SMPTE RECCOMENDED PRACTICE

Digital Object Identifier (DOI) Name and Entertainment ID Registry (EIDR) Identifier Representations

Table of Contents **Page**

Approved August 22, 2018

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 its Operations Manual.

SMPTE RP 2079 was prepared by Technology Committee 30MR.

Introduction

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

The digital object identifier (DOI) system specified in ISO/IEC 26324 provides an infrastructure for persistent unique identification of objects in the form of a DOI name. An Entertainment ID Registry (EIDR) Identifier is a particular kind of a DOI name used for global unique identification of movie and TV content. It is desirable to facilitate interchange of DOI names and EIDR Identifiers in MXF and other SMPTE engineering documents by unambiguously specifying their representation.

This revision of the specification introduces a new kind of EIDR Identifier – the EIDR Service Identifier, – in addition to the EIDR Content Identifier present in earlier revisions. It also deprecates elements that have not seen practical application.

1 Scope

This specification defines specific text and binary representations for digital object identifier (DOI) names and Entertainment ID Registry (EIDR) Identifiers. It also defines dictionary entries, e.g. types and elements, for use in MXF and other KLV-based applications.

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.

Unless otherwise specified, the order of precedence of the types of normative information in this document shall be as follows: Normative prose shall be the authoritative definition; Tables shall be next; followed by formal languages; then figures; and then any other language forms.

3 Normative References

The following standards contain provisions that, through reference in this text, constitute provisions of this recommended practice. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this recommended practice are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

ISO/IEC 7064:2003 Information Technology — Security Techniques — Check Character Systems

ISO/IEC 26324:2012, Information and Documentation — Digital Object Identifier System

SMPTE ST 335:2012, Metadata Element Dictionary Structure

SMPTE ST 2003:2012, Types Dictionary Structure

Internet Engineering Task Force (IETF) (January 2008). RFC 5234, Augmented BNF for Syntax Specifications: ABNF

Internet Engineering Task Force (IETF) (September 2016). RFC 7972, Entertainment Identifier Registry (EIDR) URN Namespace Definition

4 Canonical DOI Name

A Canonical DOI Name is defined as a DOI name string as specified in Section 4.1 of ISO/IEC 26324.

Example: `10.1000/123456` is a Canonical DOI Name with the DOI prefix `10.1000` and the DOI suffix `123456`.

Note: A DOI name can be resolved at https://doi.org/ .

5 Canonical EIDR Content Identifier

A Canonical EIDR Content Identifier shall be a DOI name string as specified in ISO/IEC 26324, conforming to the `EIDR-CONTENT-ID` syntax specified below (using ABNF as specified in IETF RFC 5234):

```
EIDR-CONTENT-ID          = EIDR-CONTENT-ID-PREFIX "/" EIDR-CONTENT-ID-SUFFIX
EIDR-CONTENT-ID-PREFIX   = "10.5240"
EIDR-CONTENT-ID-SUFFIX   = 5*5(4*4HEXDIG "-") CHECK

DIGIT         = %x30-39                                          ; 0-9
HEXDIG        = DIGIT / "A" / "B" / "C" / "D" / "E" / "F"
ALPHA         = %x41-5A / %x61-7A                                ; A-Z / a-z
CHECK         = DIGIT / ALPHA
```

`CHECK` shall be the Mod 37,36 check character as specified in ISO/IEC 7064, computed over the 20 hexadecimal digits `HEXDIG` of `EIDR-CONTENT-ID-SUFFIX`.

Canonical EIDR Content Identifiers shall be case insensitive.

Example: `10.5240/5FD4-FEE1-22F5-583E-FECC-O` and `10.5240/5fd4-FEE1-22F5-583E-fecc-o` correspond to the same EIDR Content Identifier.

Note 1: The check character is computed over the DOI suffix alone: if the prefix is not `10.5240`, the DOI name is not a Canonical EIDR Content Identifier.

