TA Document 1999045

AV/C Information Block Types Specification Version 1.0

July 23, 2001

Sponsored by:
1394 Trade Association

Accepted for Release by:
1394 Trade Association Board of Directors

Abstract:
This specification is a reference to general information block structures used in AV/C Devices supporting the AV/C Descriptor Mechanism.

Keywords:
info block, descriptor

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
Regency Plaza Suite 350
2350 Mission College Blvd.
Santa Clara, Calif. 95054, 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

List of figures

List of tables

Change history

The following table shows the change history for this specification.

Table 1.1 – Content change

Category	Description
Technical	Generation ID: Enhancements to AV/C General 3.0 version 1.1 is included into 01_{16}.
Technical	Object entry descriptor specified by object ID: This section is newly added.
Technical	6.3.9 Relative Segment HMSF Count: This section is newly added.
Technical	6.4 Position Info Block: list_descritor_reference field is enhanced as descriptor_reference field.
Technical	6.6 Character Code Info Block: ITTS, Korean, Chinese, ISO646, Shift-JIS and Japanese EUC are newly added.
Technical	6.7 Language Code Info Block: ISO639 is newly added.
Technical	7. Subunit Type: Subunit type encoding table is updated.
Editorial	Updated document to new template format
Editorial	Sections 3, 4 & 5 are removed. They were added to AV/C General Descriptor Mechanism.

1. Overview

1.1 Purpose

This document defines general information block structures that are used in various AV/C (sub) units. See references [R1]and [R9] fore more information about AV/C (sub) units. The information in this document has been developed as part of the work to define the AV/C Disc subunit and command set in reference [R11]. As a result of that effort, some new enhancements for the information block to the general AV/C descriptor mechanism were created. See reference [R10] about AV/C descriptor mechanism.

1.2 Scope

This document describes information blocks that are designed to be used across various different types of units and subunits. Refer to each unit and subunit specification for information blocks specific to that unit or subunit. See references [R11], [R12], [R13], [R14], [R15] and [R16].

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.

2.1 Reference sources

All listed references are available at various web sites. Some web sites require membership to access the references, and other sites require payment for each reference. The following sites contain the references used in this document. The reader is encouraged to always consult these sites for information on the latest versions of specifications mentioned here, as well as specifications that may be developed in the future.

[R1] 1394TA web site, http://www.1394TA.org. This web site is kept up to date with the latest released and draft versions of AV/C specifications. You need to be a member to access draft specifications.

[R2] International Electro-technical Commission web site, http://www.iec.ch. This web site contains specifications that must be purchased.

[R3] IEEE standards web site. http://standards.ieee.org. This web site contains specifications that must be purchased.

[R4] Sony Corporation web site. http://www.sony.co.jp. This web site contains the information of Sony Corporation. The following is also the contact information of Sony Corporation to obtain specifications. 6-7-35 Kitashinagawa, Shinagawa-ku, Tokyo, 141-0001 Japan, Fax+81-3-5448-7835

2.2 Specifications

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.

[R5] IEEE Std 1394–1995, *Standard for a High Performance Serial Bus*, August 30 1996.

[R6] IEEE Std 1394a-2000, *Standard for High Performance Serial Bus-Amendment 1*.

[R7] ISO/IEC 13213:1994, *Control and Status Register (CSR) Architecture for Microcomputer Buses*, First Edition, October 5, 1994.

[R8] IEC 61883-1, *Consumer audio/video equipment – Digital Interface*, 1998-02.

[R9] TA Document 1999026, *AV/C Digital Interface Command Set General Specification, Version 4.0* Draft

[R10] TA Document 1999025, *AV/C Descriptor Mechanism Specification, Version 1.0* Draft.

[R11] TA Document 19998013, *AV/C Disc Subunit Specification, Version 1.0*, January 29, 1999

[R12] TA Document 1999002, *AV/C Disc Media Type Specification – CD-DA, Version 1.0*, April 9, 1999

[R13] TA Document 1999040, *AV/C Disc Media Type Specification – SACD 1.0*, July 10, 1999

[R14] TA Document1999029, *AV/C Disc Subunit Enhancements for Hard Disk Drive Specification*, July 10, 2000.

[R15] TA Document 1999005, *AV/C Bulletin Board Subunit General Specification Version 1.0*, August 4, 1999

[R16] TA Document 1999007 *AV/C CA Subunit Specification Version 1.0* April 6, 1999

[R17] *MD control application specification version 1.0*, Sony Corporation

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.

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 information block: Information blocks (also called info blocks) are enhancements to the descriptor model. Each information block contains a collection of related data fields; info blocks can also contain other info blocks.

3.2.2 Segment: A segment is part of an object. The following diagram illustrates a segmented object:

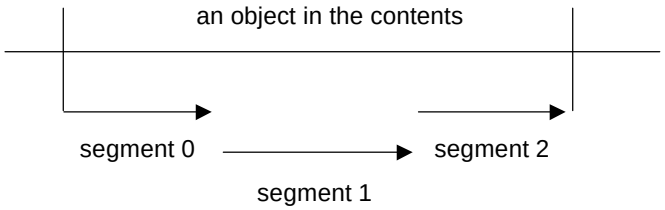

Figure 3.1 – segment

Segments are useful as a means of specifying areas or locations within the scope of a single object.

3.3 Acronyms and abbreviations

AV/C Audio Video Control

IEEE The Institute of Electrical and Electronics Engineers, Inc.

4. General AV/C Information Block Specifications

This section contains detailed specifications for the general information block types defined by this specification. Specific subunit types may have additional info block definitions, noted in the appropriate documentation. See Annex A:Subunit Defined Information Blocks.

The following table contains general AV/C information block type specification values, names, and comments.

