



TA Document 2001013

AV/C Disc Subunit - DVR-Blue Media Type Specification 1.0

December 11, 2001

Sponsored by:

1394 Trade Association

Accepted for Release by:

1394 Trade Association Board of Directors.

Abstract:

This document describes the DVR-Blue media-specific part of the Disc General Subunit Specification.

Keywords:

Audio, Video, 1394, Digital, Interface, DVR-Blue.

Copyright © 1996-2002 by the 1394 Trade Association.
1111 South Main Street, Suite 122, Grapevine, TX 76051, USA
<http://www.1394TA.org>
All rights reserved.

Permission is granted to members of the 1394 Trade Association to reproduce this document for their own use or the use of other 1394 Trade Association members only, provided this notice is included. All other rights reserved. Duplication for sale, or for commercial or for-profit use is strictly prohibited without the prior written consent of the 1394 Trade Association.

1394 Trade Association Specifications are developed within Working Groups of the 1394 Trade Association, a non-profit industry association devoted to the promotion of and growth of the market for IEEE 1394-compliant products. Participants in working groups serve voluntarily and without compensation from the Trade Association. Most participants represent member organizations of the 1394 Trade Association. The specifications developed within the working groups represent a consensus of the expertise represented by the participants.

Use of a 1394 Trade Association Specification is wholly voluntary. The existence of a 1394 Trade Association Specification is not meant to imply that there are not other ways to produce, test, measure, purchase, market or provide other goods and services related to the scope of the 1394 Trade Association Specification. Furthermore, the viewpoint expressed at the time a specification is accepted and issued is subject to change brought about through developments in the state of the art and comments received from users of the specification. Users are cautioned to check to determine that they have the latest revision of any 1394 Trade Association Specification.

Comments for revision of 1394 Trade Association Specifications are welcome from any interested party, regardless of membership affiliation with the 1394 Trade Association. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally, questions may arise about the meaning of specifications in relationship to specific applications. When the need for interpretations is brought to the attention of the 1394 Trade Association, the Association will initiate action to prepare appropriate responses.

Comments on specifications and requests for interpretations should be addressed to:

Editor, 1394 Trade Association
1111 South Main Street, Suite 122
Grapevine, TX 76051
USA

1394 Trade Association Specifications are adopted by the 1394 Trade Association without regard to patents which may exist on articles, materials or processes or to other proprietary intellectual property which may exist within a specification. Adoption of a specification by the 1394 Trade Association does not assume any liability to any patent owner or any obligation whatsoever to those parties who rely on the specification documents. Readers of this document are advised to make an independent determination regarding the existence of intellectual property rights, which may be infringed by conformance to this specification.

Table of contents

- 1. Overview 10
 - 1.1 Purpose..... 10
 - 1.2 Scope..... 10
- 2. References 11
- 3. Definitions 12
 - 3.1 Conformance levels..... 12
 - 3.2 Glossary of terms 12
 - 3.3 Acronyms and abbreviations 12
- 4. DVR-Blue model..... 13
 - 4.1 Device function 13
 - 4.2 Contents list structure..... 14
 - 4.3 Profiles 15
 - 4.3.1 Basic DVR-Blue Video Player/Recorder profile 16
 - 4.3.2 Basic DVR-Blue Video Player/Recorder with detailed performance list support..... 17
 - 4.3.3 Basic DVR-Blue Video Player/Recorder with associated data list support 18
 - 4.3.4 Basic DVR-Blue Video Player/Recorder that supports both detailed performance list and associated data list..... 19
- 5. Subunit identifier descriptor 20
 - 5.1 Size value 20
 - 5.2 Root_object_list_ID 20
 - 5.3 Disc subunit dependent information..... 20
 - 5.4 DVR-Blue supported_media_type_specification 21
 - 5.4.1 DVR-Blue Implementation_profile_ID 21
 - 5.4.2 media_type_attributes field 21
 - 5.4.3 DVR-Blue type_dependent_information 22
- 6. Other descriptors 23
 - 6.1 Status descriptor 23
 - 6.1.1 General Disc Subunit Status Area Info Block..... 23
 - 6.1.2 Destination Plug Status Area Info Block 23
 - 6.1.3 Source Plug Status Area Info Block..... 23
 - 6.2 Root contents list..... 24
 - 6.3 Object Descriptor for DVR-Blue video contents 24
 - 6.4 Root associated data list 25
 - 6.5 Child directory object in root associated data list..... 25
 - 6.6 Associate data list descriptor 26
 - 6.6.1 DVR-Blue Title database object 26
 - 6.6.2 Thumbnail picture object(Digital Still Image Object)..... 26
 - 6.7 Root performance list..... 26
 - 6.8 Child directory object for root performance list..... 27
 - 6.9 Performance list 27
 - 6.10 Performance object in performance list..... 28
- 7. Commands..... 29
 - 7.1 Mandatory commands in all profiles (AV/C Disc subunit)..... 29
 - 7.1.1 CONFIGURE..... 29
 - 7.1.2 ERASE..... 29

- 7.1.3 IMPORT/EXPORT MEDIUM.....29
- 7.1.4 PLAY30
- 7.1.5 RECORD.....30
- 7.1.6 SEARCH - POSITION.....30
- 7.1.7 SET PLUG ASSOCIATION30
- 7.1.8 STOP30
- 7.2 Mandatory commands in all profiles (AV/C General)31
 - 7.2.1 OPEN DESCRIPTOR31
 - 7.2.2 OPEN INFO BLOCK.....31
 - 7.2.3 PLUG INFO31
 - 7.2.4 READ DESCRIPTOR.....32
 - 7.2.5 READ INFO BLOCK32
 - 7.2.6 WRITE INFO BLOCK.....32
- 7.3 Additional mandatory commands for associated data list support.32
 - 7.3.1 OBJECT NUMBER SELECT32
 - 7.3.2 RECORD OBJECT32
- 7.4 Additional mandatory commands for detailed performance list support.....33
 - 7.4.1 CREATE DESCRIPTOR33

- Annex A: Bibliography (informative)35
 - A.1 Bibliography35

- Annex B: New media type, list type, object type (normative).....36
 - B.1 Supported_media_type36
 - B.2 Media_type36
 - B.3 List_type37
 - B.4 Object_type.....37

- Annex C: New info block (normative)38
 - C.1 DVR-Blue specific info block (88 16₁₆)38
 - C.2 New image_format value(“JPEG”) of Image Format Info Block.....40

- Annex D: Optional commands41
 - D.1 COMBINE.....41
 - D.2 CREATE AV TRACK41
 - D.3 CREATE DESCRIPTOR42
 - D.4 DISC STATUS42
 - D.5 DIVIDE42
 - D.6 INCREMENT OBJECT POSITION NUMBER42
 - D.7 MONITOR42
 - D.8 OBJECT NUMBER SELECT42
 - D.9 RECORD(LOOP).....42
 - D.10 RECORD OBJECT43
 - D.11 REHEARSAL.....43
 - D.12 UNDO.....43

- Annex E: Descriptor examples (informative).....44
 - E.1 Profile ID examples.....44
 - E.1.1 [Example-1] Basic DVR-Blue player/recorder45
 - E.1.2 [Example-2] Basic DVR-Blue player/recorder with detailed performance list support.....46
 - E.1.3 [Example-3] Basic DVR-Blue player/recorder with associated data list support47
 - E.1.4 [Example-4] Basic DVR-Blue player/recorder with both detailed performance list and associated data list support.....48
 - E.2 contents descriptor example49
 - E.2.1 root contents list.....49

E.2.2 Video contents object entry	54
E.2.3 Root performance list	62
E.2.4 Root Associated data list	67
E.3 Status descriptor.....	70
E.3.1 General_disc_subunit_status_info_block	70
E.3.2 Destination Plug Status Area Info Block.....	71
E.3.3 Source Plug Status Area Info Block	74
Annex F: DVR-Blue Use Examples (informative).....	78
F.1 Descriptor Access Examples.....	78
F.1.1 Method to read the Subunit Identifier Descriptor or the Status Descriptor.....	78
F.1.2 Method to read an object entry descriptor.....	79
F.1.3 Method to read the header part of the list descriptor for contents description.	80
F.1.4 Method to read the whole list descriptor for contents description	81
F.1.5 Method to read a info block in the Subunit Identifier Descriptor or the Status Descriptor....	82
F.1.6 Method to read a info block in an object entry	83
F.1.7 Method to write info block	84
F.2 DVR-Blue player/recorder control example	85
F.2.1 Method to find DVR-Blue player/recorder	85
F.2.2 Method to check availability of DVR-Blue player/recorder	85
F.2.3 Method to check medium change or contents change.....	85
F.2.4 Method to check remaining capacity	86
F.2.5 Method to get list of contents.....	86
F.2.6 Method to RECORD specific contents on destination plug #j.....	87
F.2.7 Method to "RECORD PAUSE"	87
F.2.8 Method to STOP record operation on destination plug #j	88
F.2.9 Method to PLAY specific object on source plug #i at the beginning	88
F.2.10 Method to "PLAY PAUSE"	88
F.2.11 Method to SEARCH specific position.	88
F.2.12 STOP PLAY operation on source plug #i.....	89
F.2.13 ERASE specific video object.....	89
F.2.14 Change plug association.....	89
Annex G: Audio stream format field in DVR-Blue specific info_block (informative).....	90
G.1 [example-1] Japanese BS-Digital stream.	90
G.2 [example-2] combined stream. (AAC and AC3).....	90

List of figures

Figure 4.1 – DVR-Blue player/recorder	13
Figure 4.2 – AV/C DVR-Blue player/recorder model	13
Figure 4.3 – DVR-Blue title structure example	14
Figure 4.4 – contents descriptor example	15
Figure 4.5 – contents descriptor structure example (1)	16
Figure 4.6 – contents descriptor structure example (2)	17
Figure 4.7 – contents descriptor structure example (3)	18
Figure 4.8 – contents descriptor structure example (4)	19
Figure 5.1 – DVR-Blue type_dependent_information	22
Figure C.1 – DVR-Blue specific info block	38
Figure C.2 – audio stream format field	38
Figure E.1 – Basic DVR-Blue player/recorder profiles	44
Figure E.2 – profile ID example-1 (<i>supported_media_type_specification</i>)	45
Figure E.3 – profile ID example-2 (<i>supported_media_type_specification</i>)	46
Figure E.4 – profile ID example-3 (<i>supported_media_type_specification</i>)	47
Figure E.5 – profile ID example-4 (<i>supported_media_type_specification</i>)	48
Figure E.6 – root contents list	49
Figure E.7 – Time Stamp Info Block for descriptor modification	50
Figure E.8 – Default playlist info block	50
Figure E.9 – Disc_capacity_info_block	51
Figure E.10 – Name info block	52
Figure E.11 – Video contents object entry example	54
Figure E.12 – Video signal mode info block	55
Figure E.13 – Video Stream Format Subtype Info Block	56
Figure E.14 – Size Indicator Info Block (“H_M_S_F”)	57
Figure E.15 – Time Stamp Info Block (<i>Content_creation_date_and_time</i>)	57
Figure E.16 – DVR-Blue specific info block	58
Figure E.17 – Program Attributes Info Block	59
Figure E.18 – Name info block	61
Figure E.19 – root performance list	62
Figure E.20 – object entry in root performance list	63
Figure E.21 – performance list	64
Figure E.22 – performance list for without detailed performance list support profile	65
Figure E.23 – performance entry	67
Figure E.24 – root associated data list	67
Figure E.25 – object entry in root associated data list	68
Figure E.26 – associated data list	69
Figure E.27 – General_disc_subunit_status_info_block	70
Figure E.28 – Media_and_edit_status_info_block	70
Figure E.29 – Destination Plug Status Area Info Block	71
Figure E.30 – Destination plug <i>Plug_status_info_block</i>	71
Figure E.31 – Operating Mode Info Block	72
Figure E.32 – Plug Configuration Info Block	72
Figure E.33 – Object_and_plug_type_specific_information for MPEG2-TS object	73
Figure E.34 – Position Info Block for destination plug	73
Figure E.35 – Source Plug Status Area Info Block	74
Figure E.36 – Source plug <i>Plug_status_info_block</i>	75
Figure E.37 – Operating Mode Info Block	75
Figure E.38 – Plug Configuration Info Block	76
Figure E.39 – Object_and_plug_type_specific_information for video object	76
Figure E.40 – Position Info Block for video contents in source plug status	77
Figure G.1 – audio stream format field example (1)	90

Figure G.2– audio stream format field example (2) 90
Figure G.3 – combined stream example 90
Figure G.4 – audio stream format field example (3) 90

List of tables

Table 4.1 – relationship between DVR-Blue format and AV/C DISC subunit descriptor.....	14
Table 5.1 – the size value	20
Table 5.2 – Disc subunit attributes value.....	20
Table 5.3 – DVR-Blue Implementation_profile_ID.....	21
Table 5.4 – media_type_attributes field	21
Table 5.5 – can_record field	21
Table 5.6 – DVR-Blue version	22
Table 7.1 – DVR-Blue video player/recorder commands (Disc).....	29
Table 7.2 – DVR-Blue video recorder commands (General)	31
Table 7.3 – DVR-Blue video commands (Disc).....	32
Table 7.4 – additional DVR-Blue video commands for (General)	33
Table B.1 – supported_media_type field.....	36
Table B.2 – media_type field.....	36
Table B.3 – new list_type for DVR-Blue	37
Table B.4 – new object_type for DVR-Blue	37
Table C.1 – parameter for JPEG in image_format_info_block	40
Table D.1 – DVR-Blue video recorder optional commands (DISC commands).....	41

Change history

Version 1.0 (December 11, 2001)

Original version.

1. Overview

1.1 Purpose

This document defines the DVR-Blue media type specification for the AV/C Disc subunit. This document is used in conjunction with the AV/C Disc General Specification.

1.2 Scope

The scope of this revision of the standard is to define the requirements for the DVR-Blue Video Player/Recorder basic profiles. The command set and its operational features are defined in this specification.

2. References

The following standards contain provisions, which through reference in this document, constitute provisions of this standard. All the standards listed are normative references. Informative references are given in Annex A. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

- [R1] IEEE Std 1394-1995, Standard for a High Performance Serial Bus.
- [R2] IEC 61883-1, Consumer audio/video equipment – Digital interface – Part 1: General.
- [R3] AV/C Digital Interface Command Set General Specification, Version 3.0. TA document number 1998003.
- [R4] Enhancements to the AV/C General Specification 3.0. Version 1.0. TA document number 1998010.
- [R5] AV/C Disc Subunit General Specification, Version 1.0. TA document number 1998013.
- [R6] AV/C Disc Subunit Enhancements for Hard Disk Drive Specification. TA document number 1999029.

3. Definitions

3.1 Conformance levels

3.1.1 expected: A key word used to describe the behavior of the hardware or software in the design models *assumed* by this Specification. Other hardware and software design models may also be implemented.

3.1.2 may: A key word that indicates flexibility of choice with *no implied preference*.

3.1.3 shall: A key word indicating a mandatory requirement. Designers are *required* to implement all such mandatory requirements.

3.1.4 should: A key word indicating flexibility of choice with a strongly preferred alternative. Equivalent to the phrase *is recommended*.

3.1.5 reserved fields: A set of bits within a data structure that are defined in this specification as reserved, and are not otherwise used. Implementations of this specification shall zero these fields. Future revisions of this specification, however, may define their usage.

3.1.6 reserved values: A set of values for a field that are defined in this specification as reserved, and are not otherwise used. Implementations of this specification shall not generate these values for the field. Future revisions of this specification, however, may define their usage.

3.1.7 Controller Read/Write: Each field in a descriptor can either be read/write(R/write/ignore(R/W/I) read/ignore(R/I), or read(R) by an external controller. This qualifier indicates whether the field can be written using WRITE DESCRIPTOR and WRITE INFO BLOCK commands with the partial_replace subfunction.

NOTE —The IEEE is investigating whether the “may, shall, should” and possibly “expected” terms will be formally defined by IEEE. If and when this occurs, draft editors should obtain their conformance definitions from the latest IEEE style document.

3.2 Glossary of terms

3.2.1 byte: Eight bits of data, used as a synonym for octet.

3.2.2 CSR Architecture: A convenient abbreviation of the following reference (see clause 2): ISO/IEC 13213: 1994 [ANSI/IEEE Std 1212, 1994 Edition], Information Technology—Microprocessor systems—Control and Status Register (CSR) Architecture for Microcomputer Buses.

3.2.3 quadlet: Four bytes of data.

3.2.4 DVR-Blue: A high capacity optical disc for video recording.

3.3 Acronyms and abbreviations

- 1) AV/C Audio Video Control
- 2) IEEE The Institute of Electrical and Electronics Engineers, Inc.

4. DVR-Blue model

4.1 Device function

The basic DVR-Blue player/recorder functionality is to record/play/edit MPEG-TS stream via i.LINK or analog AV input/output plug.

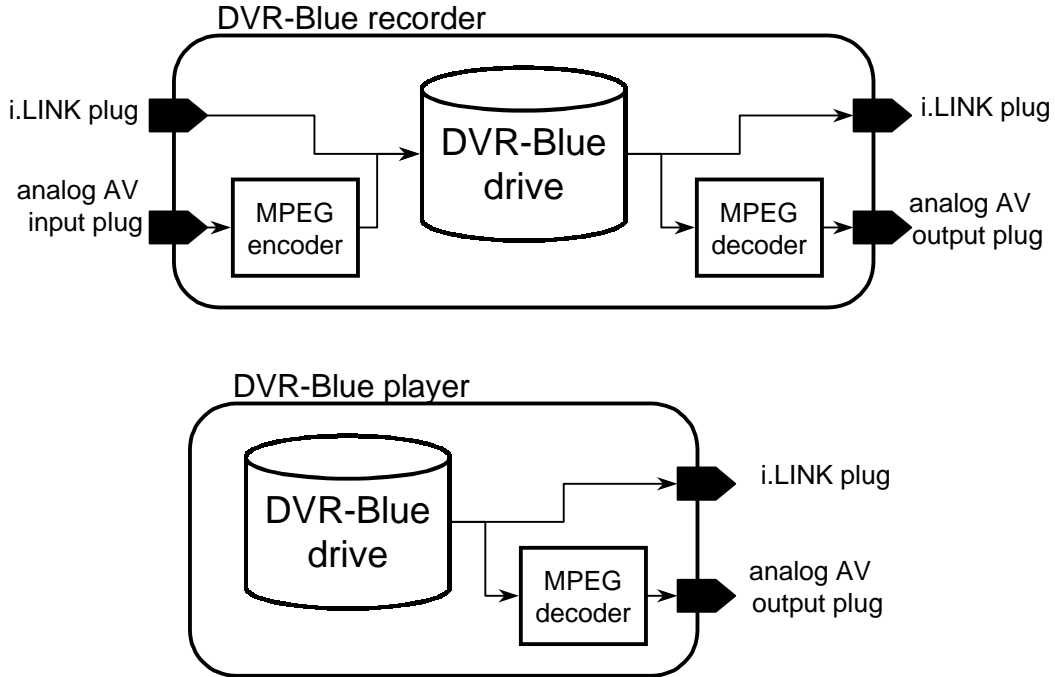


Figure 4.1 – DVR-Blue player/recorder

The AV/C model of DVR-Blue player/recorder is according to generic AV/C Disc subunit model.

