7	
0xC	0x8	
0xC	0x9	
0xC	0xA	
0xC	0xB	
0xC	0xC	
0xC	0xD	
0xC	0xE	
0xC	0xF	
0xD	*	**Reserved**
0xD	0x0	
0xD	0x1	
0xD	0x2	
0xD	0x3	
0xD	0x4	
0xD	0x5	
0xD	0x6	
0xD	0x7	
0xD	0x8	
0xD	0x9	
0xD	0xA	
0xD	0xB	
0xD	0xC	
0xD	0xD	
0xD	0xE	
0xD	0xF	

Content_nibble_level_1 Large genre classification	Content_nibble_level_2 Middle genre classification	Description
0xE	*	**Extension**
0xE	0x0	Appendix information for BS/terrestrial digital broadcast program
0xE	0x1	Extension for broadband CS digital broadcasting
0xE	0x2	
0xE	0x3	Appendix information for server-type program
0xE	0x4	Appendix information for IP broadcast program
0xE	0x5	
0xE	0x6	
0xE	0x7	
0xE	0x8	
0xE	0x9	
0xE	0xA	
0xE	0xB	
0xE	0xC	
0xE	0xD	
0xE	0xE	
0xE	0xF	
0xF	*	**Others**
0xF	0x0	
0xF	0x1	
0xF	0x2	
0xF	0x3	
0xF	0x4	
0xF	0x5	
0xF	0x6	
0xF	0x7	
0xF	0x8	
0xF	0x9	
0xF	0xA	
0xF	0xB	
0xF	0xC	
0xF	0xD	
0xF	0xE	
0xF	0xF	Others

(Note) All items without denote in the description column are regarded as not defined.

Annex J (Informative)

Additional specification related to data component

In the data component descriptor and data content descriptor, there are fields to denote necessary information for the extension of id_number, storing supplement information and gaining data. Syntax of these fields is specified in each data component. Denoted places of the data component system and syntax are shown in table J-1.

Table J-1 Denoted places of data component system and syntax

<table>
<tr><th rowspan="2">Data component
(data_component_id)</th><th rowspan="2">Operational guidelines</th><th colspan="3">Places of syntax (reference)[*1]</th></tr>
<tr><th>Standard</th><th>additional_data_component_info of data component descriptor</th><th>sector_byte of data content descriptor</th></tr>
<tr><td>ARIB-XML-base multimedia coding
(0x0007)</td><td>TR-B15</td><td rowspan="2">STD-B24</td><td>Sub-clause 9.3.2, Vol. 2; Appended specification C.1, Vol. 3</td><td>Sub-clause 9.3.3, Vol. 2; Appended specification C.2, Vol. 3</td></tr>
<tr><td>ARIB-Subtitle& teletext coding
(0x0008)</td><td>TR-B14, B15</td><td>Sub-clause 9.6.1, Part 3, Vol. 1</td><td>Sub-clause 9.6.2, Part 3, Vol. 1</td></tr>
<tr><td>ARIB-Data download
(0x0009)</td><td>TR-B14, B15</td><td>STD-B21</td><td></td><td>This descriptor is not used</td></tr>
<tr><td>G-guide (G-Guide Gold)
(0x000A)</td><td></td><td>Private</td><td></td><td></td></tr>
<tr><td>BML for 110°E CS
(0x000B)</td><td>TR-B15 Part 2</td><td rowspan="3">STD-B24</td><td rowspan="3">Sub-clause 9.3.2, Vol. 2; Appended specification C.1, Vol. 3</td><td rowspan="2">Sub-clause 9.3.3, Vol. 2; Appended specification C.2, Vol. 3</td></tr>
<tr><td>Multimedia coding for digital terrestrial broadcasting (A profile)
(0x000C)</td><td rowspan="2">TR-B14</td></tr>
<tr><td>Multimedia coding for digital terrestrial broadcasting (C profile)
(0x000D)</td><td>Relevant descriptor not used</td></tr>
<tr><td>Multimedia coding for digital terrestrial broadcasting (P profile)
(0x000E out of use)</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Multimedia coding for</td><td></td><td>STD-B13</td><td>Sub-clause 5.3, Refer-</td><td></td></tr>
</table>

digital terrestrial broad-casting (E profile) (0x000F out of use)			ence, Vol. 3	
Real-time data service (Mobile profile) (0x0010 out of use)	TR-B26 (abol-ished)	STD-B24	Sub-clause 9.3.2, Vol. 2; Appended specification C.1, Vol. 3	Sub-clause 9.3.3, Vol. 2; Appended specifica-tion C.2, Vol. 3
Accumulation-type data service (Mobile profile) (0x0011 out of use)				
Subtitle coding for digi-tal terrestrial broadcast-ing (C profile) (0x0012)	TR-B14	STD-B24	Sub-clause 9.6.1, Part 3, Vol. 1	Relevant descriptor not used
Multimedia coding for digital terrestrial broad-casting (P2 profile) (0x0013 out of use)	TR-B13	STD-B24	Sub-clause 9.3.2, Vol. 2; Appended specification C.1, Vol. 3	Relevant descriptor not used
Data carousel scheme for TYPE2 content transmission (0x0014)	TR-B27	STD-B27	Relevant information not used	Sub-clause 11.3.2.4, Vol. 4
DSM-CC section scheme for transmission of program start time information (0x0015)			Relevant information not used	Relevant descriptor not used
ARIB-Program index coding (da-ta_component_id is not defined)		STD-B10	Sub-clause 6.4.1, Part 3	Sub-clause 6.4.2, Part 3
ARIB-Descriptive lan-guage type metadata coding (0x0016)	TR-B27	STD-B38	Sub-clause 3.5.1.1, Chapter 3	Sub-clause 3.5.1.1, Chapter 3
SKY Perfect data down-load (0x0017)		not re-leased		
SKY Perfect on-demand (0x0018)		not re-leased		
ARIB-Application exe-cution engine (0x0019)		STD-B23	Sub-section 10.6.(1), Part 2	Sub-section 10.6.(2), Part 2

ARIB-Application information table (0x001A)			Sub-section 10.6.(3), Part 2	This descriptor is not used
Multimedia coding for terrestrial multimedia broadcasting using ISDB-Tmm system (X profile) (0x001B)	TR-B33	STD-B24	Sub-clause 9.3.2, Vol. 2; Appended specification C.1, Vol. 3	Relevant descriptor not used
Subtitle & teletext coding for terrestrial multimedia broadcasting using ISDB-Tmm system (0x001C)	TR-B33	STD-B24	Sub-clause 9.6.1, Part 3, Vol. 1;	Relevant descriptor not used
Application control (0x001D)		STD-B24	Sub-clause 7.3, Vol. 4	Relevant descriptor not used
IPTVF hybrid system application (0x001E)		IPTVFJ STD-0011		
Narrowcast coding (0x001F)				

*1: Denoted places may be subject to change due to revision of the specifications.

Annex K (Normative)

Subdescriptors used in the carousel compatible composite descriptor

The descriptors in the module information area and the private area defined in the data carousel transmission scheme (Chapter 6 of ARIB STD-B24 Part 3) are used in the subdescriptor area of the carousel compatible composite descriptor. The tag values of the subdescriptors are listed in table K-1.

This Annex specifies the functions of those subdescriptors with a circle marked in the "Definition" column of table K-1, regarding their use for service information.

Table K-1 Subdescriptors used in the carousel compatible composite descriptor

Tag value	Subdescriptor	Function	Definition
0x01	Type descriptor	Type of contents (such as MIME)	○
0x02	Name descriptor	File name of accumulated contents	○
0x03	Info descriptor	(Undefined)	
0x04	Module_link descriptor	(Undefined)	
0x05	CRC32 descriptor	(Undefined)	
0x06	Reserved for future use		
0x07	Download estimate time descriptor	(Undefined)	
0x08 – 0x70	Reserved for future use		
0x71	Cache priority descriptor	(Undefined)	
0x72 – 0x7F	Reserved for future use		
0x80 – 0xBF	Selectable for provider-defined descriptor		
0xC0	Expire descriptor	Time of expiration of accumulated contents	○
0xC1	Activation Time descriptor	(Undefined)	
0xC2	Compression Type descriptor	(Undefined)	
0xC3	Control descriptor	(Undefined)	
0xC4	Provider Private descriptor	Specific auxiliary information is described by network and broadcasting service providers.	○
0xC5	Store Root descriptor	The directory where contents are accumulated in the accumulation device is specified.	○
0xC6	Sub Directory descriptor	The subdirectory where contents are accumulated within the directory specified by StoreRoot is specified.	○
0xC7	Title descriptor	The name of accumulated contents, with the aim of showing it to viewers, is described.	○

0xC8	Data Encoding descriptor	(Undefined)	
0xC9	TS descriptor with time stamp	(Undefined)	
0xCA	Route certificate descriptor	(Undefined)	
0xCB	Encrypt descriptor	(Undefined)	
0xCC	ACG descriptor	(Undefined)	
0xCD – 0xEE	Reserved for future use		
0xEF	Reserved for Transport Location descriptor		
0xF0 – 0xFE	Reserved for descriptor tags inserted in the private area in each coding scheme		

K.1 Type descriptor

The Type descriptor (see table K-2) indicates the type of object addressed by the carousel compatible composite descriptor containing this descriptor.

Table K-2 Type descriptor

Syntax	Number of bits	Identifier
type_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0; i<N; i++){		
text_char	8	uimsbf
}		
}		

Semantics for the Type descriptor:

text_char: This is an 8-bit field. A series of areas indicates the type of media complying with RFC1521 and RFC1590. The method of specifying media type used in XML-based multimedia coding is defined by Specification C of STD-B24 Part 2. Regardless of the specifications in Annex A, the character coding of this descriptor follows the specifications in Chapter 6 of ARIB STD-B24 Part 3.

K.2 Name descriptor

The Name descriptor (see table K-3) indicates the file name for accumulating the object addressed by the carousel compatible composite descriptor containing this descriptor.

Table K-3 Name descriptor

Syntax	Number of bits	Identifier
name_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0; i<N; i++){		
text_char	8	uimsbf
}		
}		

Semantics for the Name descriptor:

text_char: this is an 8-bit field. A series of areas indicates the file name for accumulating the applicable object. Regardless of the specifications in Annex A, the character coding of this descriptor follows the specifications in Chapter 6 of ARIB STD-B24 Part 3.

K.3 Expire descriptor

The Expire descriptor (see table K-4) indicates the time of expiration of the object addressed by the carousel compatible composite descriptor containing this descriptor. For example, an objected accumulated in a receiver having an accumulation device will be erased at the time of expiration. The time of expiration is not set if this descriptor is not used.

Table K-4 Expire descriptor

Syntax	Number of bits	Identifier
expire_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
time_mode	8	uimsbf
If(time_mode == 0x01){		
MJD_JST_time	40	bslbf
}		
else if(time_mode == 0x04){		
reserved_future_use	8	bslbf
passed_seconds	32	uimsbf
}		
}		

Semantics for the Expire descriptor:

time_mode (time mode): Indicates the method of specifying the time of expiration (see table K-5).

Table K-5 Time mode

time_mode	Time specifying method	meaning
0x00	—	Reserved for future use
0x01	MJD_JST_time	Absolute time based on the Modified Julian Date and Japan Standard Time
0x02	—	Reserved for future use
0x03	—	Reserved for future use
0x04	passed_seconds	Elapsed time after downloading (sec)
0x05 – 0xFF	—	Reserved for future use

MJD_JST_time: This 40-bit field, which is coded when time_mode = "0x01", indicates the time of expiration based on the Modified Julian Date (MJD) and Japan Standard Time (JST) (see Annex C). This field is coded as 16 bits corresponding to the 16 least significant bits of MJD followed by 24 bits coded as 6 digits in 4-bit binary coded decimal (BCD).

passed_seconds: This 32-bit field, which is coded when time_mode = "0x04", indicates the time of expiration based on the elapsed time (in sec) after accumulation.

K.4 ProviderPrivate descriptor

The ProviderPrivate descriptor (see table K-6) describes specific auxiliary information on the object addressed by the carousel compatible composite descriptor containing this descriptor, according to the rules defined by the scope of each network or broadcasting service provider.

Table K-6 ProviderPrivate descriptor

Syntax	Number of bits	Identifier
provider_private_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
private_scope_type	8	bslbf
scope_identifier	32	bslbf
for(i = 0; i< N; i++){		
private_byte	8	bslbf
}		
}		

Semantics for the ProviderPrivate descriptor:

private_scope_type: This 8-bit field indicates the type of identifier that shows the scope of this descriptor.

scope_identifier: This 32-bit field is used to indicate the scope identifier value of each scope type (see table K-7).

Table K-7 Private scope types and scope identifiers

private_scope_type	scope_identifier	Number of bits	Identifier	Description
0x00	-	-	-	Reserved for future use
0x01	network_id	16	uimsbf	Network identifier is used as the scope of this descriptor.
	Padding	16	bslbf	
0x02	network_id	16	uimsbf	Service identifier is used as the scope of this descriptor.
	service_id	16	uimsbf	
0x03	network_id	16	uimsbf	Broadcaster identifier is used as the scope of this descriptor.
	broadcaster_id	8	uimsbf	
	Padding	8	bslbf	
0x04	bouquet_id	16	uimsbf	Bouquet identifier is used as the scope of this descriptor.
	Padding	16	bslbf	
0x05	information_provider_id	16	uimsbf	Information provider identifier is used as the scope of this descriptor.
	Padding	16	bslbf	
0x06	CA_system_id	16	uimsbf	CA system identifier is used as the scope of this descriptor.
	Padding	16	bslbf	
0x07 – 0xFF	-	-	-	Reserved for future use

Note: "1" is set to all bits in "padding".

private_byte: This is an 8-bit field. A series of areas describes auxiliary information based on the rules defined by each scope.

K.5 StoreRoot descriptor

The StoreRoot descriptor (table K-8) indicates the reference directory where the object addressed by the carousel compatible composite descriptor containing this descriptor is accumulated. It also indicates whether to renew or add to the existing object in the same directory when accumulating.

Table K-8 StoreRoot descriptor

Syntax	Number of bits	Identifier
store_root_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
update_type	1	bslbf
reserved	7	bslbf
for(i = 0;i< N;i++){		
store_root_path	8	uimsbf
}		
}		

Semantics for the StoreRoot descriptor:

update_type: This 1-bit field indicates whether to erase the content of the directory specified by store_root_path before accumulating the applicable object. Accumulation starts after erasing the existing content when updata_type is "1", and without erasing when updata_type is "0".

store_root_path: This is an 8-bit field. A series of areas indicates the reference directory where the applicable object is accumulated in the accumulation device, using the character coding defined in Chapter 9 of STD-B24 Part 2.

K.6 Subdirectory descriptor

The Subdirectory descriptor (see table K-9) indicates the subdirectory, within the reference directory specified by StoreRoot in the accumulation device, to accumulate the object addressed by the carousel compatible composite descriptor containing this descriptor. When the Subdirectory descriptor is not used, the reference directory specified by the StoreRoot descriptor is effective.

Table K-9 Subdirectory descriptor

Syntax	Number of bits	Identifier
subdirectory_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i = 0;i< N;i++){		
subdirectory_path	8	uimsbf
}		
}		

Semantics for the Subdirectory descriptor:

subdirectory_path: This is an 8-bit field. A series of areas indicates the subdirectory where the applicable object is accumulated within the directory specified by the StoreRoot descriptor, using the character coding defined in Chapter 9 of ARIB STD-B24 Part 2.

K.7 Title descriptor

The Title descriptor (see table K-10) indicates the name to be recognized by viewers as a character string when the object addressed by the carousel compatible composite descriptor containing this descriptor is accumulated.

Table K-10 Title descriptor

Syntax	Number of bits	Identifier
title_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
ISO_639_language_code	24	bslbf
for(i=0; i<N; i++){		
text_char	8	uimsbf
}		
}		

Semantics for the Title descriptor:

ISO_639_language_code: This 24-bit field denotes the language used in the subsequent text_char area. The language coding follows the alphabetic 3-character coding defined in ISO 639-2(21). Each character is coded into 8 bits according to ISO 8859-1(23) and inserted in the order into the 24-bit field.

text_char: This is an 8-bit field. A series of areas indicates the name presented to viewers for the applicable object. Regardless of the specifications in Annex A, the character coding of this descriptor follows the specifications in Chapter 6 of ARIB STD-B24 Part 3.

Annex L (Normative)

Composite descriptor for tag value extension

When the number of usable descriptors needs to be increased, the method of using composite descriptors shown in this Annex shall be used to code the descriptors. A subdescriptor shall be placed in each composite descriptor.

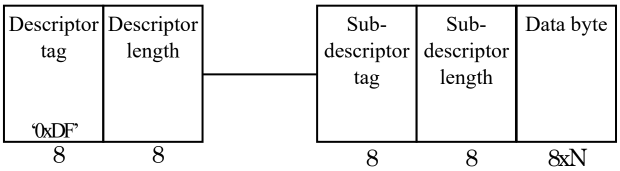


Figure L-1 Data structure of composite descriptor for tag value extension

Table L-1 Composite descriptor for tag value extension

Syntax	Number of bits	Identifier
composite_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
sub_descriptor()		
}		

Table L-2 Subdescriptor

Syntax	Number of bits	Identifier
sub_descriptor(){		
sub_descriptor_tag	8	uimsbf
sub_descriptor_length	8	uimsbf
for(i=0; i<N; i++){		
data_byte;	8	uimsbf
}		
}		

Annex M (Informative)

Allocation of conditional access system identifiers

The conditional access system identifier (CA_system_id) shall be specified, registered, and released by the standardization organization. The allocated conditional access system identifiers are listed in table M-1.

Table M-1 Allocation of conditional access system identifiers

Name of conditional access system	CA_system_id	Operational guidelines (reference)
SKY Perfect conditional access system	0x0001	Proprietary specification of SKY Perfect JSAT
Hitachi system	0x0003	JCL SPEC-005[*1]
Secure Navi system	0x0004	JCL SPEC-005[*1]
ARIB conditional access system	0x0005	ARIB TR-B14, B15
ARIB conditional access system (Second generation)		ARIB TR-B39
Matsushita CATV conditional access system	0x0006	JCL SPEC-005[*1]
Cable Labs access control system	0x0007	JCL SPEC-001-01[*1] JCL SPEC-002[*1] JCL SPEC-007[*1]
u-CAS system	0x0008	
PowerKEY system	0x0009	
ARIB conditional access system B	0x000A	ARIB TR-B26 (abolished)
PIsys conditional access system	0x000B	
MULTI2-NAGRA system	0x000C	
IPTV Forum Marlin system	0x000D	IPTV specification IP broadcasting specification
ARIB content protection system	0x000E	ARIB TR-B14
ConPas system	0x000F	ARIB TR-B33
MULTI2-NAGRA(Merlin) system	0x0010	

*1 Operational specification of JCTA Japan Cable Laboratories

Annex N (Informative)

Allocation of network identifiers

The network identifier (network_id) shall be specified, registered, and released by the standardization organization. The allocated network identifiers are listed in table N-1.

Table N-1 Allocation of network identifiers

Name of conditional access system	network_id	Operational guidelines (reference)
PerfecTV! service	0x0001	Proprietary specification of SKY Perfect JSAT
DIRECTV	0x0002	
SKY service	0x0003	Proprietary specification of SKY Perfect JSAT
BS digital broadcasting	0x0004	ARIB TR-B15 Part 1
U-Satellite Broadcasting	0x0005	
SKY Perfect CS1 (wide-band CS digital broadcasting system)	0x0006	ARIB TR-B15 Part 2
SKY Perfect CS2 (wide-band CS digital broadcasting system)	0x0007	ARIB TR-B15 Part 2
Access Television Network	0x0008	
SPACE DiVA	0x0009	
SKY Perfect (advanced narrow-band CS digital broadcasting)	0x000A	Proprietary specification of SKY Perfect JSAT
Advanced BS digital broadcasting	0x000B	ARIB TR-B39 Part 1
Advanced broad band CS digital broadcasting	0x000C	ARIB TR-B39 Part 2
Mobile Broadcasting	0x0100 (out of use)	ARIB TR-B26 (abolished)
Multimedia broadcasting	0x0200	ARIB TR-B33
IPBC Plala, IPBC KDDI, IPBC SB (IPTV Forum IP broadcasting)	0x7780 – 0x778F	IPTV specification IP broadcasting specification
Terrestrial digital television broadcasting	0x7880 – 0x7FE8*2	ARIB TR-B14
Tokyo Seg#1 – Seg#8 (Terrestrial digital sound broadcasting)	0x8090 – 0x8097 (out of use)	ARIB TR-B13 (abolished)
Osaka Seg#1 – Seg#8 (Terrestrial digital sound broadcasting)	0x8098 – 0x809F (out of use)	ARIB TR-B13 (abolished)
V-Low multimedia broadcasting	0x8100 – 0x8106	ARIB TR-B38
Fukuoka ubiquitous district experimental broadcasting	0x8371, 0x8380 (out of use)	
Digital radio (local service)	0x83FF (out of use)	ARIB TR-B13 (abolished)
Independent broadcasting by cable television operators in the terrestrial digital	0x7C1D – 0x7F5F*3	JCL SPEC-006*1 JCL SPEC-007*1

broadcasting network		
Advanced cable independent broadcasting	0xFFF7	JLabs SPEC-035*5
IP independent broadcasting	0xFFF8	JLabs SPEC-028*5
Advanced JC-HITS Trans-Modulation	0xFFF9	JLabs SPEC-019*5
Digital broadcasting advanced ReMUX	0xFFFA	JLabs SPEC-017*5 JLabs SPEC-018*5
Video on demand service exploiting interlocking data broadcasting (BML-VOD)	0xFFFB(out of use)	MEI BML-VOD SPEC 1.0*4
Analogue to digital system conversion	0xFFFC (out of use)	JCL SPEC-008*1
JC-HITS Trans-Modulation	0xFFFD	JCL SPEC-005*1
Digital broadcasting ReMUX	0xFFFE	JCL SPEC-003*1 JCL SPEC-004*1
Kagoshima Cable Television	0xFFFF	Proprietary specification of Kagoshima Cable Television

*1 Operational specification of JCTA Japan Cable Laboratories
SPEC-003 "Operational specifications for ReMUX digital broadcasting (independent broadcasting)"
SPEC-004 "Operational specifications for ReMUX digital broadcasting (i-HITS)"
SPEC-005 "Operational specifications for JC-HITS Trans-Modulation"
SPEC-006 "Operational specifications for Pass-Through terrestrial digital broadcasting and independent broadcasting"
SPEC-007 "Operational specifications for Trans-Modulation terrestrial broadcasting and independent broadcasting"
SPEC-008 "Operational specifications for analog to digital system conversion"

*2 See ARIB TR-B14 Volume 7 for allocation within this range

*3 Only 53 pieces which last 4bits are “D” and “F”. See JCL SPEC-006 Volume 2 and JCL SPEC-007 Part 2 for allocation within this range

*4 Proprietary specification of Matsushita Electric Industrial Co., Ltd.