Table 4.1 – General AV/C Information Block Types

info block type	info block name	comments
00 00_{16}	vendor_specific_info_block	Format and contents defined by a specifier ID.
00 01_{16}	size_indicator_info_block	The size of an object.
00 02_{16}	position_indicator_info_block	The position of an AV stream.
00 03_{16}	position_info_block	Describes the position of a data stream.
00 04_{16}-00 07_{16}	time_stamp_info_block	The date and time of a significant event (creation, modification, etc.).
00 08_{16}	character_code_info_block	The character code of text associated with this info block.
00 09_{16}	language_code_info_block	The language code of text associated with this info block.
00 $0A_{16}$	raw_text_info_block	Raw text bytes.
00 $0B_{16}$	name_info_block	Specifies the name of the entity that contains this info block.
00 $0C_{16}$	description_info_block	A description of the entity that contains this info block.
00 $0D_{16}$	image_info_block	Reference to a digital still image.
00 $0E_{16}$	image_format_info_block	Describes the format of a digital still image object.
00 $0F_{16}$	descriptor_reference_info_block	Encapsulates a descriptor_identifier structure in an info block.
00 10_{16}	number_of_items_info_block	Specifies the current “count” or number of items in the context of this info block.
00 11_{16}	descriptor_capacity_info_block	Specifies the storage characteristics of descriptive information.
00 12_{16} - 7F FF_{16}	reserved for future definition of general AV/C info block	

4.1 Vendor Specific Info Block (00 00_{16})

The *vendor_specific_info_block* contains data bytes whose format and interpretation is determined by a specifier identification (e.g., some organization or entity other than the subunit vendor). It has the following format:

Table 4.2 – vendor_specific_info_block

Address Offset	Contents
00 00_{16}	compound_length
00 01_{16}	
00 02_{16}	info_block_type = 00 00_{16} (vendor_specific_info_block)
00 03_{16}	
00 04_{16}	primary_fields_length
00 05_{16}	
00 06_{16}	specifier_id
00 07_{16}	
00 08_{16}	
00 09_{16}	vendor_specific_data
:	
:	

4.1.1 Field Definitions

specifier_id: The *specifier_id* field is the 24-bit IEEE/RAC provided company ID value.

vendor_specific_data: The *vendor_specific_data* field contains bytes whose format and interpretation are determined by the entity indicated by the *specifier_id*. This field could be used by controllers, for example, to allow users to attach various kinds of data to objects, lists, or any other structure which can contain this information block type.

It is the responsibility of the entity represented by *specifier_id* to devise a scheme for identifying the info blocks which it owns, in order to differentiate among possibly several instances of these blocks in a given context.

4.2 Size Indicator Info Block (00 01_{16})

The size_indicator_info_block specifies the size of the object(s) which are specified by the entity which contains this info block. For example, if this info block is in a object descriptor, it indicates the size of that object. If it's in a list header, it specifies the TOTAL size of all content objects, which are referenced by that list. It has the following format:

Table 4.3 – size_indicator_info_block

Address Offset	Contents
00 00_{16}	compound_length
00 01_{16}	
00 02_{16}	info_block_type = 00 01_{16} (size_indicator_info_block)
00 03_{16}	
00 04_{16}	primary_fields_length
00 05_{16}	
00 06_{16}	size_indicator_type
00 07_{16}	size_indicator_type_specific
:	
:	

4.2.1 Field Definitions

size_indicator_type: The *size_indicator_type* field indicates how the size is being specified. The current specification methods are as follows:

Table 4.4 – Size indicator type

Type Value	Size Indicator Type
00_{16}	H_M_S_F – the size is specified by the hours, minutes, seconds and frames of the content encoded in BCD.
01_{16}	raw_byte_count – the size is specified as the total number of bytes encoded in binary.
02_{16}	clock_time – the size, in hours, minutes, seconds and fractions of a second encoded in BCD.
all other values	reserved for future specification

size_indicator_type_specific: The *size_indicator_type_specific* field contains the size specification in the format indicated by *size_indicator_type*. The following formats are defined:

Table 4.5 – size_indicator_type=00_{16} (H_M_S_F format)

Address Offset	Contents
00 00_{16}	hours(MSB)
00 01_{16}	(LSB)
00 02_{16}	minutes
00 03_{16}	seconds
00 04_{16}	frames

hours, minutes, seconds and frames: The *hours*, *minutes*, *seconds* and *frames* fields specify the total hours, minutes, seconds and frames of the entity being described. The *hours* field is two bytes to accommodate very large items. All fields are encoded as BCD format. If a field is not supported (such as the frames count), then it shall be specified as FF_{16}. If the number of hours fits into the LSB of the *hours* field, then the MSB shall be set to 00_{16}.

The unit of frame rate depends on the signal format of AV stream to which this info block is associated.

Table 4.6 – size_indicator_type = 01_{16} (raw_byte_count format)

Address Offset	Contents
00 00_{16}	number_of_size_bytes
00 01_{16}	size
:	
:	

number_of_size_bytes: The *number_of_size_bytes* indicates how many bytes are used for the size field.

size: The *size* field specifies the total number of bytes in the entity being described.

Table 4.7 – size_indicator_type = 02_{16} (clock_time format)

Address Offset	Contents
00 00_{16}	hours(MSB)
00 01_{16}	(LSB)
00 02_{16}	minutes
00 03_{16}	seconds
00 04_{16}	x10ms

hours, minutes, seconds, x10ms: The *hours*, *minutes*, *seconds* and *x10ms* fields specify the size of an entity in hours, minutes, seconds and x10ms.

The *x10ms* field specifies a value which must be multiplied by 10 milliseconds to indicate the current fraction of the *seconds* field.

If the subunit is not able to support all of the fields in this structure (such as *x10ms*), then it shall specify a value of FF_{16} for the unsupported fields.

4.3 Position Indicator Info Block (00 02_{16})

The position indicator info block is used to specify the position, possibly in several ways. It has the following format:

Table 4.8 – position_indicator_info_block

Address Offset	Contents
$00\ 00_{16}$	compound_length
$00\ 01_{16}$	
$00\ 02_{16}$	info_block_type = $00\ 02_{16}$ (position_indicator_info_block)
$00\ 03_{16}$	
$00\ 04_{16}$	primary_fields_length
$00\ 05_{16}$	
$00\ 06_{16}$	indicator_type
$00\ 07_{16}$	indicator_type_specific
:	
:	

4.3.1 Field Definitions

indicator_type: The *indicator_type* field specifies how the position is being indicated; there are several possible choices, as shown in the following table:

Table 4.9 – Indicator Type

Type Value	Indicator Type	to specify	to report
00_{16}	relative_HMSF_count	available	available
01_{16}	relative_segment_count	available	available
02_{16}	absolute_HMSF_count	available	available
03_{16}	relative_byte_count	available	available
04_{16}	absolute_byte_count	available	available
05_{16}	relative_progress_stage	not used	available
06_{16}	relative_clock_time	available	available
07_{16}	absolute_clock_time	available	available
08_{16}	relative_segment_HMSF_count	available	available
all other values	reserved for future specification	---	---

indicator_type_specific: The *indicator_type_specific* field contains the position data in the format specified by the *indicator_type* field. The following sections provide details on each defined type.