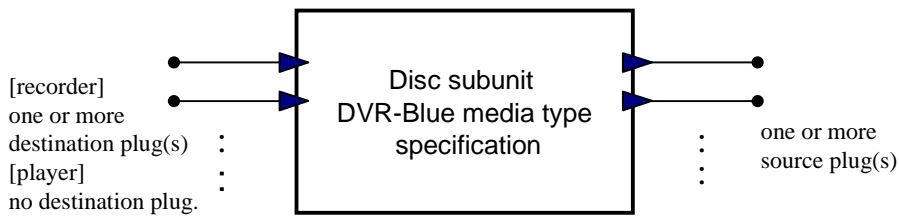


Figure 4.2 – AV/C DVR-Blue player/recorder model

4.2 Contents list structure

DVR-Blue Video player/recorder treats two different types of title, real title and virtual title. A real title is corresponding to the MPEG-TS stream that is recorded on DVR-Blue media. A virtual title is a collection of pointers, each of which point to a part of real title. Each title may have associated data. (Ex. DVR-Blue title database, thumbnail picture etc.)

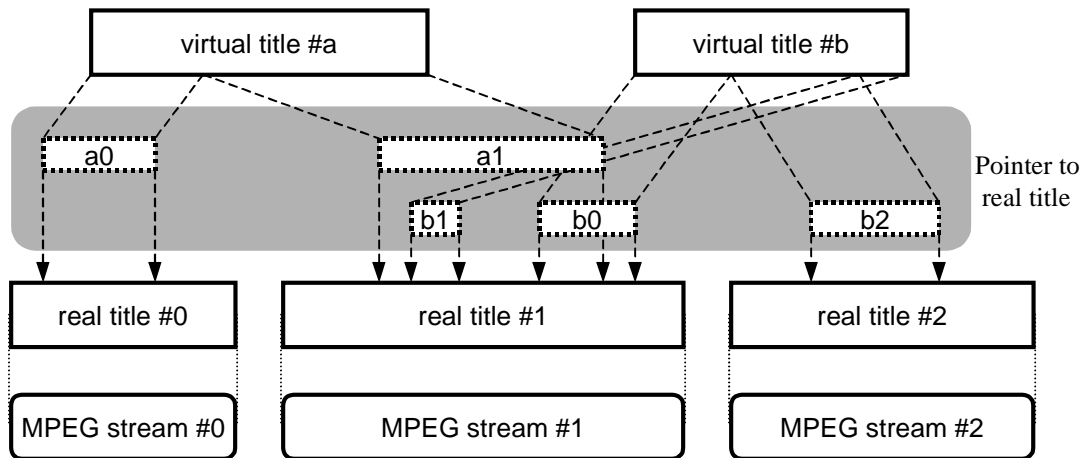


Figure 4.3 – DVR-Blue title structure example

In DVR-Blue media type specification, the following relationship between DVR-Blue format and AV/C DISC subunit descriptor structure is adopted.

Table 4.1 – relationship between DVR-Blue format and AV/C DISC subunit descriptor

DVR-Blue format	AV/C Disc subunit DVR-Blue media type specification
Real title	Video Content Object (AV track)
Virtual title	Performance List
Catalog of real title	Root Contents List
Catalog of virtual title	Root Performance List

NOTE —DVR-Blue player/recorder shall playback virtual title, so DVR-Blue player/recorder shall support the root performance list.

Figure 4.4 is an example of the Descriptor structure corresponding to the DVR-Blue Title structure shown in Figure 4.3.

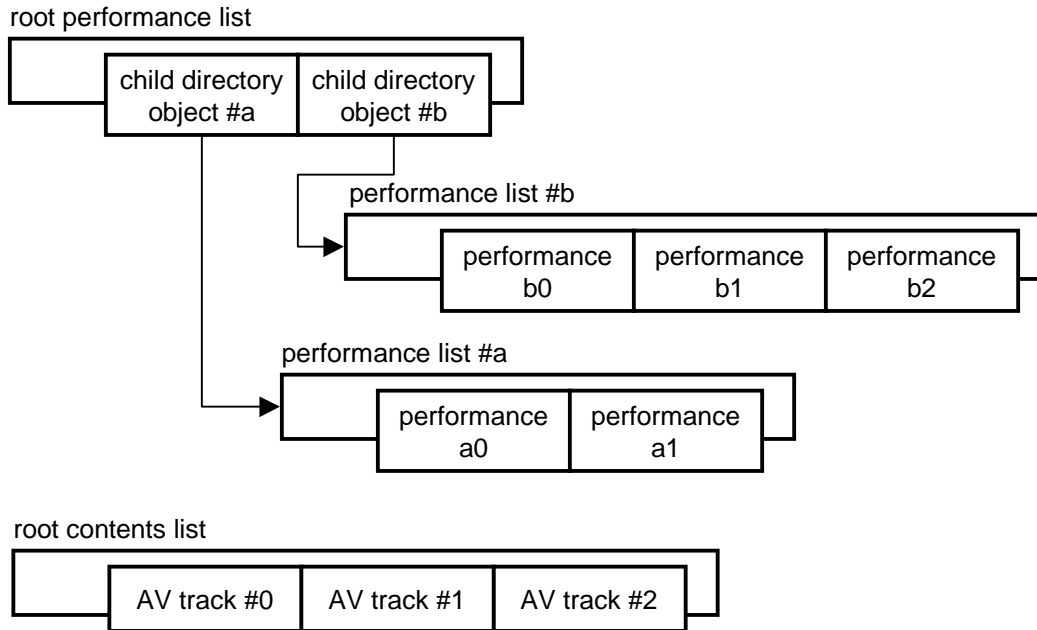


Figure 4.4 – contents descriptor example

4.3 Profiles

In this document, three profiles are defined:

- Basic DVR-Blue Video Player/Recorder
Support minimum descriptors and command set.
- Basic DVR-Blue Video Player/Recorder with detailed performance list support
Support Basic profile and detailed performance list.
- Basic DVR-Blue Video Player/Recorder with associated data list support
Support Basic profile and associated data list.

NOTE – If a subunit supports both detailed performance list and associated data list, the Extended Implementation Profiles mechanism is used. (see 5.4.1, 5.4.3 and Annex E.1)

4.3.1 Basic DVR-Blue Video Player/Recorder profile

The Descriptor structure example for Basic DVR-Blue Video Player/Recorder is as follows:

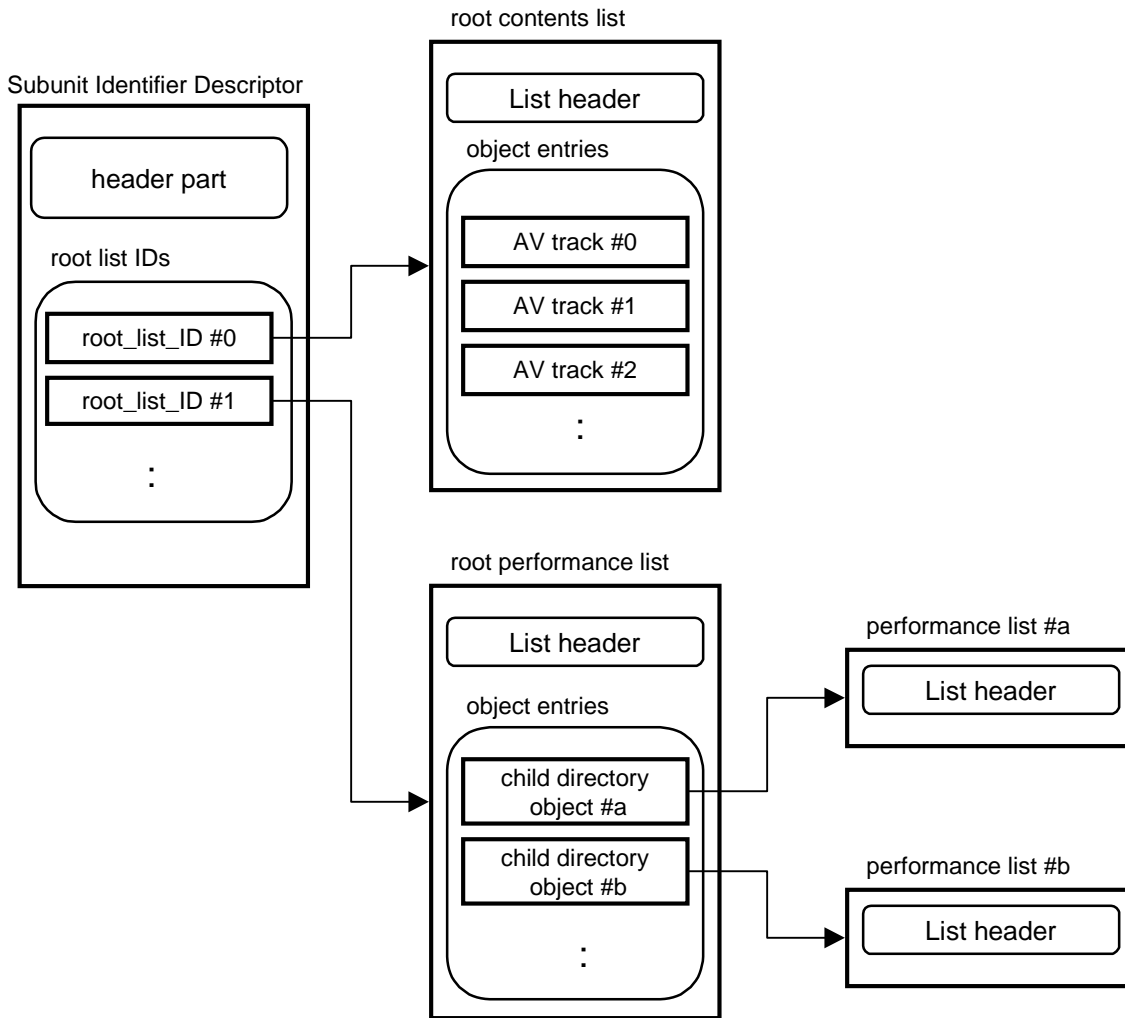


Figure 4.5 – contents descriptor structure example (1)

NOTE – The Performance lists without entry are required, because the child directory object needs to specify an appropriate *child_list_ID*.

4.3.2 Basic DVR-Blue Video Player/Recorder with detailed performance list support

The Descriptor structure example for Basic DVR-Blue Video Player/Recorder with detailed performance list support is as follows:

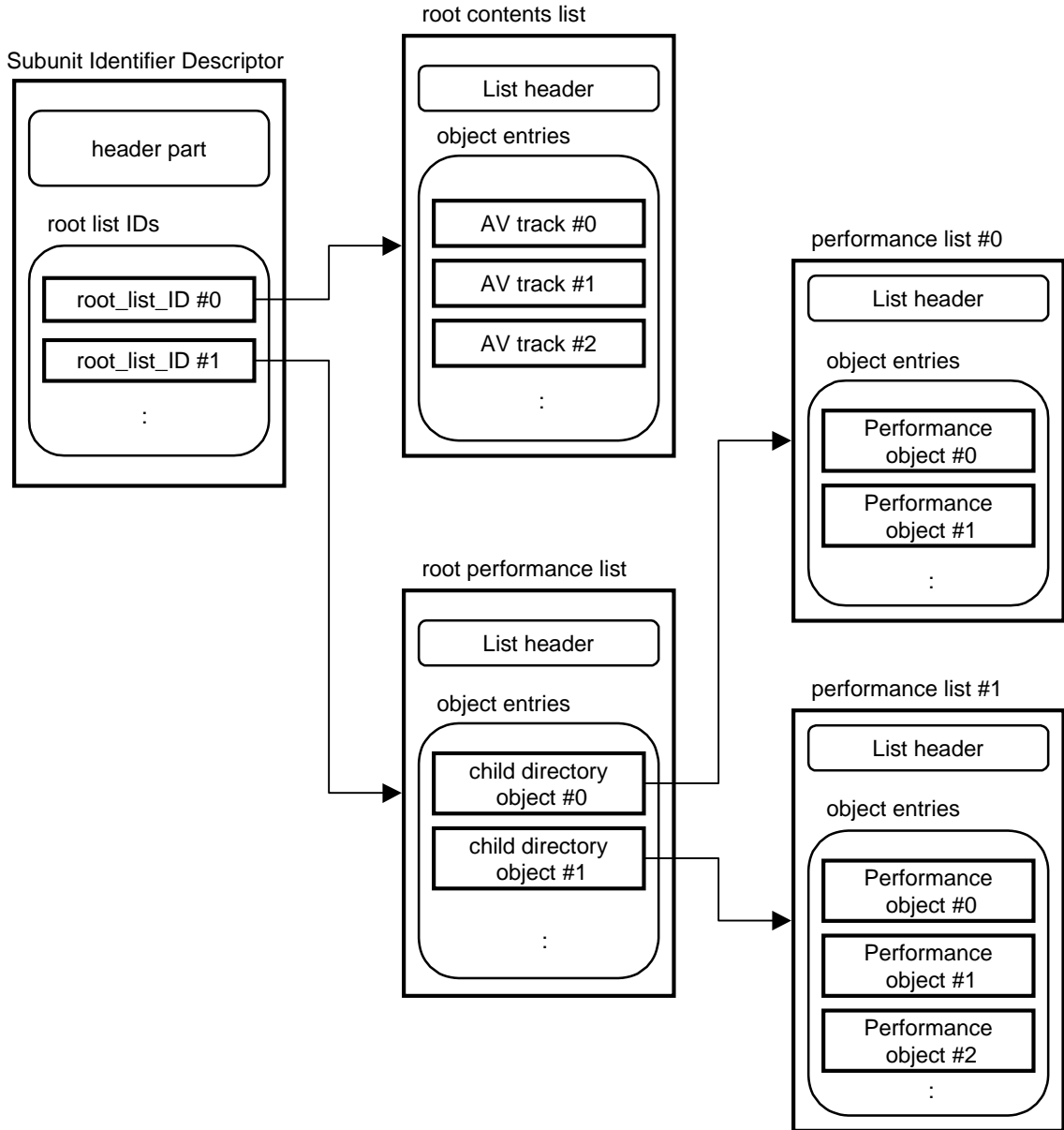


Figure 4.6 – contents descriptor structure example (2)

4.3.3 Basic DVR-Blue Video Player/Recorder with associated data list support

The Descriptor structure example for Basic DVR-Blue Video Player/Recorder with detailed performance list support is as follows:

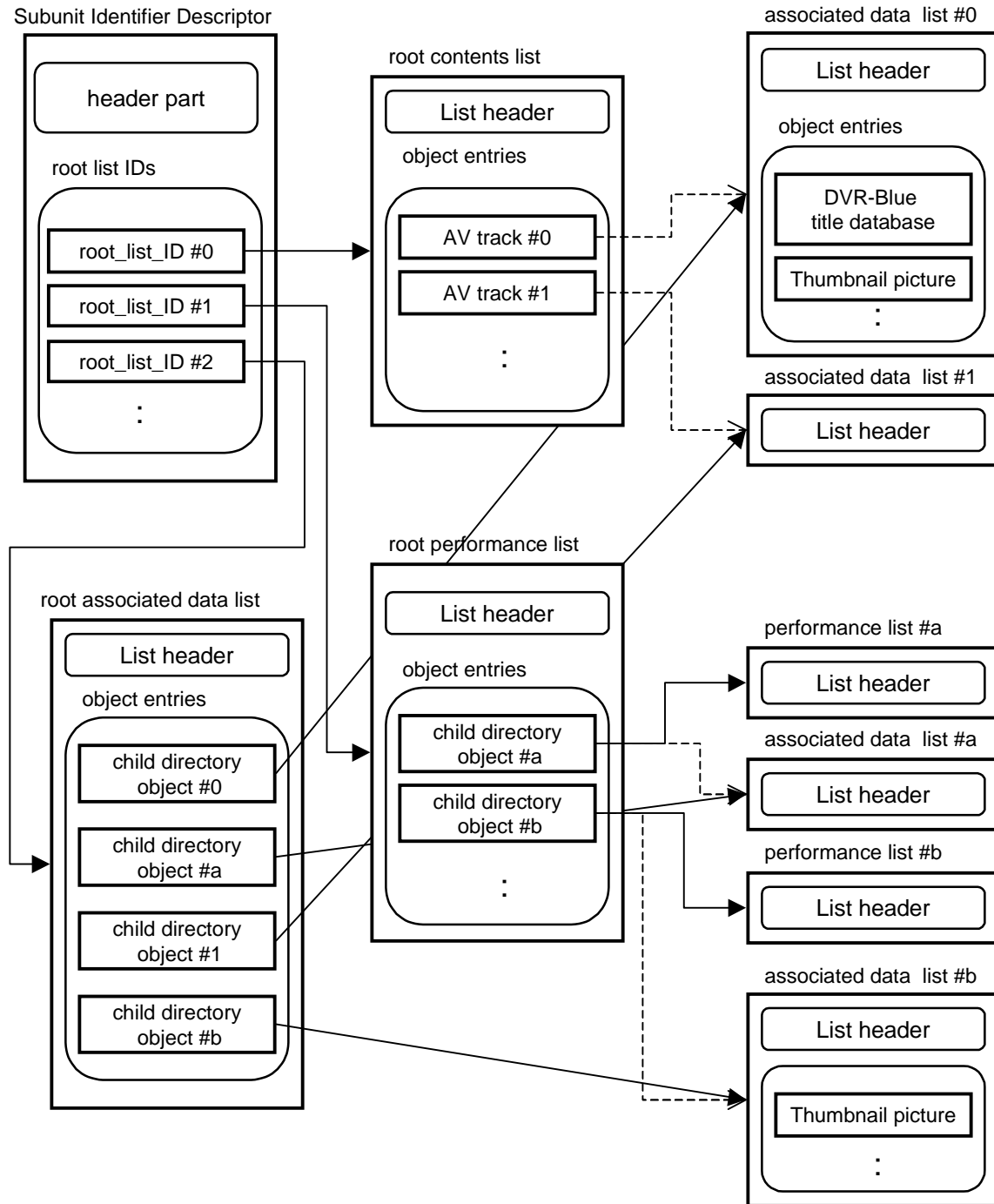


Figure 4.7 – contents descriptor structure example (3)

4.3.4 Basic DVR-Blue Video Player/Recorder that supports both detailed performance list and associated data list.

The Descriptor structure example for Basic DVR-Blue Video Player/Recorder with detailed performance list and associated data list support is as follows:

