SMPTE 298-2009
Revision of
ANSI/SMPTE 298M-1997

SMPTE STANDARD

Universal Labels for Unique Identification of Digital Data

Table of Contents

Approved
June 5, 2009

Foreword

SMPTE (the Society of Motion Picture and Television Engineers) is an internationally-recognized standards developing organization. Headquartered and incorporated in the United States of America, SMPTE has members in over 80 countries on six continents. SMPTE's Engineering Documents, including Standards, Recommended Practices and Engineering Guidelines, are prepared by SMPTE's Technology Committees. Participation in these Committees is open to all with a bona fide interest in their work. SMPTE cooperates closely with other standards-developing organizations, including ISO, IEC and ITU.

SMPTE Engineering Documents are drafted in accordance with the rules given in Part XIII of its Administrative Practices.

SMPTE Standard 298 was prepared by Technology Committee 30MR.

Intellectual Property

At the time of publication no notice had been received by SMPTE claiming patent rights essential to the implementation of this Standard. However, attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. SMPTE shall not be held responsible for identifying any or all such patent rights.

Introduction

This section is entirely informative and does not form an integral part of this Engineering Document.

This standard defines Universal Labels that can be used by any organization that wishes to label data in a manner that is universally unambiguous, globally unique, and traceable to the authorizing organization that defines their meaning. They are a form of metadata that is intended to be associated with other data or metadata for the purpose of identifying that data or metadata or of specifying the way that data or metadata is to be interpreted.

Universal Labels are intended to function across all types of digital communications protocols, message structures, and storage formats and to allow the intermixture of data of any sort. These labels may be attached to the data they identify and travel with them through communications channels.

The meanings of the Universal Labels are defined by organizations that are part of a hierarchical structure, beginning with the International Organization for Standardization (ISO) and the International Telecommunication Union (ITU), which share the top level. Each organization in the hierarchy is automatically assigned part of a specified name space to use in defining labels for data types and forms under its responsibility. SMPTE is one such organization.

1 Scope

This standard defines Universal Labels as a means to identify the type and encoding of data within a general-purpose data stream or file. These Universal Labels can be used by any organization that wishes to label data in a manner that is universally unambiguous, globally unique, and traceable to the authorizing organization.

While the primary use of Universal Labels is to identify the type and encoding of data within a general-purpose data stream or file, other meanings may be associated with specific labels; such additional meanings are beyond the scope of this standard and remain completely within the control of the particular organizations that define those labels.

A specific form of Universal Label, called an SMPTE-administered Universal Label is defined. It is the mechanism used to identify and distinguish data contained in data streams or files that have been generated according to the provisions of other SMPTE Standards and Recommended Practices.

2 Conformance Notation

Normative text is text that describes elements of the design that are indispensable or contains the conformance language keywords: "shall", "should", or "may". Informative text is text that is potentially helpful to the user, but not indispensable, and can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in this document is, by default, normative, except: the Introduction, any section explicitly labeled as "Informative" or individual paragraphs that start with "Note:"

The keywords "shall" and "shall not" indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

The keywords, "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document.

The keyword “reserved” indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword “forbidden” indicates “reserved” and in addition indicates that the provision will never be defined in the future.

A conformant implementation according to this document is one that includes all mandatory provisions ("shall") and, if implemented, all recommended provisions ("should") as described. A conformant implementation need not implement optional provisions ("may") and need not implement them as described.

3 Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this standard. 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 edition of the standards indicated below.

ISO/IEC 8824-1:2002 (ITU-T X.680), Information Technology — Abstract Syntax Notation One (ASN.1) — Specification of Basic Notation

ISO/IEC 8825-1:2002 (ITU-T X.690), Information Technology — ASN.1 Encoding Rules — Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER), and Distinguished Encoding Rules (DER)

SMPTE 2029-2009, Uniform Resource Names for SMPTE Resources

4 Glossary of Terms

The following sub-clauses define the most important terms used in this standard:

4.1 Universal Label

A variable-length label, defined by this standard and administered by an organization that is registered internationally by ISO/ITU or one of their constituent organizations.