Some position indicators are used by controllers to specify a position in a command frame, while others are used by the subunit to report a current position, either in a command frame or in a descriptor structure. Where appropriate, these restrictions are noted in the command or data structure definitions.

hours, minutes, seconds and frames: All *hours, minutes, seconds and frames* entries are encoded in BCD format, with the most significant field located in the most significant bits of its field.

The position_indicator_ info_ block is used to indicate a position in a AV stream associated with an object or list descriptor. The object or list descriptor is not specified in this info block. But it is assumed the list descriptor is known in the context where this info block appears. If the object or list descriptor need to be specified, the Position Info block containing this info block shall be used.

For example, the position_ indicator_ info_ block may indicate a current position in the current AV contents being played, if it is used in status descriptor.

4.3.2 Relative HMSF Count (00_{16})

The relative HMSF count type position data has the following format. The position indicator info block with this type position data is used to indicate a position in the object referred by the *object_position_number*.

Address Offset	msb							lsb
00_{16}	object_position_number							
:								
n								
n+1	+ / -	hours						
n+2	minutes							
n+3	seconds							
n+4	frames							

Figure 4.1 – Object Position-Relative HMSF Count Specification

4.3.2.1 Field Definitions

object_position_number: When the list is specified, the *object_position_number* field specifies the position number of the object in the list. When an object is specified, the *object_position_number* field is not used and set to 0. Its size, in bytes, is determined by the *size_of_object_position* field in the subunit identifier descriptor.

+/-: The +/- bit indicates the plus / minus sign of the following hours, minutes, seconds and frames information. If this bit is set to 1, the sign is minus. If this bit is set to 0, the sign is plus.

hours, minutes, seconds and frames: The *hours, minutes, seconds and frames* fields specify the hours, minutes, seconds and frames of the position. If the subunit is not able to support all of the fields in this structure (such as frames), then it shall specify a value of FF_{16} for the unsupported fields.

The resolution of the frame depends on the data format of the AV stream referred by this info block.

The starting point and the way of measurement of hours, minutes, seconds and frames depend on the data format of the AV stream referred by this info block.

4.3.3 Relative Segment Count (01_{16})

The relative segment counter specification indicates the current position of the stream relative to the beginning of the object, measured as the segment. This specification has the following format:

Table 4.10 – Relative Segment Count Specification

Address Offset	Contents
00 00_{16}	object_position_number
:	
:	
:	segment_number
:	

4.3.3.1 Field Definitions

object_position_number: When the list is specified, the *object_position_number* field specifies the position number of the object in the list. When an object is specified, the *object_position_number* field is not used and set to 0. Its size, in bytes, is determined by the *size_of_object_position* field in the subunit identifier descriptor.

segment_number: The *segment_number* field (two bytes) specifies the segment. Please refer to the definition of **segment** in section 3.2 for more details. If the track has not been segmented, then the subunit shall report a value of FF FF_{16} for this field.

4.3.4 Absolute HMSF Count (02_{16})

The absolute HMSF count type position data has the following format. The position indicator info block with this type position data is used to indicate a position in the list.

Table 4.11 – Absolute HMSF Count Specification

Address Offset	Contents
00_{16}	hours(MSB)
01_{16}	(LSB)
02_{16}	minutes
03_{16}	seconds
04_{16}	frames

4.3.4.1 Field Definitions

hours, minutes, seconds and frames: The *hours, minutes, seconds and frames* fields are as described above.

The resolution of the frame depends on the data format of the AV stream referred by this info block.

The starting point and the way of measurement of hours, minutes, seconds and frames depend on the data format of the AV stream referred by this info block.

4.3.5 Relative Byte Count (03_{16})

The relative byte count position specification indicates the position of the stream relative to the beginning of the object, measured in bytes. This specification has the following format:

Table 4.12 – Relative Byte Count Specification

Address Offset	Contents
00 00_{16}	object_position_number
:	
:	
:	length_of_byte_offset
:	byte_offset
:	
:	

4.3.5.1 Field Definitions

object_position_number: The *object_position_number* is as described above.

length_of_byte_offset: The *length_of_byte_offset* field specifies the number of bytes used for the *byte_offset* field.

byte_offset: The *byte_offset* field specifies the position as an offset from the beginning of the object, measured in bytes.

4.3.6 Absolute Byte Count (04_{16})

The absolute byte count position specification indicates the length of data relative to the beginning of the list, which contains the object being referenced. It has the following format:

Table 4.13 – Absolute Byte Count Specification

Address Offset	Contents
00 00_{16}	byte_count_length
:	byte_count
:	
:	

4.3.6.1 Field Definitions

byte_count_length: The *byte_count_length* field specifies the number of bytes in the *byte_count* field.

byte_count: The *byte_count* field indicates the position represented as the total number of bytes from the beginning of the list.

4.3.7 Relative Progress Stage (05_{16})

The relative progress stage specification indicates the current status of the stream using a simple progress indicator. This specification has the following format:

Table 4.14 – Relative Progress Stage Specification

Address Offset	Contents
00 00_{16}	object_position_number
:	
:	
:	progress_stage

4.3.7.1 Field Definitions

progress_stage: The *progress_stage* field specifies the stage of the progress as indicated by the following table:

Table 4.15 – progress_stage

progress_stage	meaning
00_{16}	not started
01_{16}	ongoing
all other values	reserved for future specification

4.3.8 Relative Clock Time (06_{16})

The relative Clock Time type position data has the following format. The position indicator info block with this type position data is used to indicate a position in the object referred by the object_position_number.

Address Offset	msb							lsb
00_{16}	object_position_number							
:								
:								
:	+ / -	hours						
:	minutes							
:	seconds							
:	x10ms							

Figure 4.2 – Object Position-Relative Clock Time Specification

4.3.8.1 Field Definitions

object_position_number: When the list is specified, the *object_position_number* field specifies the position number of the object in the list. When an object is specified, the *object_position_number* field is not used and set to 0. The number of bytes for this field is specified by the *size_of_object_position* field of the subunit identifier descriptor.