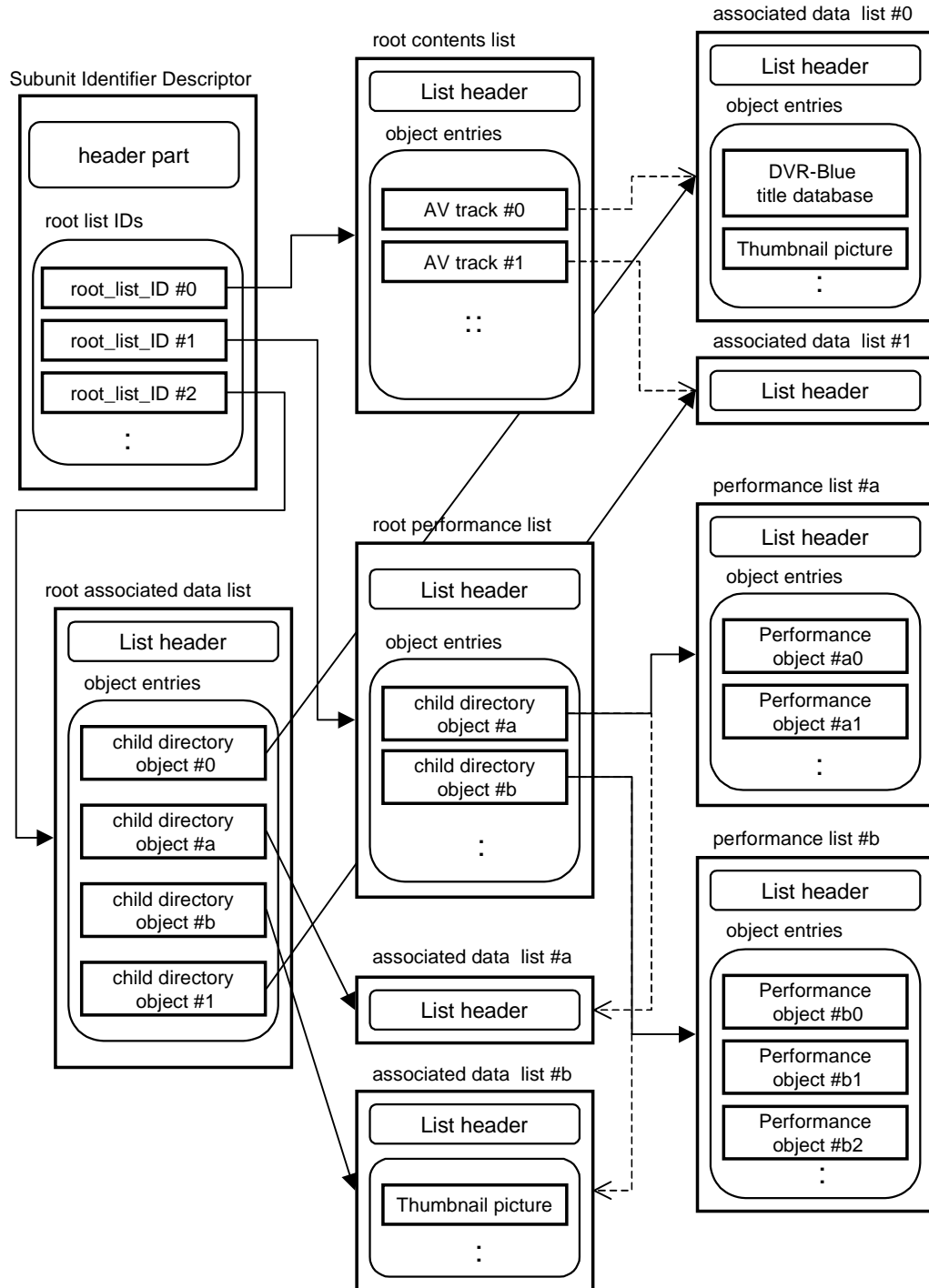


Figure 4.8 – contents descriptor structure example (4)

5. Subunit identifier descriptor

5.1 Size value

Table 5.1 – the size value

Field name	Value
Size_of_list_ID	02 ₁₆
Size_of_object_ID	02 ₁₆
Size_of_object_position	02 ₁₆

5.2 Root_object_list_ID

For all profiles, the mandatory list IDs in the *Root_object_list_ID* field of the subunit identifier descriptor are as follows:

- list ID of root contents list (30 00₁₆)
- list ID of root performance list (30 01₁₆)

For the “Basic DVR-Blue Video player/recorder with associated data list support” profile, the following root list ID is mandatory as well:

- list ID of root associated data list (30 02₁₆)

5.3 Disc subunit dependent information

Table 5.2 – Disc subunit attributes value

Attribute name	Value
has_more_attributes	0 ₂
supports_copyright	1 ₂

The *disc_subunit_version* field shall contain a value of 11₁₆ or higher. A value of 11₁₆ indicates conformance to AV/C Disc Subunit General, version 1.0[R5] and AV/C Disc Subunit Enhancements for Hard Disk Drive, version 1.0[R6].

5.4 DVR-Blue supported_media_type_specification

5.4.1 DVR-Blue Implementation_profile_ID

Table 5.3 – DVR-Blue Implementation_profile_ID

Implementation_profile_ID	Meaning
10 ₁₆	Basic DVR-Blue Video Player/Recorder
11 ₁₆	detailed performance list support
12 ₁₆	associated data list support
FF ₁₆	Extended Implementation Profiles Exist
All others	Reserved for future specification

NOTE – Usage of *Implementation_profile_ID* is described in Annex E.

5.4.2 media_type_attributes field

Table 5.4 – media_type_attributes field

Name	Attribute value
has_more_attributes	0 ₂
Supports_hierarchical_storage	0 ₂
Supports_two_sided_media	0 ₂

The *can_record* field indicates DVR-Blue recordable capability.

Table 5.5 – can_record field

DVR-Blue subunit capability	can_record field value	Meaning
DVR-Blue Recorder	1 ₂	This subunit can play/record/edit AV track on DVR-Blue media
DVR-Blue Player	0 ₂	This subunit can play AV track on DVR-Blue media

5.4.3 DVR-Blue type_dependent_information

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	1	R	DVR-Blue_version
00 01 ₁₆	4	R	Maximum_number_of_recordable_objects
00 02 ₁₆			
00 03 ₁₆			
00 04 ₁₆			
00 05 ₁₆	1	R	Number_of_extended_implementation_profiles (= n)
00 06 ₁₆	1	R	Extended_implementation_profile_ID[0]
00 07 ₁₆	1	R	Extended_DVR-Blue_version[0]
:			:
:			:
:	1	R	Extended_implementation_profile_ID[n-1]
:	1	R	Extended_DVR-Blue_version[n-1]

Figure 5.1 – DVR-Blue type_dependent_information

The *DVR-Blue_version* field indicates the version number of the DVR-Blue media type specification, to which the *implementation_profile_id* specified for this subunit, if not a multi-profile DVR-Blue conforms. This field contains an ordinal value, which is increased with each revision of the DVR-Blue media type specification. If the *implementation_profile_id* is set to FF₁₆, indicating that extended implementation profiles exist, then the *DVR-Blue_version* field may also be set to FF₁₆. (Annex E.1)

Table 5.6 – DVR-Blue version

DVR-Blue_version	Meaning
00 ₁₆	Version 1.0 of DVR-Blue media type specification
All others	Reserved for future version

6. Other descriptors

6.1 Status descriptor

6.1.1 General Disc Subunit Status Area Info Block

This info block is as described in the AV/C Disc Subunit General, version 1.0[R5]. It contains General Disc Subunit Status Area Info Block (88 00₁₆) with the nested Media and Edit Status Info Block (88 04₁₆).

The undo and difference operations may not be supported by a DVR-Blue media type specification. If it is not supported, the *undo_status* and *difference* fields are FF₁₆ and 0₂ respectively.

6.1.2 Destination Plug Status Area Info Block

The Destination Plug Status Area Info Block structure is as defined in AV/C Disc Subunit General, version 1.0[R5]. The supported info blocks for DVR-Blue media type specifications are:

- Plug Status Info Block (88 05₁₆)
- Operating Mode Info Block (88 06₁₆)
- Plug Configuration Info Block (88 07₁₆)

Refer to AV/C Disc Subunit General, version 1.0[R5] for the definition of the structure of this info block. The *AV_object_type* field and *object_and_plug_type specific information* is given in AV/C Disc Subunit Enhancement for Hard Disc Drive, version 1.0[R6].

- Position Info Block (00 03₁₆).

The structure of the info block is as specified in Enhancement to the AV/C Digital Interface Command Set General Specification, Version 3.0[R3]. It contains the Position Indicator Info Block (00 02₁₆). The position of the current recording point of the AV object will be returned. In case the root contents list is assigned with the destination plug, the DVR-Blue media specification supports indicator type *relative_HMSF_count*. It is to be noted that the indicated time may not be frame accurate, as accuracy of position is implementation dependent.

6.1.3 Source Plug Status Area Info Block

The Source Plug Status Area Info Block structure is as defined in AV/C Disc Subunit General, version 1.0[R5].

The supported info blocks for DVR-Blue media type specification are:

- Plug Status Info Block (88 05₁₆)
- Operating Mode Info Block (88 06₁₆)
- Plug Configuration Info Block (88 07₁₆)

Refer to AV/C Disc Subunit General specification, version 1.0[R5] for the definition of the structure of this info block. The *AV_object_type* field and *object_and_plug_type specific information* is given in AV/C Disc Subunit Enhancement for Hard Disk Drive, version 1.0 [R6].

- Position Info Block (00 03₁₆).

The structure of the Info Block is as specified in Enhancement to the AV/C General Specification 3.0, version 1.0 [R4]. It contains Position Indicator Info. Block (00 02₁₆). The position of the current play point of the AV object will be returned. In case the AV content (AV content object, root contents list, entry in root performance list, or performance list) is assigned with the source plug, DVR-Blue media specification supports the indicator type *relative_HMSF_count*. It is to be noted that the indicated time may not be frame accurate, as accuracy of position is implementation dependent.

6.2 Root contents list

The root contents list of DVR-Blue media type specification is corresponding to the catalog of the real play list in DVR-Blue format.

The List ID of root contents list is 30 00₁₆.

The *Attributes* field in the header part of root contents list is:

- *has_more_attributes* = 0
- *has_object_ID* = 1

NOTE – *has_child_ID* = 0 (this field is for object entry and not valid in the list descriptor.)

The Info Blocks supported for the root contents list are:

- Time Stamp Info Block for descriptor modification (00 07₁₆)
The *stamp_type* bit is set to 1₂ and the *time_stamp_data* field is used as 54bit counter.
- Default Playlist Info Block (80 0B₁₆)
Set list ID of root contents list.
- Disc Capacity Info Block (80 02₁₆)
All *capacity_format_indicator* fields are set to 01₁₆ (“bytes”).
- Name Info Block (00 0B₁₆).
The Name Info block is a nested info block and additionally contains Character Code Info block (00 08₁₆), Language Code Info Block (00 09₁₆) and Raw Text Info Block (00 0A₁₆).

6.3 Object Descriptor for DVR-Blue video contents

The object descriptor structure within the Root Contents List is as described in AV/C Digital Interface Command Set General, version 3.0[R3]. The DVR-Blue media type specification is intended to read and write Video Content Objects as described in the AV/C Disc Subunit Enhancements for Hard Disk Drive, version 1.0[R4].

The *Attributes* field:

- *has_more_attributes* = 0

NOTE – *has_object_ID* = 0 (this field is for object list and not valid in object entry.)

The supported info blocks for the video contents object descriptor are:

- Video Signal Mode Info Block (88 12₁₆).
- Video Stream Format Subtype Info Block (88 13₁₆).
- Size Indicator Info Block (00 01₁₆).
 - The supported type value is 00₁₆ (indicating HMSF type).
- Time Stamp Info Block (00 04₁₆).
 - This is the content creation time.
- DVR-Blue specific info block (88 16₁₆)
- Program Attributes Info Block (88 14₁₆).
- Name Info Block (00 0B₁₆).

The Name Info block is a nested info block and additionally contains Character Code Info block (00 08₁₆), Language Code Info Block (00 09₁₆) and Raw Text Info Block (00 0A₁₆).

The DVR-Blue media specification shall be able to reference object entries by position within the list (20₁₆) and by object ID within the list (21₁₆), as described in the AV/C Digital Interface Command Set General, version 3.0[R3].

6.4 Root associated data list

NOTE – In AV/C general 4.0, a list descriptor shall belong to a descriptor hierarchy, root element of which is root list. So that root associated data list is defined in this section. A controller need not to take care about this root associated data list, because relationship between “AV title” and “associated data list” is described in the DVR-Blue specific info block.

List ID of root associated data list is 30 02₁₆.

List type of root associated data list is 87₁₆ (“Associated data lists”)

The *Attributes* field in the header part of root associated data list is:

- *has_more_attributes* = 0
- *has_object_ID* = 0 (No object ID)

NOTE – *has_child_ID* = 0 (this field is for object entry and not valid in list descriptor)

The Info Blocks supported for the root performance lists are:

- Time Stamp Info Block for descriptor modification (00 07₁₆)
 - The *stamp_type* bit is set to 1₂ and the *time_stamp_data* field is used as 54bit counter.

6.5 Child directory object in root associated data list

The *Attributes* field:

- *has_more_attributes* = 0

NOTE – *has_object_ID* = 0 (this field is for object list and not valid in object entry)

The *child_list_ID* field indicates the *list ID* of corresponding associated data list.

6.6 Associate data list descriptor

The Associated Data List Descriptor is pointed from the *associated data list list_ID* field in DVR-Blue specific info block, and indicates the existence of associated data files that are related to the AV track. (For example: thumbnail picture, DVR-Blue title database which shows characteristics of MPEG stream of DVR-Blue specification.)

List type: 87_{16} (“Associated data lists”)

List specific attribute:

- *has_more_attributes* = 0
- *has_object_ID* = 1

NOTE – *has_child_ID* = 0 (this field is for object entry and not valid in list descriptor.)

6.6.1 DVR-Blue Title database object

Object type: $A0_{16}$ (“DVR-Blue Title database object”)

Object attribute:

- *has_more_attributes* = 0
- *has_child_ID* = 0

NOTE – *has_object_ID* = 0 (this field is for object list and not valid in object entry.)

6.6.2 Thumbnail picture object(Digital Still Image Object)

The thumbnail picture object of DVR-Blue uses Digital Still Image Object of the disc subunit.

Image format is JPEG. (Annex. C.2)

6.7 Root performance list

A root performance list is a catalogue of performance lists. In the case of DVR-Blue, the root performance list is corresponding to a catalogue of virtual play lists.

List ID of root performance list is $30\ 01_{16}$.

The *Attributes* field in header part of root performance list is:

- *has_more_attributes* = 0
- *has_object_ID* = 1

NOTE – *has_child_ID* = 0 (this field is for object entry and not valid in list descriptor)

The Info Blocks supported for the root performance lists are:

- Time Stamp Info Block for descriptor modification (00 07₁₆)

The *stamp_type* bit is set to 1₂ and the *time_stamp_data* field is used as a 54bit counter.

6.8 Child directory object for root performance list

The *Attributes* field:

- *has_more_attributes* = 0
- *has_object_ID* = 0 (this field is for object list and not valid in object entry.)

The *child_list_ID* field indicates the *list ID* of the corresponding performance list.

The supported info blocks for the child directory object in root performance list are:

- Video Signal Mode Info Block (88 12₁₆).
- Video Stream Format Subtype Info Block (88 13₁₆).
- Size Indicator Info Block (00 01₁₆).
 - The supported type value is 00₁₆ (indicating HMSF type).
- Time Stamp Info Block (00 04₁₆).
 - This is the creation time of performance list.
- DVR-Blue specific info block (88 16₁₆).
- Program Attributes Info Block (88 14₁₆).
- Name Info Block (00 0B₁₆).

The Name Info block is a nested info block and additionally contains Character Code Info block (00 08₁₆), Language Code Info Block (00 09₁₆) and Raw Text Info Block (00 0A₁₆).

6.9 Performance list

A performance list of DVR-Blue media type specification is corresponding to a virtual play list in DVR-Blue format.

The Performance list *list_specific_information_Attributes*:

- *has_more_attributes* = 0
- *has_object_ID* = 1

NOTE – *has_child_ID* = 0 (this field is for object list and not valid in object entry.)

The *AV_object_type* field is set to 83₁₆ (“Video Content Object”).

The Info Blocks supported for the performance lists are:

- Time Stamp Info Block for descriptor modification (00 07₁₆)

The *stamp_type* bit is set to 1₂ and the *time_stamp_data* field is used as a 54bit counter.

- Size Indicator Info Block (00 01₁₆).
- The *supported type* value is 00₁₆ (indicating HMSF type).

6.10 Performance object in performance list

The *Attributes* field :

- *has_more_attributes* = 0

NOTE – *has_object_ID* = 0 (this field is for object list and not valid in object entry.)

The supported info blocks for the performance object are:

- *descriptor_reference_info_block*: (“AV content object reference”)
specify the Video Content Object in root contents list.
- *content_entry_point* (*position_indicator_info_block*)
formats : 02₁₆ (“hours:minutes:seconds:frames”).
- *content_exit_point* (*position_indicator_info_block*)
formats : 02₁₆ (“hours:minutes:seconds:frames”).

7. Commands

The following table contains commands that a DVR-Blue Player/Recorder shall implement. These commands are a subset of category A commands in the AV/C Disc Subunit General, version 1.0[R5]. The command and response structure are as described in AV/C Disc Subunit General, version 1.0[R5], and AV/C Disc Subunit Enhancements for Hard Disk Drive, version 1.0[R6].

7.1 Mandatory commands in all profiles (AV/C Disc subunit)

Table 7.1 – DVR-Blue video player/recorder commands (Disc)

Command	Opcode	Defined ctypes			Comments
		C	S	N	
CONFIGURE	D1 ₁₆	X	–	–	Prepare the subunit for a recording or playback operation
ERASE	40 ₁₆	X*	–	–	Erase the entire AV contents of the disc subunit, just a specified track, or a specified portion of the disc storage space
IMPORT/EXPORT MEDIUM	C1 ₁₆	X	–	–	Put the disc into or remove it from the drive mechanism
PLAY	C3 ₁₆	X	–	–	Begin playing the disc (immediate response)
RECORD (NEW)	C2 ₁₆	X*	–	–	Record a streaming object (audio track, etc.)
SEARCH	50 ₁₆	X	–	–	Perform a relative or absolute search for the specified location on the media
SET PLUG ASSOCIATION	D3 ₁₆	X	–	–	Associate a List or Object with a Source or Destination Plug
STOP	C5 ₁₆	X	–	–	Stop the current operation

X : this command is mandatory for DVR-Blue Player/Recorder.