4.2 SMPTE-Administered Universal Label

A fixed-length (16-byte) universal label, defined by this standard and administered by SMPTE.

5 Label Structure

Within this standard, the type and format or other characteristics of messages exchanged in a general-purpose data stream shall be identified using the ISO/ITU registration system as specified by ASN.1 in ISO/IEC 8824-1. A universal label is composed of an ISO/ITU identifier or path specification to an authorized administrative organization within the ISO/ITU name space, followed by a data type identifier defined by that organization and located within its subsection of the name space.

6 Label Definition and Name Space

A universal label shall be an “object identifier” and its name space shall conform to the definition of object identifiers specified by ISO/IEC 8824-1.

Note: The ISO/ITU identifiers are organized in a hierarchy that is registered internationally by ISO, ITU, and their constituent organizations. The root or initial part of the identifier hierarchy is specified in ISO/IEC 8824-1 Annexes B, C, and D. See Annex C of this standard for a further description of the ISO/ITU identifier hierarchy registration system.

7 Notation

A universal label shall utilize the notation specified by ISO/ITU 8824-1, ASN.1, for the object identifier type. ASN.1 specifies a human readable notation for the precise representation of object identifiers within the ISO/ITU identifier hierarchy. A universal label includes an identifier; the identifier is represented by a sequence of unsigned integer components, enclosed within braces, that designate a location in the identifier name space starting at the root of the ISO/ITU hierarchy and selecting which branch to take as each level of the hierarchy is descended.

Names consisting of strings of lower-case characters may be used in addition to or, in certain circumstances, in place of the integer components. Names assigned in annexes B, C, and D of ISO/ITU 8824-1 may be used in place of the integer components at the root of the sequence. Names assigned by an identified organization may be used, followed by their equivalent integer values enclosed in parentheses. See Annex C of this standard for a further description and example of the ISO/ITU identifier notation.

8 Universal Label Encoding

Universal labels shall be encoded according to the basic encoding rules (BER) as specified by ISO/IEC 8825-1, with the additional requirement that the encoding of the object identifier value shall use either “primitive” encoding or “constructed” encoding at the option of the authorizing organization.

Universal labels, as defined in this standard, when administered by an organization registered within the ISO/ITU identifier hierarchy, shall use either the primitive encoding or the constructed encoding with a length that is a multiple of 4 bytes.

SMPTE-administered universal labels shall be encoded using primitive encoding only, and their values shall be defined so that the primitive encoding of the entire label has a fixed length of 16 bytes.

8.1 Primitive Encoding

The primitive encoding of a universal label value shall be as specified by ISO/IEC 8825-1 (BER) sub-clause 8.19, “Encoding of an Object Identifier Value.” (See Annex D for a description of primitive encoding) If necessary, an organization shall supplement its object identifier with up to three null values (corresponding to additional branches in the ISO/ITU hierarchy) so that the encoded length of the universal label is an integer multiple of 4 bytes.

The encoded label consists of an object identifier (OID) tag (06_{16} per ASN.1) and a length followed by a sequence of sub-identifier bytes. The first two integer components of the object identifier are coded within the first sub-identifier byte. Each additional integer component of the object identifier is encoded by a sequence of one or more sub-identifier bytes. Following is an example of the primitive encoding of the universal label having the value {itu(0) recommendation(0) t(20) 4}, or simply {0 0 20 4}.

OID tag	Length	Contents
06_{16}	03_{16}	00_{16} 14_{16} 04_{16}

A SMPTE-administered universal label shall be encoded using primitive encoding and shall have a fixed 4-byte prefix that identifies the SMPTE organization in the ISO/ITU hierarchy, followed by a 12-byte sub-identifier string assigned and administered by SMPTE. Following is an example of the 16-byte SMPTE-administered universal label having the value {iso(1) organization(3) smpte(52) committee(18) standard (10) identifier(1) 0 0 0 0 0 0 0 0 0}, or (as decimal numbers) {1 3 52 18 10 1 0 0 0 0 0 0 0 0 0}

OID tag	Length	SMPTE	Sub-identifiers
06_{16}	$0E_{16}$	$2B34_{16}$	12_{16} $0A_{16}$ 01_{16} 00_{16} 00_{16} 00_{16} 00_{16} 00_{16} 00_{16} 00_{16} 00_{16} 00_{16} 00_{16}

Note: SMPTE-administered universal labels are zero-padded to create 16-byte encoded labels. The use of this example does not imply that such a structure will ultimately be approved by SMPTE.