+/-: The +/- bit indicates the plus / minus sign of the following hours, minutes, seconds and x10ms information. If this bit is set to 1, the sign is minus. If this bit is set to 0, the sign is plus.

hours, minutes, seconds and x10ms: The *hours*, *minutes*, *seconds* and *x10ms* fields specify the hours, minutes, seconds and x10ms of the position.

The *x10ms* field specifies a value, which must be multiplied by 10 milliseconds to indicate the current fraction of the *seconds* field.

If the subunit is not able to support all of the fields in this structure (such as *x10ms*), then it shall specify a value of FF_{16} for the unsupported fields.

The starting point and the way of measurement of hours, minutes, seconds and x10ms depend on the data format of the AV stream referred by this info block.

4.3.9 Absolute Clock Time (07_{16})

The absolute clock time specification indicates the position relative to the beginning of the list, which contains the object being referenced. It has the following format:

Table 4.16 – Absolute Clock Time Position Specification

Address Offset	Contents
00_{16}	hours(MSB)
01_{16}	hours(LSB)
02_{16}	minutes
03_{16}	seconds
04_{16}	x10ms

4.3.9.1 Field Definitions

hours, minutes, seconds and x10ms: The *hours*, *minutes*, *seconds* and *x10ms* fields are as described above for the relative clock time.

The starting point and the way of measurement of hours, minutes, seconds and x10ms depend on the data format of the AV stream referred by this info block.

4.3.10 Relative Segment HMSF Count (08_{16})

The relative segment HMSF count type position data has the following format.

Address Offset	msb							lsb
00_{16}	object_position_number							
:								
n								
n+1	segment_number							
n+2								
n+3	+ / -	hours						
n+4	minutes							
n+5	seconds							
n+6	frames							

Figure 4.3 – Relative Segment HMSF Count

4.3.10.1 Field Definitions

object_position_number: The position indicator info block with this type of position data is used to indicate a position in the object referred by the *object_position_number*.

When the list is specified, the *object_position_number* field specifies the position number of the object in the list. When an object is specified, the *object_position_number* field is not used and set to 0. Its size, in bytes, is determined by the *size_of_object_position* field in the subunit identifier descriptor.

segment_number: The *segment_number* field (two bytes) specifies the segment.

+/-: The +/- bit indicates the plus / minus sign of the following hours, minutes, seconds and frames information. If this bit is set to 1, the sign is minus. If this bit is set to 0, the sign is plus.

hours, minutes, seconds and frames: The *hours, minutes, seconds and frames* fields specify the hours, minutes, seconds and frames of the position. If the subunit is not able to support all of the fields in this structure (such as frames), then it shall specify a value of FF_{16} for the unsupported fields.

The resolution of the frame depends on the data format of the AV stream referred by this info block.

The starting point and the way of measurement of hours, minutes, seconds and frames depend on the data format of the AV stream referred by this info block.

4.4 Position Info Block (00 03_{16})

The *position_info_block* describes the “position” of the AV stream. It can describe this location in several possible ways.

The subunit implementation decides which position specifications it can support, and reports them all in this info block by nesting any number of *position_indicator_info_block* structures.

The following diagram illustrates the *position_info_block* structure:

Table 4.17 – position_info_block

Address Offset	Contents
00 00_{16}	compound_length
00 01_{16}	
00 02_{16}	info_block_type = 00 03_{16} (position_info_block)
00 03_{16}	
00 04_{16}	primary_fields_length
00 05_{16}	
00 06_{16}	descriptor_reference
:	
:	
:	nested_position_indicator_info_blocks (secondary_fields)
:	
:	

4.4.1 Field Definitions

compound_length: The *compound_length* field specifies the number of bytes for the remainder of this information block (including any nested information blocks which may occur after the last primary field). Note that there are some nested information blocks shown for this structure; controllers should be prepared for any number and type of them to be found while parsing.

primary_fields_length: The *primary_fields_length* specifies the number of bytes for the remaining primary fields of this structure (through the *descriptor_reference* field in this case). All nested info blocks shall appear after this.

descriptor_reference: The *descriptor_reference* field specifies the object, or the list which contains the object currently being played or recorded. The format of this field is according to the descriptor reference structures mentioned in the reference [R10]. Generally, any object or list reference method is valid, but there may be restrictions imposed by certain subunit specifications. For information on possible restrictions, consult the appropriate subunit specification documentation.

position_indicator_info_block: The *secondary_fields* contains one or more *position_indicator_info_block* structures, in sequential order (not nested inside of each other). Each one of these indicates the position of the stream in some manner (by the total bytes, by the HMSF time, etc.). Each info block contains one position reporting structure; if several reporting methods are desired, then several position indicator info blocks will be used.

The position indicator info blocks are defined in the section titled Position Indicator Info Block (00 02_{16}) on page 15. When an object is referred to by *descriptor_reference*, only relative type *position_indicator_info_blocks* can be used.

The *position_indicator_info_block* structures must be reported in ascending order, based on the *indicator_type* field of the structure. For example, if the subunit only supports type 00_{16} and 02_{16}, then it reports them in that order.

4.5 Time Stamp Info Block (00 04 – 00 07_{16})

The time_stamp_info_block is used to describe the time when the event occurred.

Table 4.18 – Time Stamp Info Blocks structure

Address Offset	Contents
00_{16}	compound_length
01_{16}	
02_{16}	info_block_type = 00 04_{16} - 00 07_{16}
03_{16}	
04_{16}	primary_fields_length
05_{16}	
06_{16}	time_stamp_data
:	
:	

Four kinds of the time stamp info blocks are defined as follows:

Table 4.19 – type of time stamp

info_block_type	type of time stamp
00 04_{16}	content_creation_date_and_time
00 05_{16}	content_modification_date_and_time
00 06_{16}	descriptor_creation_date_and_time
00 07_{16}	descriptor_modification_date_and_time

4.5.1 Field Definitions

content_creation_date_and_time: The *content_creation_date_and_time* specifies recording date and time information about contents on the media.

content_modification_date_and_time: The *content_modification_date_and_time* specifies when the last change occurred to the entity which is represented by this time stamp info block.

descriptor_creation_date_and_time, descriptor_modification_date_and_time: The *descriptor_creation_date_and_time* and *descriptor_modification_date_and_time* variants indicate the creation and modification date, respectively, of the descriptor structure which contains this info block.