X* : this command is mandatory for DVR-Blue Recorder, but not implemented on DVR-Blue Player.

7.1.1 CONFIGURE

In a DVR-Blue player/recorder, the CONFIGURE command is used for plug configuration, and the *subfunction* 00₁₆ (“Reset to the Default Configuration”) and 01₁₆ (“Set the Configuration”) shall be supported.

7.1.2 ERASE

In a DVR-Blue recorder, *erase_type* 00₁₆ (“Erase All the Contents”) and 01₁₆ (“Erase the Specified Object”) shall be supported. Both the descriptor and AV contents are deleted.

7.1.3 IMPORT/EXPORT MEDIUM

The operation of this command is as specified in AV/C Disc Subunit General, version 1.0[R5].

7.1.4 PLAY

In a DVR-Blue player/recorder, the following *subfunctions* shall be supported:

- *Subfunction_1*: 75₁₆ (“Normal Play”) and 7D₁₆ (“Forward Pause”).

7.1.5 RECORD

In a DVR-Blue recorder the following *subfunctions* shall be supported:

- *Subfunction_1* : 75₁₆ (“Record at Normal Speed”) and 7D₁₆ (“Forward Pause”).
- *Subfunction_2* : 00₁₆ (“Record New”).

7.1.6 SEARCH - POSITION

In a DVR-Blue player/recorder, the *subfunction* 00₁₆ (“Search Type is Position”) shall be supported. The search indicator value is 00₁₆ i.e. search by *Relative_HMSF_count*.

NOTE — The search position may not be frame accurate and is only an estimate of the desired position.

7.1.7 SET PLUG ASSOCIATION

In a DVR-Blue player/recorder, the *subfunction* 01₁₆ (“Set a Specified List/Plug Association”) shall be supported. No default association is supported. The associated List ID descriptor identifier supports the *descriptor_type* 10₁₆ (“Object List Descriptor – specified by list ID”), the *descriptor_type* 20₁₆ (“Object Entry Descriptor – specified by Position”) and *descriptor_type* 21₁₆ (“Object Entry Descriptor – specified by an Object ID”).

7.1.8 STOP

The operation of this command is as specified in AV/C Disc Subunit General, version 1.0[R5]. The position where play operation ends shall be maintained by the DVR-Blue player/recorder at least until the DVR-Blue player/recorder enters a power off condition. This behavior emulates VCR operation.

7.2 Mandatory commands in all profiles (AV/C General)

DVR-Blue Player/Recorder supports the following additional commands. These commands are a subset of commands in the AV/C Digital Interface Command Set General, version 3.0[R3]

Table 7.2 – DVR-Blue video recorder commands (General)

Command	Opcode	Defined ctypes			Comments
		C	S	N	
OPEN DESCRIPTOR	08 ₁₆	X	–	–	Gains the right to access the descriptor.
OPEN INFO BLOCK	05 ₁₆	X	–	–	Gains the right to access the info block.
PLUG INFO	02 ₁₆	–	X	–	Information about serial bus and external plugs.
READ DESCRIPTOR	09 ₁₆	X	–	–	Read data from the descriptor.
READ INFO BLOCK	06 ₁₆	X	–	–	Read the specified info block.
WRITE INFO BLOCK	07 ₁₆	X*	–	–	Write data into specified info block.

X : this command is mandatory for DVR-Blue Player/Recorder.

X* : this command is mandatory for DVR-Blue Recorder, but may not implemented on DVR-Blue Player.

7.2.1 OPEN DESCRIPTOR

The operation of this command is as specified in AV/C Digital Interface Command Set General, version 3.0[R3].

7.2.2 OPEN INFO BLOCK

The operation of this command is as specified in Enhancements to the AV/C General Specification 3.0, version 1.0[R4].

NOTE — (Informative) This command is mandatory because the Enhancements to the AV/C General Specification 3.0, version 1.0 specifies its requirement when WRITE INFO BLOCK is mandatory. This requirement is expected to change in the next revision of the AV/C General or Enhancements to the AV/C General document. Consequently, it is recommended that OPEN DESCRIPTOR is the preferred method to gain access rights as the OPEN INFO BLOCK command would then become optional.

7.2.3 PLUG INFO

The DVR-Blue player is required to support one source plug at a minimum. The DVR-Blue recorder is required to support one source and one destination plug at a minimum. A DVR-Blue player/recorder may support additional source and destination plug(s) optionally.

7.2.4 READ DESCRIPTOR

The operation of this command is as specified in AV/C Digital Interface Command Set General, version 3.0.[R3]

7.2.5 READ INFO BLOCK

The operation of this command is as specified in Enhancements to the AV/C General Specification 3.0, version 1.0.[R4]

7.2.6 WRITE INFO BLOCK

The operation of this command is as specified in Enhancements to the AV/C General Specification 3.0, version 1.0[R3] with the following limitations. It is to be noted that DVR-Blue recorder creates Info Blocks at the time of object creation on “Record” command execution. It is to be noted that data in some of the info blocks can not be determined by the subunit. In these cases, such data may be initialized with values, which may be implementation dependent.

7.3 Additional mandatory commands for associated data list support.

Table 7.3 – DVR-Blue video commands (Disc)

Command	Opcode	Defined ctypes			Comments
		C	S	N	
OBJECT NUMBER SELECT	0D ₁₆	X	–	–	Select one or more objects for transmission, (response after completion)
RECORD OBJECT	56 ₁₆	X*	–	–	Record a non-streaming object (still image, etc.)

X : this command is mandatory for DVR-Blue Player/Recorder.

X* : this command is mandatory for DVR-Blue Recorder, but not implemented on DVR-Blue Player.

7.3.1 OBJECT NUMBER SELECT

The operation of this command is as specified in AV/C Disc Subunit General, version 1.0[R3].

The source_plug value FE₁₆ is not supported.

In a DVR-Blue player/recorder the following *subfunctions* shall be supported:

- *Subfunction*: D3₁₆ (“NEW”).
- *Subfunction*: C0₁₆ (“CLEAR”).

7.3.2 RECORD OBJECT

The operation of this command is as specified in AV/C Disc Subunit General, version 1.0[R5].

7.4 Additional mandatory commands for detailed performance list support.

Table 7.4 – additional DVR-Blue video commands for (General)

Command	Opcode	Defined ctypes			Comments
		C	S	N	
CREATE DESCRIPTOR	CC ₁₆	X*	–	–	Create new descriptors

X*: this command is mandatory for DVR-Blue Recorder, but may not implemented on DVR-Blue Player.

7.4.1 CREATE DESCRIPTOR

The CREATE DESCRIPTOR command is used for adding a new performance list to the root performance list. The operation of this command is as specified in Enhancements to the AV/C General Specification 3.0, version 1.0[R4].

This page is left intentionally blank

Annexes

Annex A: Bibliography (informative)

A.1 Bibliography

[B1] DVR-Blue System Description version 1.0

Annex B: New media type, list type, object type (normative)

B.1 Supported_media_type

The *supported_media_type* field identifies the type of media. The upper byte indicates the family, while the lower byte specifies more details about specifications or format. The following table illustrates the defined values for the supported media type:

Table B.1 – supported_media_type field

Supported_media_type (MSB)	Value	Supported_media_type (LSB)	Value
DVR-Blue	0B ₁₆	DVR-Blue	01 ₁₆

NOTE – This B.1 section is to be added to section 6 of the next revision of AV/C Disc Subunit General Specification version 1.0.[R5]

B.2 Media_type

The *media_type* field indicates the format of the information on this disc. The upper byte indicates the media family, while the lower byte specifies more detailed information. It is encoded as follows:

Table B.2 – media_type field

Media_type (MSB)	Value	Media_type (LSB)	Value
DVR-Blue	0B ₁₆	DVR-Blue	01 ₁₆
		Other	0E ₁₆
		Reserved for future extension	All other value

The *other* value for *media_type* (the LSB) means that the disc is something other than a recognized AV format. When this value is reported, it indicates that the subunit is able to recognize existence of the disc, but it contains information, which is not recognizable to the subunit.

NOTE – This B.2 section is to be added to section 9.3 of the next revision of AV/C Disc Subunit General Specification version 1.0.[R5]

B.3 List_type

Table B.3 – new list_type for DVR-Blue

list name	list type value	Comments
Associated data lists	87 ₁₆	The associated data lists contain associated data objects, which are specifying a set of associated data for an AV content. The root associated data list contains child directory objects, which are corresponding to associated data lists. The root and child lists have the same list type.

NOTE – This B.3 section is to be added to section 9.1 of the next revision of AV/C Disc Subunit General Specification version 1.0.[R5]

B.4 Object_type

Table B.4 – new object_type for DVR-Blue

Object name	Object type value	Comments
DVR-Blue Title database object	A0 ₁₆	This object represents a DVR-Blue specific database for AV track.

NOTE – This B.4 section is to be added to section 8.1 of the next revision of AV/C Disc Subunit General Specification version 1.0.[R5]

Annex C: New info block (normative)

C.1 DVR-Blue specific info block (88 16₁₆)

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	2	R	Compound length
00 01 ₁₆			
00 02 ₁₆	2	R	Info_block_type = (88 16 ₁₆ : DVR-Blue specific)
00 03 ₁₆			
00 04 ₁₆	2	R	Primary_fields_length = 00 05 ₁₆
00 05 ₁₆			
00 06 ₁₆	1	R	Audio stream format
00 07 ₁₆	2	R	Associated data list list_ID
00 08 ₁₆			
00 09 ₁₆	2	R	Reserved = 00 00 ₁₆
00 0A ₁₆			

Figure C.1 – DVR-Blue specific info block

The *Audio stream format* field shows audio encoding formats that are needed when this AV track is playback.

Address Offset	msb						Lsb
00 ₁₆	valid	MPEG	AC3	AAC	L-PCM	Reserved = 000 ₂	

Figure C.2 – audio stream format field

The *valid* flag indicates the validity of the lower 7 bits of this *audio stream format field*, and is valid when set to 1.

The *MPEG* flag indicates that an MPEG decoder is needed to play this AV track, and valid when set to 1.

The *AC3* flag indicates that a Dolby Digital decoder is needed to play this AV track, and valid when set to 1.

The *AAC* flag indicates that an AAC decoder is needed to play this AV track, and valid when set to 1.

The *L-PCM* flag indicates that a linear PCM decoder is needed to play this AV track, and valid when set to 1.

In case the DVR-Blue player/recorder does not have any information about audio decoder requirements, set all flags in the *Audio stream format* field to indicate “not valid”.

In case the information about audio decoder requirement is distinguished from the other info block (ex.. *Video Stream Format Subtype Info Block*), DVR-Blue player/recorder may set all the flags in *Audio stream format* field to indicate “not valid”.

NOTE – This field does NOT indicate that multi audio streams exist simultaneously within an AV object. The DVR-Blue format permits to combine two AV tracks that contain different audio stream format. (ex. AV object A with MPEG audio, AV object B with AAC).

The *associated data list list_ID* field shows the list ID of the associate data list. If there is no associated data for this video object, set this field as FF FF₁₆.

NOTE – This C.1 section is to be added to section 11 of the next revision of AV/C Disc Subunit General Specification version 1.0.[R5]

C.2 New image_format value(“JPEG”) of Image Format Info Block

Table C.1 – parameter for JPEG in image_format_info_block

Field	Value
Image_format	00 ₁₆ (JPEG)
Image_format_specific	00 ₁₆ (reserved)

NOTE – This C.2 section is to be added to section 6.12 of the next revision of Enhancements to the AV/C General Specification 3.0 Version 1.0[R4].

Annex D: Optional commands

The following table contains commands that a DVR-Blue Player/Recorder may implement. These commands are a subset of category A commands in the AV/C Disc Subunit General, version 1.0[R5] and AV/C Disc Subunit Enhancements for Hard Disk Drive, version 1.0[R6]. The command and response structure are as described in AV/C Disc Subunit General, version 1.0[R5] and AV/C Disc Subunit Enhancements for Hard Disk Drive, version 1.0[R6].

Table D.1 – DVR-Blue video recorder optional commands (DISC commands)

Command	Opcode	Defined ctypes			Comments
		C	S	N	
COMBINE	41 ₁₆	X	–	–	Combine two tracks into a single track
CREATE AV TRACK	D5 ₁₆	X	–	–	This command is used to create a track of a given size.
CREATE DESCRIPTOR	CC ₁₆	X*	–	–	Create new descriptors
DISC STATUS	D0 ₁₆	–	–	X	Request notification when the status of the subunit changes
DIVIDE	42 ₁₆	X	–	–	Separate a specified track into two tracks
INCREMENT OBJECT POSITION NUMBER	51 ₁₆	X	–	–	Divide a track while recording
MONITOR	C6 ₁₆	X	–	–	Listen to what is being recorded
OBJECT NUMBER SELECT	0D ₁₆	X*	–	–	Select one or more objects for transmission, (response after completion)
RECORD(LOOP)	C2 ₁₆	X	–	–	Records on the selected AV track in loop mode.
RECORD OBJECT	56 ₁₆	X*	–	–	Record a non-streaming object (still image, etc.)
REHEARSAL	C7 ₁₆	X	–	–	Playback a few positions continuously
UNDO	44 ₁₆	X	–	–	Undo the most recent editing operation(s)

X : this command is optional for DVR-Blue Player/Recorder.

X* : this command is optional for DVR-Blue Player/Recorder in some profiles.

D.1 COMBINE

In a DVR-Blue recorder, the COMBINE command is optional. The operation of this command is as specified in AV/C Disc Subunit General, version 1.0[R5]. In a DVR-Blue recorder, this command is used to combine two same object type objects.

D.2 CREATE AV TRACK

In a DVR-Blue recorder, the CREATE AV TRACK command is optional. When CREATE AV TRACK is supported it will create both a new AV track and a new AV Content Object descriptor. Right after an AV track is created by the CREATE AV TRACK command, the AV track is not recorded. When the controller issues a command to set a play back position in the not recorded part in the AV track, the subunit may return a REJECTED response. When the controller issues a command to set a recording position in the not

recorded part, except the beginning position of the not recorded part, the subunit also may return a REJECTED response.

D.3 CREATE DESCRIPTOR

In a DVR-Blue Player/Recorder, the CREATE DESCRIPTOR command is optional. When the controller adds a new performance list, the CREATE DESCRIPTOR is used. If supported, at least subfunction of 00_{16} (“Create a New Descriptor”) is required.

D.4 DISC STATUS

In a DVR-Blue Player/Recorder, the DISC STATUS command is optional. When DISC STATUS is supported, at least *subfunction* of 00_{16} (“Full Status”) is required. This will cause a notification to occur whenever the status descriptor changes.

D.5 DIVIDE

In a DVR-Blue Recorder, the DIVIDE command is optional. The operation of this command is as specified in AV/C Disc Subunit General, version 1.0[R5]. In a DVR-Blue player/recorder, this command is used to divide one object into two objects which object types are same.

D.6 INCREMENT OBJECT POSITION NUMBER

In a DVR-Blue Recorder, the INCREMENT OBJECT POSITION NUMBER command is optional. The operation of this command is as specified in AV/C Disc Subunit General, version 1.0[R5]. In a DVR-Blue player/recorder, this command is used to create a new object when recording is going on.

D.7 MONITOR

In a DVR-Blue Recorder Profile, the MONITOR command is optional. When MONITOR is supported, at the very least *subfunction* 60_{16} (“Monitoring Off”) and 70_{16} (“Monitoring On”) are required.

D.8 OBJECT NUMBER SELECT

If a DVR-blue player/recorder does not support associated data list, the OBJECT NUMBER SELECT command is optional.

D.9 RECORD(LOOP)

In a DVR-Blue Recorder, the loop *subfunction* is optionally supported.

Subfunction1 : 75_{16} (“Record at Normal Speed”) and $7D_{16}$ (“Forward Pause”).

Subfunction2 : 06_{16} (“Record Loop”).

D.10 RECORD OBJECT

If a DVR-blue recorder does not support the associated data list, the RECORD OBJECT command is optional.

D.11 REHEARSAL

In a DVR-Blue Player/Recorder, the RESERVE command is optional. The operation of this command is as specified in AV/C Disc Subunit General, version 1.0[R5].

D.12 UNDO

In a DVR-Blue Player/Recorder, the UNDO command is optional. When UNDO is supported, the *undo_status* and *difference* fields take on appropriate values based on implementation. If supported, UNDO will be implemented at least for the ERASE and WRITE INFO BLOCK commands.

Annex E: Descriptor examples (informative)

E.1 Profile ID examples

The DVR-Blue drive capability is described by combination of profile and record capability. The profile structure is shown in the following figure:

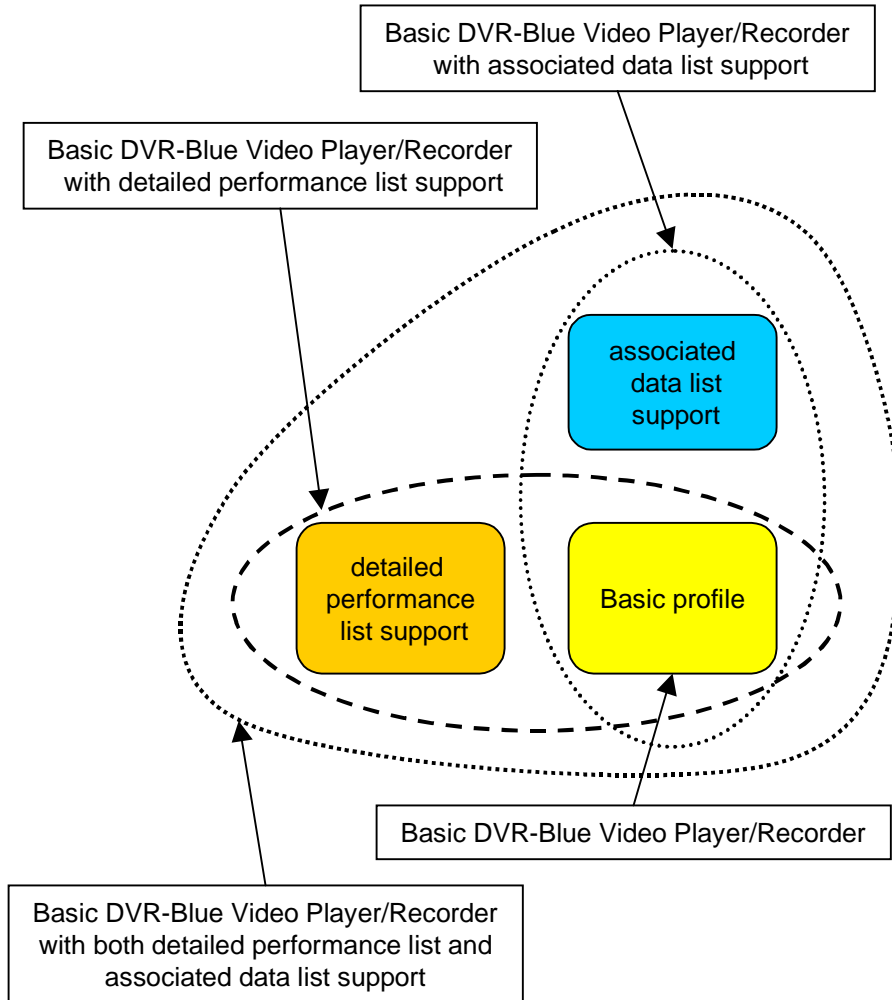


Figure E.1 – Basic DVR-Blue player/recorder profiles

In the DVR-Blue *supported_media_type_specification* field of Disc subunit Identifier Descriptor, the *Implementation_profile_ID* field and the *extended_profile_ID[i]* fields indicate profile and the *can_record* flag in *media_type_attributes* indicates record capability. Examples of *supported_media_type_specification* for the four profiles are shown in the following sections.