8.2 Constructed Encoding

When an organization specifies alternative encoding rules for its data type identifiers, a constructed encoding of a universal label value may be used. The constructed encoding of a universal label value shall be as specified by ISO/IEC 8825-1 (BER) sub-clause 8.1, General Rules for Encoding, regarding constructed types. The encoded label consists of the following sequence: a constructed object identifier (CID) tag with the constructed bit set (26_{16}), a length, a primitively encoded object identifier (OID – as described in section 9.1 of

this standard), an byte string identifier (OCT) tag, and a primitively encoded byte string. If necessary, an organization shall supplement its object identifier with up to three null values (corresponding to additional branches in the ISO/ITU hierarchy) so that the encoded length of the universal label has a length that is an integer multiple of 4 bytes. The primitively encoded byte string consists of an identifier tag (04_{16}), a length, and a sequence of bytes. (See annex E for a detailed description of constructed encoding.)

For example, {1 3 64 “00 10 0A FF”} represents a constructed-encoding form of the universal label having an organization identifier value of {iso(1) organization(3) hypothetical-organization(64)}, and a data type identifier value specified by the organization to be encoded as 00_{16} 10_{16} $0A_{16}$ FF_{16}:

CID tag	Length	OID tag	Length	Org ID	OCT tag	Length	Contents
26_{16}	$0A_{16}$	06_{16}	02_{16}	$2B_{16}40_{16}$	04_{16}	04_{16}	00_{16} 10_{16} $0A_{16}$ FF_{16}

Annex A (Informative)
Bibliography

RFC 5119, A Uniform Resource Name (URN) Namespace for the Society of Motion Picture and Television Engineers (SMPTE), published by the Internet Engineering Task Force (IETF)

Annex B (Normative)
Plain Text Form of SMPTE-Administered Universal Label

Universal labels are described in this standard using the notation defined by ISO/IEC 8824-1 for object identifiers. They are encoded according to the basic encoding rules of ISO/IEC 8825-1, yielding a byte string.

Certain applications may require that a SMPTE-administered universal label be represented in a plain text format, either for clarity or for cross-reference to a data stream containing the actual universal label. In these applications, the plain text representation of the universal label should use the urn format as defined in standard SMPTE 2029.

Note: An example of a universal label using the SMPTE 2029 urn format is:

urn:smpte:ul:060E2B34.01010101.07020101.01040000

Annex C (Informative)
ISO/ITU Identifier Hierarchy Registration System and Notation

ISO/ITU identifiers are organized in a hierarchy that is registered and administered internationally by ISO, ITU, and their constituent organizations. The roots (prefixes) of the hierarchy are illustrated in Figure C.1.