Note that changes to contents and descriptors can be independent from each other.

time_stamp_data: The *time_stamp_data* field contains the time stamp. It has the following format:

Table 4.20 – The *time_stamp_data* field format

Address Offset	msb							lsb
00_{16}	valid	stamp_type	year (MSB)					
01_{16}	year (LSB)							
02_{16}	month							
03_{16}	day							
04_{16}	hours							
05_{16}	minutes							
06_{16}	seconds							

valid: The *valid* bit indicates whether this time stamp is currently valid (= 1) or not (= 0).

stamp_type: If the *stamp_type* bit of the MSB is set to one, the structure contains a 54-bit counter which is incremented on each event, which is represented by the time stamp. If the msb is zero, the structure contains a time stamp as shown above.

date and time fields: The date and time fields are encoded as BCD values. Note that for the MSB of the year, only the higher-order two bits are used to encode the millennium, so this particular scheme is only valid until the year 3999.

For both versions of the time stamp, the *reserved* bit shall be set to 0 and treated according to the rules for reserved fields as described in [R9].

4.6 Character Code Info Block (00 08_{16})

The *character_code_info_block* describes the character code specification used to encode text in a structure. It has the following format:

Table 4.21 – character_code_info_block

Address Offset	Contents
00 00_{16}	compound_length
00 01_{16}	
00 02_{16}	info_block_type = 00 08_{16} (character_code_info_block)
00 03_{16}	
00 04_{16}	primary_fields_length
00 05_{16}	
00 06_{16}	character_code_type
00 07_{16}	character_code_type_specific
:	
:	

4.6.1 Field Definitions

character_code_type: The *character_code_type* field specifies the character code used for the text. It can take on one of the following values:

Table 4.22 – character_code_type

Value	Definition	references
00_{16}	ASCII	ISO 646(1983) Information processing – ISO 7bit coded character set for information interchange
01_{16}	ISO-8859	
02_{16}	MS-JIS	RIAJ document RIS506 Music Shift JIS Kanji character set
03_{16}	ITTS	IEC 61866, Audiovisual systems – interactive text transmission system
04_{16}	Korean	KS C 5601-1989, Standard Codes of Korean Characters for Information Interchange
05_{16}	Chinese	GB2312-80, Codes of Chinese Graphic Characters for Information Interchange, Primary Set
06_{16}	ISO 646	ISO 646
07_{16}	Shift JIS	JIS X 0208:1997
08_{16}	Japanese EUC	
80_{16}	MD-specific	
all others	reserved for future definition	

character_code_type_specific: The character_code_type_specific field provides more detail on the character code being used. The following structures are defined:

4.6.2 ASCII

There is no character code type-specific information defined for the ASCII type.

4.6.3 ISO-8859

The ISO-8859 type-specific information is defined as follows:

Table 4.23 – ISO-8859 character_code_type_specifc_info

value	meaning
00_{16}	ISO8859-1
01_{16}	ISO8859-2
02_{16}	ISO8859-3
03_{16}	ISO8859-4
04_{16}	ISO8859-5
05_{16}	ISO8859-6
06_{16}	ISO8859-7
07_{16}	ISO8859-8
08_{16}	ISO8859-9
09_{16}	ISO8859-10
all other values	reserved for future definition

The length of character_code_type_specific_info is 1byte.

4.6.4 MS-JIS

There is no type-specific information defined for the MS-JIS character code type.

4.6.5 ITTS

The ITTS type-specific information is defined as follows:

Table 4.24 – ITTS character_code_type_specific_info

value	meaning
00_{16}	Latin-based alphanumeric character set
01_{16}	Extended Latin-based alphanumeric character set
02_{16}	Mosaic and lined graphics font set
03_{16}	Japanese font table 1
04_{16}	Japanese font table 2
all other values	reserved for future definition

4.6.6 Korean

There is no type-specific information defined for the Korean character code type.

4.6.7 Chinese

There is no type-specific information defined for the Chinese character code type.

4.6.8 ISO 646

There is no type-specific information defined for the ISO 646 character code type.

4.6.9 Shift JIS

The shift JIS character code type-specific information is defined as follows:

Table 4.25 – Shift JIS specific character_code_type_specifc_info

value	meaning	References
00_{16}	Generic Shift JIS	JIS X 0208:1997
all other values	reserved for future definition	

4.6.10 Japanese EUC

The Japanese EUC type-specific information is defined as follows:

Table 4.26 – Japanese EUC character_code_type_specific_info

Value	Definition	References
00_{16}	eucJP	UI-OSF Application Platform Profile for Japanese Environment Version1.1
01_{16}	Japanese EUC for BS Digital	Enhanced Japanese EUC character set for Japanese BS digital
All other values	Reserved for future definition	

4.6.10.1 Japanese EUC character set for BS digital

Japanese EUC for BS digital follows JIS X 0202 and the encoding method is defined as follows;

Table 4.27 – se EUC character set for BS digital

Code set	Character set
G0	JIS X 0201 Roman character
G1	Extended JIS Kanji for BS digital
G2	none
G3	reserved

JIS X 0211 C0 control character set is invoked into the C0 area of the code system.

JIS X 0211 C1 control character set is invoked into the C1 area of the code system.

JIS X 0201 Roman character set is designated into G0 set, and the G0 set is invoked into the GL area of the code system.

Extended JIS Kanji for BS digital character set which is defined as follows is designated into G1 set, and the G1 set is invoked into the GR area of the code system.

Table 4.28 – Extended JIS Kanji for BS digital character set

Row	Character set
Rows 1 to 84	ARIB STD-B5 Kanji rows 1 to 84
Rows 85 to 94	Extra characters defined in Annex E.4 of ARIB TR-B15

A two-byte code character set can be designated into G3 set for future extension, and the G3 set is invoked into GR area of the code system by SS3(08/15) control character according to the code extension technique based on ARIB TR-B15.

Control characters that are supported in this character set are limited to APR(00/13), SS3(08/15) and SP(02/0).

