SMPTE STANDARD

Unique Digital Media Identifier (C4 ID)

Table of Contents

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 Standards Operations Manual. This SMPTE Engineering Document was prepared by Technology Committee 30MR.

Approved
July 21, 2017

Intellectual Property

At the time of publication, no notice had been received by SMPTE claiming patent rights essential to the implementation of this Engineering Document. 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

C4 IDs are unambiguous, universally unique and consistent identifiers for files or units of data. Two organizations will compute exactly the same C4 IDs for identical data. C4 IDs have no need for a central registry or other external information.

C4 IDs are derived directly from the digital material they identify, so, unlike assigned IDs that are linked to the identified object by an external agency, C4 IDs are perpetually and unambiguously linked to the identified object and can always be regenerated with absolute certainty at any time.

This standard describes the C4 Base58 character set, the format and construction of the C4 ID for a single contiguous unit of data (e.g. a file) and method of construction of a C4 ID for non-contiguous units of data (e.g. some or all of the files in a folder or all or some of the data in a package).

C4 IDs for non-contiguous units of data are especially useful when archiving or transferring data because data verification can be done folder-by-folder instead of file-by-file. A single change of one bit in a sub-folder many levels below a folder having a C4 ID of this type would cause a change in the value of the highest level C4 ID. So if the C4 ID is the same for a folder in one place as a folder in another place, it indicates the data in the folder and all of its children have the same data.

At the highest level, C4 IDs can also assure all of the applications and other files beyond film or video data match.

C4 ID can be used to reduce the transmission costs of data across a network. It can do this in at least three important ways:

1. By keeping C4 ID values for all relevant files on remote and local systems, re-transmission of duplicate files can be avoided.

2. By confirming C4 ID values after transmission, systems can detect transmission errors.

3. By computing the C4 ID for a file immediately upon creation, sites can compare the results of deterministic processes without transferring the resulting files. For example, unpacking an archive file (or restoring a backup) can be validated by comparing C4 ID values of the resulting files with those computed at the time the archive was created.

NB: C4 is an abbreviation of Cinema Content Creation Cloud, reflecting its origin in cinema creation. The application of C4 ID is generic to all digital media files and is not constrained to cinema files. Because it is self-identifying, and for compatibility with already existing implementations, the prefix "c4" is maintained in the standard formulation, as is the "C4 ID" nickname in reference to that formulation.

1 Scope

This standard describes the C4 Base58 character set, the format and construction of the C4 ID for a single contiguous unit of data (e.g. a file) and method of construction of a C4 ID for non-contiguous units of data (e.g. collection of files or blocks in a stream).

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.

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; then formal languages; then figures; and then any other language forms.

3 Normative References

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

ISO/IEC 8859-1:1998, Information Technology – 8-Bit Single-Byte Coded Graphic Character Sets – Part 1: Latin Alphabet No. 1

ISO/IEC 10118-3:2004 Information technology – Security techniques – Hash-functions – Part 3: Dedicated hash-functions

4 Terms and Definitions

For the purposes of this document, the following terms and definitions apply.

4.1 C4 ID

unique digital media identifier defined by this standard

4.2 Secure Hash Algorithm

SHA

Secure Hash Algorithm. As in SHA-2, SHA-512 and other algorithms.

5 C4 ID Specifications

5.1 C4 Base58

C4 Base58 shall be a positional numeral system with a radix (base) of 58 and shall represent integer values 0 through 57, inclusive, using the case-sensitive characters "123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz" in that order from the Latin set 1 as defined by ISO/IEC 8859-1. (These are the characters for the numerals 1-9, followed by the upper-case letters A-Z, followed by the lower-case letters a-z, excluding the characters “0” zero, “O” upper-case oh, “I” upper-case, and “l” lower-case el.)

Note: The above encoding can be matched with the regular expression as described in ISO/IEC/IEEE 9945:2009:

```
[1-9A-HJ-NP-Za-km-z]
```

5.2 C4 Digest

A C4 Digest shall be a 64-byte SHA-512 message digest of the digital data unit, as specified in ISO/IEC 10118-3, and shall be interpreted as an unsigned big-endian integer.