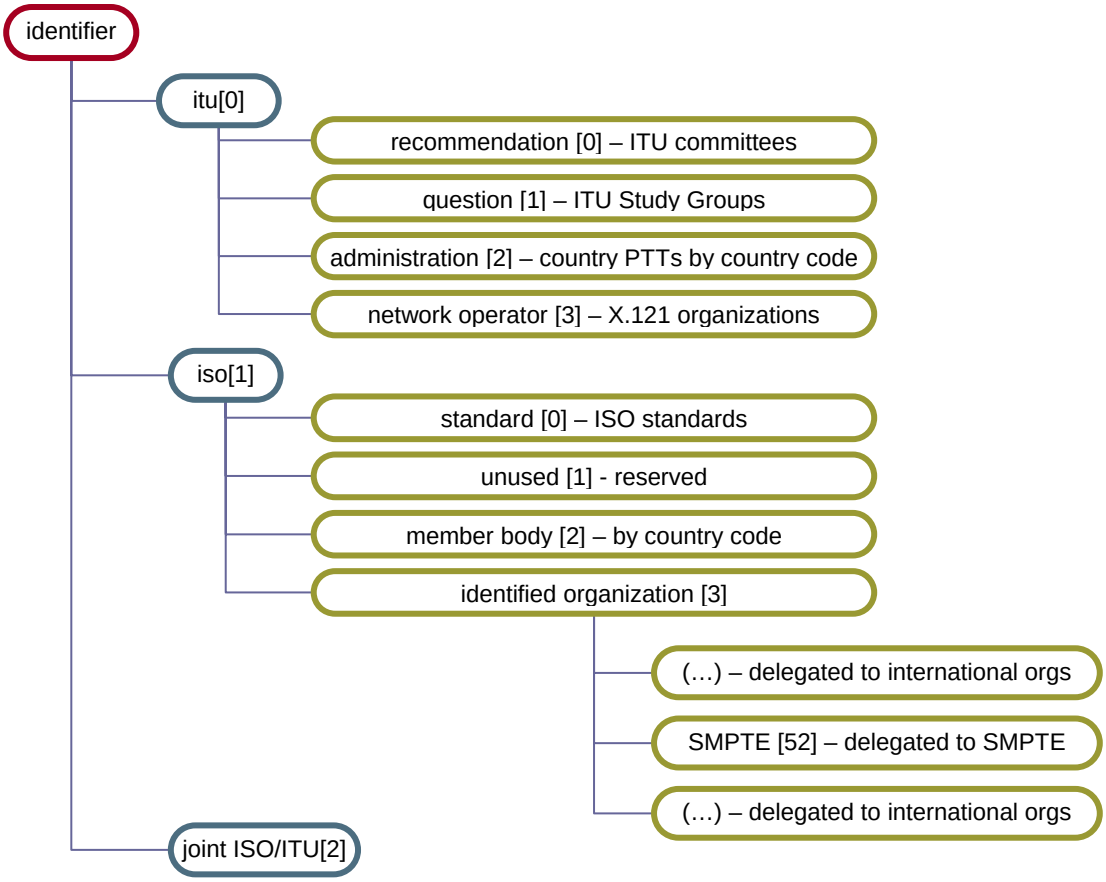

Figure C.1 – ISO/ITU identifier hierarchy

A defined notation is used to specify names in the ISO/ITU name space hierarchy. A name is a sequence of unsigned integers that designates a location in the name space, starting at the root and selecting which branch to take as each level is descended. In ASN.1 and its related encoding standards, the human readable notation for the sequence of unsigned integers is a sequence of decimal numbers within braces. Names are sometimes assigned to the unsigned integers and may be used in their place. (See ISO 8824 clause 29 for a description of the notation of the object identifier type.)

The following examples are all valid identifiers to represent the ITU T.4 (Group 3 Fax) specification:

{itu(0) recommendation(0) t(20) 4}

{0 0 20 4}

{itu recommendation t(20) 4}

The initial symbol “itu” (value 0) indicates the name is an International Telecommunication Union (ITU)-administered name. The second symbol “recommendation” (value 0) indicates the name is an ITU recommendation. ITU recommendations are further classified into series by letter a through z, represented numerically by 1 – 26. Thus 20 indicates series t. The branch below this identifies the document within that series. In this case, value 4 identifies the committee document that specifies Group 3 Fax. For example, using this name in a message header would identify a message as containing Group 3 Fax data.

All ISO standards are registered as {iso(1) standard(0)}. Sovereign bodies are registered under {iso(1) member-body(2) country-code}. ISO delegates registration authority to international organizations (e.g., SMPTE) and companies under {iso(1) organization(3)} so that individual organizations can autonomously administer a portion of the name space.

Annex D (Informative)
Primitive Encoding

As defined by this standard, the primitive encoding of a universal label is specified by the basic encoding rules (BER) for the encoding of an object identifier value. The encoding of an object identifier follows the BER general encoding for a data value with definite length and short form that consists of an OID (object identifier) byte, a size byte, a code byte, and sub-id (sub-identifier) bytes, as illustrated in Figure D.1.