The term Canonical EIDR Identifier is a synonym of the term Canonical EIDR Content Identifier.

6 Canonical EIDR Service Identifier

A Canonical EIDR Service Identifier shall be a DOI name string as specified in ISO/IEC 26324, conforming to the `EIDR-SERVICE-ID` syntax specified below (using ABNF as specified in IETF RFC 5234):

```
EIDR-SERVICE-ID                = EIDR-SERVICE-ID-PREFIX "/" EIDR-SERVICE-ID-SUFFIX
EIDR-SERVICE-ID-PREFIX         = "10.5239"
EIDR-SERVICE-ID-SUFFIX         = 4*4HEXDIG "-" 4*4HEXDIG
```

The syntax 4HEXDIG is defined in Section 5.

Canonical EIDR Service Identifiers shall be case insensitive.

7 Canonical DOI Name Type

The Canonical DOI Name Type registry entry, as defined in SMPTE ST 2003, shall be as specified in Table 1.

An instance of the Canonical DOI Name Type shall be a Canonical DOI Name as specified in Section 4.

Table 1 – DOI Name Type Definition

Name	Canonical DOI Name Type
Symbol	CanonicalDOINameType
UL	`urn:smpte:ul:060e2b34.01040101.01200700.00000000`
Definition	DOI name as specified in Section 4.1 of ISO/IEC 26324
Type Kind	Rename
Type Size	[variable]
Base Type	UTF16String
Type Qualifiers	[n/a]
Facets	[n/a]
Defining Document	SMPTE RP 2079
Context Scope	AbstractContext

8 Canonical EIDR Identifier Type

The Canonical EIDR Identifier Type registry entry, as defined in SMPTE ST 2003, shall be as specified in Table 2.

An instance of the Canonical EIDR Identifier Type shall be a Canonical EIDR Content Identifier as specified in Section 5.

Note: The CanonicalEIDRIdentifierType was introduced by earlier versions of this specification at a time when only EIDR Content Identifiers were used, and were generally called "EIDR Identifiers". As a result, the name and symbol CanonicalEIDRIdentifierType do not reflect the fact that the type refers to EIDR Content Identifiers exclusively. The name and symbol of CanonicalEIDRIdentifierType are preserved in order to maintain compatibility with existing implementations. Other types, e.g. CanonicalEIDRServiceIdentifierType are used to refer to other kinds of EIDR Identifiers.

Table 2 – Canonical EIDR Identifier Type Definition

Name	Canonical EIDR Identifier Type
Symbol	CanonicalEIDRIdentifierType
UL	`urn:smpte:ul:060e2b34.01040101.01200800.00000000`
Definition	Canonical EIDR Content Identifier NOTE: This type refers to EIDR Content Identifiers exclusively. Other types, e.g. CanonicalEIDRServiceIdentifierType, are used to refer to other kinds of EIDR Identifiers.
Type Kind	Rename
Type Size	[variable]
Base Type	CanonicalDOINameType
Type Qualifiers	[n/a]
Facets	[n/a]
Defining Document	SMPTE RP 2079
Context Scope	AbstractContext

9 Canonical EIDR Service Identifier Type

The Canonical EIDR Service Identifier Type registry entry, as defined in SMPTE ST 2003, shall be as specified in Table 2.

An instance of the Canonical EIDR Service Identifier Type shall be a Canonical EIDR Service Identifier as specified in Section 5.

Table 3 – Canonical EIDR Service Identifier Type Definition

Name	Canonical EIDR Service Identifier Type
Symbol	CanonicalEIDRServiceIdentifierType
UL	`urn:smpte:ul:060e2b34.01040101.01200b00.00000000`
Definition	Canonical EIDR Service Identifier
Type Kind	Rename
Type Size	[variable]
Base Type	CanonicalDOINameType
Type Qualifiers	[n/a]
Facets	[n/a]
Defining Document	SMPTE RP 2079
Context Scope	AbstractContext

10 Deprecated Elements and Types (informative)

The Canonical EIDR Identifier and Canonical DOI Name elements specified in earlier revisions of this specification are deprecated, since elements cannot generally be reused across groups. Instead, applications are encouraged to define their own elements using the types defined herein.