*5 Operational specification of Japan Cable Laboratories
SPEC-017 "Operational specifications for advanced ReMUX digital broadcasting (independent broadcasting)"
SPEC-018 "Operational specifications for advanced ReMUX digital broadcasting (i-HITS)"
SPEC-019 "Operational specifications for advanced JC-HITS Trans-Modulation"
SPEC-028 "Operational specifications for IP broadcasting (independent broadcasting)"
SPEC-035 “Operational specifications for advanced cable independent broadcasting (ACAS corresponding)”

Annex O (Normative)

Stream type not specified in ISO/IEC 13818-1

The allocation of 0x80 – 0xFF as user private in the allocation of stream type of ISO/IEC 13818-1 is shown in table O-1.

User private is divided to the area (ARIB standard area) to identify streams specified by ARIB standard and the area (ARIB user private) to identify streams individually specified by companies. In case that a stream not uniquely identified by stream type is used, a registration descriptor shall be used.

Table O-1 Allocation of stream_type

stream_type	Description
0x80 – 0x90	Undefined (ARIB standard area)
0x91	Unidirectional Lightweight Encapsulation (ULE), IETF RFC 4326
0x92 – 0xC3	Undefined (ARIB standard area)
0xC4 – 0xFF	ARIB user private

Annex P (Informative)

Allocation of platform identifiers

The platform identifier (platform_id) shall be specified, registered, and released by the standardization organization. The allocated platform identifiers are listed in table P-1.

Table P-1 Allocation of platform identifiers

platform_id	Platform	Operator
0x000000	Reservation	
0x000001 – 0xFFEFFF	Managed by DVB	
0xFFF000	Multimedia broadcasting	Japan Mobilecasting, Inc.
0xFFF001 – 0xFFEFFE	Undefined (managed by ARIB)	
0xFFFFFF	Reservation	

Explanation

1. How to standardize SI

In the July 24th, 1995 partial report to the government; the necessity for standardized SI concerning the broadcast service, the multiplexing and arrangement of individual program etc. in order to facilitate program selection by viewers was reported. On the condition that a) the signal transmission format be MPEG section data and format and b) the SI presentation be only in text form; a non-governmental organization was to realize this, ensuring flexible reaction to future innovations, In response to this, the transmission-path-coding committee in the new-broadcast-system special group in the Association of Radio Industries and Businesses has established SI as a non-governmental standard according to the following basic.

(1) Early realization and international compatibility

DVB-SI* is a SI standard which has been presented at various international committees and is being standardized by Europe. The DVB-SI standard was selected as the basic method because a) the DVB* method basically adheres to condition reported above, b) has adequate SI functions although some changes are necessary to adapt to broadcasting conditions in Japan, c) makes early introduction of digital broadcasting possible, 4) makes widespread use of receiving circuits through international compatibility possible.

(2) Media independence

SI should be media independent as far as possible. Target of DVB-SI is independence for total broadcasting media. The ARIB standardization scope includes CS digital broadcasting transmission media but parts of the transmission media such as cable TV, etc. have not been considered. Those will be specified additionally by the related organization when it becomes necessary.

(3) Convenience for users

Digital broadcasting signals were standardized to be capable of providing greatly improved convenience for users, compared to conventional broadcast. In particular, signals providing minimum function necessary for broadcasting were classified as "mandatory" and signals providing functions "as needed" by broadcasting service providers were classified as "optional".

(4) Signal extensibility

Taking into consideration future developments in technology, and also to ensure flexible development of the broadcasting industry, service providers are allowed to independently define original signals in addition to the standardized signals as long as they are within the ISO/IEC13818-1 scope. These independent service provider defined signals should be registered and released to the public in order to ensure transparency of broadcasting signal and make "common" receivers possible.

Also accordingly, part of mandatory signal, can be substituted by service provider defined signals, if these signals contain functions already defined in the mandatory signals.

SI informs the viewer about multiplexed program information; simplifies program selection and complements PSI (Program Specific Information) in MPEG-2 Systems. In order to establish the standard; it was necessary to map descriptors describing information service details to individual tables, including tables defined in PSI, so PSI tables newly defined for SI is also described.

2. Extension of SI and allocation of descriptor

With the development of digital broadcasting and the development and practical use of new services, addition and updating of this standard will be made. Since SI is regarded as independent throughout the broadcasting media and the identifiers of the service are allocated fixedly, there should be careful discussion considering international trends of identifier allocation when revising the standard.

In the case of independent service provider defined signals, it is assumed that tables and descriptors will be added in accordance with service development. In this case also, the signals should be registered and released to the public.

In principle, identifier values for service provider defined signals should be unique throughout a network, and it is the responsibility of the network manager to supervise.

This principle can be implemented by a receiver with a software switching function which switches identifiers for each network. However, unconditional switching may cause problems, making network-to-network common use of service provider signals difficult. Study should be carried out regarding which method to select by viewing further industrial trends.

3. Extension of SI

With the development of digital broadcasting and accumulation of viewing experience with the new service, there might arise some need to update the standard to provide more efficient, user-friendly SI. In this case, extension of the specification should be made such as addition of tables or descriptors, or addition of transmission tables of a descriptor when necessary, after deliberation by the committee. In the case of these extensions, compatibility with former specifications should be considered and the IRD should be designed so that former functions are not obstructed by the extension signal.

4. Publication and registration of service provider defined signal

Broadcast program organization differs from service provider to service provider. In order to secure individual and flexible program organization, basic signals and universal information are standardized in ARIB and transmitted SI which reflect program organization of a particular service provider are approved as extensions of the ARIB standard.

Tables and descriptors defined by service providers are regarded as basic signals for broadcasting and should comply with the publication rules of a public broadcasting system. Identifier values allocated to service provider defined signals are related closely to software design of receivers and as described in the previous clause might possibility extended beyond the scope of a single network, so they and their data structure should be publicized.

Registration procedures to register broadcasting systems particular to certain service providers will be established elsewhere. Registration and publication of service provider defined SI, data structure, identifier values etc. by this same registration system will be required.

Meaning of "publication" used herein does not include unconditional release/publication of intellectual propriety rights of the method owned by the registrant.

5. Operational standard of the identifier

Unified management for the allocation of identifier and identifier values is necessary in order to avoid confusion such as overlapping of values, etc., Part 1, table 7-1 are management guidelines.

For allocation of the identifier related to signal specified in non-governmental standard, it is necessary to be in accordance with this table. For the unified management of the identifier, it is recommended to unify also in the non-governmental standardization organization. When multiple organizations are related, overlapping of the value should be avoided by adjusting the range of the used identifier value, etc.

"Standardization organization" denoted in Part 1 table 7-1 is a general expression and does not indicate any specific group.

DVB: Abbreviation of Digital Video Broadcasting. It means a non-governmental group studying digital broadcasting methods in Europe, or its digital broadcasting system. Examination of SI "DVB-SI (DVB-SI)" and broadcasting of satellite, cable, and digital terrestrial broadcasting are made. Examined results are standardized in "ETSI", the standardization organization in Europe.

Reference materials

(1) Electric Communication Engineering Committee report
Technical condition of digital satellite broadcasting system (using 27MHz bandwidth) using 12.2 to 12.75 GHz in "Technical conditions related to digital broadcasting method"
(July, 1995)

(2) Electric Communication Engineering Committee report
Technical conditions of BS digital broadcasting system (using digital satellite broadcasting 11.7 to 12.2 GHz bandwidth) (Feb. 9, 1998)

(3) Electric Communication Engineering Committee report
Technical condition of digital terrestrial television broadcasting system (May 24, 1999)

(4) Electric Communication Engineering Committee report
Technical condition of digital satellite broadcasting system (using 34.5MHz bandwidth) using 12.2 to 12.75 GHz in "Technical conditions related to digital broadcasting method"
(Feb. 28, 1999)

(5) Electric Communication Engineering Committee report
Technical conditions of digital terrestrial sound broadcasting system (Nov. 29, 1999)

(6) Information and Communication Council report
Technical conditions related to advanced satellite digital broadcasting (July 29, 2008)

(7) Information and Communication Council report
Technical conditions related to multimedia broadcasting for mobile terminal (Oct. 16, 2009)

(8) Information and Communication Council report
"Technical conditions related to a satellite main broadcasting and a satellite general broadcasting" in "Technical conditions related to an UHDTV broadcasting system"
(March 25, 2014)

(9) Information and Communication Council report
"Technical conditions related to the enhancement of picture quality of an UHDTV broadcasting system and so on" in "Technical conditions related to an UHDTV broadcasting system"
(May 24, 2016)

(10) Ministerial Ordinance No.87 of the Ministry of Internal Affairs and Communications in Jun 29, 2011. "Standard transmission system for digital broadcasting among standard television broadcasting and the like"

(Revised in Dec. 10, 2013, in Jul. 3, 2014, in Jul. 29, 2016)

(June 29, 2011)

(11) Ministerial Notification No. 233 of the Ministry of Internal Affairs and Communications in 2014 “Structure and delivery procedure of rerated information, delivery procedure of PES packet, section format, TS packet, IP packet, ULE packet, MMTP packet, compressed IP packet and TLV packet, structure of transmission control signal and identifier, structure of emergency information descriptor and emergency warning broadcasting”

(July 3, 2014)

(12) ARIB STD-B1 Ver. 3.2 "Digital receiver for digital satellite broadcasting services using communication satellites (Recommended specification)"

(April 2018)

(13) ARIB STD-B16 Ver. 1.1 "Standard common IRD standard specification for CS digital broadcasting" (Feb. 1999)

(14) ARIB STD-B32 Ver. 3.10 "Video coding, audio coding and multiple system of digital system" (April 2018)

(15) ARIB STD-B21 Ver. 5.10 "Receiver unit for digital broadcasting (Recommended specification)" (April 2018)

(16) ARIB STD-B23 Ver. 1.2 "Application execution engine platform for digital broadcasting

(July 2009)

(17) ARIB STD-B24 Ver. 6.4 "Data coding and transmission systems for digital broadcasting

(July 2017)

(18) ARIB STD-B25 Ver. 6.6 "Conditional access system specifications for digital broadcasting"

(April 2018)

(19) ARIB STD-B38 Ver. 2.3 "Coding, transmission, and storage control systems for server-type broadcasting" (July 2013)

(20) ARIB STD-B55 Ver. 1.3 “Transmission system for area broadcasting”

(Mar. 2014)

(21) ARIB TR-B14 Ver. 6.3 "Operational guidelines for digital terrestrial television broadcasting"

(Jan. 2018)

(22) ARIB TR-B15 Ver. 7.5 "Operational guidelines for BS/broadband CS digital broadcasting" (Jan. 2018)

(23) ARIB STD-B35 Ver. 2.0 “Operational guidelines for area broadcasting” (Sep. 2015)

(24) Rec. ITU-T H.222.0 (03/2017)|ISO/IEC 13818-1:2018 ”Information Technology - Generic Coding of Moving Pictures and Associated Audio Information: Systems”

(25) Rec. ITU-T H.264 (02/2014)|ISO/IEC 14496-10:2014 ”Advanced video coding for generic audiovisual services” (Newest version: H.264 (V12) (04/2017))

(26) Rec. ITU-T H.265 (04/2013)|ISO/IEC 23008-2:2013 ”High efficiency video coding” (Newest version: H.265 (V5) (Feb. 2018))

(27) ISO 639-2 (1998) "Codes for the representation of names of languages - Part 2: Alpha-3 code"

(28) ISO 3166-1 (2013) " Codes for the representation of names of countries and their subdivisions - Part 1: Country codes "

(29) ISO 8859-1 (1998) "Information processing - 8-bit single-byte coded graphic character sets - Part 1: Latin alphabet No.1"

(30) ETSI ETS 300 468 Edition 2（1997-01） ”Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems” (Newest version: ETSI EN 300 468 V1.15.1 (2016-03)）

(31) ETSI ETR 211 Edition 2 (1997-08) "Digital Video Broadcasting (DVB); Guidelines on Implementation and Usage of Service Information" (Newest version：ETSI TS 101 211 V1.12.1 (2013-12)）

(32) ETSI EN 301 192 V1.5.1 (2009-11) “Digital Video Broadcasting (DVB); Specification for data broadcasting” (Newest version: ETSI EN 301 192 V1.6.1 (2015-08))

(33) ETSI TS 101 162 V1.2.1 (2009-07) “Digital Video Broadcasting (DVB); Allocation of Service Information (SI) and Data Broadcasting Codes for Digital Video Broadcasting (DVB) Systems”（Newest version：ETSI TS 101 162 V1.8.1 (2017-01) Allocation of identifiers and codes for Digital Video Broadcasting (DVB) systems）

(34) Rec. ITU-R BO.1408-1 (04/2002) "Transmission system for advanced multimedia services provided by integrated services digital broadcasting in a broadcasting-satellite channel"

(35) Rec. ITU-R BT.1306-6 (12/2011) "Error-correction, data framing, modulation and emission methods for digital terrestrial television broadcasting" (Newest version: Rec. ITU-R BT.1306-7 (06/2015))

<BLANK>

Part 3

DATA STRUCTURE AND DEFINITION OF EXTENSION INFORMATION OF SERVICE INFORMATION

Part 3
DATA STRUCTURE AND DEFINITION OF EXTENSION INFORMATION OF SERVICE INFORMATION

CONTENTS

1. Purpose

Part 3 of this standard is established to specify detail syntax of extension information of SI, basing on the SI defined in Ministerial Ordinance No. 87 of the Ministry of Public Management, Home Affairs, Posts and Telecommunications "Standard transmission system of digital broadcast of standard TV broadcasting, etc." in 2011.

2. Scope

Part 3 of this standard applies to extension information of SI specified in Part 1.

3. Definitions and abbreviations

3.1 Definitions

This standard applies the following definitions in addition to the definitions set forth in Part 2.

Local event (program segment event): Part of the event (program) subdivided by time line or program component, etc.

Node: Node of graph defined to describe relation of the event (program) and/or the local event (program segment event), which is encoded as extension information of SI. Node itself has no meaning, but a meaning is given from the relation with other node or node description.

Information provider: Organization who provides information encoded in extension information of SI to audience.

Material information: Basic information for materials composing programs, which is link to related information on material type, material name, material code used for specifying the material from outside, usage conditions and entity data etc.

3.2 Abbreviations

This standard applies the following abbreviations in addition to abbreviations set forth in Part 2.

LIT	Local event Information Table
ERT	Event Relation Table
ITT	Index Transmission information Table
STC	System Time Clock
NPT	Normal Play Time
PTS	Presentation Time Stamp
URL	Uniform Resource Locator
HTML	Hyper Text Markup Language

4. Explanation of extension information of SI

4.1 Organization of extension information of SI

In addition to the basic information of SI defined in Part 2, data for describing the relation among programs, information of contents smaller than program and relation among them are expected.

The EIT in basic information of SI (Part 2) describes information individually in the unit of the event (program). Extension information of SI defined in Part 3 of this standard can describe the relation among events and information of the local event and relation among local events, which are smaller parts of the events.

Information to describe the relation among multiple events and/or local events is called a program group index, and information to describe information of local event in one program or the relation among local events is called a program segment index. The program group index and program segment index are called a program index as a whole.

Extension information of SI consists of the following three tables in addition to the EIT and ST defined in basic information of SI.

Local event Information Table (LIT):

The LIT includes information related to the local event (program segment event) such as name, start time and duration of a local event

Event Relation Table (ERT):

The ERT includes information related to node indicating attribute or group of event (program) and/or the local event (program segment event), and information of relation of those nodes.

It indicates relation of the events by using with the EIT and indicates relation of the local events by using with the LIT. By using with both EIT and LIT, it can indicate relation of both events and local events.

Program Index Transmission information Table (ITT):

The ITT includes auxiliary information related to program transmission, such as the relation between the STC and the time information for identifying local event (program segment event). These information are given an exclusive table dividing with the LIT, because there may be information, that is fixed at the moment of program transmission or values differing in each time of program transmission.

Flexible table structure and compatible extension for the future are possible by using descriptors.

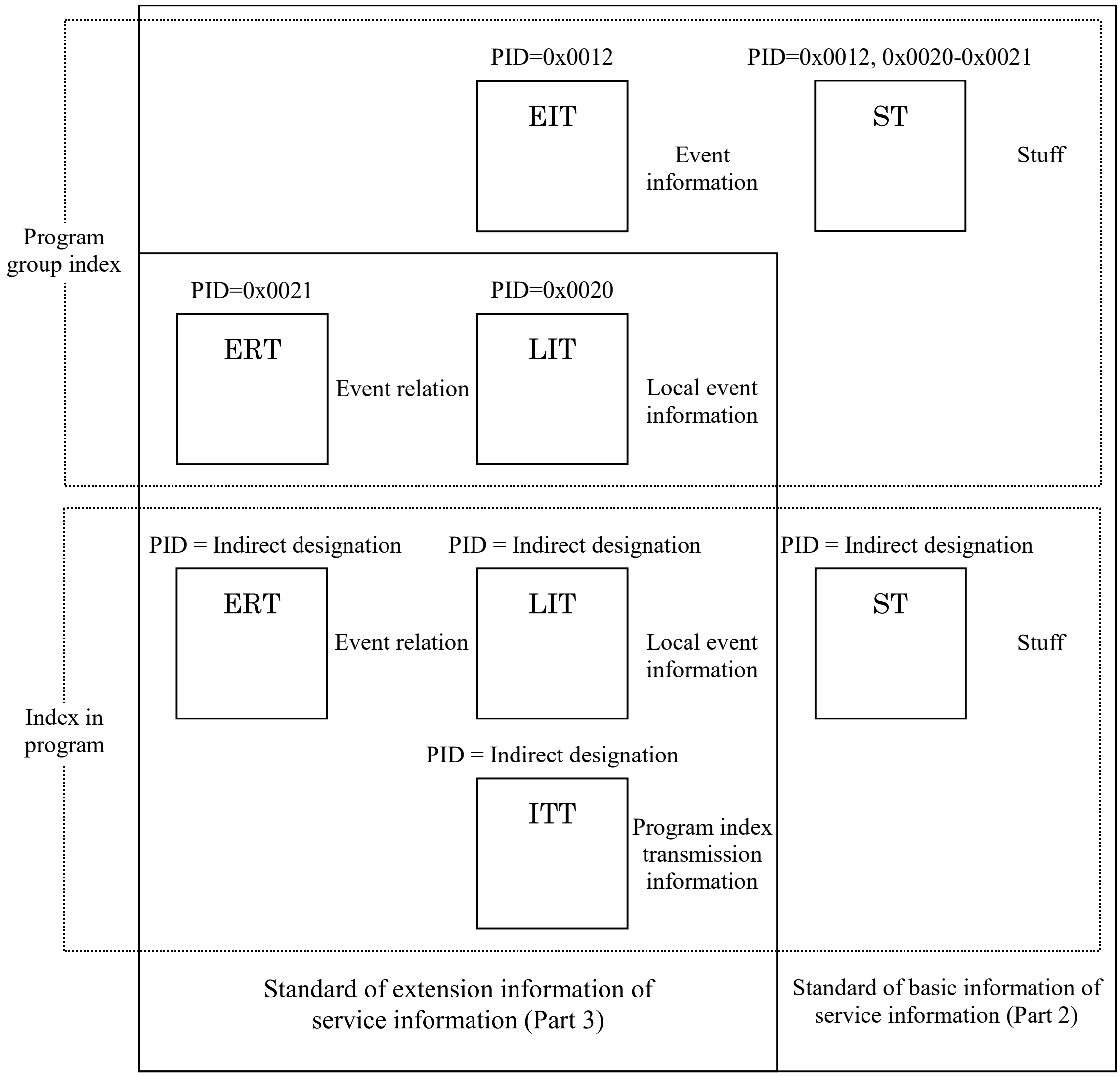


Figure 4-1 Organization of extension information of SI

4.2 Program group index

Program group index provides grouping information of the program (event) and assists in selecting or searching a program by this grouping information. The program group index enables grouping of programs in various viewpoints such as series program group like a series TV drama, program group with same contents such as broadcasting and rebroadcasting, and group of recommended programs, etc.