5.3 C4 ID

The C4 ID shall be a 90-character string of alphanumeric characters from the Latin set 1 as defined by ISO/IEC 8859-1, and shall consist of:

A. C4 Prefix: The 2-character string “c4” (the lower-case letter “c” followed by the numeral “4”).

B. C4 Suffix: An 88-character string representing the unique message digest of the identified digital data unit. The C4 Suffix shall be calculated as follows:

1. Encode the 64-byte C4 Digest as a C4 Base58 integer having 88 digits.

Annex A provides an example of pseudocode for computing a C4 ID from a C4 Digest and the inverse.

Note: The above encoding can be matched with the regular expression:

```
c4[1-9A-HJ-NP-Za-km-z]{88}
```

Examples: Sample C4 IDs and corresponding decimals:

"c411111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111"

0

"c411111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111BukQL"

123456789

5.4 Calculating a C4 ID for Non-contiguous Blocks of Data

For non-contiguous blocks of data, such as a set of files, a single C4 ID shall be derived as follows:

A. Create a C4 Digest List from a C4 Digest for each block of data or file.

B. Sort the C4 Digest List in ascending order. Remove duplicates.

C. Repeat until the C4 Digest List contains only one C4 Digest:

   a. If the number of C4 Digests in the C4 Digest List is odd, remove the last C4 Digest from the list and retain it separately.

   b. Pair adjacent C4 Digests in the C4 Digest List.

   c. Create a new C4 Digest List by, for each pair, computing a new C4 Digest as follows:

      i. Order the C4 Digests in the pair in ascending order;

      ii. Concatenate the C4 Digests together into a single 128-byte block;

      iii. Calculate a C4 Digest for the 128-byte block.

   d. Append the C4 Digest retained in step a, if any, to the new C4 Digest List.

D. Encode the remaining C4 Digest as a C4 ID. This shall be the C4 ID representing the non-contiguous blocks of data.

Annex B includes a graph showing the resulting tree of digests.

Note: When this method is applied to a single block of data, the method returns the C4 ID of that single block of data.

Annex A Pseudocode to Compute a C4 ID and a C4 Digest (informative)

A.1 Compute C4 ID from C4 Digest

```
SET c4digest to SHA512(input-bytes)

SET i to 0
SET intdigest to 0

COMMENT: Convert the digest to an integer.
LOOP WHILE i < 64
    intdigest = 256  * intdigest + c4digest[i]
    i = i + 1
END LOOP

SET c4Base58 to "123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz"
SET base to 58
SET result to ""

COMMENT: Convert the integer to a C4 Base58 string.
LOOP WHILE LENGTH(result) < 88
  digit = intdigest MODULO base
  intdigest = intdigest DIVIDE base
  result = c4Base58[digit] + result
END LOOP

result = "c4" + result
```

A.2 Compute C4 Digest from C4 ID

```
SET c4id to the input C4 ID
SET c4Base58 to "123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz"
SET base to 58
SET result to 0
SET i to 2

COMMENT: Convert a C4 Base58 string to an integer.
LOOP WHILE i < 90
  temp = INDEX OF c4id [i] IN c4Base58
  result = result * base + temp
  i = i + 1
END LOOP

COMMENT: Convert the integer to a digest.
SET i to 63

LOOP WHILE i >= 0
  c4digest[i]= result MODULO 256
  result = result DIVIDE 256
  i = i - 1
END LOOP
```

Annex B Diagrams of C4 IDs for Non-contiguous Blocks of Data (Informative)

This section provides an example calculation of the C4 ID for noncontiguous blocks of data.