11 URI Representations

11.1 Canonical DOI Name

ISO/IEC 26324 specifies multiple possible URI representations for Canonical DOI Names.

11.2 Canonical EIDR Content Identifier and Canonical EIDR Service Identifier

The URI representation of a Canonical EIDR Content Identifier or a Canonical EIDR Service Identifier should conform to IETF RFC 7972, which specifies the URN representation for EIDR Identifiers.

Example: According to IETF RFC 7972, the URI representation of the Canonical EIDR Content Identifier `10.5240/5FD4-FEE1-22F5-583E-FECC-O` is `urn:eidr:10.5240:5FD4-FEE1-22F5-583E-FECC-O`.

Note: A URN representation is also a URI representation since all URNs are also valid URIs.

12 Binary Representations

12.1 Canonical EIDR Content Identifier

12.1.1 Full Binary Representation

The full binary representation of a Canonical EIDR Content Identifier shall be a 152-bit binary integer constructed as follows:

- the most significant 64 bits shall be equal to the integer 0x31302e353234302f (US-ASCII encoding of the string "10.5240/");
- the next 20 4-bit nibbles in order of decreasing significance shall each be equal to the integer representation of a single hexadecimal digit `HEXDIG` of `EIDR-CONTENT-ID-SUFFIX`, starting from the leftmost one; and
- the least significant 8 bits shall be an integer equal to the ASCII-encoded value of either the lowercase or uppercase representation of the `CHECK` character.

Note: This specification does not specify the byte ordering of the full binary representation, which is left to each application.

Example: The full binary representation of the Canonical EIDR Content Identifier `10.5240/5FD4-FEE1-22F5-583E-FECC-O` is `0x31302E353234302F5FD4FEE122F5583EFECC4F`

12.1.2 Compact Binary Representation

The compact binary representation of a Canonical EIDR Content Identifier shall be a 96-bit binary integer constructed as follows:

- the most significant 16 bits shall be equal to the integer 0x1478 (5240 in decimal); and

- the final 20 4-bit nibbles in order of decreasing significance shall each be equal to the integer representation of a single hexadecimal digit HEXDIG of EIDR-CONTENT-ID-SUFFIX, starting from the leftmost one.

Note 1: When generating a Canonical EIDR Content Identifier from a compact EIDR Content Identifier, the CHECK character will need to be computed.

Note 2: This specification does not specify the byte ordering of the compact binary representation, which is left to each application.

Example: The compact binary representation of the Canonical EIDR Content Identifier 10.5240/5FD4-FEE1-22F5-583E-FECC-O is 0x14785FD4FEE122F5583EFECC

12.2 Compact Binary Representation of Canonical EIDR Service Identifier

The compact binary representation of a Canonical EIDR Service Identifier shall be a 96-bit binary integer constructed as follows:

- the most significant 16 bits shall be equal to the integer 0x1477 (5239 in decimal);
- the next 8 4-bit nibbles in order of decreasing significance shall each be equal to the integer representation of a single hexadecimal digit HEXDIG of EIDR-SERVICE-ID-SUFFIX, starting from the leftmost one; and
- the last 12 4-bit nibbles shall be equal to 0x0.

Note: This specification does not specify the byte ordering of this representation, which is left to each application.

Example: The compact binary representation of the Canonical EIDR Service Identifier 10.5239/250C-5152 is 0x1477250C5152000000000000.

Bibliography (Informative)

Entertainment Identifier Registry (5 October 2017). EIDR: ID Format Ver. 1.51 (http://eidr.org/documents/EIDR_ID_Format.pdf)