Reference:
JIS X 0201:1997 7-bit and 8-bit coded character sets for information interchange[1]
JIS X 0202:1998 Character code structure and extension techniques[1]
JIS X 0211:1994 Character code structure and extension techniques[1]
ARIB STD-B5 Version 1.0: Data Multiplex Broadcasting System for the Conventional Television using the Vertical Blanking Internal.[2]
ARIB TR-B15 Version1.0: Operational Guidelines for Digital Satellite Broadcasting Services using Broadcasting Satellites.[2]

UI-OSF Application Platform Profile for Japanese Environment Version1.1[3]

1) JIS publications are available from Japanese Standards Association, 1-24, Akasaka 4-Chome, Minato-ku, Tokyo, 107-8440, Japan

2) ARIB publications are available from Association of Radio industries and Businesses, Nittochi Bldg. 14F, 1-4-1, Kasumigaseki, Chiyoda-ku, Tokyo 100-0013, Japan.

3) This publication is available from the Open Group Japan Vendors Council. The contact person is specified at the website: http://www.opengroup.or.jp/jvc/jvc_top.html

4.6.11 MD-Specific

The MD-specific character code type-specific information is defined as follows:

Table 4.29 – D-specific character_code_type_specifc_info

value	meaning
00_{16}	MD1
01_{16}	MD2
02_{16}	MD3
03_{16}	MD4
04_{16}	MD5
05_{16}	MD6
06_{16}	MD7
all other values	reserved for future definition

The length of character_code_type_specific_info is 1byte.

For details on the MD character code sets, please refer to reference [R17].

4.7 Language Code Info Block (00 09_{16})

The *language_code_info_block* specifies the language of the encoded text. It is defined as follows:

Table 4.30 – language_code_info_block

Address Offset	Contents
$00\ 00_{16}$	compound_length
$00\ 01_{16}$	
$00\ 02_{16}$	info_block_type = $00\ 09_{16}$ (langauge_code_info_block)
$00\ 03_{16}$	
$00\ 04_{16}$	primary_fields_length
$00\ 05_{16}$	
$00\ 06_{16}$	language_code_type
$00\ 07_{16}$	language_code_type_specific
:	
:	

4.7.1 Field Definitions

language_code_type: The *language_code_type* field specifies the language code used for the text. It can take on one of the following values:

Table 4.31 – language_code_type

Value	Definition	references
00_{16}	EBU Tech 3258-E (1991)	EBU Tech 3258 (1991) Specification of the systems of the MAC/packet family
01_{16}	ISO 639	ISO 639
all others	reserved for future definition	

language_code_type_specific: The *language_code_type_specific* field provides more detail on the language code being used. The following structures are defined:

4.7.2 EBU Tech 3258-E(1991)

The type specific info for EBU Tech 3258-E is one byte in length; the table of Language Codes is specified in annex 1 of Part 5 of EBU Tech 3258-E(1991).

4.7.3 ISO 639

The type specific info for ISO 639 is 2 bytes in length; the table of Language Codes is specified in No. ISO 639.

4.8 Raw Text Info Block ($00\ 0A_{16}$)

The *raw_text_info_block* contains only text data bytes. The format of the data bytes depends on the presence of *character_code_info_block* and *language_code_info_block* structures, which must immediately precede it. If those structures do not exist, then the default coding is minimal ASCII English.

The *raw_text_info_block* has the following format:

Table 4.32 – raw_text_info_block

Address Offset	Contents
00 00_{16}	compound_length
00 01_{16}	
00 02_{16}	info_block_type = 00 $0A_{16}$ (raw_text_info_block)
00 03_{16}	
00 04_{16}	primary_fields_length
00 05_{16}	
00 06_{16}	raw_text_data
:	
:	

4.8.1 Field Definitions

compound_length: The *compound_length* field specifies the number of bytes for the remainder of this information block (including any nested information blocks which may occur after the last primary field). Note that there are no nested information blocks shown for this structure, but controllers should be prepared for any blocks to be found in case this structure is extended in the future.

primary_fields_length: The *primary_fields_length* specifies the number of bytes for the remaining primary fields of this structure (through the *raw_text_data* field in this case). Any nested info blocks would appear after this.

raw_text_data: The *raw_text_data* field contains the text bytes. As mentioned above, if there are no character or language code info blocks accompanying this structure, then the default is minimal ASCII English.

Note that the character code and language code info blocks are not nested inside of the raw text info block; they accompany it inside of another block structure, such as the *name_info_block*. If character and language code blocks are accompanied, they MUST appear in this order: {character code, language code, raw text}. Other info blocks may precede or succeed them in the structure.

4.9 Name Info Block (00 $0B_{16}$)

The *name_info_block* specifies a name for the container of the information block. For example, if this info block is contained in a list, then it represents the name of the list. If that list represents some other entity (such as an audio disc), then the info block represents the name of that entity (the audio disc).

There may be several of these name info blocks in a given context, for example, one for each language.

The name info block has the following format:

Table 4.33 – name_info_block

Address Offset	Contents
$00\ 00_{16}$	compound_length
$00\ 01_{16}$	
$00\ 02_{16}$	info_block_type = $00\ 0B_{16}$ (name_info_block)
$00\ 03_{16}$	
$00\ 04_{16}$	primary_fields_length
$00\ 05_{16}$	
$00\ 06_{16}$	name_data_reference_type
$00\ 07_{16}$	name_data
:	
:	

4.9.1 Field Definitions

compound_length: The *compound_length* field specifies the number of bytes for the remainder of this information block (including any nested information blocks which may occur after the primary field). Note that there are no nested information blocks shown for this structure, but controllers should be prepared for any blocks to be found in case this structure is extended in the future.

primary_fields_length: The *primary_fields_length* specifies the number of bytes for the remaining primary fields of this structure (through the *name_data* field in this case). Any nested info blocks would appear after this.

name_data_reference_type: The *name_data_reference_type* field specifies how the name data is referenced. It can have one of the following values:

Table 4.34 – name_reference_type

name_reference_type	Meaning
00_{16}	Immediate – the name data is encoded directly in this structure.
01_{16}	Referenced – the name data is referenced by pointing to a list or object descriptor which contains text data.
all others	Reserved for future definition.

name_data: The *name_data* field contains the immediate or reference information. Its format is dependent on the value of *name_data_reference_type*. The following formats are defined:

Table 4.35 – name_data field for name_data_reference_type = 00_{16} (immediate)

Address Offset	Contents
00 00_{16}	name_data_attributes
00 01_{16}	maximum_number_of_characters
00 02_{16}	
00 03_{16}	character_code_info_block
:	
:	
:	language_code_info_block
:	
:	
:	raw_text_info_block
:	
:	

name_data_attributes: The *name_data_attributes* field specifies the attributes of this structure (including the character and language codes and the text).