Program group index is provided by the EIT defined in basic information of SI (Part 2) and the ERT defined in extension information of SI (Part 3). The EIT defines events (programs) and describes grouping information of events in character or code of the program group defined in the ERT. The ERT defines the program group and describes its attributes in text. The ERT can also express the relation among program groups.

In the program group index, not only events (programs) but also local events (program segment events) can be objects of the grouping. In this case, the LIT is used to define the local events.

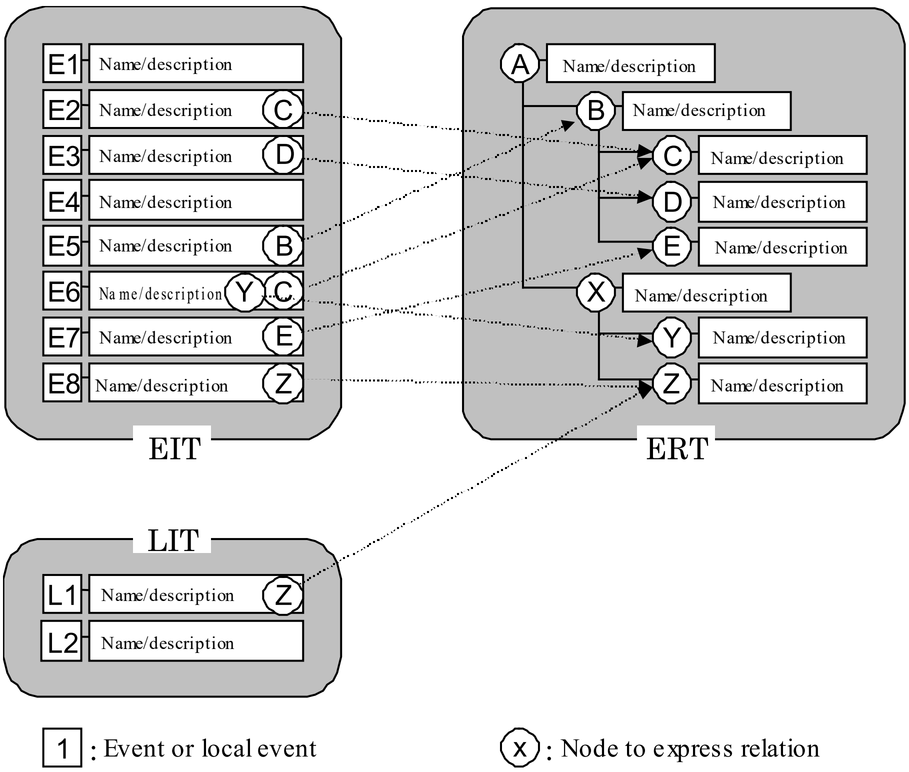


Figure 4-2 Outline of program group index

4.3 Program segment index

Program segment index provides information to assist in selecting or searching local events (program segment events). Furthermore, grouping information of local events is provided and selecting or searching local events is assisted by this grouping information.

The program segment index is provided by the LIT and the ERT defined in extension information of SI (Part 3). The LIT defines the local event and also describes grouping information defined in the ERT by code. The ERT defines grouping information of local events and describes the grouping information in text. The ERT can also express the relation among groups.

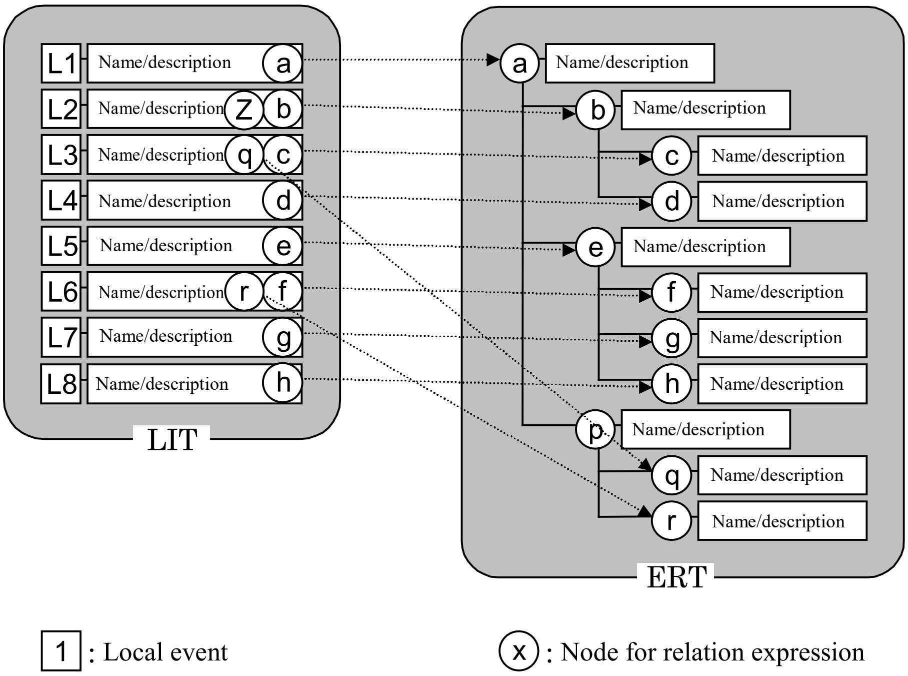


Figure 4-3 Outline of program internal index

5. Program index encoding method

5.1 Table used for program index encoding

The following tables are defined as extension information of SI in Part 3 of this standard for encoding program index.

(1) Local event information table (LIT)

(2) Event relation table (ERT)

(3) Index transmission information table (ITT)

The following tables defined as basic information of SI (Part 2) are also used.

(4) Event information table (EIT)

(5) Stuffing table (ST)

Syntax and semantics of each table (1), (2), (3) are described in the following clauses.

[Note]: Symbols, abbreviations and description method of the syntax used in this standard is in accordance with clauses 2.2 and 2.3 in ISO/IEC 13818-1.

5.1.1 Local event Information Table (LIT)

The LIT is information related to the local event (program segment event) included in each event (program). Each sub_table includes all description related to the local event of one program and composed of local event information section, for which the values of table_id, event_id service_id, transport_id, original_network_id, and version_number coincide.

Syntax of the local event information section is shown in table 5-1.

Table 5-1 Local event information sections

Syntax	No. of bits	Identifier
local_event_information_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
event_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
service_id	16	uimsbf
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
for(i=0;i<N;i++){		
local_event_id	16	uimsbf
reserved_future_use	4	bslbf
descriptors_loop_length	12	uimsbf
for(j=0;j<M;j++){		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

Semantics for the local event information section:

table_id: This field indicates the local event information section and shall be set to 0xD0. See tables 6-1 and 6-2.

section_syntax_indicator: Section syntax indicator is a 1-bit field which shall be set to "1".

section_length: This is a 12-bit field. It specifies the number of bytes of the section, starting immediately following the section_length field and including the CRC. The section_length shall not exceed 4093 so that the entire section has a maximum length of 4096 bytes.

event_id: This 16-bit field indicates the event_id (uniquely assigned in a service) of the event, that the local event information section describes.

version_number: This 5-bit field is the version number of the sub_table. The version_number shall be incremented by 1 when a change in the information carried within the sub_table occurs. When it reaches value 31, it wraps around to 0. When the current_next_indicator is set to "1", then the version_number shall be that of the currently applicable sub_table defined by the table_id and event_id. When the current_next_indicator is set to "0", then the version_number shall be that of the next applicable sub_table defined by the table_id and event_id.

current_next_indicator: This 1-bit indicator, when set to "1" indicates that the sub_table is the currently applicable sub_table. When the bit is set to "0", it indicates that the sub_table sent is not yet applicable and shall be the next sub_table to be valid.

section_number: This 8-bit field gives the number of the section. The section_number of the first section in the sub_table shall be "0x00". The section_number shall be incremented by 1 with each additional section with the same table_id, event_id, service_id, transport_stream_id and original_network_id.

last_section_number: This 8-bit field specifies the number of the last section of the sub_table to which this section belongs.

service_id: This 16-bit field indicates the service_id number (uniquely assigned in a network) of the service to which the event described by the local event information section belongs. The service_id is the same as the program_number in the corresponding program map section.

transport_stream_id: This 16-bit field indicates the transport_stream_id (uniquely assigned in a network) of the transport stream where to which the event described by the local event information section belongs.

original_network_id: This 16-bit field indicates the originating_network_id of the original_network to which the event described by the local event information section belongs.

local_event_id: This 16-bit field serves as a label to identify the local_event (program segment event).

descriptors_loop_length: This 12-bit field gives the total length in byte of the following descriptors.

CRC_32: This 32-bit field contains the CRC value for the entire section.

5.1.2 Event Relation Table (ERT)

The ERT describes the relation among the events (programs) and/or local events (program segment events). The event relation table consists of sub_tables. The sub_table specifies the relation among the events and/or local events for a particular use, and it is constructed by the event relation section in which values of table_id, event_relation_id, information_provider_id and version_number coincide.

Event relation section is indicated in table 5-2.

Table 5-2 Event relation section

Syntax	No. of bits	Identifier
event_relation_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
event_relation_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
information_provider_id	16	uimsbf
relation_type	4	uimsbf
reserved_future_use	4	bslbf
for(i=0;i<N;i++){		
node_id	16	uimsbf
collection_mode	4	uimsbf
reserved_future_use	4	bslbf
parent_node_id	16	uimsbf
reference_number	8	uimsbf
reserved_future_use	4	bslbf
descriptors_loop_length	12	uimsbf
for(j=0;j<M;j++){		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

Semantics for the event relation section:

table_id: Table field indicates the event relation section and shall be set to 0xD1. See tables 6-1 and 6-2.

section_syntax_indicator: Section syntax indicator is a 1-bit field which shall be set to "1".

section_length: This is a 12-bit field. It specifies the number of bytes of the section, starting immediately following the section_length field and including the CRC. The section_length shall not exceed 4093 so that the entire section has a maximum length of 4096 bytes

event_relation_id: This is a 16-bit field and serves as a label to identify the event relation.

version_number: This 5-bit field is the version number of the sub_table. The version_number shall be incremented by 1 when a change in the information carried within the sub_table occurs. When it reaches value 31, it wraps around to 0. When the current_next_indicator is set to "1", then the version_number shall be that of the currently applicable sub_table defined in table_id and event_relation_id. When the current_next_indicator is set to "0", then the version_number shall be that of the next applicable sub_table defined in table_id and event_relation_id.

current_next_indicator: This 1-bit indicator, when set to "1", indicates that the sub_table is the currently applicable sub_table. When the bit is set to "0", it indicates that the sub_table sent is not yet applicable and shall be the next sub_table to be valid.

section_number: This 8-bit field gives the number of the section. The section_number of the first section in the sub_table shall be "0x00". The section_number shall be incremented by 1 with each additional section with the same table_id, event_relation_id and information_provider_id.

last_section_number: This 8-bit field specifies the number of the last section of the sub_table to which this section belongs.

information_provider_id: This 16-bit field identifies the information provider who specifies the event relation.

relation_type: This 4-bit field indicates the type of the relation described by the event relation section. See table 5-3.

Table 5-3 Relation type

relation_type	Semantics
0x0	Reserved
0x1	Relation for the contents description (Indicates tree structure to describe contents)
0x2	Relation for navigation (Indicates tree structure to assist display and selection)
0x3-0xF	Reserved for future use

node_id: This 16-bit field serves as a label to identify the node used to describe the relation among the event and/or local event. Node identifier "0x0000" is reserved for a special node to describe the event relation sub_table. The node identifier "0xFFFF" is not used.

collection_mode: This 4-bit field indicates the characteristics of the collection of events, local events and nodes which refer to this node by the parental_node_id, node_relation_descriptor or reference_descriptor. See table 5-4.

Table 5-4 Collection mode

collection_mode	Semantics
0x0	Group (bag)
0x1	Concatenation (sequential)
0x2	Selection (alternate)
0x3	Parallel
0x4-0xF	Reserved for future use

parent_node_id: This 16-bit field indicates the node_id_of the parent node when the node refers another node in the event relation sub_table as a parent of the tree structure. When the parent node is not specified by this field, "0xFFFF" shall be coded.

reference_number: This 8-bit field specifies the priority of reference in the collection of events, local events and nodes which refers to the same node.

descriptors_loop_length: This 12 bit field gives the total length in byte of the following descriptors.

CRC_32: This 32-bit field contains the CRC value for the entire section.

5.1.3 Index transmission information table (ITT)

The index transmission information table describes information to be used for transmission of program index.

Index transmission information table consists of sub_tables. The sub_table is a table including information for transmission of program index of an event (program), and constructed of program index transmitting information section. See table 5-5.

Table 5-5 Index transmitting section

Syntax	No. of bits	Identifier
index_transmission_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
event_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
reserved_future_use	4	bslbf
descriptors_loop_length	12	uimsbf
for(i=0;i<N;i++){		
descriptor()		
}		
CRC_32	32	rpchof
}		

Semantics for the program index transmitting information section:

table_id: This field indicates the event relation section and shall be set to 0xD2. See table 6-2.

section_syntax_indicator: Section syntax indicator is a 1-bit field which shall be set to "1".

section_length: This is a 12-bit field. It specifies the number of bytes of the section, starting immediately following the section_length field and including the CRC. The section_length shall not exceed 4093 so that the entire section has a maximum length of 4096 bytes

event_id: This 16-bit field identifies the event (program). It indicates the event identifier of the event, by which the program index transmission information section is provided.

version_number: This 5-bit field is the version number of the sub_table. The version_number shall be incremented by 1 when a change in the information carried within the sub_table occurs. When it reaches value 31, it wraps around to 0. When the current_next_indicator is set to "1", then the version_number shall be that of the currently applicable sub_table defined in table_id and event_id. When the current_next_indicator is set to "0", then the version_number shall be that of the next applicable sub_table defined in table_id and event_id.

current_next_indicator: This 1-bit indicator, when set to "1" indicates that the sub_table is the currently applicable sub_table. When the bit is set to "0", it indicates that the sub_table sent is not yet applicable and shall be the next sub_table to be valid.

section_number: This 8-bit field gives the number of the section. The section_number of the first section in the sub_table shall be "0x00". The section_number shall be incremented by 1 with each additional section having the same table_id, event_id, service_id, transport_stream_id and original_network_id.

last_section_number: This 8-bit field specifies the number of the last section of the sub_table to which this section belongs.

descriptors_loop_length: This 12-bit field gives the total length in bytes of the following descriptors.

CRC_32: This 32-bit field contains the CRC value for the entire section.

5.2 Descriptor used for program index encoding

The following identifiers are defined as extension information of SI in Part 3 of this standard for encoding program index.

(1) Basic local event descriptor

(2) Reference descriptor

(3) Node relation descriptor

(4) Short node information descriptor

(5) STC reference descriptor

(6) Material information descriptor

The following descriptors defined in basic information of SI (Part 2) are also used as a standard.

(7) Short event descriptor

(8) Extended event descriptor

(9) Hyperlink descriptor

(10) Stuffing descriptor

Syntax and semantics of each descriptor of (1) to (6) are described in the following clauses.

5.2.1 Basic local event descriptor

The basic local event descriptor used in the LIT indicates segmentation information of the local event (program segment event), such as start time, duration and component identifier, etc. See table 5-6.

Table 5-6 Basic local event descriptor

Syntax	No. of bits	Identifier
basic_local_event_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
reserved_future_use	4	bslbf
segmentation_mode	4	uimsbf
segmentation_info_length	8	uimsbf
if(segmentation_mode == 0){		
}		
else if(segmentation_mode == 1){		
reserved_future_use	7	bslbf
start_time_NPT	33	uimsbf
reserved_future_use	7	bslbf
end_time_NPT	33	uimsbf
}		
else if(segmentation _mode < 6){		
start_time	24	uimsbf
duration	24	uimsbf
if(segmentation_info_length == 10){		
start_time_extension	12	uimsbf
reserved_future_use	4	bslbf
duration_extension	12	uimsbf
reserved_future_use	4	bslbf
}		
}		
else{		
for(i=0; i< M; i++){		
reserved	8	bslbf
}		
}		
for(i=0; i<N; i++){		
component_tag	8	uimsbf
}		
}		

Semantics for the basic local event descriptor:

segmentation_mode: This 4-bit field specifies the coding type of the segmentation information such as time and hour, etc. in the basic local event descriptor. See table 5-7.

Table 5-7 Segmentation mode

segmentation_mode	Name	Semantics
0x0	Invalid	Segmentation information is not designated in the basic local event descriptor
0x1	NPT	Designated by NPT form
0x2	Relative time	Designate relative time from the start time of program in hour, minute, and second (ms.) form
0x3	Relative time (STC reference descriptor is used together)	Designate relative time from the start time of program in hour, minute, and second (ms.) form
0x4	JST time	Designate JST time of broadcasting in hour, minute, and second (ms.) form
0x5	JST time (STC reference descriptor is used together)	Designate JST time of broadcasting in hour, minute, and second (ms.) form
0x6-0xF	reserved_future_use	Reserved for future use.

segmentation_info_length: This 8-bit field specifies the byte length of the subsequent segmentation information.

start_time_NPT: This 33-bit field specifies the start time of the local event in NPT form.

end_time_NPT: This 33-bit field specifies the end time of the local event in NPT form.

start_time: This 24-bit field expresses the unit of seconds or the greater time unit of the start time of the local event. Using six 4-bit binary-coded decimal numbers (BCD), the time is coded in the order of hours, minutes and seconds. When no start time is defined (for example, the start time remains undetermined, or it is not open yet), all bits in this field shall be set to "1".

duration: This 24-bit field expresses the unit of seconds or the greater time unit of duration of the local event. Using six 4-bit binary-coded decimal numbers (BCD), the time duration is coded in the order of hours, minutes and seconds. When no time duration is defined (for example, the time duration remains undetermined, or it is not open yet), all bits in this field shall be set to "1". The value for this field shall be set to "0" to indicate a point on the time base.

start_time_extension: This 12-bit field expresses units smaller than seconds of the start time of the local event. Using three 4-bit binary-coded decimal numbers (BCD), the time is coded in milliseconds. When no start time is defined, all bits in this field shall be set to "1". This field is omitted when no specification is made down to the millisecond level of accuracy.

duration_extension: This 12-bit field expresses units smaller than seconds of the time duration of the local event. Using three 4-bit binary-coded decimal numbers (BCD), the time is coded in milliseconds. When no time duration is defined, all bits in this field shall be set to "1". The value for this field shall be set to "0" to indicate a point on the time base. This field is omitted when no specification is made down to the millisecond level of accuracy.

component_tag: This 8-bit field serves as a label to identify the component stream within this local event. The component stream to which the corresponding value of this component tag is assigned in the PMT belongs to this local event. This field could be omitted if all the component streams belong to this local event. This field has the value of "0xFF" if none of the component streams belong to this local event. "0xFF" is used only for this case, and is not used for the stream identifier descriptor.

5.2.2 Reference descriptor

The reference descriptor used in EIT or LIT associates the event or the local event with the event relation sub_table. The reference descriptor refers to the event relation sub_table (omitted in some cases) and indicates that event or local event placed with this descriptor has attribute indicated by reference node. See table 5-8.

Table 5-8 Reference descriptor

Syntax	No. of bits	Identifier
reference_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
information_provider_id	16	uimsbf
event_relation_id	16	uimsbf
for(i=0;i<N;i++){		
reference_node_id	16	uimsbf
reference_number	8	uimsbf
last_reference_number	8	uimsbf
}		
}		

Semantics for the reference descriptor:

information_provider_id: This 16-bit field indicates the information provider id of the event relation sub_table to which the referred node belongs.

event_relation_id: This 16-bit field indicates the event relation id of the event relation sub_table to which the referred node belongs.

reference_node_id: This 16-bit field indicates the node id of the referred node.

reference_number: This 8-bit field specifies the reference priority of the nodes to be referred to. If the referred node is the node that indicates the event or the local event itself, it should be "0x00". If the referred node indicates the parent node of the event or the local event, the reference_number should be specified by the value calculated based on the following equation.

The value should be "0xFF" when the reference priority is not designated.

reference_number = mod(actual reference priority order – 1, 254) + 1

last_reference_number: This 8-bit field indicates the maximum value of the reference_number of the referred node. The last_reference_number should be specified buy the value calculated based on the following equation. The value should be "0xFF" when the last reference priority is not designated.

last_reference_number = mod(actual last reference priority order – 1,254) + 1

The last_reference_number should not be equal to the reference_number, except when the actual reference order coincides with the last reference priority order. Therefore, when there is a possibility that the encoded value of reference order equals the encoded value of the last reference order, "0xFF" is set to the last_reference_number.

5.2.3 Node relation descriptor

The node relation descriptor is used to describe the referencing relation of nodes in event relation table (ERT). If the referencing is the basic referencing relation that is only made to the parent node and the parent node is located in the same event relation identifier, the parent node identifier field of ERT section is used to express the node relation. See table 5-9.

Table 5-9 Node relation descriptor

Syntax	No. of bits	Identifier
node_relation_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
reference_type	4	uimsbf
external_reference_flag	1	bslbf
reserved_future_use	3	bslbf
if(external_reference_flag == 1){		
information_provider_id	16	uimsbf
event_relation_id	16	uimsbf
}		
reference_node_id	16	uimsbf
reference_number	8	uimsbf
}		

Semantics for the node relation descriptor:

reference_type: This 4-bit field indicates the reference attribute for the node indicated by the reference_node_id. See table 5-10.