E.1.1 [Example-1] Basic DVR-Blue player/recorder

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆ 00 01 ₁₆	2	R	Supported_media_type = 0B 01 ₁₆ (DVR-Blue)
00 02 ₁₆	1	R	Implementation_profile_ID = 10 ₁₆
00 03 ₁₆	1	R	Media_type_attributes = 0000 000R ₂
00 04 ₁₆ 00 05 ₁₆	2	R	Type_dependent_length = 00 06 ₁₆
00 06 ₁₆	1	R	DVR-Blue version = 00 ₁₆
00 07 ₁₆ 00 08 ₁₆ 00 09 ₁₆ 00 0A ₁₆	4	R	Maximum_number_of_recordable_objects = XX XX XX XX ₁₆
00 0B ₁₆	1	R	Number_of_extended_implementation_profiles = 0

Figure E.2 – profile ID example-1 (supported_media_type_specification)

In this example, only one *Implementation_profile_ID*(10₁₆) is defined, so the extended implementation profiles are not used.

DVR-Blue player: The *can_record* flag (R) is set as 0₂.

DVR-Blue recorder: The *can_record* flag (R) is set as 1₂.

E.1.2 [Example-2] Basic DVR-Blue player/recorder with detailed performance list support

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	2	R	Supported_media_type = 0B 01 ₁₆ (DVR-Blue)
00 01 ₁₆			
00 02 ₁₆	1	R	Implementation_profile_ID = FF ₁₆
00 03 ₁₆	1	R	Media_type_attributes = 0000 000R ₂
00 04 ₁₆	2	R	Type_dependent_length = 00 0A ₁₆
00 05 ₁₆			
00 06 ₁₆	1	R	DVR-Blue version = FF ₁₆
00 07 ₁₆	4	R	Maximum_number_of_recordable_objects = XX XX XX XX ₁₆
00 08 ₁₆			
00 09 ₁₆			
00 0A ₁₆			
00 0B ₁₆	1	R	Number_of_extended_implementation_profiles = 2
00 0C ₁₆	1	R	Extended_implementation_profile_ID[0] = 10 ₁₆
00 0D ₁₆	1	R	Extended_DVR-Blue_version[0] = 00 ₁₆
00 0E ₁₆	1	R	Extended_implementation_profile_ID[1] = 11 ₁₆
00 0F ₁₆	1	R	Extended_DVR-Blue_version[1] = 00 ₁₆

Figure E.3 – profile ID example-2 (supported_media_type_specification)

In this example, two profile_IDs(10₁₆ and 11₁₆) are required, because “Basic DVR-Blue player/recorder with detailed performance list support profile” has upper compatibility with “Basic DVR-Blue player/recorder profile”.

So, the *Implementation_profile_ID* field is set to FF₁₆ and two extended implementation profiles are defined.

DVR-Blue player: The *can_record* flag (R) is set as 0₂.

DVR-Blue recorder: The *can_record* flag (R) is set as 1₂.

E.1.3 [Example-3] Basic DVR-Blue player/recorder with associated data list support

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	2	R	Supported_media_type = 0B 01 ₁₆ (DVR-Blue)
00 01 ₁₆			
00 02 ₁₆	1	R	Implementation_profile_ID = FF ₁₆
00 03 ₁₆	1	R	Media_type_attributes = 0000 000R ₂
00 04 ₁₆	2	R	Type_dependent_length = 00 0A ₁₆
00 05 ₁₆			
00 06 ₁₆	1	R	DVR-Blue version = FF ₁₆
00 07 ₁₆	4	R	Maximum_number_of_recordable_objects = XX XX XX XX ₁₆
00 08 ₁₆			
00 09 ₁₆			
00 0A ₁₆			
00 0B ₁₆	1	R	Number_of_extended_implementation_profiles = 2
00 0C ₁₆	1	R	Extended_implementation_profile_ID[0] = 10 ₁₆
00 0D ₁₆	1	R	Extended_DVR-Blue_version[0] = 00 ₁₆
00 0E ₁₆	1	R	Extended_implementation_profile_ID[1] = 12 ₁₆
00 0F ₁₆	1	R	Extended_DVR-Blue_version[1] = 00 ₁₆

Figure E.4 – profile ID example-3 (supported_media_type_specification)

In this example, two profile_IDs(10₁₆ and 12₁₆) are required, because “Basic DVR-Blue recorder with associated data list support profile” has upper compatibility with “Basic DVR-Blue recorder profile”. So, the *Implementation_profile_ID* field is set to FF₁₆ and two extended implementation profiles are defined.

DVR-Blue player: The *can_record* flag (R) is set as 0₂.

DVR-Blue recorder: The *can_record* flag (R) is set as 1₂.

E.1.4 [Example-4] Basic DVR-Blue player/recorder with both detailed performance list and associated data list support

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	2	R	Supported_media_type = 0B 01 ₁₆ (DVR-Blue)
00 01 ₁₆			
00 02 ₁₆	1	R	Implementation_profile_ID = FF ₁₆
00 03 ₁₆	1	R	Media_type_attributes = 0000 000R ₂
00 04 ₁₆	2	R	Type_dependent_length = 00 0C ₁₆
00 05 ₁₆			
00 06 ₁₆	1	R	DVR-Blue version = FF ₁₆
00 07 ₁₆	4	R	Maximum_number_of_recordable_objects = XX XX XX XX ₁₆
00 08 ₁₆			
00 09 ₁₆			
00 0A ₁₆			
00 0B ₁₆	1	R	Number_of_extended_implementation_profiles = 3
00 0C ₁₆	1	R	Extended_implementation_profile_ID[0] = 10 ₁₆
00 0D ₁₆	1	R	Extended_DVR-Blue_version[0] = 00 ₁₆
00 0E ₁₆	1	R	Extended_implementation_profile_ID[1] = 11 ₁₆
00 0F ₁₆	1	R	Extended_DVR-Blue_version[1] = 00 ₁₆
00 10 ₁₆	1	R	Extended_implementation_profile_ID[2] = 12 ₁₆
00 11 ₁₆	1	R	Extended_DVR-Blue_version[2] = 00 ₁₆

Figure E.5 – profile ID example-4 (supported_media_type_specification)

In this example, Three profile_IDs(10₁₆, 11₁₆ and 12₁₆) are required, because this profile include both “Basic DVR-Blue recorder with detailed performance list support profile” and “Basic DVR-Blue recorder with associated data list support profile”. So, the *Implementation_profile_ID* field is set to FF₁₆ and three extended implementation profiles are defined.

DVR-Blue player: The *can_record* flag (R) is set as 0₂.

DVR-Blue recorder: The *can_record* flag (R) is set as 1₂

E.2 contents descriptor example

This section shows examples of the DVR-Blue contents descriptor structure.

E.2.1 root contents list

The following figure is an example of root contents list:

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	2	R	Descriptor length = XX XX ₁₆
00 01 ₁₆			
00 02 ₁₆	1	R	List_type = 80 ₁₆
00 03 ₁₆	1	R	Attributes = 0001 1000 ₂
00 04 ₁₆	2	R	Size_of_list_specific_information = XX XX ₁₆
00 05 ₁₆			
00 06 ₁₆	2	R	Non_info_block_fields_length = 00 04 ₁₆
00 07 ₁₆			
00 08 ₁₆	1	R	Disc_subunit_list_attributes = 0000 00XX ₂
00 09 ₁₆	2	R	Media_type = 0B 01 ₁₆ ("DVR-Blue")
00 0A ₁₆			
00 0B ₁₆	1	R	Disc_recordable_information = XX01 0000 ₂
:	<i>Time Stamp Info Block for descriptor modification</i>		
:	<i>Default playlist info block</i>		
:	<i>Disc capacity info block</i>		
:	<i>Name info block</i>		
:	2	R	Number_of_entries = n
:			
:	<i>Object entry entry descriptor[0]</i>		
	:		
:	<i>Object entry entry descriptor[n-1]</i>		

Figure E.6 – root contents list

NOTE – When the RECORD NEW operation is started, the new object entry is added.

Address	Length, bytes	External Read/Write	Contents		
00 00 ₁₆ 00 01 ₁₆	2	R	Compound_length = 00 0B ₁₆		
00 02 ₁₆ 00 03 ₁₆	2	R	Info_block_type = 00 07 ₁₆ (Time Stamp Info Block)		
00 04 ₁₆ 00 05 ₁₆	2	R	Primary_fields_length = 00 07 ₁₆		
00 06 ₁₆ : 00 0C ₁₆	7	R	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="width: 20px;">*a</td> <td style="width: 20px;">*b</td> </tr> </table> Time_stamp_data (54bit unsigned integer counter type)	*a	*b
*a	*b				

*a : valid field

*b : stamp_type field = 1₂ (binary counter)

Figure E.7 – Time Stamp Info Block for descriptor modification

In the case of root contents list, the *valid* field = 1₂ (“valid”).

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆ 00 01 ₁₆	2	R	Compound_ length = 00 06 ₁₆
00 02 ₁₆ 00 03 ₁₆	2	R	Info_Block_type = 80 0B ₁₆
00 04 ₁₆ 00 05 ₁₆	2	R	Primary_fields_length = 00 02 ₁₆
00 06 ₁₆ 00 07 ₁₆	2	R	Default_playlist_ID = 30 00 ₁₆ (list ID of root contents list.)

Figure E.8 – Default playlist info block

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	2	R	Compound_length = 00 1A ₁₆
00 01 ₁₆			
00 02 ₁₆	2	R	Info_Block_type = 80 02 ₁₆
00 03 ₁₆			
00 04 ₁₆	2	R	Primary_fields_length = 00 16 ₁₆
00 05 ₁₆			
00 06 ₁₆	1	R	Capacity_format_indicator = 01h ("bytes")
00 07 ₁₆	2	R	Disc_total_playback_capacity_length = 00 05 ₁₆
00 08 ₁₆			
00 09 ₁₆	5	R	Disc_total_playback_capacity = XX XX XX XX XX ₁₆
:			
00 0D ₁₆			
00 0E ₁₆	2	R	Disc_maximum_recording_capacity_length = 00 05 ₁₆
00 0F ₁₆			
00 10 ₁₆	5	R	Disc_maximum_recording_capacity = XX XX XX XX XX ₁₆
:			
00 14 ₁₆			
00 15 ₁₆	2	R	Disc_remaining_recording_capacity_length = 00 05 ₁₆
00 16 ₁₆			
00 17 ₁₆	5	R	Disc_remaining_recording_capacity = XX XX XX XX XX ₁₆
:			
00 1B ₁₆			

Figure E.9 – Disc_capacity_info_block

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆ 00 01 ₁₆	2	R	Compound_length = XX XX ₁₆
00 02 ₁₆ 00 03 ₁₆	2	R	Info_block_type = 00 0B ₁₆ (name Info Block)
00 04 ₁₆ 00 05 ₁₆	2	R	Primary_fields_length = XX XX ₁₆
00 06 ₁₆	1	R	Name_data_reference_type = 00 ₁₆
00 07 ₁₆	1	R	Name_data_attributes = 0000 0011 ₂
00 08 ₁₆ 00 09 ₁₆	2	RWI	Maximum_number_of_characters = XX XX ₁₆
Character code info block			
00 0A ₁₆ 00 0B ₁₆	2	R	Compound_length = 00 06 ₁₆
00 0C ₁₆ 00 0D ₁₆	2	R	Info_block_type = 00 08 ₁₆ (character code Info Block)
00 0E ₁₆ 00 0F ₁₆	2	R	Primary_fields_length = 00 02 ₁₆
00 10 ₁₆	1	RWI	Character code type = XX ₁₆
00 11 ₁₆	1	RWI	Character code type specific = XX ₁₆
Language code info block			
00 12 ₁₆ 00 13 ₁₆	2	R	Compound_length = 00 07 ₁₆
00 14 ₁₆ 00 15 ₁₆	2	R	Info_block_type = 00 09 ₁₆ (language code Info Block)
00 16 ₁₆ 00 17 ₁₆	2	R	Primary_fields_length = 00 03 ₁₆
00 18 ₁₆	1	RWI	Language code type = XX ₁₆
00 19 ₁₆ 00 1A ₁₆	2	RWI	Language code type specific = XX XX ₁₆
Raw text info block			
00 1B ₁₆ 00 1C ₁₆	2	R	Compound_length = 00 04 ₁₆ + a
00 1D ₁₆ 00 1E ₁₆	2	R	Info_block_type = 00 0A ₁₆ (raw text Info Block)
00 1F ₁₆ 00 20 ₁₆	2	R	Primary_fields_length = a
00 21 ₁₆ :	a	RWI	Raw_text_data

Figure E.10 – Name info block

The following are cases of *character_code_info_block* and *Language_code_info_block* :

Case 1:

If a DVR-Blue player/recorder supports only one specific character code and language code, the DVR-Blue player/recorder sets the *character_code_info_block* field and *Language_code_info_block* field as the appropriate value and the controller can't modify these fields.

For example, the Japanese model for BS-digital, initial value is set as follows:

Character code type field: 08₁₆ (“Japanese EUC”)

Character code type specific field: 01₁₆ (“Japanese EUC for BS digital”)

Language code type field = 01₁₆ (“ISO 639”)

Language code type specific field = 6A61₁₆ (“ja”)

Case 2:

If a DVR-Blue player/recorder has a preferred character code and language code, the DVR-Blue player/recorder set the *character_code_info_block* field and *Language_code_info_block* field as the appropriate value and the controller may modify these fields.

Otherwise, the controller shall modify these fields and DVR-Blue player/recorder sets the *Character code type* field and *Character code type specific* field to FF₁₆, as initial value.

The initial value of the *primary_fields_length* field in *raw_text_info_block* is 0.

The value of each field in *name_info_block* is revised by a controller.

E.2.2 Video contents object entry

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	2	R	Descriptor length = XX XX ₁₆
00 01 ₁₆			
00 02 ₁₆	1	R	Entry_type = 83 ₁₆ (Video Content object)
00 03 ₁₆	1	R	Attributes = 0000 1000 ₂
00 04 ₁₆	2	R	Object ID
00 05 ₁₆			
00 06 ₁₆	2	R	Size_of_entry_specific_information = XX XX ₁₆
00 07 ₁₆			
Entry specific information			
00 08 ₁₆	2	R	Non_info_block_fields_length = 00 01 ₁₆
00 09 ₁₆			
00 0A ₁₆	1	R	Disc_subunit_object_attributes = 0000 00XX ₂
:	Video signal mode info block		
:	Video Stream Format Subtype Info Block		
:	Size Indicator Info Block		
:	Time Stamp Info Block		
:	DVR-Blue specific info block		
:	Program Attributes Info Block		
:	Name info block		
:	(Other optional info blocks)		

Figure E.11 – Video contents object entry example

The following info blocks are mandatory for Player/Recorder DVR-Blue and should be described in this order:

- 1) Video signal mode info block
- 2) Video Stream Format Subtype Info Block
- 3) Size Indicator Info Block
- 4) Time Stamp Info Block
- 5) DVR-Blue specific info block
- 6) Program Attributes Info Block

7) Name info block

Other info blocks are optional and shall be described in the *optional info blocks* area.

When “RECORD NEW” operation is started, a new video object entry is added at the end of root contents list with all mandatory info blocks (and optional info blocks, if they exist).

NOTE – The DVR-Blue player/recorder does not support any method to add new info blocks, thus when a new object entry is created, all necessary info blocks shall be provided with a new entry.

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆ 00 01 ₁₆	2	R	Compound_length = 00 09 ₁₆
00 02 ₁₆ 00 03 ₁₆	2	R	Info_block_type = 88 12 ₁₆ (Video Signal Mode Info Block)
00 04 ₁₆ 00 05 ₁₆	2	R	Primary_fields_length = 00 05 ₁₆
00 06 ₁₆	1	R	Video_signal_mode = 01 ₁₆ (MPEG2-TS)
00 07 ₁₆	1	R	Frame_rate_code & reserved bits = XXXX XXXX ₂
00 08 ₁₆ 00 09 ₁₆ 00 0A ₁₆	3	R	Video_signal_bit_rate = XX XX XX ₁₆

Figure E.12 – Video signal mode info block

The *Video_signal_mode* field is specified by DVR-Blue player/recorder. Controller can not modify this field. Write command on this field is accepted, but not affected.

All fields in *video_signal_mode_info_block* are maintained by DVR-Blue player/recorder. When recording, the plug configuration of the destination plug is reflected in this info block.

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆ 00 01 ₁₆	2	R	Compound_length = 00 22 ₁₆
00 02 ₁₆ 00 03 ₁₆	2	R	Info_block_type = 88 13 ₁₆ (Video Stream Format Subtype Info Block)
00 04 ₁₆ 00 05 ₁₆	2	R	Primary_fields_length = 00 1E ₁₆
00 06 ₁₆	1	RWI	Validity_flags = XXXX XX00 ₂
00 07 ₁₆ : 00 0A ₁₆	4	RWI	Format identifier = XX XX XX XX ₁₆
00 0B ₁₆ 00 0C ₁₆	2	RWI	Original Network ID = XX XX ₁₆
00 0D ₁₆ 00 0E ₁₆	2	RWI	Transport Stream ID = XX XX ₁₆
00 0F ₁₆ 00 10 ₁₆	2	RWI	Service ID = XX XX ₁₆
00 11 ₁₆ 00 12 ₁₆ 00 13 ₁₆	3	RWI	Country Code = XX XX XX ₁₆
00 14 ₁₆ : 00 23 ₁₆	16	RWI	Stream_format_name = XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX ₁₆

Figure E.13 – Video Stream Format Subtype Info Block

The information in the *Video_stream_format_subtype* info block of a DVR-Blue recorder is revised by a controller. When new video contents object is recorded, it is strongly recommend for the controller to set effective value in this info block.

In the case where a DVR-Blue recorder can not set an appropriate value for the *stream_format_subtype* info block, it uses initial value of each field as follows:

Validity_flags field

DVR-Blue player/recorder sets the initial value to 00₁₆. (all flags are set to 0)

Format Identifier field

DVR-Blue player/recorder sets the initial value to all FF₁₆.