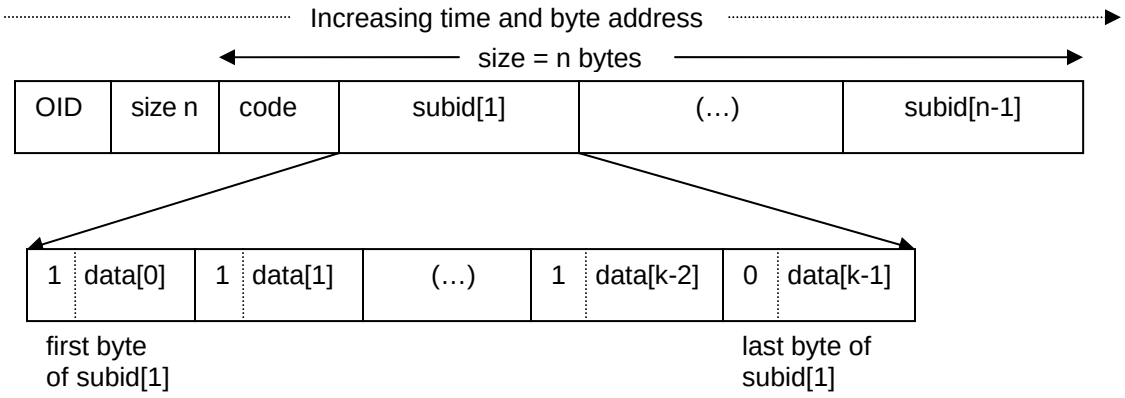

Figure D.1 — Object identifier encoding

The OID byte has a value of 06_{16}. Bit 6 of the OID byte indicates whether constructed or primitive encoding is used, with bit 6=0 indicating primitive encoding. The size byte specifies the number of additional bytes used to encode the value of the identifier, which, in the short form, can specify from 0 to 127 bytes.

The value of an object identifier consists of a sequence of unsigned integer components that represent the location of the identifier within the ISO/ITU hierarchy. Since the first two integer components that represent the root of the ISO/ITU hierarchy have values that are restricted to integers less than 16, these two integer values can be packed into a single code byte using the formula specified by equation D.1, where id[0] and id[1] represent the first and second integer component values, respectively.

$$code = (40_{10} \times id[0]) + id[1] \quad \text{(D.1)}$$

The code-value ranges generated by the code byte encoding are tabulated in Table D.1.

The remaining integer components of the object identifier are encoded as additional sub-identifier bytes, sub-id[1] through sub-id[n-1], as illustrated in figure D.1. Each integer component is encoded as one or more sub-id bytes depending on the value of the component. Each sub-id byte codes 7 bits of the integer value; therefore, a single sub-id byte can code an integer component with a value from 0 to 127. Integer component values greater than 127 are coded using additional sub-id bytes with the most significant bits of the integer value coded first. Bit 8 of the sub-id bytes is used to specify the end of each integer coding sequence. The most significant bit in the first through the next-to-last byte of each sub-id sequence is one; the most significant bit in the last byte of each sub-id sequence is zero.

Within each sub-identifier, the first-through-last bytes are used to encode the most- through least-significant portions of the unsigned integer object identifier component. Within Figure D.1, for example, the data[0]-through-data[k-1] bytes are the most- through least-significant portions of the unsigned integer being encoded.

Table D.1 – Identifier code field values

Organization	Code	Name	Description
ITU	0	Recommendation	ITU committees
	1	Question	ITU study groups
	2	Administration	Country PTTs (country code)
	3	Network operator	X.121 organizations
	4–39	—	Reserved
ISO	40	Standard	ISO standards
	41	—	Reserved
	42	Member body	Member bodies (country code)
	43	Identified organization	Organizations
	44–79	—	Reserved
Joint ISO/ITU	80–127	—	(Delegated to ANSI)

For example, the 16-byte SMPTE-administered universal label for a hypothetical SMPTE document 128 would be printed as {1 3 52 128 ...} which starts with the 6-byte "$06_{16}0E_{16}2B_{16}34_{16}81_{16}00_{16}$" sequence. This encoding is illustrated in Figure D.2.