Table 5-10 Reference type

reference_type	Semantics
0x0	Reference to parent node
0x1 – 0xF	Reserved for future use

external_reference_flag: The value "0" indicates that the node to be referred to by the reference node id is located in the same event relation table, while the value "1" indicates that the node to be referred to by the reference node id is located in the other event relation table.

information_provider_id: This 16-bit field designates information provider identifier of the sub_table when the referred node belongs to different event relation sub_table.

event_relation_id: This 16-bit field designates event relation identifier of the sub_table when the referred node belongs to different event relation sub_table.

reference_node_id: This 16-bit field identifies the node to be referred to.

reference_number: This 8-bit field specifies the reference priority of the nodes to be referred to by the reference_node_id. The value "0xFF" may be used if there is no need to specify the priority order. "0x00" is not used.

5.2.4 Short node information descriptor

The short node information descriptor used in the event relation table (ERT) expresses the node name as well as the descriptions on the node definition in the textual format. The short node information descriptor used in the EIT expresses the node name and the description related to the node, of the node to be referred to by the event, in textual format. See table 5-11.

Table 5-11 Short node information descriptor

Syntax	No. of bits	Identifier
short_node_information_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
ISO_639_language_code	24	bslbf
node_name_length	8	uimsbf
for(i=0;i< node_name_length;i++){		
node_name_char	8	uimsbf
}		
text_length	8	uimsbf
for(i=0;i<text_length;i++){		
text_char	8	uimsbf
}		
}		

Semantics for the short node information descriptor:

ISO_639_language_code: This 24-bit field indicates the language of the subsequent character information field in a form of three alphabetical characters specified by ISO639-2[2]. Each character is encoded in eight bits in accordance with ISO8859-1[3] and inserted into the 24-bit field in the same order as that of the character code.

EXAMPLE: Japan has 3-character code "jpn", which is coded as:

"0110 1010 0111 0000 0110 1110"

node_name_length (Node name length): This 8-bit field indicates the byte length of the following node name.

node_name_char: This is an 8-bit field. The series of character information indicates the node name.

text_length: This 8-bit field indicates the byte length of the following node description.

text_char: This is an 8-bit field. The series of character information provide an explanation of the node.

5.2.5 STC reference descriptor

The STC reference descriptor describes the corresponding relation between the time information described in the LIT and the STC to enable precise synchronizing of the event component in program segment index. See table 5-12.

Table 5-12 STC reference descriptor

Syntax	No. of bits	Identifier
STC_reference_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
reserved_future_use	3	bslbf
external_event_flag	1	bslbf
STC_reference_mode	4	uimsbf
if(external_event_flag == 1){		
external_event_id	16	uimsbf
external_service_id	16	uimsbf
external_network_id	16	uimsbf
}		
if(STC_reference_mode == 0){		
}		
else if(STC_reference_mode == 1){		
reserved_future_use	7	bslbf
NPT_reference	33	uimsbf
reserved_future_use	7	bslbf
STC_reference	33	uimsbf
}		
else if(STC_reference_mode == 3 || STC_reference_mode == 5){		
time_reference	24	uimsbf
time_reference_extention	12	uimsbf
reserved_future_use	11	bslbf
STC_reference	33	uimsbf
}		
else{		
for(i=0; i<N; i++){		
reserved	8	bslbf
}		
}		
}		

Semantics for the STC reference descriptor:

external_event_flag: Set this field to "1" when the information of the STC reference descriptor is the reference information of the stream which is broadcasted as a different program from this program index.

external_event_id: This 16-bit field designates the event_id of the broadcasting program which the STC reference descriptor indicates.

external_service_id: This 16-bit field designates the service_id of the broadcasting program which the STC reference descriptor indicates

external_network_id: This 16-bit field designates the original _network_id of the broadcasting program which the STC reference descriptor indicates.

STC_reference_mode: This 4-bit field designates the reference type of the time in the STC reference descriptor. See table 5-13. Generally, the mode corresponding to the segmentation mode of the basic local event descriptor shall be used.

Table 5-13 STC reference mode

STC_reference_mode	Name	Semantics
0x0	Invalid	No relation is specified
0x1	NPT	Designate relation with NTP and STC
0x2	Undefined	Reserved for future use
0x3	Relative time	Relation between relative time from the start of the program (hour, minute, second, ms) and STC is designated.
0x4	Undefined	Reserved for future use
0x5	JST time	Relation between JST time (hour, minute, second, ms) and STC is designated.
0x6-0xF	Undefined	Reserved for future use

STC_reference: This 33-bit field indicates the STC value corresponding to the time designated with the NPT reference value or time reference value (extension) in 90kHz unit.

NPT_reference: This 33-bit field indicates the NPT expression time referring to the STC.

time_reference: This 24-bit field indicates unit of more than a second either the relative time in the expression of hour, minute, second and millisecond expression referring to the STC or the JST time. Using six 4-bit binary-coded decimal numbers (BCD), the time is coded in the order of hours, minutes and seconds.

time_reference_extension: This 12-bit field indicates units of less than a second, either the relative time in the expression of hour, minute, second and millisecond expression referring to the STC or the JST time. Using three 4-bit binary-coded decimal numbers (BCD), the time is coded in milliseconds. The value "0" is specified when no specification is made down to the millisecond level of accuracy.

5.2.6 Material information descriptor

The material information descriptor (see table 5-14) describes basic information for materials composing programs, which is link to related information on material type, material name, material code etc. Materials compose programs or parts of a program and the duration of the material is described in event or local event.

Table 5-14 Material information descriptor

Syntax	No. of bits	Identifier
material_information_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
descriptor_number	4	uimsbf
last_descriptor_number	4	uimsbf
number_of_material_set	8	uimsbf
for(i=0; i<N;i++) {		
material_type	8	uimsbf
material_name_length	8	uimsbf
for(j=0; j<N1;j++) {		
material_name_char	8	uimsbf
}		
material_code_type	8	uimsbf
material_code_length	8	uimsbf
for(j=0; j<N2;j++) {		
material_code_char	8	uimsbf
}		
material_period_flag	1	bslbf
reserved_future_use	7	bslbf
if (material_period_flag == 1) {		
material_period	24	uimsbf
}		
material_url_type	8	uimsbf
material_url_length	8	uimsbf
for(j=0; j<N3;j++) {		
material_url_char	8	uimsbf
}		
reserved_future_use_length	8	uimsbf
for(j=0; j<N4;j++){		
reserved_future_use	8	bslbf
}		
}		
}		

Semantics for the material information descriptor:

descriptor_number: This 4-bit field describes the number of a descriptor. This is used for relating to information which cannot be described in one descriptor. The first descriptor_number in a set of related material information descriptor is set to “0x0”. The descriptor_number is added by 1 every addition of a material information descriptor in this section.

last_descriptor_number: This 4-bit field indicates the number of the last material information descriptor (the descriptor with the maximum descriptor_number) in the set of related descriptor which includes this descriptor.

number_of_material_set: This 8-bit field indicates the number of succeeding loop of material information for describing basic information of every part of material in case that the material is composed of plural parts of material.

material_type: This 8-bit field indicates type of material and is coded according to table 5-15.

Table 5-15 Allocation of material_type

material_type	Table
0x00～0x7F	Reserved for future use
0x80～0xFF	Defined by service providers

material_name_length: This 8-bit field indicates byte length of material name.

material_name_char: This 8-bit field indicates character description of material name. Annex B is referred to regarding coding method of character information.

material_code_type: This 8-bit field indicates type of code for specifying material and is coded according to table 5-16.

Table 5-16 Allocation of material_code_type

material_code_type	Table
0x00	Out of use
0x01	IEC 62227
0x02~0x7F	Reserved for future use
0x80～0xFF	Defined by service providers

material_code_length: This 8-bit field indicates byte length of material code.

material_code_char: This 8-bit field indicates character description of material code. Annex B is referred to regarding coding method of character information.

material_period_flag: This 1-bit field indicates existence of succeeding material period in case that this is set to ‘1’. A material period is equal to a period of an event or local event including this descriptor in case that this is set to ‘0’.

material_period: This 24-bit field indicates unit more than second. This is coded in the order of hour, minute and second using 6 digits of BCD which is a 4-bit binary coded decimal coding system.

material_url_type: This 8-bit field indicates type of link information of entity data and URL for usage conditions on a material, and is coded according to table 5-17.

Table 5-17 Allocation of material_url_type

material_url_type	Table
0x00~0x7F	Reserved for future use
0x80～0xFF	Defined by companies

material_url_length: This 8-bit field indicates byte length of material link.

material_url_char: This 8-bit field indicates character description of material link. Annex B is referred to regarding coding method of character information.

reserved_future_use_length: This 8-bit field indicates byte length of succeeding reserved_future_use.

5.2.7 Allocation of the tag value and possible locations of the descriptors

Table 5-18 shows allocation of the possible locations of the descriptors.

Table 5-18 Allocation of the tag value and possible locations of the descriptors in the index encoding system

Descriptor	Tag value	EIT	LIT	ERT	ITT
Stuffing descriptor	0x42	o	o	o	o
Short event descriptor	0x4D	o	o		
Extended event descriptor	0x4E	o	o		
Material information descriptor	0x67		o		
Hyperlink descriptor	0xC5	o	o	o	
Basic local event descriptor	0xD0		o		
Reference descriptor	0xD1	o	o		
Node relation descriptor	0xD2			o	
Short node information descriptor	0xD3	o		o	
STC reference descriptor	0xD4				o

6. Program index transmission system

6.1 Transmission of program group index

Each table of program group index is transmitted by the same method as tables of EIT in basic information of SI (Part 2) and transmitted PID are specified directly.

When grouping the local event (program segment event) as the program group index, LIT is transmitted. The PID transmitting the LIT in this case is also specified directly.

Table 6-1 Table ID and PID used for program and program group index

Table	Table ID	PID used for transmission
EIT	0x4E-0x6F	0x0012
LIT	0xD0	0x0020
ERT	0xD1	0x0021

6.2 Transmission in program segment index

Each table of the program segment index is transmitted as one program component in ISO/IEC 13818-1[2] and transmitted PID is specified indirectly by the PMT. To identify that the program component is each table of the index in the PMT, the data component descriptor specified as basic information of SI (Part 2) is used.

The PID used to transmit each table section is shown in table 6-2.

Table 6-2 Table ID and PID used for index in program

Table	Table ID	PID used for transmission
LIT	0xD0	Indirect designation by PMT
ERT	0xD1	Indirect designation by PMT
ITT	0xD2	Indirect designation by PMT

6.3 Identifier used for transmission of program index

6.3.1 Stream type

The value of stream_type given to section signal transmitting program index "0x05", is used indicating private section in ISO/IEC 13818-1[4] as shown in table 6-3. The stream type is encoded in PMT, etc.

Table 6-3 Stream type

Value	Semantics
0x05	ISO/IEC 13818-1 private sections

6.3.2 Data component identifier

The standardization organization specifies the value of data coding method identifier (data_component_id) given to the transmission of program index. Data component identifier is encoded in data component descriptor, etc.

6.3.3 Service type

The value of service type given to the service added to the program index uses the value indicating main service, and encoded in accordance with table 6-4. For example, when index information is added to the digital TV service, "0x01" is used which indicates digital TV service, the main service.

When providing the program index as an independent service, "0xC0" is used as a value of service type.

Table 6-4 Service type

Value	Semantics
0x01	Digital TV service
0x02	Digital audio service
0xA1	Special video service
0xA2	Special audio service
0xA3	Special data service
0xA4	Engineering download service
0xA5	Promotion video service
0xA6	Promotion audio service
0xA7	Promotion data service
0xA8	Data service for accumulation beforehand
0xA9	Data service exclusive for accumulation
0xAA	Book mark list data service
0xC0	Data service

6.4 Descriptor used for program index transmission

The data component descriptor and data contents descriptor are used for transmission of the program index in accordance with the basic information of SI specified in part 2.

When transmitting the program segment index in other time (other event) or other program channel (other service) than the program body, or when providing the program segment index as an independent service, the hyperlink descriptor is used in accordance with the basic information of SI. Standard placement of these descriptors is shown in table 6-5.

Table 6-5 Placement of descriptor used for index transmission

Descriptor	Tag value	CAT	PMT (1st)	PMT (2nd)	NIT	BAT	SDT	EIT
Data component descriptor	0xFD			o				
Hyperlink descriptor	0xC5							o
Data contents descriptor	0xC7							o

Definitions of additional identification information of the data component descriptor and the program index transmission system for the selector area of the data contents descriptors are made in the following clauses.

6.4.1 Data component descriptor

Additional identification information area of the data component descriptor is used for the program index transmission, and the table identification information is shown in table 6-6.

Table 6-6 Table identifier information

Syntax	No. of bits	Identifier
table_identifier_info(){		
for(i=0; i<N; i++){		
table_id	8	uimsbf
}		
}		

Semantics definition of fields in the table identifier information:

table_id: This 8-bit field indicates the table_id of the table or sub_table transmitting in that component. When multiple tables are transmitting, multiple table_id can be specified.

6.4.2 Data content descriptor

When transmitting the program index, index transmission information such as table transmission status and size are described using selector area of the data content descriptor. Index transmission information is shown in table 6-7.

Table 6-7 Index transmission information

Syntax	No. of bits	Identifier
index_transmission_info(){		
start_time_offset	24	bslbf
end_time_offset	24	bslbf
version_updating_indicator	1	bslbf
interim_version_indicator	1	bslbf
reserved	6	bslbf
index_version	16	uimsbf
cycle_time	32	uimsbf
reserved	2	bslbf
leak_rate	22	uimsbf
table_size	32	uimsbf
}		

Semantics definition of fields in index transmission information:

start_time_offset: This 24-bit field specifies the offset time of the index information transmission when starting index information transmission preceding the event starting time. Using six 4-bit binary coded decimal numbers (BCD), the time is coded in the order of hours, minutes and seconds. When transmission is not made before the event, all bits in this field shall be set at "0". When transmission time before the event is not defined, all bits in this field shall be set at "1".

end_time_offset: This 24-bit field specifies duration of index information transmission when continuing index information transmission after the event end time. Using six 4-bit binary coded decimal numbers (BCD), the time is coded in the order of hours, minutes and seconds. When transmission is not made after the event, all bits in this field shall be set at "0". When transmission time after the event is not defined, all bits in this field shall be set at "1".

version_updating_indicator: This 1-bit flag indicates that the index information is updated within transmission time. When updating of the index information is not made in the event, this field is set to "0" and when updating is made, this field is set to "1".

interim_version_indicator: This 1-bit flag indicates that the index information is interim information. When the index of this event is interim information, that is, when broadcasting of updated information is scheduled in other event, this field is set to "1". When updated information other than the index of (final version) of the event is not broadcast, this field is set to "0".

index_version: This 16-bit field indicates the index information version (differing with the version number of the section). When the index information is updated in the event, it indicates the final version. When the version is not specified, all bits shall be set at "1".

cycle_time: This 32-bit field indicates the upper limit (the uppermost value) of the cycle which the sub_table is transmitted, in ms unit. When table transmission is made, this sub_table is completed when section of hours indicated here is gathered. It can be used as time out hour in the IRD. When cycle time is not specified, all bits shall be set at "1".

leak_rate: This 22-bit field indicates leak rate (size of data which should be taken out per unit time from transport buffer) of the sub_table. Unit shall be 50 byte/sec.

table_size: This 32-bit field indicates the upper limit (the uppermost value) of the sub_table in byte units. When multiple sub_tables are transmitted, it indicates the upper limit of the total. When size is not specified, all bits shall be set at "1".

Annex A (Normative)

Program index protection system

Program index protection system is specified herein, to suppress the use of the program segment index, which is against the service provider or program producer's will. This function is optional.

The LIT is protected beforehand and transmitted. The IRD store the LIT, which is protected. The protected program index information cannot be used in this condition, but when the program signal is decoded at the order of the service providers or at the program producers' will, the program index enables the information to work in the correct order and the program index information becomes available.

A.1 Protection of program index information

When transmitting the LIT, start time and continuation time of the local event is protected beforehand and then transmitted using the local event information section. Protection herein is made by the following methods:

(1) Value not defined is set

(2) Value with low accuracy including tolerance is set

A.2 Enable program index information

Protected program index information enable by overwriting new information using index enabling information. To enforce the program index protection function, ciphering is used in some cases when encoding index enabling information. When encoding the index enabling information, it should be specified otherwise in the service provider specification, etc. An example of index enabling information is shown in table A-1.

Table A-1 Index enabling information

Syntax	No. of bits	Identifier
index_enable_info(){		
local_event_id	16	uimsbf
enable_info_type	4	uimsbf
enable_info_priority	4	uimsbf
if(enable_info_type==1){		
start_time	24	uimsbf
duration	24	uimsbf
}		
if(enable_info_type==2){		
start_time	24	uimsbf
duration	24	uimsbf
start_time_extension	12	uimsbf
reserved_future_use	4	bslbf
duration_extension	12	uimsbf
reserved_future_use	4	bslbf
}		
}		

Semantic definition of fields in index enabling information:

local_event_id: This 16-bit field indicates the local event to operate the enabling information.

enable_info_type: Indicates information to protect and enable the index. This field classifies the syntax on and after the enabling information priority field.

Table A-2 Enabling information type

Value	Semantics
0x0	Reserved for future use
0x1	Time information (sec. unit)
0x2	Time information (ms unit)
0x3 – 0xF	Reserved for future use

enable_info_priority: Indicates priority when setting multiple enabling information to the same local event. On the IRD side, the enabling information is worked to the LIT when the value of the decoded enabling information priority is greater than the previous value of the decoded enabling information priority. (When the value of the decoded enabling information priority is not greater than the previous value of decoded enabling information priority, the decoded enabling information is cancelled.)

start_time : This 24-bit field specifies the value to overwrite as the local event start time of the LIT. Using six 4-bit binary-coded decimal numbers (BCD), the start time is coded in the order of hours, minutes and seconds.

duration: This 24-bit field specifies the value to overwrite as the local event duration of the LIT. Using six 4-bit binary-coded decimal numbers (BCD), the duration is coded in the order of hours, minutes and seconds.

start_time_extension: This 12-bit field specifies the value to overwrite as the local event start time extension of the LIT. Using three 4-bit binary-coded decimal numbers (BCD), the start time extension is coded in milliseconds.

duration_extension: This 12-bit field specifies the value to overwrite as the local event duration extension of the LIT. Using three 4-bit binary-coded decimal numbers (BCD), the duration extension is coded in milliseconds.

A.3 Transmission of index enabling information

Index enabling information is transmitted by either of the methods as shown below. The IRD is decoded in accordance with the program signal decoding.

The closer the layer to transmit index enabling information approaches the grade of the presentation layer, the stronger the index protection function becomes, generally. However, decoding process of the index enabling information becomes more complex accordingly. Transmission method of the index enabling information should be operated considering the balance of the strength of protection function and complexity of the decoding process.

(1) Transmission by section type

When transmitting the index enabling information using the section type, private descriptor of the service provider standard is placed using the ITT, or the private table of the service provider standard is used.

Though the protection function is not so strong, decoding process is the easiest (decode material for index method is available) and the transmission method does not depend on a service encoding method.

(2) Transmission by independent PES

When the index enabling information is transmitted using the data transmission method of independent PES, it should be in accordance with the transmission method of ARIB STD-B24 "Data Coding and Transmission Specification for Digital Broadcasting".

As the transmission method does not depend on the service encoding method and the strength of the protection function is almost the same as method (3), independent PES should be set for the index protection method.

(3) Transmission by PES header

When transmitting the index enabling information using the PES private data area of the PES packet header such as video and audio PES, it should be in accordance with ISO/IEC 13818-1[4].

The protection method is the strongest among those transmission methods, which do not depend on the service encoding methods, but index enabling information, which can be transmitted is limited to a maximum of 16 bytes.

(4) Transmission by video or audio PES

When transmitting the index enabling information using the data transmission method of video PES or audio PES, it should be in accordance with the transmission method of ARIB STD-B24 "Data Coding and Transmission Specification for Digital Broadcasting".

Though it offers the strongest protection, the transmission method depends on the service encoding method.

Annex B (Normative)

Coding of character

Characters and control codes used in extended information of SI are in accordance with the 8-unit character code specified in sub-clause 7.1, section 7, part 2 of Vol. 1 of the ARIB STD-B24 "Data Coding and Transmission Specification for Digital Broadcasting”. However, the details of the character set shall be specified in the operational guidelines of service providers.

Reference materials

[1] ARIB STD-B24 Ver. 5.9 "Data Coding and Transmission Specification for Digital Broadcasting" (Mar. 2014)

[2] ISO 639-2 (1998) "Codes for the representation of names of languages - Part 2: Alpha-3 code"

[3] ISO 8859-1 (1998) "Information processing - 8-bit single-byte coded graphic character sets - Part 1: Latin alphabet No.1"

[4] Rec. ITU-T H.222.0|ISO/IEC 13818-1(06/2012) "Information Technology – Generic Coding of Moving Pictures and Associated Audio: Systems"

Appendix

GUIDELINE FOR THE OPERATION METHOD OF SI (SERVICE INFORMATION)

Appendix
GUIDELINE FOR THE OPERATION METHOD OF SI (SERVICE INFORMATION)

CONTENTS

<BLANK>

Preface

This appendix is established as a guideline of SI specified in Part 1 and Part 2 for various attentions and conditions of transmission in actual operation, and is not a part of the standard.

1. How to use SI table

This chapter contains some guidelines on the usage of the Service Information (SI) table.