Table 4.36 – name_data_attributes

bits	definition	meaning
xxxx xxx1	user_modifiable	1 = the user may modify this text 0 = this text is read-only
xxxx xx1x	stored_on_media	This bit is only available for storage subunit . If it is not storage subunit, this bit is set to 0. 1 = this text is stored on the media 0 = this text is stored in the subunit

maximum_number_of_characters: The *maximum_number_of_characters* field specifies a limitation, if any, on the number of **characters**, **not bytes**, for the immediate text. In some subunit implementations, or in some media specifications, there may be limits to fields such as disc and track titles, etc.

If the target does not define a *maximum_number_of_characters* field, then this field shall be set to FF FF_{16}.

character_code_info_block, language_code_info_block* and *raw_text_info_block: The *character_code_info_block*, *language_code_info_block* and *raw_text_info_block* are nested info blocks which describe the character code, language code and text bytes for this name data. There shall be exactly one of each of these info blocks in the structure, OR there may be only the *raw_text_info_block*; in the latter case, the default text format is minimal English ASCII as noted in the *raw_text_info_block* description.

NOTE —Because the character code, language code and raw text info blocks are part of the name_data field, the primary_fields_length value includes the bytes used for these two info blocks.

Table 4.37 – name_data field for name_data_reference_type = 01_{16} (referenced)

Address Offset	Contents
00 00_{16}	descriptor_identifier
:	
:	

descriptor_identifier: The *descriptor_identifier* field contains a reference to either a list of objects which contain text data, or to a single object which contains text data. The format of this field is as defined in the

general AV/C command OPEN DESCRIPTOR (see reference [R10]). The following descriptor types are valid for use in the *name_info_block* structure:

Table 4.38 – valid descriptor types

valid descriptor types	meaning	comments
10_{16}	List – reference by list ID	General – defined in ref [R10]
20_{16}	Object – reference by position	General – defined in ref [R10]
21_{16}	Object – reference by ID	General – defined in ref [R10]

4.10 Description Info Block (00 $0C_{16}$)

The description info block contains a user-readable textual description of the entity which contains this info block. For example, if it's in an object descriptor, then it's a description of that object.

The format of this info bock is the same as for the *name_info_block* structure, except for the *info_block_type* field. Note that there can be several of these description blocks, one for each supported language (depending on the implementation as described for the *name_info_block*).

4.11 Image Info Block (00 $0D_{16}$)

The *image_info_block* field contains a reference to an image object that represents the item, which contains this info block. It has the following format:

Table 4.39 – image_info_block

Address Offset	Contents
00 00_{16}	compound_length
00 01_{16}	
00 02_{16}	info_block_type = 00 $0D_{16}$ (image_info_block)
00 03_{16}	
00 04_{16}	primary_fields_length
00 05_{16}	
00 06_{16}	image_reference
:	
:	

4.11.1 Field Definitions

image_reference: The *image_reference* field contains a reference to a digital still image object. The format of this field is one of the descriptor identifiers described in reference [1]. The following descriptor types are valid for use in the *image_info_block* structure:

Table 4.40 – valid descriptor types

valid descriptor types	meaning	comments
10_{16}	List – reference by list ID	General – defined in ref [R10]
20_{16}	Object – reference by position	General – defined in ref [R10]
21_{16}	Object – reference by ID	General – defined in ref [R10]

4.12 Image Format Info Block (00 $0E_{16}$)

The *image_format_info_block* describes the image format of a digital still image object. It has the following structure:

Table 4.41 – image_format_info_block

Address Offset	Contents
00 00_{16}	compound_length
00 01_{16}	
00 02_{16}	info_block_type = 00 $0E_{16}$ (image_format_info_block)
00 03_{16}	
00 04_{16}	primary_fields_length
00 05_{16}	
00 06_{16}	image_format
00 07_{16}	image_format_specific
:	
:	

4.12.1 Field Definitions

image_format: The *image_format* field indicates the type of image this object descriptor represents. The following table specifies the currently defined values:

Table 4.42 – image_format

image_format	format	comments
80_{16}	MD1	Indicates the MiniDisc-audio MD1 image format. For details on the MD1 image code sets, please refer the reference [R17]. .
all others i	------	Reserved for future definition.

image_format_specific: The *image_format_specific* field contains further information about the image format. The interpretation of this field depends on the *image_format* field.

Image formats may be defined by the subunit specification (and documented here).

4.13 Descriptor Reference Info Block (00 $0F_{16}$)

The *descriptor_reference_info_block* is used to hold a *descriptor_identifier* structure, as defined in the OPEN DESCRIPTOR command description. Note that the general AV/C descriptor model defines various *descriptor_identifier* structures, and each subunit type may define type-specific structures as well.

Table 4.43 – descriptor_reference_info_block

Address Offset	Contents
00 00_{16}	compound_length
00 01_{16}	
00 02_{16}	info_block_type = 00 $0F_{16}$ (descriptor_reference_info_block)
00 03_{16}	
00 04_{16}	primary_fields_length
00 05_{16}	
00 06_{16}	descriptor_identifier
:	
:	

4.13.1 Field Definitions

descriptor_identifier: The *descriptor_identifier* field points to a descriptor reference, as defined either by the AV/C General Specification (version 3.0 or higher), or by a subunit-specific definition. Generally, the format of this descriptor reference may be any of the defined descriptor reference types, but there may be subunit-type or media-format restrictions. The type of descriptor referred to, and the means of making the reference (by position in a list, etc.) will be dictated by the context in which this info block is used.

4.14 Number of Items Info Block (00 10_{16})

The *number_of _items_info_block* field indicates the current number of "items" within the context of this info block. For example, when it is used in a *list_specific_information* area, it indicates the number of objects in that list.

NOTE —The use of the number_of_items_info_block does not change the currently defined list structure; there is still a number_of_entries field immediately following the list_specific_information area.

This info block has the following format:

Table 4.44 – number_of_items_info_block

Address Offset	Contents
00 00_{16}	compound_length
00 01_{16}	
00 02_{16}	info_block_type = 00 10_{16} (number_of_items_info_block)
00 03_{16}	
00 04_{16}	primary_fields_length
00 05_{16}	
00 06_{16}	number_of_items
:	
:	

4.14.1 Field Definitions

number_of_items: The *number_of_items* field contains a count value. The number of bytes used for this field is specified by the *size_of_object_position* field in the subunit identifier descriptor.