Network information field

DVR-Blue player/recorder sets the initial value to all FF₁₆.

Stream_format_name field

DVR-Blue player/recorder sets the initial value to all FF₁₆.

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	2	R	Compound_length = 00 0A ₆
00 01 ₁₆			
00 02 ₁₆	2	R	Info_block_type = 00 01 ₁₆ (size indicator Info Block)
00 03 ₁₆			
00 04 ₁₆	2	R	Primary_fields_length = 00 06 ₁₆
00 05 ₁₆			
00 06 ₁₆	1	R	size_indicator_type = 00 ₁₆ ("H_M_S_F")
00 07 ₁₆	2	R	Hour = XX XX ₁₆
00 08 ₁₆			
00 09 ₁₆	1	R	Minutes = XX ₁₆
00 0A ₁₆	1	R	Seconds = XX ₁₆
00 0B ₁₆	1	R	Frames = XX ₁₆

Figure E.14 – Size Indicator Info Block ("H_M_S_F")

The value of Size Indicator Info Block ("H_M_S_F") is set by a DVR-Blue player/recorder. Accuracy of size information is subunit dependent.

"no information" value for *H_M_S_F* field : all FF₁₆

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	2	R	Compound_length = 00 0B ₁₆
00 01 ₁₆			
00 02 ₁₆	2	R	Info_block_type = 00 04 ₁₆ (Time Stamp Info Block)
00 03 ₁₆			
00 04 ₁₆	2	R	Primary_fields_length = 00 07 ₁₆
00 05 ₁₆			
00 06 ₁₆	2	RWI	Year = XX XX ₁₆
00 07 ₁₆			
00 08 ₁₆	1	RWI	Month = XX ₁₆
00 09 ₁₆	1	RWI	Day = XX ₁₆
00 0A ₁₆	1	RWI	Hours = XX ₁₆
00 0B ₁₆	1	RWI	Minutes = XX ₁₆
00 0C ₁₆	1	RWI	Seconds = XX ₁₆

Figure E.15 – Time Stamp Info Block (Content_creation_date_and_time)

The value of the *content_creation_date_and_time* info block is set by a controller.

In the case where a DVR-Blue player/recorder can maintain the *contents creation date and time* field, initial value is appropriate date and time. Otherwise, a DVR-Blue recorder sets the initial value to all FF₁₆.

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	2	R	Compound length = 00 09 ₁₆
00 01 ₁₆			
00 02 ₁₆	2	R	Info_block_type = (88 16 ₁₆ : DVR-Blue specific)
00 03 ₁₆			
00 04 ₁₆	2	R	Primary_fields_length = 00 05 ₁₆
00 05 ₁₆			
00 06 ₁₆	1	R	Audio stream format = XXXX XXXX ₂
00 07 ₁₆	2	R	Associated data list list_ID = XX XX ₁₆
00 08 ₁₆			
00 09 ₁₆	2	R	Reserved = 00 00 ₁₆
00 0A ₁₆			

Figure E.16 – DVR-Blue specific info block

The value of each field in the DVR-Blue specific info block is set by a player/recorder DVR-Blue player/recorder.

In the case where the DVR-Blue Recorder does not have any information about audio stream, set the *Audio stream format* field to 0000 0000₂ (“not valid”) as the initial value.

In the case of an associated data list supported profile, the *Associated data list list_ID* field contains an appropriate list ID, otherwise (i.e. associated data list is not supported) set this field as FF FF₁₆.

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆ 00 01 ₁₆	2	R	Compound_length = XX XX ₁₆
00 02 ₁₆ 00 03 ₁₆	2	R	Info_block_type = 88 14 ₁₆ (Program Attributes Info Block)
00 04 ₁₆ 00 05 ₁₆	2	R	Primary_fields_length = XX XX ₁₆
Program starting date & time			
00 06 ₁₆ 00 07 ₁₆	2	RWI	Year = XX XX ₁₆
00 08 ₁₆	1	RWI	Month = XX ₁₆
00 09 ₁₆	1	RWI	Day = XX ₁₆
00 0A ₁₆	1	RWI	Hour = XX ₁₆
00 0B ₁₆	1	RWI	Minute = XX ₁₆
00 0C ₁₆	1	RWI	Second = XX ₁₆
Program duration			
00 0D ₁₆ 00 0E ₁₆	2	RWI	Hours = XX XX ₁₆
00 0F ₁₆	1	RWI	Minutes= XX ₁₆
00 10 ₁₆	1	RWI	Seconds= XX ₁₆
00 11 ₁₆	1	RWI	Parental rating = XXXX XXXX ₂
00 12 ₁₆	1	RWI	Program modes = XXXX XXXX ₂
Service name			
00 13 ₁₆	1	RWI	Character code type = XX ₁₆
00 14 ₁₆	1	RWI	Character code type specific = XX ₁₆
00 15 ₁₆	1	RWI	Maximum service name length = XX ₁₆
00 16 ₁₆	1	RWI	Service name length = y
	y	RWI	(Service name)

Figure E.17 – Program Attributes Info Block

In the case of the DVR-Blue player/recorder, the *REPLAYED* field may be maintained by DVR-Blue player/recorder and other fields of the program attribute info block are revised by a controller.

In case the DVR-Blue recorder can not set an appropriate value for the *program attributes* info block, the DVR-Blue recorder sets initial value of each field as the following:

Program starting date & time field

DVR-Blue player/recorder sets the initial value to all FF₁₆.

Program duration field

DVR-Blue player/recorder sets the initial value to all FF₁₆.

Parental rating field

DVR-Blue player/recorder sets the initial value of rating field to 1₂. (“no restriction by parental rating”)

Program mode field & HD/SD field

DVR-Blue player/recorder sets the initial value to 11₂. ("No information")

REPLAYED field

In case a DVR-Blue player/recorder does not have the capability to maintain the *REPLAYED* field, the DVR-Blue player/recorder sets the initial value to 11₂ ("no information") and does not change this field by itself.

In case a DVR-Blue player/recorder has the capability to maintain the *REPLAYED* field, the DVR-Blue player/recorder sets the initial value to 00₂ ("not replayed") when Video object entry is created, and changes to 01₂ ("replayed") when the video object is playback.

The Write command on this field is accepted, but may not be affected. (depend on subunit implementation)

Reserved Field

The *Reserved* field is for a future extension, and shall be set to 0000₂.

Service name field*Case 1:*

If a DVR-Blue player/recorder supports only one specific character code, the DVR-Blue player/recorder sets the *Character code type* field and *Character code type specific* field as the appropriate value and the controller can't modify these fields.

For example, the Japanese model for BS-digital, initial value is set as follows;

Character code type field: 08₁₆ ("Japanese EUC")

Character code type specific field: 01₁₆ ("Japanese EUC for BS digital")

Case 2:

If a DVR-Blue player/recorder has a preferred character code, the DVR-Blue player/recorder sets the *Character code type* field and *Character code type specific* field as appropriate value and the controller can modify these fields.

Otherwise, the controller shall modify these fields and DVR-Blue player/recorder sets the *Character code type* field and *Character code type specific* field to FF₁₆, as initial value.

The *Maximum service name length* field is maintained by DVR-Blue player/recorder. The controller can not modify this field. The Write command on this field is accepted, but not affected.

DVR-Blue player/recorder sets the initial value of the *Service name length* field to 00₁₆.

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆ 00 01 ₁₆	2	R	Compound_length = 00 XX ₁₆
00 02 ₁₆ 00 03 ₁₆	2	R	Info_block_type = 00 0B ₁₆ (name Info Block)
00 04 ₁₆ 00 05 ₁₆	2	R	Primary_fields_length = 00 XX ₁₆
00 06 ₁₆	1	R	Name_data_reference_type = 00 ₁₆
00 07 ₁₆	1	R	Name_data_attributes = 0000 0011 ₂
00 08 ₁₆ 00 09 ₁₆	2	RWI	Maximum_number_of_characters = XX XX ₁₆
Character code info block			
00 0A ₁₆ 00 0B ₁₆	2	R	Compound_length = 00 06 ₁₆
00 0C ₁₆ 00 0D ₁₆	2	R	Info_block_type = 00 08 ₁₆ (character code Info Block)
00 0E ₁₆ 00 0F ₁₆	2	R	Primary_fields_length = 00 02 ₁₆
00 10 ₁₆	1	RWI	Character code type = XX ₁₆
00 11 ₁₆	1	RWI	Character code type specific = XX ₁₆
Language code info block			
00 12 ₁₆ 00 13 ₁₆	2	R	Compound_length = 00 07 ₁₆
00 14 ₁₆ 00 15 ₁₆	2	R	Info_block_type = 00 09 ₁₆ (language code Info Block)
00 16 ₁₆ 00 17 ₁₆	2	R	Primary_fields_length = 00 03 ₁₆
00 18 ₁₆	1	RWI	Language code type = XX ₁₆
00 19 ₁₆ 00 1A ₁₆	2	RWI	Language code type specific = XX XX ₁₆
Raw text info block			
00 1B ₁₆ 00 1C ₁₆	2	R	Compound_length = 00 04 ₁₆ + a
00 1D ₁₆ 00 1E ₁₆	2	R	Info_block_type = 00 0A ₁₆ (raw text Info Block)
00 1F ₁₆ 00 20 ₁₆	2	R	Primary_fields_length = a
00 21 ₁₆ :	a	RWI	(Raw_text_data)

Figure E.18 – Name info block

The cases of *character_code_info_block* and *Language_code_info_block* are as follows : .

Case 1:

If a DVR-Blue player/recorder supports only one specific character code and language code, the DVR-Blue player/recorder sets the *character_code_info_block* field and *Language_code_info_block* field as the appropriate value and the controller can't modify these fields.

For example, the Japanese model for BS-digital, initial value is set as follows;

Character code type field: 08₁₆ (“Japanese EUC”)
 Character code type specific field: 01₁₆ (“Japanese EUC for BS digital”)
 Language code type field = 01₁₆ (ISO 639)
 Language code type specific field = 6A61₁₆ (“ja”)

Case 2:

If a DVR-Blue player/recorder has a preferred character code and language code, the DVR-Blue player/recorder sets the *character_code_info_block* field and *Language_code_info_block* field as the appropriate value and the controller may modify these field.

Otherwise, the controller should modify these fields and DVR-Blue player/recorder sets the *Character code type* field and *Character code type specific* field to FF₁₆, as the initial value.

The value of each field in *name_info_block* is set by a controller.

The initial value of the *primary_fields_length* field in *raw_text_info_block* is 0.

E.2.3 Root performance list

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	2	R	Descriptor length = 00 XX ₁₆
00 01 ₁₆			
00 02 ₁₆	1	R	List_type = 84 ₁₆
00 03 ₁₆	1	R	Attributes = 0001 1000 ₂
00 04 ₁₆	2	R	Size_of_list_specific_information = 00 XX ₁₆
00 05 ₁₆			
00 06 ₁₆	2	R	Non_info_block_fields_length = 00 01 ₁₆
00 07 ₁₆			
00 08 ₁₆	1	R	Disc_subunit_list_attributes = 0000 00XX ₂
:	<i>Time Stamp Info Block for descriptor modification</i>		
00 22 ₁₆	2	R	Number_of_entries = n
00 23 ₁₆			
:	<i>Object entry entry descriptor[0]</i>		
:	:		
:	<i>Object entry entry descriptor[n-1]</i>		

Figure E.19 – root performance list

The structure of time stamp info block is the same as time stamp info block in the root contents list.(see Figure E.7) The *valid* field is set to 1₂. (“valid”)

E.2.3.1 Entry for root performance list (child directory object)

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	2	R	Descriptor length = XX XX ₁₆
00 01 ₁₆			
00 02 ₁₆	1	R	Entry_type = 90 ₁₆ (child directory object)
00 03 ₁₆	1	R	Attributes = 0010 1000 ₂
00 04 ₁₆	2	R	Child List ID of performance list
00 05 ₁₆			
00 06 ₁₆	2	R	Object ID
00 07 ₁₆			
00 08 ₁₆	2	R	Size_of_entry_specific_information
00 09 ₁₆			
Entry_specific_information			
00 0A ₁₆	2	R	Non_info_block_fields_length = 00 01 ₁₆
00 0B ₁₆			
00 0C ₁₆	1	R	Disc_subunit_object_attributes = 0000 00XX ₂
:	Video signal mode info block		
:	Video Stream Format Subtype Info Block		
:	Size Indicator Info Block		
:	Time Stamp Info Block		
:	DVR-Blue specific info block		
:	Program Attributes Info Block		
:	Name info block		
(Other optional info blocks)			

Figure E.20 – object entry in root performance list

The following info blocks are mandatory for Player/Recorder DVR-Blue and should be described in this order:

- 1) Video signal mode info block
- 2) Video Stream Format Subtype Info Block
- 3) Size Indicator Info Block
- 4) Time Stamp Info Block
- 5) DVR-Blue specific info block
- 6) Program Attributes Info Block

7) Name info block

The initial conditions of these info blocks are the same as described in E.2.2 Video contents object entry.

Other info blocks are optional, and shall be described in optional info blocks area, if they exist.

E.2.3.2 Performance list

In the case of a DVR-Blue player/recorder with detailed performance list support, the performance list structure is as follows:

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	2	R	Descriptor length = XX XX ₁₆
00 01 ₁₆			
00 02 ₁₆	1	R	List_type = 84 ₁₆
00 03 ₁₆	1	R	Attributes = 0000 1000 ₂
00 04 ₁₆	2	R	Size_of_list_specific_information = XX XX ₁₆
00 05 ₁₆			
00 06 ₁₆	2	R	Non_info_block_fields_length = 00 01 ₁₆
00 07 ₁₆			
00 08 ₁₆	1	R	Disc_subunit_list_attributes = 0000 00XX ₂
:	Time Stamp Info Block for descriptor modification		
:	Size_indicator_info_block (total duration of all performances in list)		
:	2	R	Number_of_entries = n
:			
:	Object entry entry descriptor[0]		
:	:		
:	Object entry entry descriptor[n-1]		

Figure E.21 – performance list

The structure of time stamp info block is the same as time stamp info block in the root contents list.(see Figure E.7) The *valid* field is set to 1₂. (“valid”)

The structure of size_indicator_info_block is the same as size_indicator_info_block in the video contents object. (see Figure E.14)

In the case of a DVR-Blue player/recorder without detailed performance list support, the entry structure in the performance list is as follows:

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	2	R	Descriptor length = XX XX ₁₆
00 01 ₁₆			
00 02 ₁₆	1	R	List_type = 84 ₁₆
00 03 ₁₆	1	R	Attributes = 0100 0000 ₂
00 04 ₁₆	2	R	Size_of_list_specific_information = XX XX ₁₆
00 05 ₁₆			
00 06 ₁₆	2	R	Non_info_block_fields_length = 00 01 ₁₆
00 07 ₁₆			
00 08 ₁₆	1	R	Disc_subunit_list_attributes = 0000 0000 ₂
:	<i>Time Stamp Info Block for descriptor modification</i>		
:	<i>size_indicator_info_block (total duration of all performances in list)</i>		
:	2	R	Number_of_entries = 00 00 ₁₆
:			

Figure E.22 – performance list for without detailed performance list support profile

Set the *skip* bit in *list attribute* to 1₂ to indicate this list does not contain detailed information.

The structure of time stamp info block is the same as time stamp info block in the root contents list.(see Figure E.7) The *valid* field is set to 0₂. (“not valid”) The *time stamp data* field is set as 3F FF FF FF FF FF FF₁₆.

The structure of size_indicator_info_block is the same as size_indicator_info_block in the video contents object. (see Figure E.14) All bytes in the *H_M_S_F* field are set as FF₁₆ to indicate “not valid”.

NOTE – In case the detailed performance list is not supported, the performance lists are dummy lists to avoid miss-linked list ID. A controller should not read/write these performance lists.

E.2.3.3 Performance entry in performance list

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	2	R	Descriptor length field
00 01 ₁₆			
00 02 ₁₆	1	R	Entry_type= 91 ₁₆ (performance entry)
00 03 ₁₆	1	R	Attribute = 0000 1000 ₂
00 04 ₁₆	2	R	Size_of_entry_specific_information
00 05 ₁₆			
00 06 ₁₆	2	R	Non_info_block_fields_length = 00 01 ₁₆
00 07 ₁₆			
00 08 ₁₆	1	R	Disc_subunit_object_attributes = 0000 00XX ₂
Descriptor reference info block			
00 09 ₁₆	2	R	Compound_length = 00 0A ₁₆
00 0A ₁₆			
00 0B ₁₆	2	R	Info_block_type = 00 0F ₁₆ (descriptor_reference_info_block)
00 0C ₁₆			
00 0D ₁₆	2	R	Primary_fields_length = 00 06 ₁₆
00 0E ₁₆			
Descriptor identifier			
00 0F ₁₆	1	RWI	Descriptor_type = 21 ₁₆
00 10 ₁₆	2	RWI	Root_list_ID
00 11 ₁₆			
00 12 ₁₆	1	RWI	List_type
00 13 ₁₆	2	RWI	Object_ID
00 14 ₁₆			
Position indicator info block (content entry point)			
00 15 ₁₆	2	R	Compound_length = 00 0A ₁₆
00 16 ₁₆			
00 17 ₁₆	2	R	Info_block_type = 00 02 ₁₆ (position indicator info block)
00 18 ₁₆			
00 19 ₁₆	2	R	Primary_fields_length = 00 06 ₁₆
00 1A ₁₆			
00 1B ₁₆	1	RW	Indicator type = 02 ₁₆ (absolute HMSF)
00 1C ₁₆	2	RW	Hours
00 1D ₁₆			
00 1E ₁₆	1	RW	Minutes
00 1F ₁₆	1	RW	Seconds
00 20 ₁₆	1	RW	Frames
Position indicator info block (content exit point)			
00 21 ₁₆	2	R	Compound_length = 00 0A ₁₆
00 22 ₁₆			
00 23 ₁₆	2	R	Info_block_type = 00 02 ₁₆ (position indicator info block)
00 24 ₁₆			
00 25 ₁₆	2	R	Primary_fields_length = 00 06 ₁₆
00 26 ₁₆			
00 27 ₁₆	1	RW	Indicator type = 02 ₁₆ (absolute HMSF)
00 28 ₁₆	2	RW	Hours
00 29 ₁₆			
00 2A ₁₆	1	RW	Minutes

00 2B ₁₆	1	RW	Seconds
00 2C ₁₆	1	RW	Frames

Figure E.23 – performance entry

E.2.4 Root Associated data list

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	2	R	Descriptor length = XX XX ₁₆
00 01 ₁₆			
00 02 ₁₆	1	R	List_type = 87 ₁₆
00 03 ₁₆	1	R	Attributes = 0000 1000 ₂
00 04 ₁₆	2	R	Size_of_list_specific_information = XX XX ₁₆
00 05 ₁₆			
00 06 ₁₆	2	R	Non_info_block_fields_length = 00 01 ₁₆
00 07 ₁₆			
00 08 ₁₆	1	R	Disc_subunit_list_attributes = 0000 0000 ₂
:	<i>Time Stamp Info Block for descriptor modification</i>		
:	2	R	Number_of_entries = n
:			
:	<i>Object entry entry descriptor[0]</i>		
:	:		
:	<i>Object entry entry descriptor[n-1]</i>		

Figure E.24 – root associated data list

NOTE – In the case of DVR-Blue player/recorder, the root associated list is a dummy list for compatibility with AV/C general 4.0 specification. A controller should not read/write the root associated data list.