1.1 Network Information Table (NIT) information

The Network Information Table (NIT) provides a grouping of Transport Streams (TSs) and the relevant tuning information. The NIT could be used during set-up procedures of the IRD and the relevant tuning information may be stored in no-volatile memory. The NIT also could be used to signal changes of tuning information. The following rules apply to the NIT:

a) transmission of the NIT is mandatory for the actual delivery system;

b) the NIT describing the actual delivery system is valid if and only if it contains applicable delivery system descriptors for the actual delivery system. This rule specifies the conditions under which the NIT contains valid information. At some transitions of broadcast delivery system boundaries, the NIT carried in a TS is allowed to describe an earlier network in the broadcast chain. A different mechanism has to be selected by the IRD to obtain the relevant tuning information for the actual delivery system. If a satellite IRD receives a satellite delivery system descriptor for the actual delivery system, then it is valid. If a cable IRD receives a satellite delivery system descriptor for the actual delivery system, then it is valid. If a cable IRD receives a satellite delivery system descriptor for the actual delivery system, then it is assumed to be invalid for the cable IRD;

c) if a valid NIT for the actual delivery system is present in the SI bit stream then it shall lost all TSs of the actual delivery system;

d) the SI stream shall have at least 8 TS packets per 10 seconds carrying NIT data or NULL packets. This rule simplifies the replacement of the NIT at broadcast delivery system boundaries. With the simple replacement mechanism, local frequency control is possible with relatively low cost equipment.

The SI uses two labels related to the concept of a delivery system, namely the network_id and the original_network_id. The latter is intended to support the unique identification of a service, contained in a TS, even if that TS has been transferred to another delivery system than the delivery system where it originated. A TS can be uniquely referenced through the path original_network_id/transport_stream_id. A service can be uniquely referenced through the path original_network_id/transport_stream_id/service_id. The network_id, thus, is not part of this path. In addition each service_id shall be unique within each original_network_id. When a service (contained inside a TS) is transferred to another delivery system, only the network_id changes, whereas the original_network_id remains unaffected.

Figure 1-1 shows an example, where two services (A and B), which originate in two different delivery systems and happen to have the same service_ids and transport_stream_ids, are transferred to a new delivery system.

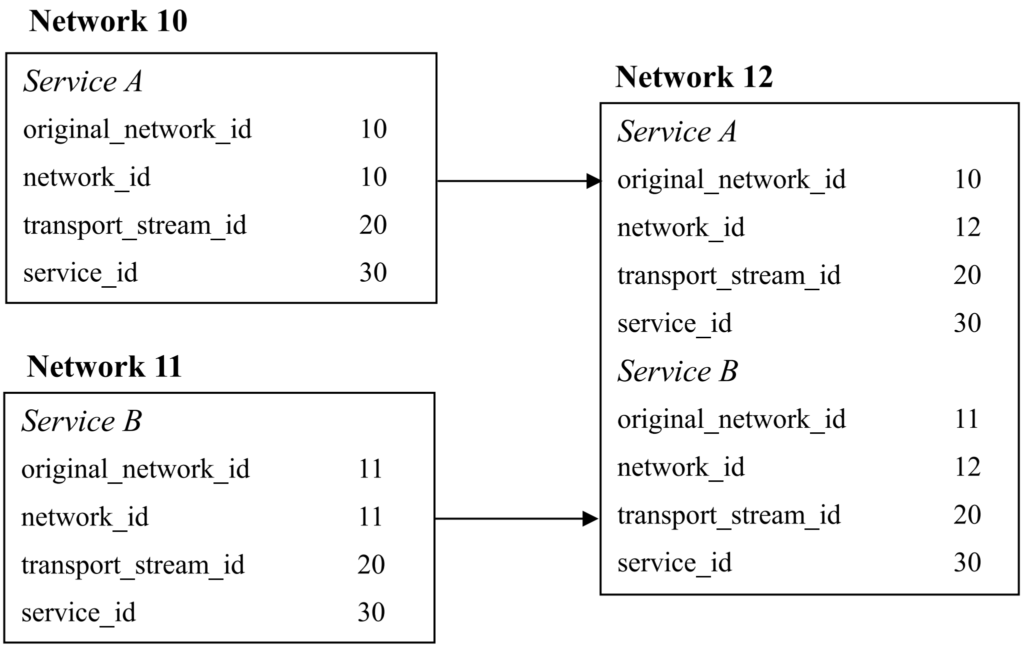


Figure 1-1 Transfer to a new delivery system

1.2 Bouquet Association Table (BAT) information

The BAT provides a grouping of services which serves as one basis on which an IRD presents the available services to a user. Transmission of the BAT is optional. The following rule improves the consistency in the SI bit streams and simplifies the processing in the IRDs.

The SI bit stream shall list in each BAT sub-table all the services belonging to that bouquet.

One service may belong to more than one bouquet. This rule creates consistency across the different TSs which are accessible to the IRD.

If it is intended for the IRD to present service information to the user grouped in bouquets, then it would be beneficial to ensure that every service is listed in one or more bouquets, or some services will be omitted from this method of presentation. A bouquet may group together services from more than one TS, which could even be carried in different networks. The IRD's acess to information on all the services of a bouquet would be facilitated if all the service referred to in the BAT were listed in the Service Description Table (SDT). Similarly, the IRD's access to these services is facilitated if NIT information is given for all TSs in which services of the bouquet occupy capacity.

1.3 Service Description Table (SDT) information

The SDT is used to list the names and other parameters of the services within TSs. For each TS a separate SDT sub-trable exists. The following rules apply in oreder to improve the acquisition of services:

- the transmission of the SDT for the actual TS is mandatory;
- the SI bit stream shall list in the SDT of a particular TS at least all the services of that TS.

In addition:

- any SDT for another TS than the actual one (i.e. with table_id = 0x46) shall list all the services of that TS;
- it is strongly recommended that service_ids, once assigned to a specific service within a network, remain unchanged in order to enable IRDs to implement features like favourite channel lists, etc.

1.4 Event Information Table (EIT) information

The Event Information Table (EIT) is used to transmit information about present, following and further future events. For each service a separate EIT sub-table exists.

1.4.1 EIT Present/Following information

The following rule simplifies the acquisition of the EIT Present/Following information. The SI specification states that an EIT section has a maximum size of 4096 bytes.

The SI bit stream shall have two sections per service for an EIT Present/Following with the section_number 0x00 reserved for the description of the present event and section_number 0x01 for the following event. These constraints do not apply in the case of an NVOD reference service which may have more than one event description of the EIT Present/Following. The event after the following event can be implied optionally, using the section_number 0x02 and after.

The SI bit stream shall have a maximum of 4096 bytes to describe a single event in a section.

The organization of the EIT Present/Following is based on the concept of present and following events. Which event is the present one can be determined using the following scheme:

a) at each instant in time, there is at most one present event;

b) when there is a present event, this event shall be described in section 0 of the EIT Present/Following;

c) when there is no present event (e.g. in the case of a gap in the schedule) an empty section 0 of the EIT Present/Following;

d) the running_status field in the description of the present event shall be given the interpretation in table 1-1.

Table 1-1 running_status of the present event

undefined	No information except the nominal status is provided. IRDs and VCRs shall treat the present event as running.
running	IRDs and VCRs shall treat the present event as running.
not running	IRDs and VCRs shall treat the present event as not running. In other words, this event is nominally the present one, but at this time has either not started or has already ended.
pausing	IRDs and VCRs shall treat the present event as pausing. In other words, this event is nominally the present one and has already started, but at this time the material being broadcast is not a part of the event itself.
starts in a few seconds	IRDs and VCRs shall prepare for the change of event status to "running" in a few seconds.

The duration of an event as encoded in the EIT shall also include the duration of all times when the event has the status "not running" or "paused". The start time of an event as encoded in the field start_time of the EIT shall be the start time of the entire event, i.e. not the start time after the pause has finished;

e) at each point in time, there shall be at most one following event:

f) if a following event exists, it shall be described in section 1 of the EIT Present/Following:

g) if no following event exists, an empty section 1 of the EIT Present/Following shall be transmitted;

h) the running_status field in the definition of the following event shall be given the following interpretation of table 1-2:

Table 1-2 running_status of the following event

undefined	No information except the nominal status is provided. IRDs and VCRs shall treat the following event as not running.
running	Not allowed.
not running	IRDs and VCRs shall treat the present event as not running.
pausing	This status is intended to indicate that the "following" event has been running at some time, but is now overlapped by another event. In such a case, during the whole time that the "following event has status "pausing", one and the same overlapping event shall be encoded in section 0 of the EIT Present/Following. Furthermore, an event which has the status "pausing" shall acquire the status "running" at a later time, then replacing the overlapping event in section 0 of the EIT Present/Following.
starts in a few seconds	IRDs and VCRs shall prepare for the status of the following event to change to running within a few seconds.

The duration of an event as encoded in the EIT shall also include the duration of all times when the event has the status "not running" or "paused". The start time of an event as encoded in the field start_time of the EIT shall be the start time of the entire event, i.e. not the start time after the pause has finished.

The start time of one event plus its duration may be smaller than the start time of the following event. In other words, gaps between events are allowed. In such a case, the following event is considered to be the event scheduled to begin after the gap. This event shall be encoded in section 1 of the EIT Present/Following. The start time and duration are scheduled times. Some broadcasts may update this information if the schedule is running late, whereas others may prefer to keep the indicated start time unchanged, e.g. to avoid having an event called "The News at 8" from being indicated as starting at 8:01:23, instead of 8:00:00.

1.4.2 EIT Schedule information

1.4.2.1 EIT Schedule structure

The EIT Schedule information is structured in such a way that it is easy to access the EIT data in a flexible manner. The EIT Schedule Tables shall obey the following rules:

a) the EIT/Schedule is distributed over 16 table_ids, being 0x50 – 0x5F for the actual TS, and 0x60 – 0x6F for other TSs, which are ordered chronologically;

b) the 256 sections under each sub-table are divided into 32 segments of 8 sections each.

Segment #0, thus, comprises sections 0 to 7, segment #1 section 8 to 15 etc.;

c) each segment contains information about events that start anywhere within a three-hour period;

d) the information about separate events is ordered chronologically within segments;

e) if only n < 8 sections of a segment are used, the information shall be placed in the first n sections of the segment. To signal that the last sections of the segment are not used, the value s0 + n -1, where s0 is the first section number of the segment, shall be encoded in the field segment_last_section_number of the EIT header. As an example, if segment 2 contains only 2 sections, the field segment_last_section_number shall contain the value 8 + 2 – 1 = 9 in those two sections;

f) segments that contain all their sections shall have the value s0 + 7 encoded in the field segment_last_section_number;

g) entirely empty segments shall be represented by an empty section, (i.e. a section which does not contain any loop over events) with the value s0 + 0 encoded in the field segment_last_sectio

h) the placing of events in segments is done referring to a time t0. t0 is "last midnight" in Japan Standard Time (JTC)

j) there are the following two methods of placing event information in segments:

1) segment #0 of table_id 0x50 (0x60 for other TSs) shall contain information about events that start between midnight and 02:59:59 of "today". Segment #1 shall contain events that start between 03:00:00 and 05:59:59, and so on. This means that the first sub_table (table_id 0x50, or 0x60 for other TSs) contains information about the first four days of the schedule, starting today at midnight.

2) segment #0 of table_id 0x50 (0x60 for other TSs) shall contain information about events that start between midnight and 02:59:59 of the first day in every month. Segment #1 shall contain events that start between 03:00:00 and 05:59:59, and so on. This means that the first sub_table (table_id 0x50, or 0x60 for other TSs) contains information about the first four days of schedule, starting the first day of every month at midnight.

k) the field last_section_number is used to indicate the end of the sub-table. Empty segments that fall outside the section range indicated by last_section_number shall not be represented by empty sections;

l) the field last_table_id is used to indicate the end of the entire EIT/Schedule structure. Empty segments that fall outside the table_id range indicated by last_table_id shall not be represented by empty sections;

m) segments that correspond to events in the past may be replaced by empty segments (see rule g));

n) the running_status field of event definitions contained in the EIT/Schedule shall be set to undefined (0x00).

1.4.2.2 EIT scrambling

The EIT Schedule Tables may be scrambled. In order to provide an association with the Conditional Access (CA) streams, it is necessary to allocate a service_id (= MPEG-2 program_number) which is used in the Program Specific Information (PSI) to describe scrambled EIT Schedule Tables. The EIT is identified in the Program Map Table (PMT) section for this service_id as a program consisting of one private stream, and this PMT section includes one or more CA_descriptors to identify the associated CA streams. The service_id value 0xFFFF is reserved for this purpose.

1.5 Time and Date Table (TDT)

The Time and Date Table (TDT) transmits the actual JTC-time coded as Modified Julian Date (MJD). It may be used to synchronize the internal clock of an IRD. The TDT shall be transmitted at least every 30 seconds. The encoded time is intended to be valid when the section becomes valid according to figure 1-2 of this standard.

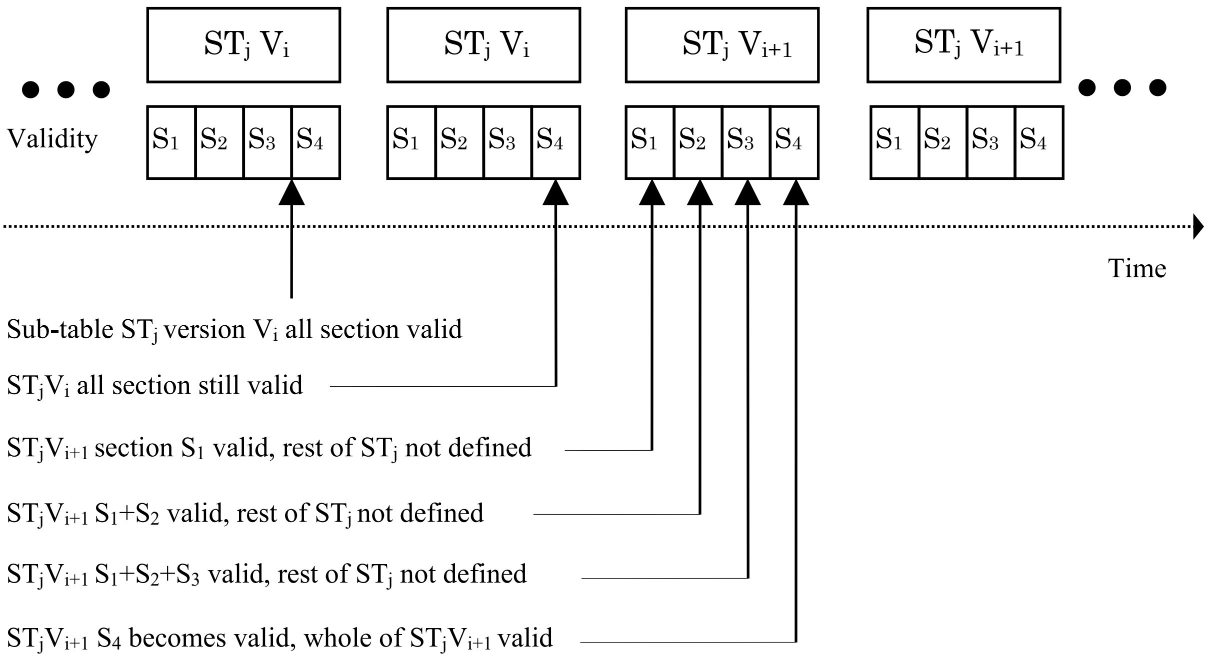


Figure 1-2 Timing of table updates and validity

1.6 Time and Data Offset Table (TOT)

The TOT transmits the time and data offset information coded as MJD and JTC (Note 1) in place of TDT. That is, either the TDT or the TOT shall be transmitted. It may be used to synchronize the internal clock of an IRD. By mapping the local time offset descriptor in TOT at local time, partial content time of the actual time (UTC+9) and indicated time to human can be transmitted. The TOT shall be transmitted at least every 30 seconds. Updating of table and timing of validity is operated in accordance with the TDT.

(Note 1) Whether the encoded time should be the transmitted time or the received time shall be specified in the operational guideline.

1.7 Running Status Table (RST)

Running status sections are used to rapidly update the running status of one or more events. Running status sections are sent out only once, at the time the status of an event changes, unlike other SI Tables which are normally repetitively transmitted. Thus there does not exist any update mechanism for RSTs. At the moment an RST is transmitted to update the running status of an event, it invalidates the running status of that event, transmitted previously by the EIT Present/Following. The following time the EIT is transmitted, it shall contain the updated running status bits.

The intended use of this optional mechanism is to enable IRDs or VCRs to implement highly accurate switching to the beginning of events by setting up a filter on Running Status Tables and waiting for the occurrence of the RST section containing the event.

1.8 Stuffing Table (ST)

A stuffing section may occur in anywhere that a section belonging to an SI Table is allowed. Stuffing Tables may be used to replace or invalidate either sub-tables or complete SI Tables. In order to guarantee consistency, all sections of a sub-table shall be stuffed. It is not allowed to replace some sections of a sub-table by stuffing some sections while keeping others.

1.9 Partial Contents Announcement Table (PCAT)

The partial contents announcement table is used to announce the schedule of partial contents to update a part of information in the specific data broadcasting contents accumulated in the IRD, etc.

Rules to maintain consistency to realize partial contents to accumulated data broadcasting contents are as follows.

- Total contents are broadcast as normal data broadcasting program that is an event. Partial contents are broadcast with the same service as total contents.
- Partial contents should always announce the partial content depending on the total contents and do not depend on the prior partial contents. For example, when it is announced in the order of:

  Total content → Partial content (A) → Partial content (B),

  the Partial content (B) does not depend on the Partial content (A).
- Version of the contents is controlled by the total announcement version (contents version) and version of the partial announcement (contents minor version) depending on it.
- A field for the content identifier (content_id) and the contents version (content_version) is operated in the selector area of the EIT data contents descriptor at the time of total announcement, for data component expressing contents which can be accumulated.
- When intending to update by overwriting a partial or total contents on an accumulated content, their content identifier should have the same value consistently.

Example: The figure below indicates the relation of version of the total contents announcement and the partial content announcement, and version of the accumulated contents gained by the result of those receptions.

	Total announcement	→	Partial content announcement	→	Partial content announcement	→	Total announcement
content_version content_minor_version	1 -		(1) 1		(1) 2		2 -
Version of accumulated content	1.0	→	1.1	→	1.2	→	2.0

Total announcement contents version at the first time is 1. In the partial content announcement following it, the partial contents of contents minor version 1 is announced having the content version 1 as a target. Then the partial content of contents minor version 2 is announced having the content version 1 as target. And then, content of the content version 2 is announced in the second total announcement at the last.

1.10 Broadcaster Information Table (BIT)

The broadcaster information table provides combination of the broadcaster existing on the original network and the relating SI transmission parameter information. The BIT can be used to know in what cycle/span the SI table including NIT is transmitted in the IRD. The BIT is applied with the following rules.

- The BIT constructs a sub_table in each original network.
- In the sub_table, a descriptor loop exists in each original network and broadcaster. In the original network descriptor area (the first descriptor area), the SI transmission parameter operated commonly in the original network is denoted. In the broadcaster descriptor area (the second descriptor area), information for each broadcaster is denoted. The information for each broadcaster means the broadcaster name, a service list provided by the broadcaster and the SI transmission parameter operated in each broadcaster.

The broadcaster name is indicated in the broadcaster name descriptor. When the broadcaster view propriety is "1", it can be used to realize functions of the IRD to indicate or select a program list for each broadcaster.

The service list for each broadcaster can be used to know the searching area of the series identification.

1.11 Network Board Information Table (NBIT)

The NBIT is a table to provide board information in the network.

The board information itself and reference information to acquire the board information can be provided by dividing the table id.

The board information is provided as noticing information to viewers. By including service id and genre codes, the IRD can give indication including service id and genre icons at the beginning of the message.

The NBIT is applied under the following rules

- The NBIT constructs a sub_table in each original network.
- Information identifier is given to information, each transmitted as board information. When certain information is transmitted and the content of the information is changed, other information identifier is given.
- The information type with service or genre information to the board information is provided with a key identifier.
- The content body of the actual board information is indicated by placing the board information descriptor in the descriptor area.

1.12 Linkage Description Table (LDT)

The LDT is provided with collecting various descriptions referred from other tables. The LDT linkage descriptor is placed to other table to indicate linkage to the LDT.

The LDT is applied under the following rules.

- The LDT constructs a sub_table in each group to collect descriptions such as service id of the representative service, etc.
- In case of linkage from other tables, the descriptor identifier indicated in the LDT descriptor and descriptor type are given as information when linked. The value of the descriptor identifier and the descriptor indicated in the descriptor type are indicated in the descriptor area in the corresponding table.
- One event may be linked to multiple collecting groups.

1.13 Address Map Table (AMT)

The AMT provides a list of multicast groups in order that an application possibly receives IP packets transmitted by TS packets and ones by communication channels without distinction.

The AMT supports ASM (Any Source Multicast) indicating a multicast group only with a group address and SSM (Source Specific Multicast) indicating a multicast group with combination of a source address and a group address.

An address mask is used for describing continuous IP addresses effectively. It is possible to describe a list of plural IP addresses with a single line shown in Figure 1-3. The mask=32 in IPv4 and the mask=128 in IPv6 are equal to not using an address mask.

Service A	234.0.1.0
Service A	234.0.1.1
	……
Service A	234.0.1.255
Service B	234.0.2.0
Service B	234.0.2.1
	……
Service B	234.0.2.255

(a) Without address mask

Service A	234.0.1.0,	mask=24
Service B	234.0.2.0,	mask=24

(b)With address mask

Figure 1-3 Example of effectively describing IP address using address mask

1.13.1 Channel selection with multicast group

It allows to select a channel with an IP multicast group using AMT.