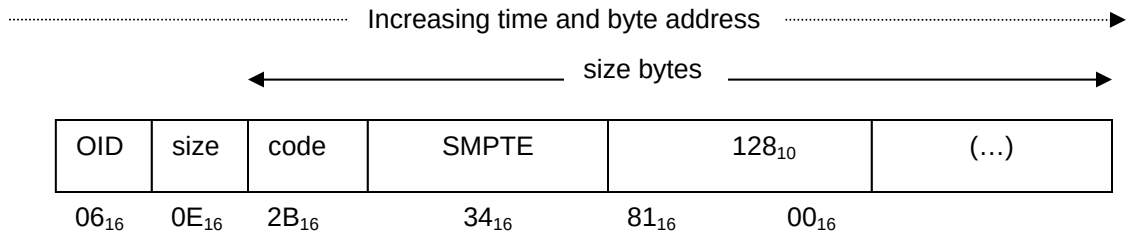

Figure D.2 – A typical, SMPTE-assigned, universal label with primitive encoding

Annex E (Informative)
Constructed Encoding

The constructed encoding of a universal label has a leading CID (constructed object identifier) and size byte. The distinct CID value of 26_{16} distinguishes the constructed encoding from the primitive encoding option. (For the primitive encoding, the first byte is 06_{16}.) These bytes are followed by a primitive object identifier (which identifies the organization) and data-type identifier (provided by the organization), as illustrated in Figure E.1.

The data-type identifier consists of an OCT (byte string) identifier, a size C byte, and other bytes. The OCT byte has a value of 04_{16}. The format and function of the remaining bytes shall be defined by the organization specified within the primitive object identifier.

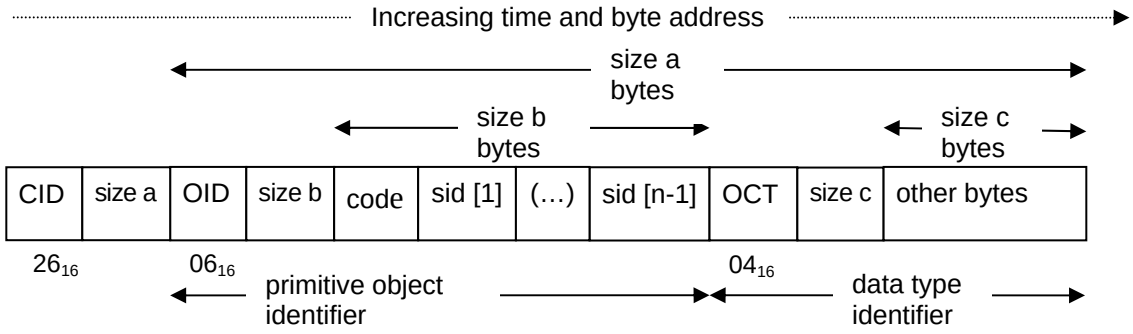

Figure E.1 – Constructed encoding format

For example, the constructed encoding for ISO standard #5 would include the organization's object identifier {iso(1) standard(1) 5} value and (for example) an assigned 6-byte value of AC_{16} DE_{16} 48_{16} 23_{16} 45_{16} 67_{16}. This encoding, with the object identifier extended to make the universal label a multiple of 4 bytes in size, is represented by {1 1 5 0 0 "AC DE 48 23 45 67"} and is encoded as illustrated in Figure E.2.

Because many organizations have larger (than 2-byte) code/sub-identifier values and may assign a larger-than 6-byte data-type identifier, the constructed form of the universal label often is expected to be longer than this 16-byte illustration.