The structure of time stamp info block is the same as time stamp info block in the root contents list.(see Figure E.7) The *valid* field is set to 1₂. (“valid”)

E.2.4.1 Object entry in root as associated data list

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	2	R	Descriptor length = XX XX ₁₆
00 01 ₁₆			
00 02 ₁₆	1	R	Entry_type = 90 ₁₆ (child directory object)
00 03 ₁₆	1	R	Attributes = 0010 1000 ₂
00 04 ₁₆	2	R	Child List ID for associated data list
00 05 ₁₆			
00 06 ₁₆	2	R	Size_of_entry_specific_information
00 07 ₁₆			
<i>Entry_specific_information</i>			
00 08 ₁₆	2	R	Non_info_block_fields_length = 00 01 ₁₆
00 09 ₁₆			
00 0A ₁₆	1	R	Disc_subunit_object_attributes = 0000 0000 ₂
:	<i>(optional info blocks)</i>		

Figure E.25 – object entry in root associated data list

E.2.4.2 Associated data list

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	2	R	Descriptor length = XX XX ₁₆
00 01 ₁₆			
00 02 ₁₆	1	R	List_type = 87 ₁₆
00 03 ₁₆	1	R	Attributes = 0001 1000 ₂
00 04 ₁₆	2	R	Size_of_list_specific_information = XX XX ₁₆
00 05 ₁₆			
00 06 ₁₆	2	R	Non_info_block_fields_length = 00 01 ₁₆
00 07 ₁₆			
00 08 ₁₆	1	R	Disc_subunit_list_attributes = 0000 00XX ₂
:	<i>Time Stamp Info Block for descriptor modification</i>		
:	2	R	Number_of_entries = n
:			
:	<i>Object entry entry descriptor[0]</i>		
:	:		
:	<i>Object entry entry descriptor[n-1]</i>		

Figure E.26 – associated data list

NOTE – Objects in the associated data list shall have *object_ID*.

The structure of time stamp info block is the same as time stamp info block in the root contents list.(see Figure E.7) The *valid* field is set to 1₂. (“valid”)

NOTE — In the case of an *associated data list support* profile, the associated data list is created when a new video contents object entry is created in the root contents list (i.e. new video object is created by RECORD NEW operation) or when a new child directory object is created in the root performance list (i.e. new performance list is created by CREATE DESCRIPTOR operation.)

The *List_ID* value of the associated data list is reflected in the *associated data list ID* field of DVR-Blue specific info block in the appropriate video contents object entry or the appropriate child directory object.

E.3 Status descriptor

This section shows examples of the DVR-Blue status descriptor structure.

E.3.1 General_disc_subunit_status_info_block

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆ 00 01 ₁₆	2	R	Compound_length = 00 0D ₁₆
00 02 ₁₆ 00 03 ₁₆	2	R	Info_block_type = 88 00 ₁₆
00 04 ₁₆ 00 05 ₁₆	2	R	Primary_fields_length = 00 09 ₁₆
00 06 ₁₆ :			<i>Media_and_edit_status_info_block</i>

Figure E.27 – General_disc_subunit_status_info_block

Address Offset	Msb				Isb
00 ₁₆ 01 ₁₆	Compound_length = 00 07 ₁₆				
02 ₁₆ 03 ₁₆	Info_block_type = 88 04 ₁₆ (media_and_edit_status infoblock)				
04 ₁₆ 05 ₁₆	Primary_fields_length = 00 03 ₁₆				
06 ₁₆	Disc_in_drive = XX ₂	Error_condition = XX ₂	Reserved = 0000 ₂		
07 ₁₆	Undo_status = XX ₁₆				
08 ₁₆	Difference = 0 ₂	Auto_update = 1 ₂	Reserved = 00 0000 ₂		

Figure E.28 – Media_and_edit_status_info_block

The *Disc_in_drive* field indicates media existence. A DVR-blue player/recorder shall maintain this field corresponding to existence of disc medium.

NOTE – In the above example, this subunit does not support the temporary contents list structure, so the *Difference* field is set as 0₂ (“no difference”) and the *Auto_update* field is set as 1₂ (“not support temporary contents list and update always occur”). If a subunit supports the temporary contents list structure, the *Difference* field and *Auto_update* field shall indicate conditions of the temporary contents list.

E.3.2 Destination Plug Status Area Info Block

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆ 00 01 ₁₆	2	R	Descriptor length field = XX XX ₁₆
00 02 ₁₆ 00 03 ₁₆	2	R	Info_block_type = 88 01 ₁₆
00 04 ₁₆ 00 05 ₁₆	2	R	Primary_fields_length
00 06 ₁₆	1	R	Number_of_destination_plugs = n
00 07 ₁₆ :			<i>Plug_status_info_block[0]</i> : <i>plug_status_info_block[n-1]</i>

Figure E.29 – Destination Plug Status Area Info Block

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆ 00 01 ₁₆	2	R	Descriptor length field = XX XX ₁₆
00 02 ₁₆ 00 03 ₁₆	2	R	Info_block_type = 88 05 ₁₆
00 04 ₁₆ 00 05 ₁₆	2	R	Primary_fields_length = XX XX ₁₆
00 06 ₁₆	1	R	Plug_number
<i>Nested structures</i>			
00 07 ₁₆ : : : : :			<i>Operating Mode Info Block (88 06₁₆)</i> <i>Plug Configuration Info Block (88 07₁₆)</i> <i>Position Info Block (00 03₁₆)</i>

Figure E.30 – Destination plug *Plug_status_info_block*

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆ 00 01 ₁₆	2	R	Descriptor length field = 00 07 ₁₆
00 02 ₁₆ 00 03 ₁₆	2	R	Info_block_type = 88 06 ₁₆
00 04 ₁₆ 00 05 ₁₆	2	R	Primary_fields_length = 00 03 ₁₆
00 06 ₁₆	1	R	Operating_mode
00 07 ₁₆ 00 08 ₁₆	2	R	Operating_mode_specific_information (FF ₁₆ pad bytes if necessary)

Figure E.31 – Operating Mode Info Block

The size of the *operating_mode_specific_information* field is 2 bytes and DVR-Blue recorder may pad FF₁₆ if necessary.

NOTE – In the case where the *Operating_mode* field is set to FF₁₆(“SUSPENDED”), this status indicates that the plug is not available.

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆ 00 01 ₁₆	2	R	Descriptor length field = XX XX ₁₆
00 02 ₁₆ 00 03 ₁₆	2	R	Info_block_type = 88 07 ₁₆
00 04 ₁₆ 00 05 ₁₆	2	R	Primary_fields_length = XX XX ₁₆
00 06 ₁₆	1	R	AV_object_type = XX ₁₆
:			<i>Object and plug type specific information</i>
:			

Figure E.32 – Plug Configuration Info Block

When a destination plug is configured for an MPEG2-TS object, the *AV_object_type* field is set to 83₁₆(“Video Content object”) and *object_and_plug_type_specific_information* is a fixed length structure (6 bytes) as follows:

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	1	R	*a Reserved = 000 0000 ₂
00 01 ₁₆	1	R	Video_signal_mode = 01 ₁₆ (MPEG2-TS)
00 02 ₁₆	1	R	Frame_rate Reserved
00 03 ₁₆	3	R	Video_signal_bit_rate (x 10 ³ bit/sec)
00 04 ₁₆			
00 05 ₁₆			

*a Increment_position_number bit = 0₂

Figure E.33 – Object_and_plug_type_specific_information_for_MPEG2-TS_object

In the above example, the increment position number function is not supported, so the *increment_position_number* field is set to 0₂.

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	2	R	Descriptor length field = 00 14 ₁₆
00 01 ₁₆			
00 02 ₁₆	2	R	Info_block_type = 00 03 ₁₆
00 03 ₁₆			
00 04 ₁₆	2	R	Primary_fields_length = 00 10 ₁₆
00 05 ₁₆			
List descriptor reference			
00 06 ₁₆	1	R	Descriptor_type = 10 ₁₆
00 07 ₁₆	2	R	List ID = 30 00 ₁₆ (root contents list)
00 08 ₁₆			
Nested position indicator info block			
00 09 ₁₆	2	R	Descriptor length field = 00 0B ₁₆
00 0A ₁₆			
00 0B ₁₆	2	R	Info_block_type = 00 02 ₁₆
00 0C ₁₆			
00 0D ₁₆	2	R	Primary_fields_length = 00 07 ₁₆
00 0E ₁₆			
00 0F ₁₆	1	R	Indicator_type = 00 ₁₆ ("relative HMSF")
00 10 ₁₆	2	R	Object_position_number
00 11 ₁₆			
00 12 ₁₆	1	R	+/- Hours
00 13 ₁₆	1	R	Minutes
00 14 ₁₆	1	R	Seconds
00 15 ₁₆	1	R	Frames

Figure E.34 – Position Info Block for destination plug

In the above example, the root contents list is assigned to the destination plug.

In the case of DVR-Blue recorder, the recording position in the object is equal to or greater than 00:00:00:00, so the “+/-“ field is always set to 0₂.

In the case where any list is not associated to the destination plug:

list ID field in List descriptor reference is set to FF FF₁₆.

Nested position indicator info block

object_position_number field is set as FF FF₁₆,

hours, Minutes, Seconds, Frames fields are set as 00₁₆.

E.3.3 Source Plug Status Area Info Block

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆ 00 01 ₁₆	2	R	Descriptor length field = XX XX ₁₆
00 02 ₁₆ 00 03 ₁₆	2	R	Info_block_type = 88 02 ₁₆
00 04 ₁₆ 00 05 ₁₆	2	R	Primary_fields_length
00 06 ₁₆ 00 07 ₁₆	1	R	Number_of_destination_plugs = n
:			<i>Plug_status_info_block[0]</i> : <i>Plug_status_info_block[n-1]</i>

Figure E.35 – Source Plug Status Area Info Block

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	2	R	Descriptor length field = XX XX ₁₆
00 01 ₁₆			
00 02 ₁₆	2	R	Info_block_type = 88 05 ₁₆
00 03 ₁₆			
00 04 ₁₆	2	R	Primary_fields_length = XX XX ₁₆
00 05 ₁₆			
00 06 ₁₆	1	R	Plug_number
<i>Nested structures</i>			
00 07 ₁₆	<i>Operating Mode Info Block (88 06₁₆)</i>		
:			
:	<i>Plug Configuration Info Block (88 07₁₆)</i>		
:			
:	<i>Position Info Block (00 03₁₆)</i>		
:			

Figure E.36 – Source plug *Plug_status_info_block*

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	2	R	Descriptor length field = 00 07 ₁₆
00 01 ₁₆			
00 02 ₁₆	2	R	Info_block_type = 88 06 ₁₆
00 03 ₁₆			
00 04 ₁₆	2	R	Primary_fields_length = 00 03 ₁₆
00 05 ₁₆			
00 06 ₁₆	1	R	Operating_mode
00 07 ₁₆	2	R	Operating_mode_specific_information (FF ₁₆ pad bytes if necessary)
00 08 ₁₆			

Figure E.37 – Operating Mode Info Block

The size of the *operating_mode_specific_information* field is 2 bytes and DVR-Blue player/recorder may pad FF₁₆ if necessary.

NOTE – In the case where *Operating_mode* field is set to FF₁₆ (“SUSPENDED”), this status indicates that the plug is not available.

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	2	R	Descriptor length field = XX XX ₁₆
00 01 ₁₆			
00 02 ₁₆	2	R	Info_block_type = 88 07 ₁₆
00 03 ₁₆			
00 04 ₁₆	2	R	Primary_fields_length = XX XX ₁₆
00 05 ₁₆			
00 06 ₁₆	1	R	AV_object_type = XX ₁₆
00 07 ₁₆			<i>Object_and_plug_type_specific_information</i>
:			
:			

Figure E.38 – Plug Configuration Info Block

The *Object_and_plug_type_specific_information* for video object is as follows:

Address	Length, bytes	External Read/Write	Contents
00 01 ₁₆	1	R	*b Reserved = 000 0000 ₂

*b *audio_mute* bit = 0₂ ("not effect")

Figure E.39 – Object_and_plug_type_specific_information for video object

In the above example, the audio mute function is not supported, so the *audio_mute* field is set to 0₂.

Address	Length, bytes	External Read/Write	Contents
00 00 ₁₆	2	R	Descriptor length field = 00 17 ₁₆
00 01 ₁₆			
00 02 ₁₆	2	R	Info_block_type = 00 03 ₁₆
00 03 ₁₆			
00 04 ₁₆	2	R	Primary_fields_length = 00 13 ₁₆
00 05 ₁₆			
List descriptor reference			
00 06 ₁₆	1	R	Descriptor_type = 21 ₁₆
00 07 ₁₆	2	R	List ID of contents list (root contents list or root performance list)
00 08 ₁₆			
00 09 ₁₆	1	R	List_type
00 0A ₁₆	2	R	Object_ID of a content object
00 0B ₁₆			
Nested position indicator info block			
00 0C ₁₆	2	R	Descriptor length field = 00 0B ₁₆
00 0D ₁₆			
00 0E ₁₆	2	R	Info_block_type = 00 02 ₁₆
00 0F ₁₆			
00 10 ₁₆	2	R	Primary_fields_length = 00 07 ₁₆
00 11 ₁₆			
00 12 ₁₆	1	R	Indicator_type = 00 ₁₆ (relative HMSF)
00 13 ₁₆	2	R	Object_position_number = 00 00 ₁₆
00 14 ₁₆			
00 15 ₁₆	1	R	+/- Hours
00 16 ₁₆	1	R	Minutes
00 17 ₁₆	1	R	Seconds
00 18 ₁₆	1	R	Frames

Figure E.40 – Position Info Block for video contents in source plug status

In the above example, a video content object is assigned to the destination plug.

In the case of DVR-Blue player/recorder, the recording position in object is equal to or greater than 00:00:00:00, so the “+/-“ field is always set to 0₂.

In the case where any list or object is not associated to a source plug:

List ID field in List descriptor reference is set to FF FF₁₆.

Nested position indicator info block:

Object_position_number field is set as FF FF₁₆,

Hours, Minutes, Seconds, Frames fields are set as 00₁₆.

Annex F: DVR-Blue Use Examples (informative)

The following methods are examples of control sequences that should be supported by a DVR-Blue Video Player/Recorder profile.

F.1 Descriptor Access Examples

F.1.1 Method to read the Subunit Identifier Descriptor or the Status Descriptor

In the case of DVR-Blue player/recorder, the following method is applied for reading Subunit Identifier Descriptor and Status Descriptor:

The reading procedure is as follows:

- 1) The controller issues OPEN DESCRIPTOR command with the following parameters:
 - descriptor identifier* : indicate the objective descriptor.
 - Subunit Identifier Descriptor : *descriptor_type* = 00_{16}
 - Status Descriptor : *descriptor_type* = 80_{16} , *reference_method* = 00_{16} (full descriptor reference)
 - subfunction* : READ OPEN or WRITE OPEN
- 2) The controller issues READ DESCRIPTOR command with the following parameters:
 - descriptor identifier* : indicate the objective descriptor.
 - data_length* = size that controller can receive.
 - (example : Async. buffer size of controller - overhead of READ DESCRIPTOR command)
 - address* = $00\ 00_{16}$
- 3) The DVR-Blue player/reorder returns the appropriate part of the specified descriptor.
 - There are three cases of an ACCEPTED response:
 - Case 1 : *read_result_status* = 10_{16}
 - The READ request can be handled with no problem.
 - Case 2 : *read_result_status* = 12_{16}
 - The READ request started in valid data space, but went beyond the end of valid data space.
 - data_length* = The actual number of bytes read.

In the above two cases, the “read whole descriptor” sequence is finished and the controller has been received with whole descriptor contents.

 - Case 3 : *read_result_status* = 11_{16}
 - The READ request was only partially satisfied due to data transfer capacity limitations.
 - data_length* = The actual number of bytes read.
 - The controller must issue additional READ command(s) to get all of the desired data until *read_result_status* = 12_{16} or 10_{16}
 - The additional READ DESCRIPTOR command(s) parameter is set as follows:
 - data_length* = size that controller can receive.

address = *address* of preceding READ DESCRIPTOR command
 + *data_length* value of preceding READ DESCRIPTOR command.

- 4) The controller issues OPEN DESCRIPTOR command with the following parameters:
 - descriptor_identifier* : indicate the objective descriptor.
 - subfunction* : CLOSE

F.1.2 Method to read an object entry descriptor

In the case of DVR-Blue player/recorder, this method is applied for reading the object entry descriptor in the list descriptor.

The reading procedures are as follows:

- 1) The controller issues OPEN DESCRIPTOR command with the following parameters:
 - descriptor_identifier* : indicate the object list descriptor, which includes the objective object entry.
 - descriptor_type* = 10_{16} (Object list descriptor - specified by list ID)
 - subfunction* : READ OPEN or WRITE OPEN
- 2) The controller issues READ DESCRIPTOR command with the following parameters:
 - descriptor_identifier* : indicate the objective object entry.
 - descriptor_type* = 20_{16} (Object entry descriptor - specified by object position)
 - or
 - descriptor_type* = 21_{16} (Object entry descriptor - specified by an object ID)
 - data_length* = size that controller can receive.
 - (example : Async. buffer size of controller - overhead of READ DESCRIPTOR command)
 - address* = $00\ 00_{16}$ (top of descriptor)
- 3) The DVR-Blue player/reorder returns the appropriate part of the specified descriptor.
 - There are three cases of an ACCEPTED response.
 - Case 1 : *read_result_status* = 10_{16}
 The READ request can be handled with no problem.
 - Case 2 : *read_result_status* = 12_{16}
 The READ request started in valid data space, but went beyond the end of valid data space.
data_length = The actual number of bytes read.
 - In the above two cases, the “read an object entry” sequence is finished and the controller has been received with whole object entry contents.
 - Case 3 : *read_result_status* = 11_{16}
 The READ request was only partially satisfied due to data transfer capacity limitations.
data_length = The actual number of bytes read.
 - The controller must issue additional READ command(s) to get all of the desired data until *read_result_status* = 12_{16} or 10_{16}
 - The additional READ DESCRIPTOR command(s) parameter is set to the following:
data_length = size that controller can receive.

address = *address* of preceding READ DESCRIPTOR command
 + *data_length* value of preceding READ DESCRIPTOR command.