IGMP or MLD which are the standard protocols of IP multicast communication is used for delivery request and delivery stop request of IP packets from the applicastion to the receiver with IP packet demultiplexing function. The multicast group is specified to be demultiplexed in a receiver by using a join message at the start of receiving and a leave message at the end of receiving.

When an application is implemented in the receiver without network, it is possible to select a channel without IGMP and MLD etc. for delivery request and delivery stop request.

[Note] IGMP：Internet Group Management Protocol, MLD：Multicast Listener Discovery

These protocols are used in case that a terminal requests to deliver and to stop delivering IP packets.

IGMPv2 (RFC 2236) : Used by IPv4 , corresponding to Any Source Multicast

IGMPv3 (RFC 3376) : Used by IPv4, corresponding to Source Specific Multicast

MLDv1 (RFC 2710) : Used by IPv6, corresponding to Any Source Multicast

MLDv2 (RFC 3810) : Used by IPv6, corresponding to Source Specific Multicast

IGMPv3 maintains upper compatibility to IGMPv2, and MLDv2 maintains upper compatibility to MLDv1.

1.14 IP/MAC Notification Table (INT)

The IP/MAC notification table (INT) provides information relating broadcasting programs to IP/MAC streams composing them. The information and relation on IP address, TS identification, service identification and the like is described in INT. The selection of MPEG-2 TS according to TS identification, service identification in NIT enables to extract streams of target IP address.

The example of service selection using INT is shown in Figure 1-4.

(1) The service identification (service_id) of the service, in which NIT is delivered, is taken from a linkage descriptor (linkage_descriptor()) located in NIT according to NIT (PID=0x0010).

(2) The PMT PID=BBB with the same program number (program_number) as the service identification (service_id) of INT is taken according to PAT (PID=0x0000).

(3) The elementary PID (elementary_PID=NNN) of INT is taken with the PID of the ES loop, in which the data broadcast id descriptor (data_broadcast_id_descriptor(), data_broadcast_id=0x000B) is located, according to PMT (PID=BBB).

(4) The TS identification (transport_stream_id) and the service identification (service_id) are taken with searching the target IP address (IP ADDR) according to INT (PID=NNN).

(5) The PMT PID=CCC with the same program number (program_number) as the service identification (service_id) of the target is taken according to PAT (PID=0x0000).

(6) The elementary PID (elementary_PID=NNN) delivering the target IP address is taken according to PMT (PID=CCC).

(7) The TS packet corresponding to the elementary PID (elementary_PID=NNN) is selected.

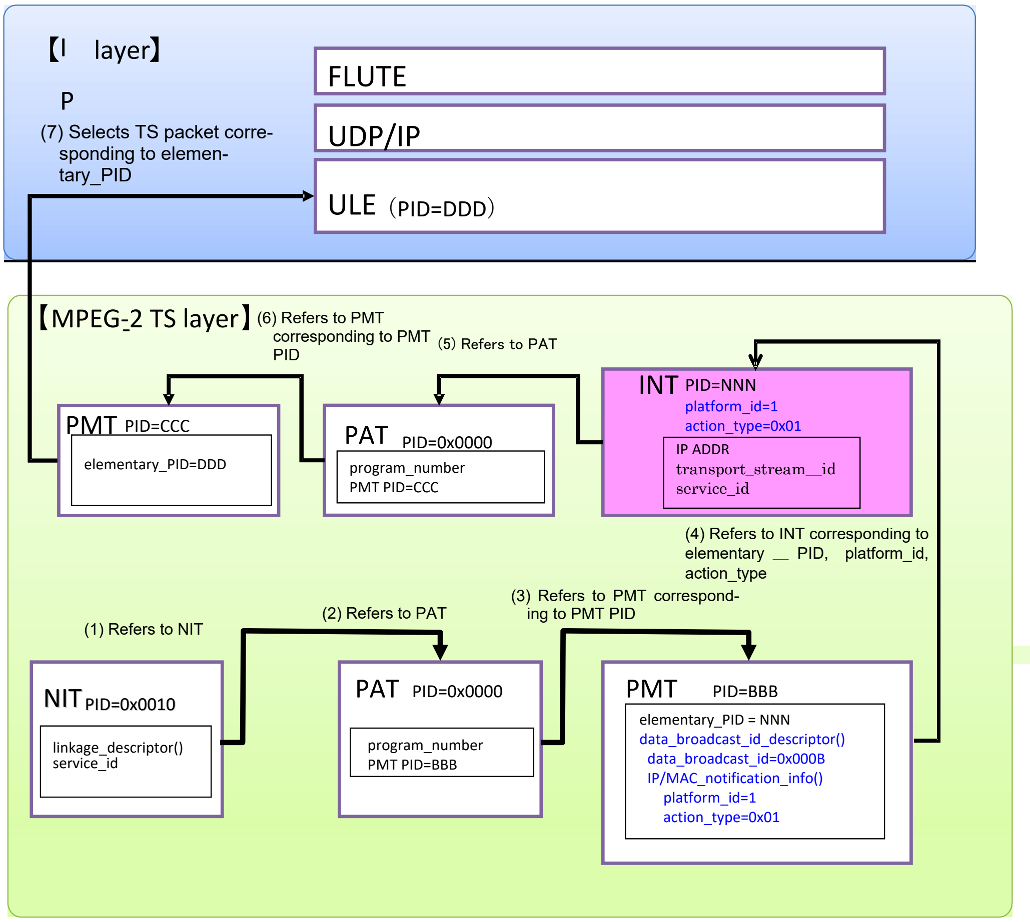


Figure 1-4 Example of service selection using INT

1.15 Table Updating Mechanism

The Section syntax used for SI has various mechanisms to indicate the updating of the SI contents.

The updating of the section is indicated by the increase of the version number field. Updating becomes effective immediately after the final byte of the CRC32 of the new version section. Therefore, the value of the current next indication shall always be "1". The section with the current next indication "0" shall not be transmitted.

2. SI descriptor allocation and usage

This chapter specifies the location where descriptors can be expected in a SI bit stream, and identifies which descriptors may occur multiple times. Descriptors which contain fundamental SI data are identified as recommended to be decoded by the IRD. The interpretation of other descriptors by the IRD is optional.

2.1 Descriptors of the Network Information Table (NIT)

The NIT is organized as shown in table 5-3, part 2 of this standard.

2.1.1 First descriptor loop

In the first loop of the NIT, the SI descriptors in this sub-clause are defined in addition to those defined in the Ministerial Ordinances and Notifications.

2.1.1.1 Linkage descriptor

This descriptor is used to give a link to another service or TS. If it appears in this loop it links to a service that is attached to the network operator. This descriptor is allowed more than once in this loop. It could, for example, point to the "123 Info channel" and to "123 Text". Transmission of this descriptor is optional. The meaning of the descriptor when it occurs here depends on the value of the linkage_type. If the linkage_type is:

a) 0x01, it refers to a service that contains information about the network. An example of the intended use is for the IRD to switch to the information service when the user requests additional information about the network;

b) 0x02, it refers to an Electronic Program Guide (EPG) for the network. Note that the IRD can only make use of this type of linkage if it can decode the EPG service. This standard does not specify the contents of such a service;

c) 0x04, it refers to a TS which carries comprehensive SI. The SI carried in the referenced TS includes at least all the SI information available on all other TSs of the network.

d) 0x0B, it refers to a TS of the service with the IP/MAC notification table (INT).

The meanings of other values of linkage_type are not defined in this context. Note that the linkage_type does not indicate the service_type of the referenced service. An example of the intended use of the linkage descriptor is that an IRD user interface could include a mechanism like "info about the network" which would make the IRD tune to the linked service after the user initiated the mechanism.

2.1.1.2 Network name descriptor

This descriptor is used to transmit the name of a physical network, e.g. "JCSAT-3", "SUPERBIRD-C" etc. This descriptor shall be used exactly once in any NIT sub-table.

2.1.2 Second descriptor loop

In the second loop of the NIT, the SI descriptors in this sub-clause are defined in addition to those defined in the Ministerial Ordinances and Notifications.

2.1.2.1 Delivery system descriptor

The delivery system descriptors are used to transmit the physical parameters for each transport multiplex in the network.

One (and only one) delivery system descriptor shall appear in each loop. IRDs shall be able to interpret the delivery system descriptor in order to tune to TSs quickly (see sub-clauses 1.1 and 4.3.1).

2.1.2.2 Service list descriptor

This descriptor is used to list the services and service_types for each TS. The services are listed identified by service_id (= MPEG-2 program_number). The transport_stream_id and original_network_id, which are necessary to identify a service uniquely, are given at the start of the descriptor loop.

The service list descriptor is allowed only once in each loop. Transmission of this descriptor is optional, but if it is present, then the service list shall be complete.

2.1.2.3 Emergency information descriptor

This descriptor is transmitted when the emergency warning broadcasting is made and includes information and function necessary for the emergency warning signal which is transmitted as an audio signal formerly.

2.1.2.4 Partial reception descriptor

This descriptor is used to indicate which service id can receive by the narrow band IRD, which receives only partial reception hierarchy when there is a service which is transmitted only in the partial reception hierarchy in digital terrestrial television broadcasting system. When there is a service transmitted in conditional access hierarchy, this transmission is mandatory.

2.1.2.5 Connected transmission descriptor

This descriptor is used to identify which connected transmission group transmits the TS when the connected transmission is made to transmit multiple segments (1-segment type, 3-segment type or 1-segment type) is made without a guard band. By using the segment information (segment form and modulation system type) by which each TS is transmitted and the terrestrial delivery system descriptor in the IRD, quick tuning to TS can be realized. When the TS connected transmission is made, transmission of this descriptor is mandatory.

2.1.2.6 TS information descriptor

This descriptor indicates, for digital terrestrial television broadcasting, the remote control key identifier to which the applicable TS should be allocated during scan operation in the initial setting of the receiver. Such operation includes grasping of receivable network identifiers, transport stream identifiers, and receivable frequency information. This descriptor also indicates the relationship between the service identifier and the transmission layer.

2.1.2.7 Service group descriptor

This descriptor provides a list of service relationships when interrelated services are provided in a Transport Stream. In the case of server-type broadcasting, a list of server-type broadcasting services operated simultaneously with the conventional broadcast is provided.

2.1.2.8 Area broadcasting information descriptor

This descriptor provides the identification, signal format and related information of stations in area limited broadcasting. This is transmitted with a terrestrial delivery system descriptor.

2.2 Bouquet association table descriptor

Composition of the BAT is indicated in table 5-4 of Part 2 in this standard. The BAT has the same structure as the NIT. The BAT gives a logical grouping of services into bouquets, which may group together services delivered by different networks. A TS may contain services from more than one bouquet within a network. Each BAT collects the services that are allocated to the specified bouquet.

2.2.1 First descriptor loop

The SI descriptors in this sub-clause have a defined meaning in the first loop of the BAT.

2.2.1.1 Bouquet name descriptor

This descriptor is used to transmit the name of the bouquet the following services are allocated to. This descriptor is allowed once in each sub-table of the BAT. It is mandatory to be transmitted in any BAT sub-table in the TS.

2.2.1.2 CA identifier descriptor

Transmission of this descriptor is optional; it is allowed to only once in this loop. It identifies one or more CA system which apply to the services in the BAT.

2.2.1.3 Country availability descriptor

This descriptor is used to indicate whether a bouquet is available in a specific country. It has no meaning in the sense of CA. However, it may be a good feature for IRDs to interpret this descriptor, not to display bouquets that are not available in order to avoid frustration of the user.

This descriptor is allowed a maximum of twice in each BAT sub-table, once to indicate a list of countries in which the bouquet is intended to be available, and once to indicate those countries in which it is not intended to be available. If the descriptor is not present, the availability status of the bouquet is undefined. Transmission of this descriptor is optional.

2.2.1.4 Linkage descriptor

This descriptor is used to give a link to another service or TS. If it appears in this loop it links to a service that is attached to the bouquet provider. The linkage_descriptor is allowed more than once in this loop. It could, for example, point to the "123 Info channel" and to "123 Text". Transmission of this descriptor is optional. The meaning of the descriptor when it occurs here depends on the value of the linkage_type. If the linkage_type is:

a) 0x01, it refers to a service that contains information about the bouquet. An example of the intended use is for the IRD to switch to the information service when the user requests additional information about the bouquet;

b) 0x02, it refers to an Electronic Program Guide (EPG) for the bouquet. Note that the IRD can only make use of this type of linkage if it can decode the EPG service. This standard does not specify the contents of such a service:

c) 0x04, it refers to a TS which carries comprehensive SI. The SI carried in the referenced TS includes at least all the SI information available on all other TSs which carry services of the bouquet.

The meanings of other values of linkage_type are not defined in this context. Note that the linkage_type does not indicate the service_type of the referenced service. An example of the intended use of the linkage descriptor is that an IRD user interface could include a mechanism like "info about the bouquet" which would make the IRD tune to the linked service after the user initiated the mechanism.

2.2.2 Second descriptor loop

The SI descriptors in this sub-clause have a defined meaning in the second loop of the BAT.

2.2.2.1 Service list descriptor

This descriptor is used to list the services and service types of each TS that belong to the bouquet of this section. This allows to find all services that belong to a specific bouquet.

The service_list_descriptor is allowed only once in each loop. It should be transmitted if a BAT exists.

2.3 Service description table descriptor

The structure of the SDT is indicated in the table 5-5, part 2 of this standard. In the SDT, there is one loop for the descriptors for each service described in the SDT. The SI descriptor in this sub-clause has a defined meaning in the loop.

2.3.1 Bouquet name descriptor

This descriptor is used to transmit the name of the bouquet the service is allocated to. This descriptor is allowed more than once in the loop because a service could belong to more than one bouquet. Transmission of this descriptor is optional in the SDT. The use of this descriptor in the SDT is wasteful of bandwidth, since the information can be conveyed more efficiently using the BAT.

2.3.2 CA identifier descriptor

If a service is generally CA protected, this descriptor may be used to transmit data of the CA-system. The CA_identifier_descriptor is not involved in any CA control function, it is an indication for the user interface software in the IRD that a service is under conditional access and which CA-system is used. Then the user interface software may decide whether this service is reachable or not. The aim of the transmission of this descriptor is to avoid frustration to users caused by services being displayed for selection that are not reachable. This descriptor is allowed only once in the loop. Transmission of this descriptor is optional.

2.3.3. Country availability descriptor

This descriptor is used to indicate whether a service is available in the specified country. It has no meaning in the sense of CA, however, it may be a good feature for IRDs to interpret this descriptor, not to display services that are not available in order to avoid frustration of the user.
This descriptor is allowed a maximum of twice in each SDT service loop, once to indicate a list of countries in which the service is intended to be available, and once to indicate those countries in which it is not intended to be available. If the descriptor is not present, the availability status of the service is undefined. It is not allowed if there is a time_shifted_service_descriptor. Transmission of this descriptor is optional.

2.3.4 Linkage descriptor

This descriptor is used to give a link to another service. If it appears in this loop it links to a service that is attached to this service. This descriptor is allowed more than once in this loop. Transmission of this descriptor is optional. The meaning of the descriptor when it occurs here depends on the value of the linkage_type. If the linkage_type is:

a) 0x01, it refers to a service that contains information about this service. An example of the intended use is for the IRD to switch to the information service when the user requests additional information about this service;

b) 0x02, it refers to an Electronic Program Guide (EPG) for this service. Note that the IRD can only make use of this type of linkage if it can decode the EPG service. This standard does not specify the contents of such a service;

c) 0x03, it refers to a CA replacement service for this service. An example of the intended use is for the IRD to switch automatically to the replacement service if the CA system denies access to this service.

d) 0x05, it refers to a replacement service for this service. An example of the intended use is for the IRD to switch automatically to this replacement service when the selected service has a running status of "not running".

The linkage_type does not indicate the service_type of the reference service. An example of the intended use of the linkage descriptor is that an IRD user interface could include a mechanism like "info about the service" which would make the IRD tune to the linked service after the user initiated the mechanism.

2.3.5 Mosaic descriptor

This descriptor may be located in the SDT and/or PMT. It is used to describe mosaic services described in sub-clause 4.2.

2.3.6 NVOD reference descriptor

This descriptor lists the services which belong to a Near Video On Demand (NVOD) service. A description of the NVOD-mechanism is given in sub-clause 4.1.

The NVOD_reference_descriptor is allowed only once in each loop and if there is no time_shifted_service_descriptor in it. It is mandatory to be transmitted if the corresponding services are described using the time_shifted_service_descriptor.

IRDs are recommended to make use of the NVOD_reference_descriptor in order to allow access to NVOD_services.

2.3.7 Service descriptor

This descriptor contains the basic textual identifications of a service such as service name and provider name. The service_descriptor is allowed only once in each loop and if there is no time_shifted_service_descriptor.

It is mandatory to be transmitted. IRDs are recommended to make use of it in order to display the service names in the user interface.

The service type defined in this standard is the service used for the following:

- Temporary (video, audio, data) service is not a regular service but is a service organizing the program temporarily.
- Engineering download service is a service to download software and data to the IRDs.
- Promotion (video, audio, data) service is to advertise contents of programs and services.
- Data service for accumulation beforehand is a service to be used without depending on the placement on the accumulation media among the service which can be viewed after the data is accumulated in the IRD.
- Data service exclusively for accumulation is an exclusive service used for maintaining the service in the designated directory of the accumulating media among the service which can be viewed after the data is accumulated in the IRD.
- Book mark list data service is a service to indicate book mark information recorded in the IRD.

2.3.8 Time shifted service descriptor

This descriptor identifies a service as a time shifted copy of another service (sub-clause 4.1). The time_shifted_service_descriptor is allowed only once in each loop, if there is no service_descriptor. It is mandatory to be transmitted for services listed in a NVOD_reference_descriptor. IRDs are recommended to be able to interpret it in order to access NVOD-events.

2.3.9 Digital copy control descriptor

This descriptor is mapped to the SDT when digital copy control information and maximum transmission rate is the same in most programs of the same service. When a program differing with this information exists, this descriptor is mapped to the PMT and/or EIT for the program differing from the information.

When this descriptor is transmitted in multiple tables, priority of information expressed by this descriptor is PMT>EIT>SDT.

2.3.10 Logo transmission descriptor

This descriptor describes service logo information, such as pointing to PNG logo data transmitted by CDT (see ARIB STD-B21), logo identifier, logo version, and the 8-unit code alphanumeric character string for simple logo. Transmission is essential in a service that refers to simple logo or PNG logo data transmitted by using CDT.

2.3.11 Content availability descriptor

This descriptor is used in combination with the digital copy control descriptor. This descriptor can be put into the SDT when information to control record and output is the same in most programs of the same service. When there is a program with different information or when this descriptor is not put into the SDT, it can be put into the PMT and/or EIT.

When this descriptor is transmitted by multiple tables, the priority of information expressed by this descriptor is in the order of PMT, EIT, and SDT.

2.4 Descriptors of the Event Information Table (EIT)

An EIT-section is organized as shown in table 5-7, part 2 of this standard. The EIT has a loop for descriptors for each event described in the EIT. The SI descriptors in this sub-clause have a defined meaning in the loop.

2.4.1 Component descriptor

This descriptor is used to specify all streams that are attached to an event. The descriptor may appear more than once in a loop since there may be more than one stream. Even if there is a time_shifted_event_descriptor, this descriptor is allowed.

It is useful to indicate which streams will be available for future events.

2.4.2 Content descriptor

This descriptor is used to classify the content of the event. Only one content descriptor may appear in the loop, but there is the possibility to transmit more than one classification term because there is a loop within the descriptor. Even if there is a time_shifted_event_descriptor, this descriptor is allowed. The content information can be provided in the EIT sub_table for the corresponding NVOD reference service. Transmission of this descriptor is optional.

2.4.3 Extended event descriptor

This descriptor is used to transmit a larger amount of textual information for an event than is possible with the short_event_descriptor. The information in extended event descriptors supplements that given in a short event descriptor. A language code is transmitted in order to indicate in which language the text is written. More than one extended_event_descriptor is allowed, for transmitting more data than one descriptor may contain (255 bytes excluding header) and for different languages. Descriptors for the same language have to be grouped together, and the last_descriptor field specifies the number of the last extended_event_descriptor for a specific language.

Even if there is a time_shifted_event_descriptor, this descriptor is allowed. Transmission of this descriptor is optional.

2.4.4 Linkage descriptor

This descriptor is used to give a link to another service. If it appears in this loop it links to a service that is attached to this event. This descriptor is allowed more than once in this loop. Transmission of this descriptor is optional. Even if there is a time_shifted_event_descriptor, this descriptor is allowed. The meaning of the descriptor when it occurs here depends on the value of the linkage_type. If the linkage_type is:

a) 0x01, the descriptor refers to a service that contains information about this event. An example of the intended use is for the IRD to switch to the information service when the user requests additional information about this event;

The meaning of other values of linkage_type is not defined in this context. Note that the linkage_type does not indicate the service_type of the referenced service. An example of the intended use of the linkage descriptor is that an IRD user interface could include a mechanism like "info about the event" which would make the IRD tune to the linked service after the user initiated the mechanism.

2.4.5 Parental rating descriptor

This descriptor is used to give a rating of the program based on age or other criteria that is used to prevent children from viewing unsuitable programs. Even if there is a time_shifted_event_ descriptor, this descriptor is allowed. The parental rating information can be provided in the EIT sub_table for the corresponding NVOD reference service. Transmission of this descriptor is optional.

2.4.6 Short event descriptor

This descriptor is used to transmit the name and a short text description for an event. A language code is transmitted in order to indicate in which language the title and the text are written. Transmission of this descriptor is mandatory, unless there is a time_shifted_event_descriptor, in which case the descriptor is allowed. This descriptor is allowed more than once in the loop for different languages. Thus it is not allowed to have more than one short_event_descriptor with the same language code.