4.15 Descriptor Capacity Info Block (00 11_{16})

The descriptor_capacity_info_block is used to indicate the storage capacity of descriptor. It has the following format:

Table 4.45 – descriptor_capacity_info_block

Address Offset	Contents
00 00_{16}	compound_length
00 01_{16}	
00 02_{16}	info_block_type = 00 11_{16}
00 03_{16}	(descriptor_capacity_info_block)
00 04_{16}	primary_fields_length
00 05_{16}	
00 06_{16}	storage_location
00 07_{16}	remaining_capacity_length
00 08_{16}	
00 09_{16}	remaining_capacity
:	
:	
:	total_capacity_length
:	
:	total_capacity
:	
:	

4.15.1 Field Definitions

storage_location: The *storage_location* field specifies where the descriptive information is stored. The currently defined values are (00_{16} = stored on the subunit) and (01_{16} = stored in the media). These two values are available only for storage subunit. If it is not storage subunit, this value set to 00_{16}. All other values are reserved for future specification.

remaining_capacity_length: The *remaining_capacity_length* field specifies the number of bytes for the following *remaining_capacity* field.

remaining_capacity: The *remaining_capacity* field specifies the size, measured in bytes, of the unused portion of storage allocated to descriptive information.

total_capacity_length: The *total_capacity_length* field specifies the number of bytes for the following *total_capacity* field.

total_capacity: The *total_capacity* field specifies the total size, in bytes, of the storage space allocated to descriptive information.

The definition of descriptor capacity should be done where this info block is used.

This page is left intentionally blank

Annexes

Annex A: Subunit Defined Information Blocks

The following table contains AV/C information block type specification values, names, and comments for specific subunit types. See each specification for more information about the definitions of the information blocks.

Table A.1 – Disc Subunit defined Info Block Types

info block type	info block name	comments
80 00_{16}	Artist Info Block	Disc General [R11]
80 01_{16}	Genre Info Block	Disc General [R11]
80 02_{16}	Disc Capacity Info Block	Disc General [R11]
80 03_{16}	AV Object Type Specific Capacity Info Block	Disc General [R11]
80 04_{16}	AV Content Identifier Info Block	Disc General [R11]
80 05_{16}	Disc Catalog Code Info Block	Disc General [R11]
80 06_{16}	File Format Info Block	Disc General [R11]
80 07_{16}	Audio Recording Parameters Info Block	Disc General [R11]
80 08_{16}	Synchro Performance List and Plug Pairs Info Block	Disc General [R11]
80 09_{16}	Reserved for disc subunit defined Info Block	
80 $0A_{16}$	Text Database Content Attributes Info Block	Disc General [R11]
80 $0B_{16}$	Default Playlist Info Block	Disc General [R11]
80 $0C_{16}$	Text Content type	Disc General [R11]
80 $0D_{16}$	Output Start Time Info Block	Disc General [R11]
80 $0E_{16}$	Presentation Start Time Info Block	Disc General [R11]
80 $0F_{16}$	Presentation End Time Info Block	Disc General [R11]
80 10_{16}	Content Entry Point Info Block	Disc General [R11]
80 11_{16}	Content exit point Info Block	Disc General [R11]
80 12_{16}	Track Number Offset Info Block	CD-DA [R12]
80 13_{16}	Album Set Info Block	SACD [R13]
80 14_{16}	Album Catalog Code Info Block	SACD [R13]
80 15_{16}	SACD specific Info Block	SACD [R13]
80 16_{16} – 80 FF_{16}	Reserved for disc subunit defined Info Block	
81 00_{16} – 87 FF_{16}	Reserved for future definition of subunit defined Info Block	

Table A.2 – Disc Subunit defined Info Block Types

info block type	info block name	comments
88 00_{16}	General Disc Subunit Status Area Info Block	Disc General [R11]
88 01_{16}	Destination Plug Status Area Info Block	Disc General [R11]
88 02_{16}	Source Plug Status Area Info Block	Disc General [R11]
88 03_{16}	Synchro Plug Group Status Area Info Block	Disc General [R11]
88 04_{16}	Media and Edit Status Info Block	Disc General [R11]
88 05_{16}	Plug Status Info Block	Disc General [R11]
88 06_{16}	Operation Mode Info Block	Disc General [R11]
88 07_{16}	Plug Configuration Info Block	Disc General [R11]
88 08_{16}	Playback Order Configuration Info Block	Disc General [R11]
88 09_{16}	Audio Level Meter Status Info Block	Disc General [R11]
88 $0A_{16}$	Monitor Status Info Block	Disc General [R11]
88 $0B_{16}$	Reserved for disc subunit defined Info Block	
88 $0C_{16}$	Position Marker Info Block	Hard Disk Drive [R14]
88 $0D_{16}$	Segment Marker Info Block	Hard Disk Drive [R14]
88 $0E_{16}$	Recording Position Info Block	Hard Disk Drive [R14]
88 $0F_{16}$	Playback Position Info Block	Hard Disk Drive [R14]
88 10_{16}	Latest Recording Mode Info Block	Hard Disk Drive [R14]
88 11_{16}	Previously Recorded Segment Info Block	Hard Disk Drive [R14]
88 12_{16}	Plug Configuration	Hard Disk Drive [R14]
88 13_{16}	Video Stream Format Subtype Info Block	Hard Disk Drive [R14]
88 14_{16}	Program Attribute Info Block	Hard Disk Drive [R14]
88 15_{16}	Object Subtype Info Block	Hard Disk Drive [R14]
88 16_{16} – 88FF_{16}	Reserved for disc subunit defined Info Block	

Table A.3 – Bulletin Board subunit defined Info Block Type(s) (See reference [R15])

info block type	info block name	comments
89 00_{16}	Subunit Resource Info Block	
89 01_{16} – 89FF_{16}	Reserved for future definition of subunit defined Info Block	

Table A.4 – CA subunit defined Info Block Types (See reference [R16])

info block type	info block name	comments
90 00_{16}	General CA Subunit Status Area Info Block	
90 01_{16}	Source Plug Status Area Info Block	
90 02_{16}	Plug Status Info Block	
90 03_{16} – FF FF_{16}	Reserved for future definition of subunit defined Info Block	