Data blocks, formatted as text strings:

```
A: "alfa"
B: "bravo"
C: "charlie"
D: "delta"
E: "echo"
F: "foxtrot"
G: "golf"
H: "hotel"
I: "india"
```

C4 IDs for the data blocks:

```
id A: c43zYcLni5LF9rR4Lg4B8h3Jp8SBwjcnyyeh4bc6gTPHndKuKdjUWx1kJPYhZxYt3zV6tQXpDs2shPsPYjgG81wZM1
id B: c42jd8KUQG9DKppN1qt5aWS3PAmdPmNutXyVTb8H123FcuU3shPxpUXsVdcouSALZ4PaDvMYzQSMYCWkb6rop9zhDa
id C: c44erLietE8C1iKmQ3y4ENqA9g82Exdkoxox3KEHops2ux5MTsuMjfbFRvUPsPdi9Pxc3C2MRvLxWT8eFw5XKbRQGw
id D: c42Sv2Wi2Qo8AKbJKnUP6YTSdz8pt9aDaf2Ltx44HF1UDdXANM8Ltk6qEzpncvmVbw6FZxgBumw9Eo2jtGyaQ5gDSC
id E: c41bviGCyTM2stoMYVTVKgBkfC6SitoLRFinp77BcmN9awdaeC9cxPy4zyFQBhmTvRzChawbECK1KBRnw3KnagA5be
id F: c427CsZdfUAHyQBS3hxDFrL9NqgKeRuKkuSkxuYTm26XG7AKAWCjViDuMhHaMmQBkvuHnsxojetbQU1DdxHjzyQw8r
id G: c41yLiwAPdsjiBAAw8AFwQGG3cAWnNbDio21NtHE8yD1Fh5irRE4FsccZvm1WdJ4FNHtR1kt5kev7wERsgYomaQbfs
id H: c44nNyaFuVbt5MCfo2PYWHpwMkBpYTbt14C6TuoLCYH5RLvAFLngER3nqHfXC2GuttcoDxGBi3pY1j3pUF2W3rZD8N
id I: c41nJ6CvPN7m7UkUA3oS2yjXYNSZ7WayxEQXWPae6wFkWwW8WChQWTu61bSeuCERu78BDK1LUEny1qHZnye3oU7DtY
```

Sorted C4 IDs:

```
1: c41bviGCyTM2stoMYVTVKgBkfC6SitoLRFinp77BcmN9awdaeC9cxPy4zyFQBhmTvRzChawbECK1KBRnw3KnagA5be
2: c41nJ6CvPN7m7UkUA3oS2yjXYNSZ7WayxEQXWPae6wFkWwW8WChQWTu61bSeuCERu78BDK1LUEny1qHZnye3oU7DtY
3: c41yLiwAPdsjiBAAw8AFwQGG3cAWnNbDio21NtHE8yD1Fh5irRE4FsccZvm1WdJ4FNHtR1kt5kev7wERsgYomaQbfs
4: c427CsZdfUAHyQBS3hxDFrL9NqgKeRuKkuSkxuYTm26XG7AKAWCjViDuMhHaMmQBkvuHnsxojetbQU1DdxHjzyQw8r
5: c42Sv2Wi2Qo8AKbJKnUP6YTSdz8pt9aDaf2Ltx44HF1UDdXANM8Ltk6qEzpncvmVbw6FZxgBumw9Eo2jtGyaQ5gDSC
6: c42jd8KUQG9DKppN1qt5aWS3PAmdPmNutXyVTb8H123FcuU3shPxpUXsVdcouSALZ4PaDvMYzQSMYCWkb6rop9zhDa
7: c43zYcLni5LF9rR4Lg4B8h3Jp8SBwjcnyyeh4bc6gTPHndKuKdjUWx1kJPYhZxYt3zV6tQXpDs2shPsPYjgG81wZM1
8: c44erLietE8C1iKmQ3y4ENqA9g82Exdkoxox3KEHops2ux5MTsuMjfbFRvUPsPdi9Pxc3C2MRvLxWT8eFw5XKbRQGw
9: c44nNyaFuVbt5MCfo2PYWHpwMkBpYTbt14C6TuoLCYH5RLvAFLngER3nqHfXC2GuttcoDxGBi3pY1j3pUF2W3rZD8N
```