2.4.7 Time shifted event descriptor

This descriptor is used to indicate that an event is the time_shifted copy of another event. Transmission of this descriptor is mandatory in case of NVOD. IRDs are recommended to decode this descriptor, without which access to the SI of NVOD events is not possible.

2.4.8 Digital copy control descriptor

This descriptor indicates digital copy control information of individual program and the maximum transmission rate.

When this descriptor is transmitted in multiple tables, priority of the information indicated by this descriptor is PMT>EIT>SDT.

2.4.9 Audio component descriptor

This descriptor is used to specify each parameter of audio stream composing an event. As multiple audio streams exist for one event in some cases, this descriptor may occur more than once in one loop. Even if there is a time_shifted_event_descriptor, this descriptor is allowed.

2.4.10 Data contents descriptor

This descriptor describes data component of the contents in the event, and component tag of the component stream. Selector_byte area in the descriptor is used to describe information of language of multimedia service or picture size, or capacity for storage, according to the form specified in each data component. The component stream composing data broadcasting contents may be transmitted in the event or in other event or service, and this descriptor describes component tag of all component streams related to the corresponding contents in the former event.

Example: The component tag of all streams necessary to indicate video/audio and related data in data contents descriptor is described when program linked data produced at the same time as the video and audio in the same Transport Stream is announced in the same event in the same service. Therefore, all component streams necessary to record data broadcasting are specified only by referring to the data content descriptor.

2.4.11 Hyperlink descriptor

This descriptor is used to describe linkage information when two related programs are made in different events and services, for the following:

a) Video audio program and related information program

b) Video audio program and index program in program

c) Video audio program and its guide information program

d) Others

When the hyperlink type is combined_data (0x01), combined_stream (0x02), index_data (0x03), or index_stream (0x04), it is recommended to link in bi-directional. It means that when the hyperlink made to other event B is made by mapping the hyperlink descriptor in the EIT of event A, it is recommended to make hyperlink to event A by mapping the hyperlink descriptor to the EIT of event B.

2.4.12 Series descriptor

This descriptor is used to identify multiple events, which are made in series. An individual series is identified with the series identifier. The IRD can use it when operating as a whole (such as reservation) for the series event group.

2.4.13 Event group descriptor

This descriptor describes grouping information for the same event of a common event, linkage information for the event relay, information of the original event when moving the event to different service.

- Common event is a broadcasting style, which the same program can be viewed whichever service is selected by describing the same ES_PID in the PMT of multiple services when broadcasting programs.
- Event relay is a broadcasting style, in which a program is broadcast continuously on a different service from midway in the program.
- Event moving is a broadcasting style, by which a program is broadcast on the service differing from the service scheduled before the broadcasting starts.

2.4.14 Component group descriptor

This descriptor is used to indicate that the component group is organized in a group, when there is a relation in multiple components composing one event. Its relation is identified with the component group type. CA setting and total bit rate description for each component group can be made. It is used for the multi-view TV (MVTV), etc.

- Multi-view (MVTV) is an application to broadcast related contents in one service by multiple video, audio and other components, simultaneously.

2.4.15 CA identifier descriptor

If a service is generally CA protected, this descriptor may be used to transmit data of the CA-system. The CA_identifier_descriptor is not involved in any CA control function, it is an indication for the user interface software in the IRD that a service is under conditional access and which CA-system is used. Then the user interface software may decide whether this service is reachable or not. The aim of the transmission of this descriptor is to avoid frustration to users caused by services being displayed for selection that are not reachable. This descriptor is allowed only once in the loop. Transmission of this descriptor is optional.

2.4.16 LDT linkage descriptor

This descriptor provides information of linkage for the descriptor collected in the LDT.

When placed in the EIT, the event information linked from the descriptor is collected to the LDT and transmitted.

2.4.17 Content availability descriptor

This descriptor, which is used in combination with the digital copy control descriptor, describes information to control the record and output of each program.

When this descriptor is transmitted by multiple tables, the priority of information expressed by this descriptor is in the order of PMT, EIT, and SDT.

2.4.18 Carousel compatible composite descriptor

This descriptor shows the accumulation control information of each program by using the descriptors in the module information area and the private area defined in the data carousel transmission scheme (Chapter 6 of ARIB STD-B24 Part 3) as subdescriptors.

More than one subdescriptor can be placed in one carousel compatible composite descriptor.

2.5 Descriptors of the Program Map Table (PMT)

In addition to the descriptors defined in ISO/IEC 13818-1, the Ministerial Ordinances and Notifications, the following SI descriptors may be used in the PMT.

2.5.1 Mosaic descriptor

This descriptor may be located in the PMT and/or SDT. Its use to describe mosaic services is described in sub-clause 4.2.

2.5.2 Stream identifier descriptor

This descriptor enables specific streams to be associated with a description in the EIT, in cases where there are more than one stream of the same type within a service. The descriptor is mandatory only if the service contains more than one stream of the same type and there are component descriptors for that type of stream within the EIT.

2.5.3 Hierarchical transmission descriptor

This descriptor indicates the relation between hierarchical streams when transmitting elementary stream composing program to prevent deterioration of transmission or discriminating information quality. Hierarchical transmission is presupposed to transmit with the same TS and the same service identifier, to improve response characteristics at user selection and for SI transmission efficiency. The hierarchical transmission description is denoted in the second loop of the PMT.

When video stream is transmitted in two-hierarchical transmission, the higher-level and lower-level streams refer to each other.

If hierarchical level has more than two levels, an undefined bit is added before the hierarchical level to use as an hierarchical level to have cyclic linkage structure from the higher level to the lower level.

2.5.4 Digital copy control descriptor

This descriptor is used to indicate a program, digital copy control information of an elementary stream composing program, and maximum transmission rate.

When the descriptor is transmitted in PMT, the component control flag should always be "0". When this descriptor exists in the first descriptor loop, this information is applied to all elementary streams composing the program. When this descriptor is in the second descriptor loop, it is designated in each elementary stream. When designation to whole program and to individual elementary stream differs, designation to individual elementary stream has the priority.

When this descriptor is transmitted in multiple tables, information priority which the descriptor indicates is in the order of PMT>EIT>SDT.

2.5.5 Emergency information descriptor

This descriptor is transmitted when the emergency warning broadcasting is made and includes necessary information and function as emergency warning signal, which is transmitted as the audio signal formerly.

2.5.6 Target region descriptor

The target region descriptor indicates the region that is the target of that service when it is placed in the first loop and the target of that component when it is placed in the second loop. When the descriptor is not encoded, it means that target of that component is all areas. When the descriptor is encoded, it is recommended that the IRDs in the target area receive this component as default.

2.5.7 Video decode control descriptor

The video decode control descriptor is placed in the second loop and used to receive still picture composed of MPEG-I picture transmitted in small transmission speed and to have smooth indication when switched to video format.

2.5.8 Country availability descriptor

This descriptor is used to indicate if the service is available in a special country.

This descriptor can be used twice at maximum within the program loop of the PMT, once to indicate the country list where the service is available and once to list the countries where the service is not available.

2.5.9 Component descriptor

This descriptor compensates for the use in the EIT to specify all streams, which composes the service and can be used in the PMT. This descriptor can be used only once in the ES loop of the PMT.

2.5.10 Parental rating descriptor

This descriptor is used to rate the program during broadcast based on age or other judgment standard to prevent young people from viewing inappropriate programs.

2.5.11 Linkage descriptor

This descriptor is used to give a link to another service. If it appears in the first loop, it links to other service that is attached to this service. This descriptor is allowed only once in the first loop. Transmission of this descriptor is optional. The meaning of the descriptor depends on the value of the linkage_type.

If the linkage_type is:

- 0x03, it refers to a CA substitution service for this service.

The meanings of other values of linkage_type are not defined in this context. Note that linkage_type does not indicate the service_type of the referenced service. An example of the intended use of the linkage descriptor is that when access to this service is denied by the conditional access system, and when the CA substitution service exists to the selected service, information necessary to switch to the CA substitution service can be transmitted.

2.5.12 Content availability descriptor

This descriptor, which is used in combination with the digital copy control descriptor, describes information to control the record and output of each program and the elementary streams that constitute the program.

When this descriptor is in the first descriptor loop, the information applies to all the elementary stream that constitutes the program. When this descriptor is in the second descriptor loop, specific information is applied to each elementary stream. When specifications applied are different between the entire program and each elementary stream, priority is given to the specifications for each elementary stream.

When this descriptor is transmitted by multiple tables, the priority of information expressed by this descriptor is in the order of PMT, EIT, and SDT.

2.5.13 Hybrid information descriptor

This descriptor describes information on obtaining communication components and meta-files to indicate communication components with hybrid delivery of broadcast and communication, and to realize synchronizing playback between broadcast components and communication components. This descriptor is in the first descriptor loop.

2.6 Descriptors of the Time Offset Table

Composition of the TOT is indicated in table 5-9 of Part 2 of this standard. The TOT includes all items defined in the TDT and adds only the descriptor area. This descriptor area can map the descriptor only when the time offset time changing date and the time (set value of time_of_change) of next time are clear, and not mapped in other case.

2.6.1 Local time offset descriptor

This descriptor is mapped to the descriptor area in the TOT to add regular offset to the transmitted hour (UTC + 9 hours) and indication hour to human, when executing local time system.

2.7 Stuffing descriptor

This descriptor can be placed anywhere, by which the descriptor is usable in SI. This descriptor is used to fill up the table or to make enabled descriptor to non-operation status for a certain reason (such as re-multiple, etc.) The IRDs should skip the stuff descriptor.

2.8 ISO 13818-1 descriptors

The following ISO/IEC 13818-1 (MPEG-2) descriptors can be expected in the SI bit streams:

- registration_descriptor;

  The registration descriptor is used for describing information uniquely identifying private data not specified by ISO/IEC 13818-1. In case that a stream not uniquely identified by stream type is used, a registration descriptor shall be used.

- private_data_indicator_descriptor;
- copyright_descriptor;
- ISO_639_language_descriptor.

  This descriptor lists the different languages in which a service/event is broadcast. This descriptor may be present in the SDT (and in the EIT). When present, the descriptor can be used by the IRD to select services or events with a language criterion. When this descriptor is used within the SI bit streams the audio type field should be set to the value 0x00 (undefined).

The meaning of other MPEG-2 descriptors is not defined if included in the SI Tables.

2.9 Unknown descriptors

If an unknown descriptor appears in a context where its meaning is not specified in this standard, or if the IRD encounters a descriptor with an unrecognized tag, the IRD is recommended to skip over that descriptor (using the length field) and proceed with decoding the following SI data.

2.10 Broadcaster information table descriptor

The structure of the BIT is specified in table 5-13, part 2 of this standard.

2.10.1 First descriptor area (Original network group)

In the first descriptor area of the BIT, the SI descriptors in this sub-clause are defined.

2.10.1.1 SI transmission parameter descriptor

When this descriptor is placed in the first descriptor area of the BIT, it is used to indicate the SI transmission parameter information operated commonly in the original network. This descriptor can be placed more than once in the same area. This is because for enabling transmission of the parameter to be used in the near future beforehand, as well as transmission of the actually enabled parameter, when changing the SI transmission parameter from a certain time. The changing time of the parameter, either enabled or disabled, is indicated with the parameter version number and update_time of the descriptor.

2.10.1.2 SI prime TS descriptor

When this descriptor is placed in the first descriptor area of the BIT (original network group), the identification information and transmission parameter of the SI prime TS of the network (TS of the special transmission style regarding the SI) is provided.

In the table description length byte, both NBIT and LDT information are provided. Even when they are the default parameter, description is not omitted, as it is the judgment reference of the table usage for the IRD. That is, when there is no description, it means that the table is not transmitted.

2.10.2 Second descriptor area (broadcaster group)

In the second descriptor area of the BIT, the SI descriptors in this sub-clause are defined.

2.10.2.1 Broadcaster name descriptor

This descriptor is used to transmit the broadcaster name. Only one descriptor can be placed for one broadcaster group.

2.10.2.2 Service list descriptor

This descriptor can provide a list of the service and service type in each broadcaster. One descriptor can be placed for one broadcaster group.

2.10.2.3 SI transmission parameter descriptor

When this descriptor is placed in the second descriptor area of the BIT, it is used to indicate the SI transmission parameter information operated commonly in the broadcaster. When the SI transmission parameter operated in the broadcaster is the same as the SI transmission parameter operated in the original network common placed in the first descriptor area, this descriptor does not have to be placed in the second descriptor area. This descriptor can be placed more than once in the same area. This is because it is for enabling transmission of the parameter to be used in the near future beforehand, as well as transmission of the actually enabled parameter, when changing the SI transmission parameter from a certain time in the broadcaster. The changing time of the parameter, either enabled or disabled, is indicated with the parameter version number and update_time of the descriptor for each broadcaster group.

2.10.2.4 Extended broadcaster descriptor

This descriptor is used for describing the extension information of broadcasters. Terrestrial broadcasters are identified in digital terrestrial television broadcasting, and terrestrial audio broadcasters are identified in digital terrestrial sound broadcasting.

A terrestrial broadcaster may share the same NVRAM in the receiver with terrestrial broadcasters who are out of the service area or broadcasters of other networks. Other than the access right to the NVRAM, this descriptor can also be used, when a mobile receiver moves out of the service area of a digital terrestrial television broadcaster, for describing information needed to tune in to terrestrial broadcasters of other areas who might be broadcasting the same program. Similar use of information is possible also in the case of a terrestrial sound broadcaster.

For the above purposes, this descriptor is used for grouping the relation of a terrestrial broadcaster with other terrestrial broadcasters and broadcasters of other networks as well as the relation of a terrestrial audio broadcaster with other terrestrial audio broadcasters and broadcasters of other networks.

2.10.2.5 Hyperlink descriptor

This descriptor is used to specify for each broadcaster the URI of the portal link destination and the URI of the authority, which allow the access of receiver units. Multiple hyperlink descriptors can be placed for one broadcaster group. The URI of the portal link destination corresponds to the URI of the BML document provided by the broadcaster for the contract between the broadcaster and the users. The authority is the character string used as the name space for each broadcaster when accumulating server-type contents in server-type broadcasting receivers.

2.11 Network board information table descriptor

The structure of the NBIT is specified in table 5-14, part 2 of this standard.

2.11.1 Board information descriptor

When this descriptor is placed in the NBIT, the title and the content of the board information are provided in text type.

2.12 Linkage description table descriptor

The structure of the LDT is shown in table 5-15, part 2 of this standard.

2.12.1 Short event descriptor

Operation of this descriptor, which is linked with the EIT using the LDT linkage descriptor, shall be in accordance with the operation of the same descriptor in the EIT.

2.12.2 Extended event descriptor

Operation of this descriptor, which is linked with the EIT using the LDT linkage descriptor, shall be in accordance with the operation of the same descriptor in the EIT.

When linking from LDT linkage descriptor to the LDT, the item name is not described in cases where the descriptor identification is in independent style.

2.13 Descriptor used in INT

The INT consists of 3 descriptor areas (see table 5-21 in Part 2)), the first descriptor area (located next to platform_descriptor_loop_length), the second descriptor area (located next to target_descriptor_loop_length) and the third descriptor area (located next to operational_descriptor_loop_length) which are called platform, target and operational, respectively.

The platform describes information on all receivers. The target is basically used with the operational for indicating information described in the operational for the receiver designated by the target.

- Platform (the first descriptor area): The platform means an IP platform (IP network) used in broadcasting and describes information on the IP platform.
- Target (the second descriptor area): The target means a service target terminal and describes information identifying a target terminal.

- Operational (the third descriptor area): The operational means operation of a terminal and describes information for indicating terminal operation (for example, reference of IP stream) of a receiver designated by the target.

3. Program Specific Information (PSI) and SI operational interaction states

For the description of a service state the following four columns of table 3-1 are relevant: Program Association Table (PAT), PMT, SDT and EIT. The possible indications given by these tables for a service are listed in table 3-1. The first three columns and the fifth column give the possible combinations of the existence of the four tables, the fourth column lists the relevant combinations of the running status bits in the SDT.

For information about the states of the running_status field in event information, see sub-clause 1.4.

Table 3-1 Service state

Service present in					State of the service
PAT	PMT	SDT	SDT running status	EIT p/f	
Yes	Yes	Yes	Running or undefined	Yes	Service is running and broadcasting
No	No	Yes	Not running or undefined	No	Service definition still exists but the elementary stream does not exist and the broadcasting is not made (stopped) e.g: before broadcasting start or after broadcasting.
Yes	Yes	Yes	Pausing	Yes	Service definition still exists and the elementary stream exists and the broadcasting is not made (stopped) e.g: Other service guide or test broadcasting during broadcasting stop time.
No	No	Yes	Start within several seconds or undefined	Yes	Service definition still exists and broadcasting will start soon (stopped)
No	No	No	-	No	Under preparation, starting to make the service or corresponding to the end status of the service (service does not exist)

* All statuses other than listed above are in transition status.

4. Application

The syntax of SI is designed so that it operates under a wide range of operation conditions. Usage of SI in some applications is described (or illustrated) herein.

4.1 NVOD service

In MPEG-2, a method to transmit multiple video programs at once on one Transport Stream is provided. This has the possibility to provide the NVOD service by one broadcast service provider. This clause explains how such service can be realized or how to describe such service in SI.

A concept to provide one service as 6 services by shifting time is shown in figure 4-1. This is the simplest form of such service. All programs are the same in all channels. (Other forms, such as inserting different commercial messages between programs, can also be made.)

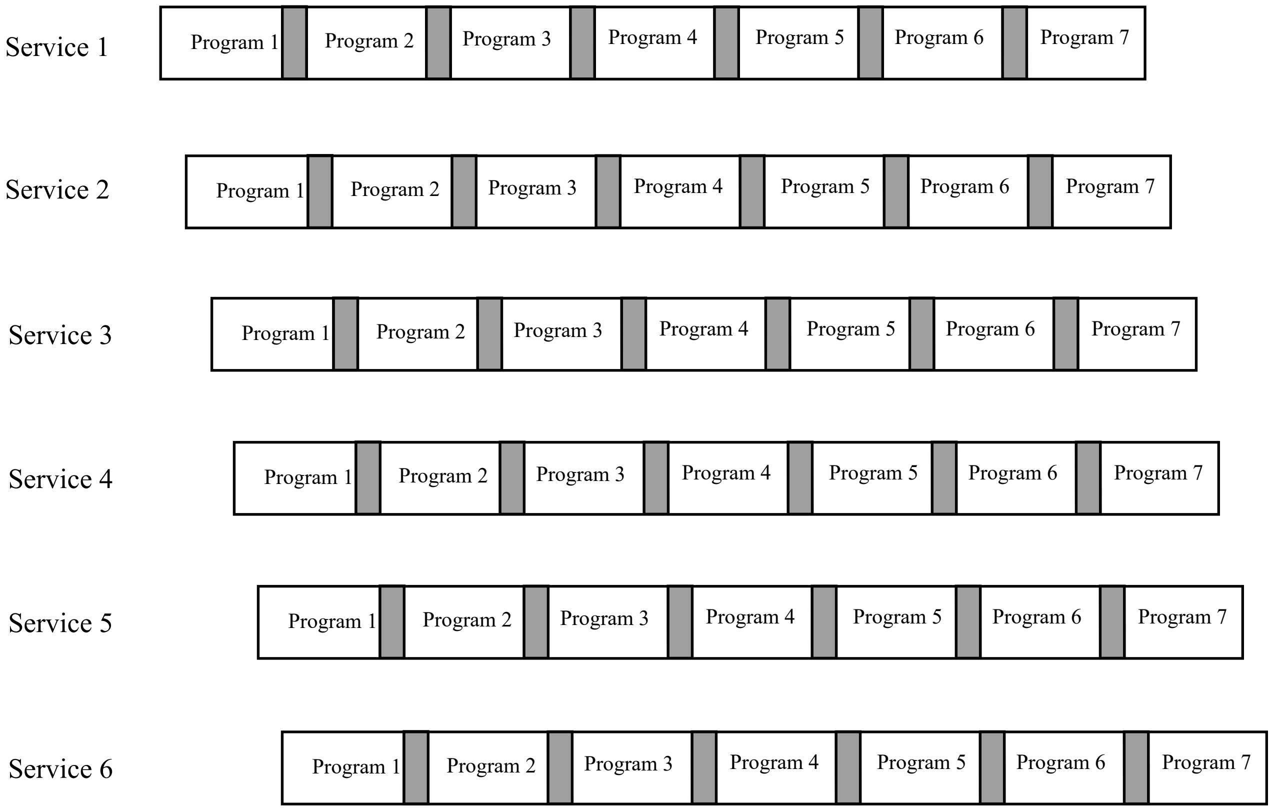


Figure 4-1 Example of NVOD service

In order to describe such NVOD service by former SI, event information table (EIT) should be transmitted 6 times repeatedly. Instead of this method, a concept of "reference service" is used herein.

"Reference service" is a kind of fictional service, and provides the means to relate the time shift services (service 1 to 6) during transmission with SI. This "reference service" is identified by the reference service identifier linked to the description common to the event in all the services belonging to the NVOD. The event information table (EIT) of the reference service always exists in the Transport Stream, by which the NVOD service is transmitted. Each time, the shift service is completely referred to Transport Stream identifier, original network identifier, and service identifier and these services are listed to the NVOD reference descriptor. Moreover, each time, the shift service is described with time shift service descriptor, which designates the reference description. Those are shown in figure 4-2.