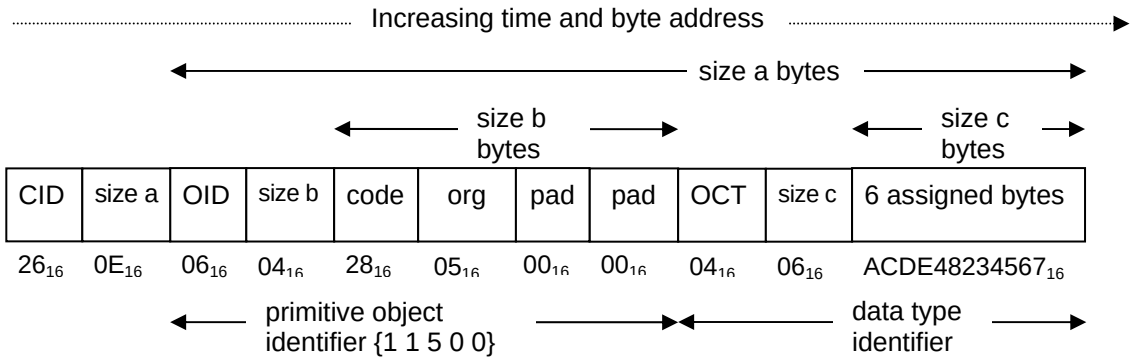

Figure E.2 – A universal label, constructed encoding

Annex F (Informative)
Universal Label Comparisons

F.1 Arbitrary Label Comparisons

Within most applications, the universal labels can be checked (for a match to several supported values) without parsing of the label contents. Equality comparisons can be performed between inputs and tables of reference universal label values. Label checking hardware need only know the data-stream location of the universal label (this might, in some applications, involve a first-byte comparison) and the label length (two more than the label's second-byte value).

Simple, fixed-length (in the following example, 64-byte) comparisons between the incoming label and a table of pre-encoded universal label values are sufficient to detect which of N (in this example, N=4) universal label values have been received, as illustrated in Figure F.1.

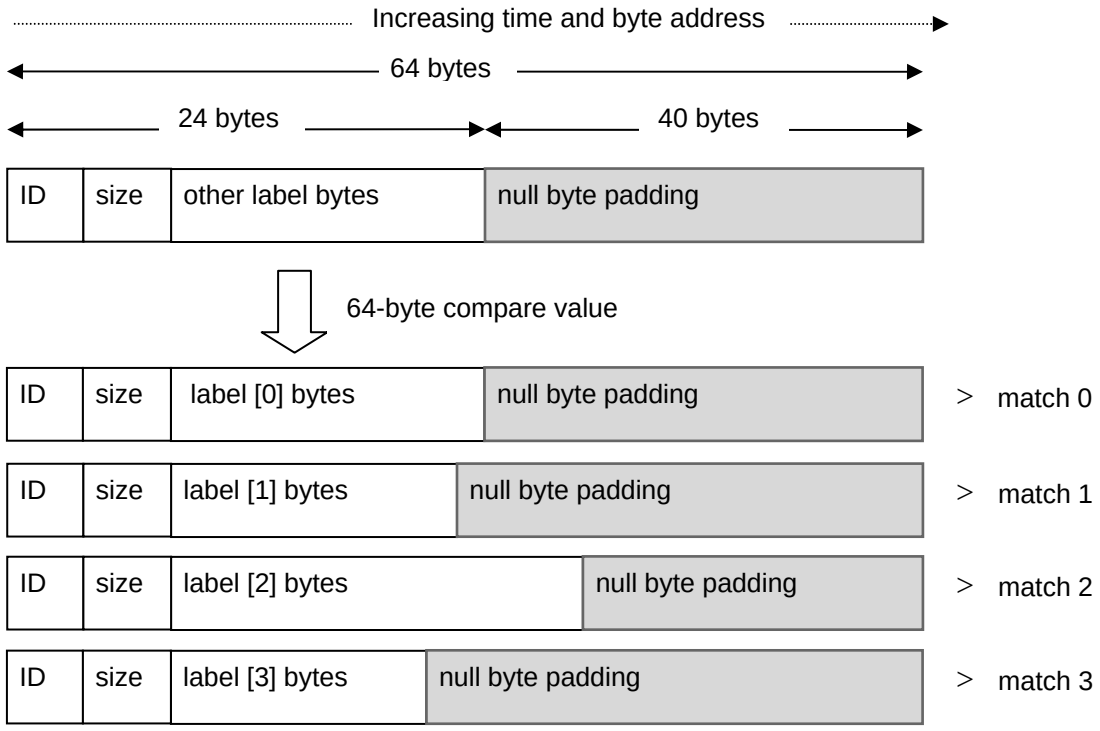

Figure F.1 – Arbitrary (up to 64-byte) label comparisons

In this example, the incoming universal label value is extended with null (zero-valued) bytes to form a fixed-length, 64-byte value. This post-padded value can be compared against a table of (also post-padded) universal label values, to determine which label has been received. The known location of the universal label's size field simplifies the post-padding operation; the size value also allows unsupported labels to be skipped when their contents do not match any supported universal label values.