After pass 1:

```
10: c42zjM4ARWVNHVkHsaiEWMAxzngUk8op167Dsm1iNpGfxdQBmhwjhWshKRqacPQw3MKwj7kAVxqBwSxADRDKQFAbtu
11: c45y4hGsfLRcoDpccf7vh8oaEvuFV5UePJmoXWg2W8fr2EqPHLxucBJMmPSXN1wv45okRdjEXkbZn1KzapPwUhYhgz
12: c41DGFq9sEb7jVmfsvPWnB8R8nENZp1xfoMbS5kK8TkCDpCT28A3wXsAbj8L5ojNLJrENh4UPmrqBCqJvRtG3oeavt
13: c453g2FnSZnHyUsM95Hs63wVTLmaJLgcB6HULNY7G6xeKggUPsdtN39e9C2qzkoMWKB9gWHVX6aigy1uSzAvyVoS7R
14: c44nNyaFuVbt5MCfo2PYWHpwMkBpYTbt14C6TuoLCYH5RLvAFLngER3nqHfXC2GuttcoDxGBi3pY1j3pUF2W3rZD8N
```

After pass 2:

```
15: c42WxVx7sogq4LSuxxbzzytXztB3GMwiqfsEPyghJnR5QYVoJ7rVu2yDTpzKTS63eEn2bH4ouhkb1CUTqNfu8RepgB
16: c45b6ZA4eu1PoCmeYXncTNGAD47sqJPoN1kMgSBsFgXQB9pwRr6u8a6hDWsBbB5x78ZENb5GsnmGejDcCo7aZ4SAsz
17: c44nNyaFuVbt5MCfo2PYWHpwMkBpYTbt14C6TuoLCYH5RLvAFLngER3nqHfXC2GuttcoDxGBi3pY1j3pUF2W3rZD8N
```

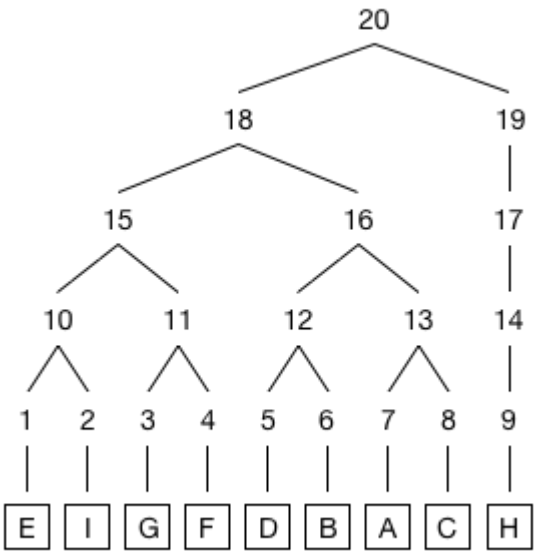
After pass 3:

```
18: c449rzjCF2bwgbWkHLWRRNNQsjxMu36ee6hU3gMr3PxX8zSPpwWZkYp27zgtgFpuBCajMtfYA6PzSmGpRJYLT6pqa5
19: c44nNyaFuVbt5MCfo2PYWHpwMkBpYTbt14C6TuoLCYH5RLvAFLngER3nqHfXC2GuttcoDxGBi3pY1j3pUF2W3rZD8N
```

After pass 4, the final C4 ID

```
20: c435RzTWWsjWD1Fi7dxS3idJ7vFgPVR96oE95RfDDT5ue7hRSPENePDjPDJdnV46g7emDzWK8LzJUjGESMG5qzuXqq
```

The combining of IDs above to produce the final ID can be visualized as follows:



Bibliography (informative)

ISO/IEC/IEEE 9945:2009 — Information technology -- Portable Operating System Interface (POSIX®) Base Specifications, Issue 7

The C4 Identification System, Universally Consistent Identification Without Communication (whitepaper) by Joshua Kolden joshua@studiopyxis.com ETC@USC