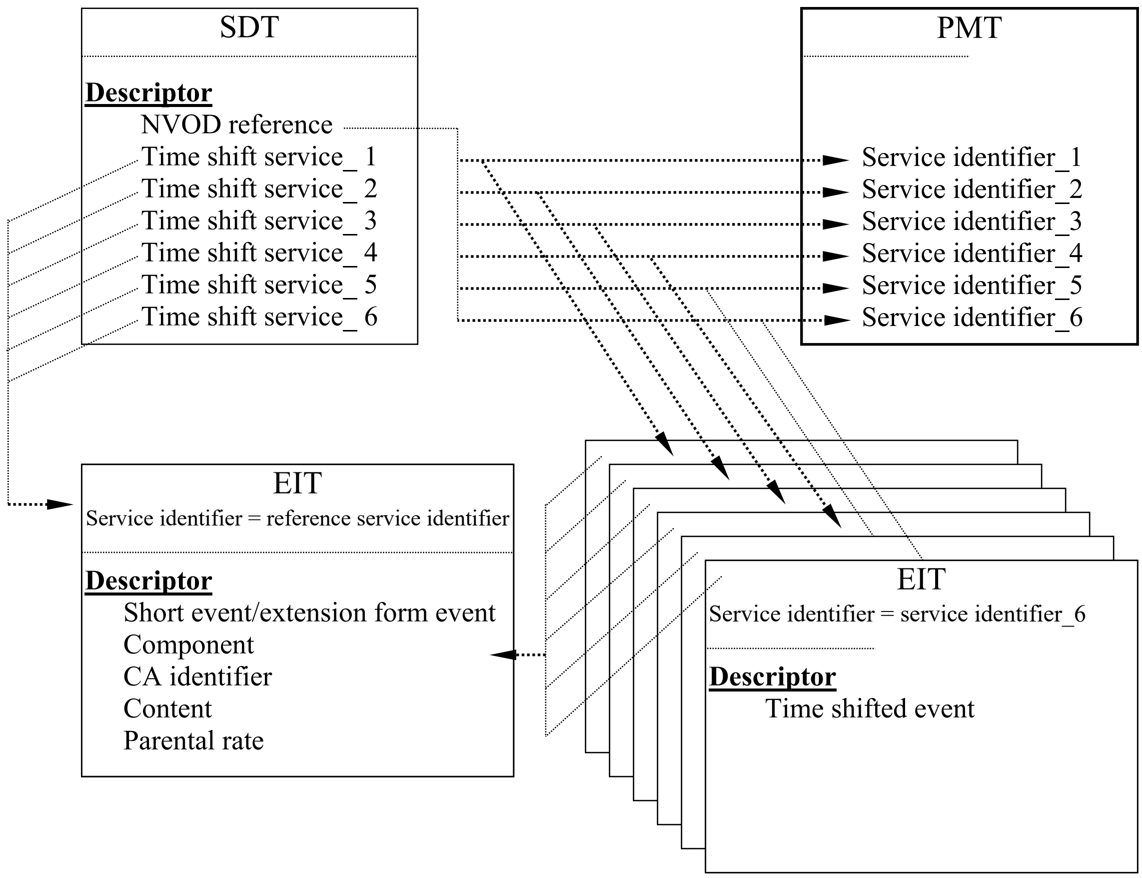


Figure 4-2 Description of SI in NVOD service

By using this method, data quantity can be reduced to 1/5. All the start times of the "reference service event information table" are set to "1" which is an invalid value, and the correct start time of each event is given in the EIT of each time shift service.

4.2 Mosaic services

4.2.1 General consideration

Mosaic services can be spread out over several TSs. A complete mosaic system can be organized in a tree structure.

A mosaic component is a collection of different video images to form a coded MPEG-2 video stream. The merging of the video images is performed at the source level, in such a way that at the display each image will occupy a specific area of the screen.

Each specific area is called a logical cell. Logical cells are composed of elementary cell(s). The mosaic screen is subdivided by a maximum of 8 × 8 elementary cells. Each elementary cell is numbered. A logical cell is a collection of elementary cells. Each logical cell is identified by a unique logical_cell_id.

The mosaic descriptor identifies the elementary cells (see figure 4-3), groups different elementary cells to form logical cells (see figure 4-4), and establishes a link between the content of all or part if the logical cell and the corresponding information carried in the SDT or EIT or BAT. Thus there is a close association between the mosaic descriptor and other SI Tables. The mosaic descriptor may be placed in either or both of the SDT and PMT sections for the mosaic service. Use in the SDT reduces the amount of interaction between the SI and MPEG Tables. However, a single mosaic service containing multiple video components can only be described by having the mosaic descriptor appearing multiple times within the PMT section. Some logical cells may have no link to SI (see figure 4-4).

0	1	2	3
4	5	6	7
9	10	11	12
13	14	15	16

Figure 4-3 Elementary cell organization

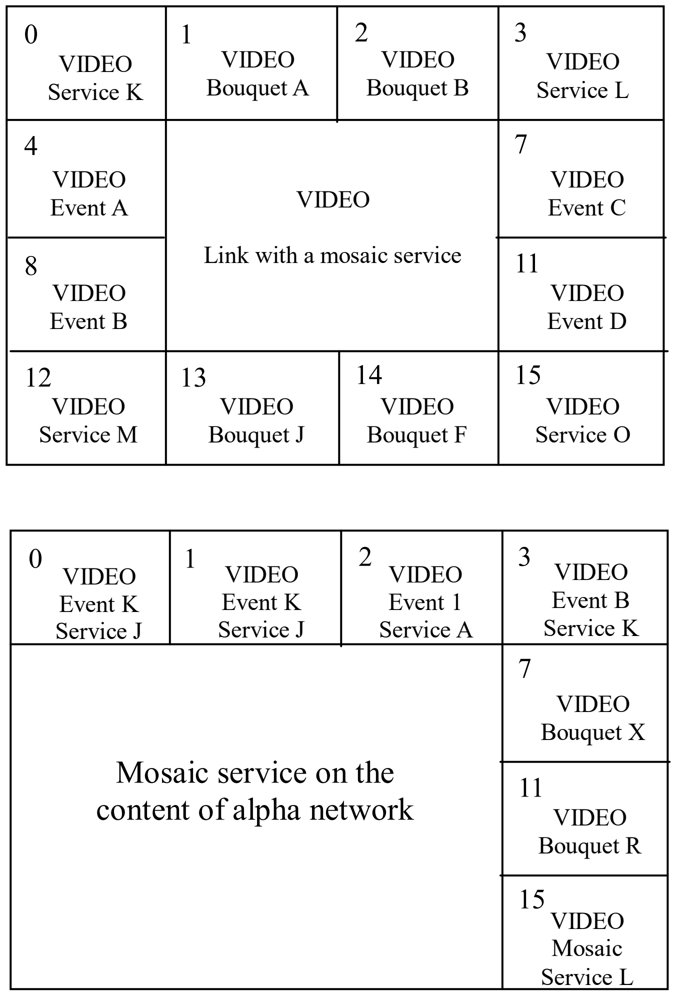


Figure 4-4 Examples of logical cell organization and content

4.2.2 Relationship between mosaic service and SI/PSI table

Algorithm to look for a mosaic service:

a) check the SDT Tables (actual TS / other TSs);

b) if a mosaic service exists:

- go to the corresponding TS;
- process the PAT and PMT;
- display the mosaic service;
- look after the content of the mosaic service:
  - if you are interested by the content of one logical cell and if it is related to:

- a bouquet: display the information of the corresponding BAT, go forward or cancel;
- a service: display the information of the corresponding SDT, process the PAT and the PMT and display the selected service;
- an event: display the information of the corresponding EIT, process the PAT and the PMT and display the selected event;
- a mosaic service: display the information of the corresponding SDT, process the PAT, the PMT, display the selected mosaic service, and go to "-look after the content of the mosaic service".

4.3 Transitions at broadcast delivery media boundaries

A very common broadcast delivery media infrastructure will be, that signals received from a satellite are converted and rebroadcast on a cable network. Depending on the size of the network, various technical options exist to facilitate these transitions.

4.3.1 Seamless transitions

A simple and low-cost solution is to remove the Quadrature Phase Shift Keying (QPSK) modulation from a satellite signal and replace it with a Quadrature Amplitude Modulation (QAM) suitable for the cable system. This mode is usually called a seamless transition. The major complication in this seamless mode is that the bit stream is left unchanged, which causes the NIT to be invalid for the actual delivery system to which the IRD is connected, for example a cable system.

Seamless transitions are supported by the SI System, with the proviso that it is readily possible for the IRD to identify whether the NIT information is valid. The rules of operation specified in subclause 1.1 permit invalid NIT data in the case that applicable delivery system descriptors are not given for the actual delivery system.

The NIT is intended to simplify the set-up and installation procedure for the viewers, and to signal changes of tuning information. However, since it is impossible after a seamless transition of a broadcast delivery media boundary always to maintain valid information in the NIT, the IRD may require mechanisms in addition to reception of the NIT to obtain tuning data.

The support of seamless network transitions is based on the definition of a unique identification mechanism for a TS. The transport_stream_id field, as specified in the MPEG-2 standard, allows

65,536 TSs to be uniquely identified. If transport_stream_id values are uniquely assigned to multiplex originators, this number is considered too small. Thus, the range of unique identifications of TSs has been extended in the SI by a field called original_netwrok_id of 16 bits. The concatenation of these 2 fields results in 4,294,967,296 unique identifiers for TSs. This gives sufficient room to allow for a unique identification of TSs without requiring a registration procedure.

Given this unique identification of the TSs, it is then possible to build IRDs that do not require a correct NIT for correct installation purposes. In order to support seamless transitions of TSs for small cable systems it is highly recommended that IRDs are able to initiate a frequency scan and store the unique TS identifiers with the sets of delivery system parameters. Within this procedure the same information as carried in the NIT can be obtained. However, the presence of a NIT does provide certain advantages for installation set-up and network management purposes.

In a seamless mode of operation, an IRD is able to detect the permitted instances of incorrect NIT data, even though no modifications to the bit stream have been made. In general, a network transition will occur between two different types of networks, e.g. from satellite to cable. In this case, the detection of an incorrect NIT is based on the value of the descriptor_tag in the NIT's delivery_system_descriptor. If the transition is between networks of the same type, the NIT should be replaced (see subclause 4.3.2) by either a valid NIT or a NIT for another type of network. After the detection of an incorrect NIT, the IRD should be able to initialize itself correctly, e.g. by using a frequency scanning procedure.

4.3.2 Non-seamless transitions without re-multiplexing

A slightly more complex option is to restore the TS packet bit stream and to perform some selective TS packet replacements in the TS. Such a packet replacement option does not require a re-timestamp operation and is of relatively low complexity. Some error handling operations need to be implemented in order to deal with the unrecoverable errors in the satellite signal and with lost TS packets. The NIT is carried in TS packets with a unique Packet Identifier (PID) value which allow the replacement function to be based on simple PID filter logic.

If a network transition is based on a TS packet replacement function, it is desirable that the new NIT information is stored and managed at the network boundary. This is the logical location of, as each network operator will demand the control over the frequency allocation in his network. For this local control to be as simple as possible, a fixed PID value is selected for TS packets carrying NIT data. A certain minimum data rate for the transmission of NIT data is specified to allow the replacement function to meet the minimum repetition time for the replacement NIT.

4.3.3 Transitions with re-multiplexing

The most complicated and expensive solution is to combine two or more TSs into a single one at the broadcast delivery media boundary. This re-multiplexing also involves the re-timing of the TS packets and the generation of a new SI data stream. The SI data in other TSs might in this case also be incorrect, which requires the checking and regeneration of the SI data in all TSs in the network. This option will only be feasible for very large networks.

4.4 Mixed multiple programming (Madara-broadcasting)

This clause explains usage of SI when mixed multiple programming (hereinafter referred to as Madara-broadcasting) broadcast by switching the HDTV or plural SDTVs in time series within the same band area is made.

4.4.1 Service image in Madara-broadcasting

There are three service images in the Madara-broadcasting: when all service_id exists all the time, when a part of SDTV service_id stops during the HDTV service broadcasting, and when the HDTV service_id and SDTV service_id are defined as different services. Usage of the SI in each case image is explained. The number of the service_id and ES_PID herein is one example.

4.4.1.1 When all service_id exists all the time

Madara-broadcasting when all service exists all the time is allocated with elementary PID (ES_PID) as shown in table 4-1 and services are broadcast as shown in table 4-5.

Table 4-1 Entry sample of ES_PID in Madara-broadcasting when all services exist all the time

service_id	ES_PID entered in PMT	
	SDTV	HDTV
0x0001	0x0030	0x0033
0x0002	0x0031	0x0033
0x0003	0x0032	0x0033

service_id	19:00	20:00	21:00
0x0001	SDTV(ES_PID=0x0030)	HDTV(ES_PID=0x0033)	SDTV(ES_PID=0x0030)
0x0002	SDTV(ES_PID=0x0031)		SDTV(ES_PID=0x0031)
0x0003	SDTV(ES_PID=0x0032)		SDTV(ES_PID=0x0032)

Figure 4-5 Service image of the Madara-broadcasting in which all services exists all the time

4.4.1.2 When a part of SDTV services stops

While the HDTV service is broadcast, Madara-broadcasting, which a part of SDTV services pause is allocated with ES_PID as shown in table 4-2 and services are broadcast as shown in figure 4-6.

Table 4-2 Entry sample of ES_PID in Madara-broadcasting which part of service pauses

service_id	ES_PID entered in PMT	
	SDTV	HDTV
0x0001	0x0030	0x0033
0x0002	0x0031	−
0x0003	0x0032	−

service_id	19:00	20:00	21:00
0x0001	SDTV(ES_PID=0x0030)	HDTV(ES_PID=0x0033)	SDTV(ES_PID=0x0030)
0x0002	SDTV(ES_PID=0x0031)	pause	SDTV(ES_PID=0x0031)
0x0003	SDTV(ES_PID=0x0032)	pause	SDTV(ES_PID=0x0032)

Figure 4-6 Service image of the Madara-broadcasting which part of services pause

4.4.1.3 When the HDTV service and SDTV service are defined as different services

The Madara-broadcasting, by which the HDTV service and SDTV services are defined as different services, is allocated with the elementary PID as shown in table 4-3 and the services are broadcast as shown in figure 4-7.

Table 4-3 Entry sample of ES_PID which the HDTV service and SDTV services are defined as different services

service_id	ES_PID entered in PMT	
	SDTV	HDTV
0x0001	0x0030	−
0x0002	0x0031	−
0x0003	0x0032	−
0x0004	−	0x0033

service_id	19:00	20:00	21:00
0x0001	SDTV(ES_PID=0x0030)	pause	SDTV(ES_PID=0x0030)
0x0002	SDTV(ES_PID=0x0031)	pause	SDTV(ES_PID=0x0031)
0x0003	SDTV(ES_PID=0x0032)	pause	SDTV(ES_PID=0x0032)
0x0004	Stop	HDTV(ES_PID=0x0033)	Stop

Figure 4-7 Service image when the HDTV service and SDTV services are defined as different services

4.4.2 Seamless switching of HDTV/SDTV

Usage of SI to switch the HDTV and SDTV seamlessly is explained.

4.4.2.1 Presupposition condition

a) PTS and DTS are synchronized between video ESs of target HDTV and SDTV.

* STC of both encoding equipment are synchronized

b) The GOP is synchronized between the video ESs of the target HDTV and SDTV.

c) The video ES on the ending transmission side should add the sequence end code (in case of MPEG-2 Video, an end-of-sequence NAL unit in case of MPEG-4 AVC and HEVC, the same hereinafter) after the final frame of the GOP transmission is finished, before ending.

d) The video ES on the starting transmission side should start as the closed GOP having a sequence header.

e) The video ES on the ending transmission side and the video ES on the starting transmission side should not be overlapped on TS.

There should be no gap which underflows buffer for the video ES of the IRD equipment.

4.4.2.2 PMT procedure

a) Updating of the PMT should be 0.5 to 2.0 sec. prior to switching control time between the HDTV and SDTV.

* Actual switching time of the video ES should be delayed 0.0 to 0.5 sec. to control time as there is a time lag in the cycle of 500.5ms in 1 GOP (in case of 15 frame) though the service control device is controlled in correct seconds generally.

b) At least the PMT before and after the switching control time of the HDTV and SDTV should include the video control descriptor.

* The IRD equipment corresponding to the seamless switching detects updating of the PMT version number, and selection of the video ES and decoding of the video are made according to the change of the video encode format of the video control descriptor.

c) The sequence_end_code_flag included in the video decode control descriptor should indicate whether or not the sequence end code exists when the transmission of the video ES indicated by the PMT ends.

4.4.2.3 Timing chart

Switching of the video ESs in a multiplied TS and version upgrade position of the PMT are shown in figure 4-8.

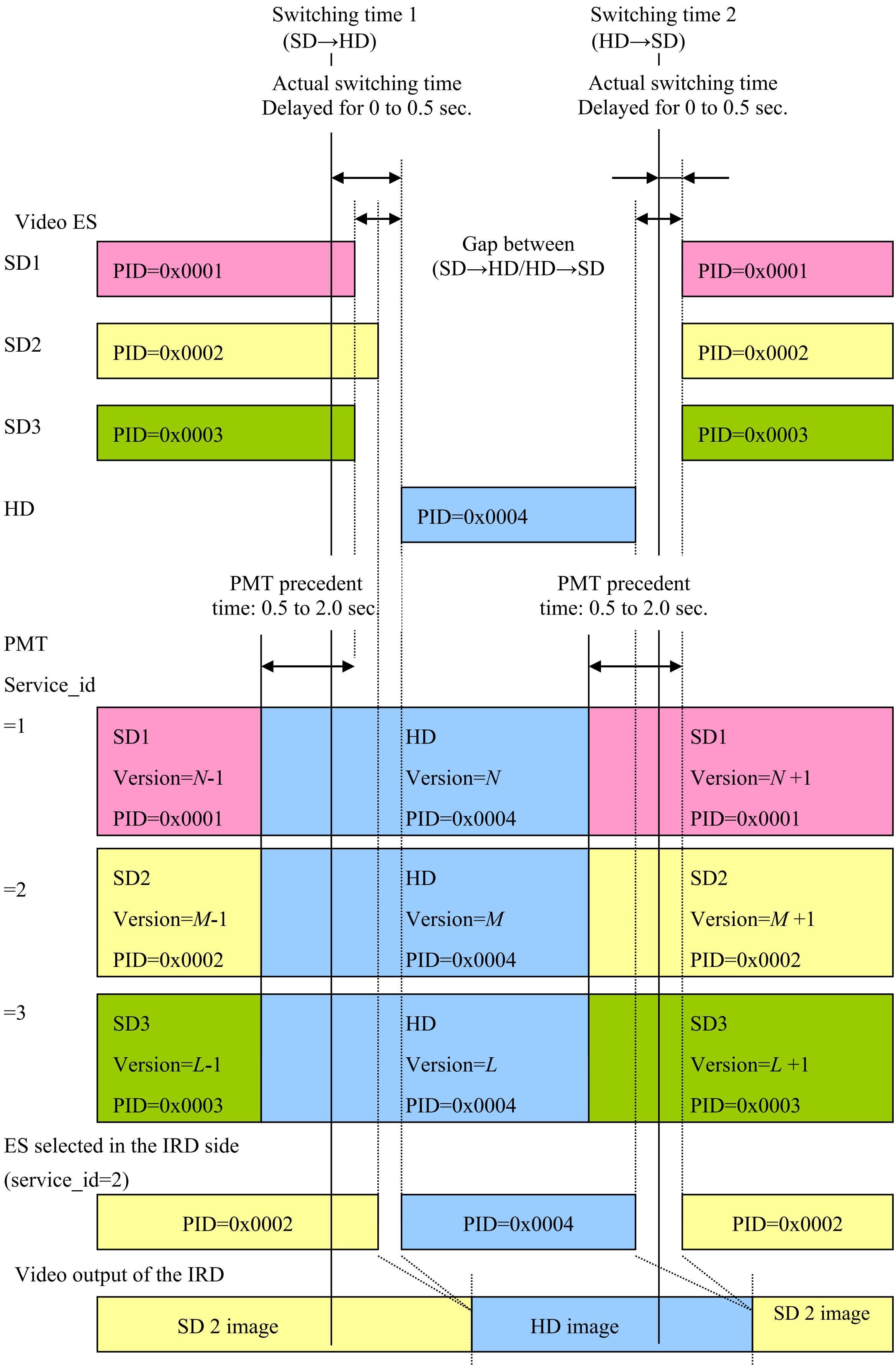


Figure 4-8 Timing chart of the seamless switching

Postscript

In the signal transmission multiplied in MPEG-2 Systems, many control signals are related, and the multiple style has a complex structure, by which they are combined each other. This appendix is attached to the standard because in organization and transmission of SI, which simplifies the users program selection, it is important to grasp and understand fully those characteristics and restricted items of those multiple system. It is recommended to use this appendix fully for smooth actual operation when the broadcast service providers and broadcast equipment production company use this standard.

This appendix is in accordance with ETSI ETR 211 "Digital broadcasting systems for television implementation guidelines for the use of MPEG-2 systems" issued as a technical document as a European area standard by ETSI, which is drafted by EP-DVB and EBU. Refer to the original technical documents when necessary.

SERVICE INFORMATION FOR
DIGITAL BROADCASTING SYSTEM

ARIB STANDARD

ARIB STD-B10　VERSION 5.13-E1
(December 5, 2019)

This Document is based on the ARIB standard of “Service Information For Digital Broadcasting System” in Japanese edition and translated into English in February, 2023.

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