- 4) The controller issues OPEN DESCRIPTOR command with the following parameters:
- descriptor identifier* : indicate the object list descriptor, which includes the objective object entry.
subfunction : CLOSE

F.1.3 Method to read the header part of the list descriptor for contents description.

This method is applied to the following lists:

- Root contents list
- Root performance list
- Performance list
- Root associated data list
- Associated data list

When READ DESCRIPTOR command with descriptor identifier specifies the list of contents description, DVR-Blue player/recorder returns the header part of list. If the controller wants to read the object entry, the controller shall specify each object entry.

The header part of these lists contain the following fields:

list_type
attributes
size_of_list_specific_information
list_specific_information
number_of_entries

The reading sequence for the list descriptor is as follows:

- 1) The controller issues OPEN DESCRIPTOR command, which specifies the root contents list.
 The parameters of OPEN DESCRIPTOR command are as follows:
descriptor identifier: indicate the objective list.
descriptor_type = 10_{16} (Object list descriptor - specified by list ID)
descriptor_type_specific_reference = $XXXX_{16}$ (list ID of the objective list)
subfunction: READ OPEN or WRITE OPEN
- 2) The controller issues READ DESCRIPTOR command with the following parameter:
descriptor identifier : indicate the objective list.
data_length = size that controller can receive.
address = 0002_{16} (address of *list_type* field)
- 3) DVR-Blue player/recorder sends the header part of the objective list with the RESPONSE frame.
 The RESPONSE procedure is the same as “Method to read whole descriptor”, thus the controller may need to issue more READ DESCRIPTOR command(s).

NOTE — In some implementation, it may be very difficult to maintain the *descriptor_length* field of these lists, thus the above sequence excludes the *descriptor_length* field.

The controller issues OPEN DESCRIPTOR command, which specifies the objective list.

The parameters of OPEN DESCRIPTOR command are as follows:

descriptor identifier : indicate the root contents list.

subfunction : CLOSE

F.1.4 Method to read the whole list descriptor for contents description

This method is applied to the following lists:

- Root contents list
- Root performance list
- Performance list
- Root associated data list
- Associated data list

Reading the whole list descriptor is as follows:

- 1) The controller issues OPEN DESCRIPTOR command, which specifies the objective list.
The parameters of OPEN DESCRIPTOR command are as follows:
descriptor identifier : indicate the objective list.
descriptor_type = 10₁₆ (Object list descriptor - specified by list ID)
descriptor_type_specific_reference = XXXX₁₆ (list ID of the objective list)
subfunction : READ OPEN or WRITE OPEN
- 2) Read the header part of the objective list. The method is the same as part 2 & 3 of “Method to read header part of root contents list descriptor.”
- 3) Parse the header part and look for the number of object entry from the *number_of_entry* field or *number_of_items* info block.
- 4) Read each object entry. The method is the same as part 2 & 3 of “Method to read an object entry descriptor”. In this situation, the descriptor type of descriptor identifier for the READ DESCRIPTOR command is 20₁₆ (Object entry descriptor - specified by object position)
- 5) The controller issues OPEN DESCRIPTOR command with the following parameters:
descriptor identifier : indicate the objective list.
Subfunction : CLOSE

F.1.5 Method to read a info block in the Subunit Identifier Descriptor or the Status Descriptor

For a DVR-Blue player/recorder, reading the primary field of info block is guaranteed.

In the case of DVR-Blue player/recorder, this method is applied for reading the info blocks in Subunit Identifier Descriptor or Status Descriptor.

The reading procedure is as follows:

- 1) The controller issues OPEN DESCRIPTOR command with the following parameters:

descriptor identifier : indicate the descriptor, which contains the objective info block.

Subfunction : READ OPEN or WRITE OPEN

- 2) The controller issues READ INFO BLOCK command with the following parameters:

info_block_reference_path : indicate objective info block.

descriptor_type = 30_{16} (Information block - specified by its type and instance count)

data_length = size that controller can receive.

(example : Async. buffer size of controller - overhead of READ INFO BLOCK command)

address = $00\ 00_{16}$

- 3) The DVR-Blue player/reorder returns the appropriate part of specified descriptor.

There are three cases of an ACCEPTED response:

Case 1 : *read_result_status* = 10_{16}

The READ request can be handled with no problem.

Case 2 : *read_result_status* = 12_{16}

The READ request started in valid data space, but went beyond the end of valid data space

data_length = The actual number of bytes read.

In the above two cases, the "read info block" sequence is finished and the controller has been received with contents in the primary field of the info block.

Case 3 : *read_result_status* = 11_{16}

The READ request was only partially satisfied due to data transfer capacity limitations.

data_length = The actual number of bytes read.

The controller must issue additional READ command(s) to get all of the desired data until *read_result_status* = 12_{16} or 10_{16}

The additional READ INFO BLOCK command(s) parameter is set as follows:

data_length = size that controller can receive.

address = address of preceding READ INFO BLOCK command

+ *data_length* value of preceding READ INFO BLOCK command.

- 4) The controller issues OPEN DESCRIPTOR command with following parameters:

descriptor identifier : indicate the descriptor, which contains the objective info block.

Subfunction : CLOSE

F.1.6 Method to read a info block in an object entry

In regards to the DVR-Blue player/recorder, reading the primary field of the info block is guaranteed. This method is applied for reading object entry in the following lists.

- Root contents list
- Root performance list
- Performance list
- Root associated data list
- Associated data list

The reading procedure is as follows:

- 1) The controller issues OPEN DESCRIPTOR command with the following parameters:
 - descriptor identifier* : indicate the object list descriptor, which contains the object entry with objective info block.
 - Subfunction* : READ OPEN or WRITE OPEN
- 2) The controller issues READ INFO BLOCK command with the following parameters:
 - info_block_reference_path* : indicate objective info block.
 - descriptor_type* = 30_{16} (Information block - specified by its type and instance count)
 - data_length* = size that controller can receive.
(example : Async. buffer size of controller - overhead of READ INFO BLOCK command)
 - address* = $00\ 00_{16}$
- 3) The DVR-Blue player/reorder returns the appropriate part of specified descriptor.
 - There are three cases of an ACCEPTED response:
 - Case 1 : *read_result_status* = 10_{16}
The READ request can be handled with no problem.
 - Case 2 : *read_result_status* = 12_{16}
The READ request started in valid data space, but went beyond the end of valid data space
data_length = The actual number of bytes read.
In the above two cases, the “read info block” sequence is finished and the controller has been received with contents in the primary field of the info block.
 - Case 3 : *read_result_status* = 11_{16}
The READ request was only partially satisfied due to data transfer capacity limitations.
data_length = The actual number of bytes read.
The controller must issue additional READ command(s) to get all of the desired data until *read_result_status* = 12_{16} or 10_{16}
The Additional READ INFO BLOCK command(s) parameter is set as follows:
 - data_length* = size that controller can receive.
 - address* = *address* of preceding READ INFO BLOCK command

+ *data_length* value of preceding READ INFO BLOCK command.

- 4) The controller issues OPEN DESCRIPTOR command with the following parameters:

descriptor identifier : indicate the object list descriptor, which contains the object entry with objective info block.

Subfunction : CLOSE

F.1.7 Method to write info block

All modifiable fields in the descriptor of DVR-Blue player/recorder are composed of info blocks.

The controller shall use the WRITE INFO BLOCK command. It is strongly recommended to modify the whole primary field at once. The *group_tag* function is optional for DVR-Blue player/recorder, thus the controller should write the primary field of the info block in one WRITE INFO BLOCK transaction.

In the case of DVR-Blue player/recorder, the modifiable info blocks exist in the following lists:

- Root contents list
- Root performance list
- Performance list
- Associated data list

So that this method is applied for info block in these list.

The write sequence is as follows:

- 1) The controller issues OPEN DESCRIPTOR command to open the root contents list. The parameters of OPEN DESCRIPTOR command are as follows:

descriptor identifier : indicate the object list descriptor, which contains the object entry with objective info block.

Subfunction = 03₁₆ (“WRITE OPEN”)

- 2) If the controller does not know the actual size of info block, the controller issues READ INFO BLOCK command.

- 3) The controller issues WRITE INFO BLOCK command. The parameters are as follows:

info_block_reference_path : point to objective info block.

group_tag = 00₁₆ (“immediate”)

replacement_data_length = length of *replacement_info_block_data* field.

address = 00 00₁₆ (top of primary field)

original_data_length : length of original primary field.

replacement_info_block_data : the new data to be written into the info block.

- 4) The controller issues OPEN DESCRIPTOR command with the following parameters:

descriptor identifier : indicate the object list descriptor, which contains the object entry with objective info block..

Subfunction : CLOSE

F.2 DVR-Blue player/recorder control example

F.2.1 Method to find DVR-Blue player/recorder

Assumption: the controller has already found the disc subunit.

<Sequence>

Check the Subunit Identifier Descriptor:

- 1) Scan *supported_media_type[i]* to find the identifier of “DVR-Blue” (type =0B 01₁₆)
- 2) Check *implementation_profile_ID* in *supported_media_type_specification[j]*, which *supported_media_type* is 0B 01₁₆ (“DVR-Blue”).
If *implementation_profile_ID* is 10₁₆ (player/recorder profile), this subunit is “BASIC DVR-Blue player/recorder”.
If *implementation_profile_ID* is FF₁₆, the DVR-Blue is compatible with the multi-profile of DVR-Blue, thus check *extended_implementation_profile_ID[k]* to know capability.
If *extended_implementation_profile_ID[k]* is 10₁₆ (Basic DVR-Blue player/recorder), this subunit is compatible with “BASIC DVR-Blue player/recorder” and may have an extended function.
If *extended_implementation_profile_ID[k]* is 11₁₆ (with detailed performance list support profile), this subunit has capability to treat detailed performance list.
If *extended_implementation_profile_ID[k]* is 12₁₆ (with associated data list support profile), this subunit has the capability of an associated data list function.

F.2.2 Method to check availability of DVR-Blue player/recorder

- 1) **Check whether medium is existing in the drive or not.**
Check *disc_in_drive* field of *media_and_edit_status_info_block*, which is a primary field of *general_disc_subunit_status_area_info_block* in The Disc Subunit Status Descriptor.

If *disc_in_drive* value is 01₂ (“installed”), medium is available.

- 2) **Check media_type is DVR-Blue ?**
Check the *media_type* field in Root Contents List *list_specific_information*, which value is 0B 01₁₆ (“DVR-Blue”).

F.2.3 Method to check medium change or contents change.

Medium change:

As described in the above section, medium change is detected by watching the *disc_in_drive* field.

Contents or medium change:

In the case of following:

- Any change/update is applied on contents in DVR-Blue Disc.
(i.e. record new object, delete object, divide object, combine objects, change object’s name, etc.)
- unload medium
- load medium

The *time_stamp_data* value of Time Stamp Info Block in the header part of root contents list is updated.

So that controller can detect modification of contents on DVR-Blue Disc by polling the *time_stamp_data*.

NOTE – DVR-Blue player/recorder shall NOT reset the *time_stamp_data* field when loading/unloading medium.

F.2.4 Method to check remaining capacity

The remaining capacity is specified in the *disc_remaining_recording_capacity* field of *disc_capacity_info_block*, which is located in the Root Contents List *optional_info_block_area*, thus the controller can know the remaining capacity using READ INFO BLOCK command.

In the case of DVR-Blue player/recorder, the format of the *disc_remaining_recording_capacity* field is “bytes.”

F.2.5 Method to get list of contents

Read the header part of root contents list and memorize time stamp info block.

NOTE — When root contents list is updated, *time_stamp_info_block* in Root Contents List *list_specific_information* is updated too. Before the controller does something to affect the contents (ERASE, write attribute, etc.), the controller should check that the time stamp value has not changed.

Read each object entry in root contents list.

NOTE — There shall be only one root contents list in DVR-Blue player/recorder.

NOTE —

- A) If there are other types of object entries other than the video object, a controller should work without a problem.

The controller shall check the *entry_type* field of object entry.

- B) If an object has a child list, a controller should work without a problem.

The controller shall check the *has_child_ID* flag in the *attribute* field of object entry.

The above two restrictions (A & B) are to take steps to meet future extensions of DVR-Blue specifications.

F.2.6 Method to RECORD specific contents on destination plug #j

Setup of destination plug #j

Use CONFIGURE command to set *video_signal_mode* in video object *object_and_plug_type_specific_information* field of *plug_configuration_info_block*, which is located in *plug_status_info_block* in the *destination_plug_status_area_info_block* of Disc Subunit Status Descriptor.

Issue SET PLUG ASSOCIATION command to associate root contents list with destination plug #j.

Issue RECORD command to destination plug #j, the parameters are:

destination_plug : destination plug #j

subfunction_1 : 75₁₆ (“forward”)

subfunction_2 : 00₁₆ (“new”)

NOTE – As soon as accepting RECORD command, the DVR-Blue player/recorder creates new video object entry with info blocks in root contents list.

NOTE – In the case of an “associated data list support” profile, as soon as accepting RECORD command, the DVR-Blue player/recorder creates a new associated data list and sets the list ID of this associated data list into DVR-Blue specific info block.

Write the attributes of a new video object by using WRITE INFO BLOCK commands.

NOTE —ERASE command designate a video object that is currently recording shall be REJECTED. .

NOTE —During recording, size information in video object entry may be inaccurate. If the controller want to know size of video object during recording, the controller should read status of destination plug.

F.2.7 Method to “RECORD PAUSE”

- In the case of changing *rec_state* from “RECORD forward” to “RECORD pause”

Issue RECORD command to destination plug #j. The parameters follow:

destination_plug : destination plug #j

subfunction_1 : 7D₁₆ (“forward pause”)

subfunction_2 : 00₁₆ (“new”)

- In the case of changing *rec_state* from “RECORD pause” to “RECORD forward”

Issue RECORD command to destination plug #j. The parameters follow:

destination_plug : destination plug #j

subfunction_1 : 75₁₆ (“forward”)

subfunction_2 : 00₁₆ (“new”)

NOTE — Changing *rec_state* does not cause creation of a new video object. RECORD operation is on going.

F.2.8 Method to STOP record operation on destination plug #j

Issue STOP command specifying destination plug #j.

NOTE — When STOP command is ACCEPTED, the value of *size_indicator_info_block* in video object becomes definite.

F.2.9 Method to PLAY specific object on source plug #i at the beginning

- 1) Setup source plug #i
- 2) Issue SET PLUG ASSOCIATION command to associate specific object with source plug #i
- 3) Issue SEARCH command to the beginning point (00:00:00.00)

NOTE — DVR-Blue player/recorder supports relative HMSF format on SEARCH command.

Issue PLAY command with “forward X1” to source plug #i

NOTE — ERASE command designated to a video object that is currently playing shall be REJECTED.

F.2.10 Method to “PLAY PAUSE”

- In the case of changing *play_state* from “PLAY forward” to “PLAY pause”.

Issue PLAY command to source plug #j. The parameters are as follows:

source_plug : source plug #j

subfunction_1 : 7D₁₆ (“forward pause”)

- In the case of changing *play_state* from “PLAY pause” to “PLAY forward”.

Issue PLAY command to source plug #i. The parameters are as follows:

source_plug : source plug #i

subfunction_1 : 75₁₆ (“forward”)

F.2.11 Method to SEARCH specific position.

Issue SEARCH command. The parameters follow:

search_type : 00₁₆ (“position”)

indicator_type : 00₁₆ (“relative HMSF count”)

F.2.12 STOP PLAY operation on source plug #i

Issue STOP command specifying source plug #i.

NOTE — The value of current pointer is maintained as it is positioned when the operation stopped.

F.2.13 ERASE specific video object

Issue ERASE command specifying the video object.

NOTE — ERASE command designated to a video object that is currently playing or recording shall be REJECTED.

F.2.14 Change plug association

Issue SET PLUG ASSOCIATION command.

NOTE – SET PLUG ASSOCIATION command should be REJECTED when *operating_mode* is not STOP.

Annex G: Audio stream format field in DVR-Blue specific info_block (informative)

Essentially, audio format in MPEG2-TS stream is self-evident when MPEG2-TS stream type is determined. In the case of DVR-Blue player/recorder, MPEG2-TS stream type is known from *Video Stream Format Subtype Info Block* in each video contents object entry.

The DVR-Blue format permits to combine different type MPEG2-TS streams, which contain different audio stream formats. (ex. AV object A with MPEG audio, AV object B with AAC).

When playing back a stream consisting of object A with MPEG audio and AV object B with AAC, both MPEG audio decoder and AAC decoder are required. This *audio stream format* field is used as an assistant of the *Video Stream Format Subtype info block* and indicates this requirement about decoder.

G.1 [example-1] Japanese BS-Digital stream.

Japanese BS-Digital only uses AAC audio stream, so that when signal type of a video contents object is determined as “Japanese BS-Digital” from Video Stream Format Subtype Info Block, *audio stream format field* in DVR-Blue specific info block is set as follows: (Two cases are listed below)

In the case where a DVR-Blue Player/Recorder does not know about Japanese BS-Digital specification. (Permitted, but not recommended)

Offset	msb						lsb
00 ₁₆	Valid =0 ₂	MPEG =0 ₂	AC3 =0 ₂	AAC =0 ₂	L-PCM =0 ₂	Reserved = 0000 ₂	

Figure G.1– audio stream format field example (1)

In the case where a DVR-Blue Player/Recorder knows about Japanese BS-Digital specification. (Recommended)

Offset	msb						lsb
00 ₁₆	Valid =1	MPEG =0 ₂	AC3 =0 ₂	AAC =1 ₂	L-PCM =0 ₂	Reserved = 0000 ₂	

Figure G.2– audio stream format field example (2)

G.2 [example-2] combined stream. (AAC and AC3)

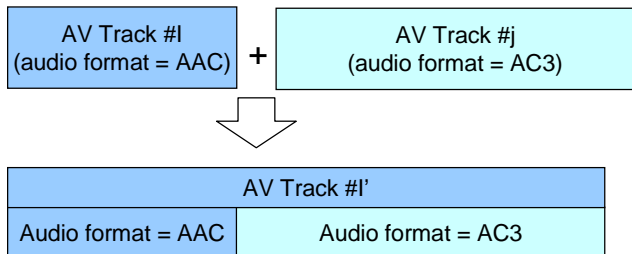


Figure G.3 – combined stream example

In the above case, both AAC and AC3 decoder are required for receiver. Information is as follows:

Offset	msb						lsb
00 ₁₆	Valid =1 ₂	MPEG =0 ₂	AC3 =1 ₂	AAC =1 ₂	L-PCM =0 ₂	Reserved = 0000 ₂	

Figure G.4 – audio stream format field example (3)