Note that any post-pad value can be used, if the value is used when generating table entries and padding input label values. Similarly, a pre-padding extension also could be used, if table entries and input labels are pre-padded in a consistent fashion.

F.2 SMPTE Label Comparisons

Label recognition is simpler when only SMPTE-administered labels are supported, as illustrated in Figure F.2. In this case, the first four bytes are compared for a SMPTE-prefix match. If these match, the remaining 12 bytes can be compared to a table of 12-byte, SMPTE-supported, sub-label values.

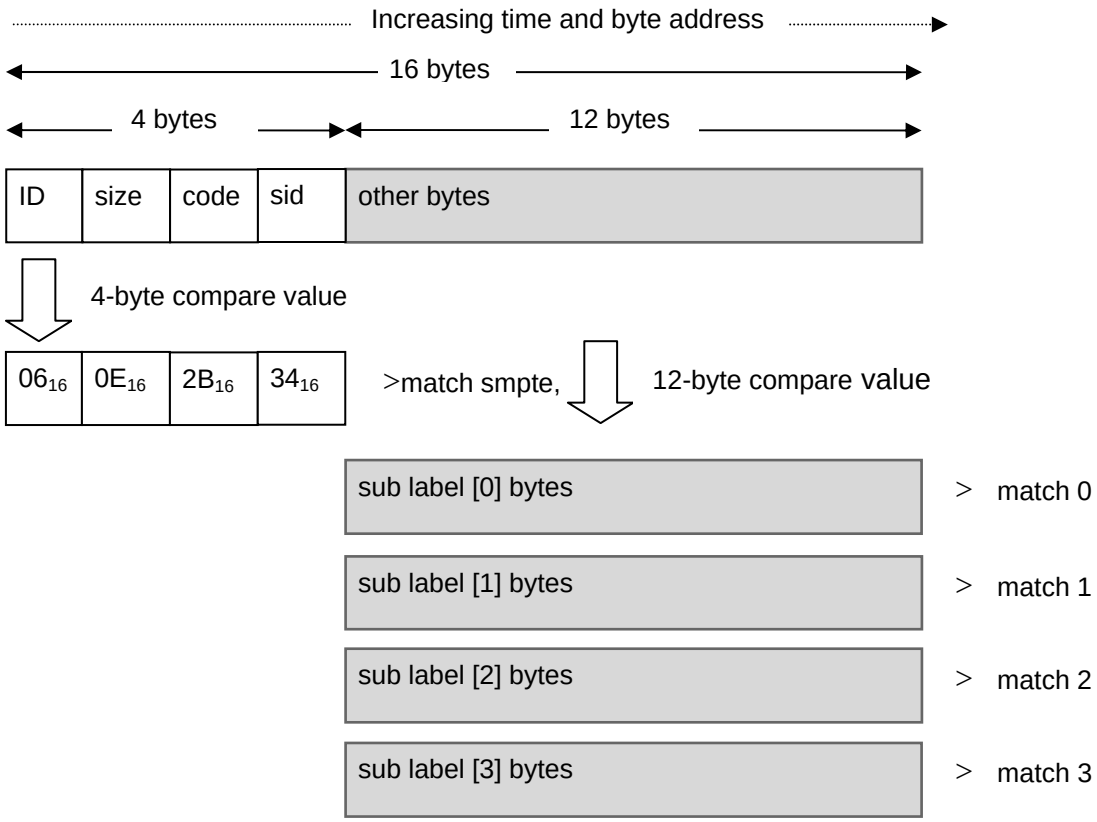

Figure F.2 – SMPTE-administered label comparisons