EssenceMark^{TM}

- SMPTE Standard based Textual Video Marker -

Original edition[1]

Abstract

EssenceMark is a simple textual video marker that may be associated with a particular point of interest (POI) in audiovisual essence. This allows a user to quickly access the particular point as desired. EssenceMark is designed based on one of the SMPTE (Society of Motion Picture and Television Engineers) standard-based KLV (Key-Length-Value) metadata items, and thus it can be handled by a wide range of applications and devices that support the relevant standards. This paper explains the EssenceMark and its applications.

Table of Contents

[1] This is original text of an article published in SMPTE Motion Imaging Journal [1]

1 Introduction

Metadata has been paid strong attention over a decade in the broadcast and professional industry. In earlier days, the introduction of time code has drastically changed the video editing environment, i.e., compared with the old method where video tape was physically cut and spliced, a semiautomatic video editing with a frame level accuracy has been achieved by electrically specifying the in- and out- points of the frame without ambiguity.

The term “metadata”, which is often defined as “data about audiovisual (AV) essence” in this context, is still vague: If one specific application relates to AV essence, then metadata specifically designed for the application should exist. In other words, as many metadata as existing applications would exist. Among these kinds of metadata, POI (Point Of Interest) metadata has been recognized as one of the most fundamental, and therefore most applications consider it because of the large data size in the AV essence, especially for broadcast and professional uses, where access to a desired point of AV essence has been always cumbersome.

In spite of the importance of POI metadata and the fact that it is commonly desired and widely handled by many applications, a major problem often exists in its reusability. For instance, if one application extracts the POI metadata, there is generally no way for the subsequent application to efficiently inherit the POI information. It is evident that the more valuable the metadata is, the more cost its extraction requires, such as POI information that can be extracted only through interaction with and/or judgment by a human being. Therefore, it is urgent that a scheme for the POI metadata interoperability be established for a wide range of applications and devices.

EssenceMark has been developed to address this urgent demand: EssenceMark by itself is just a simple video (AV essence) marker with arbitrary text information attached. However, because it is based on the well-known SMPTE standards [2] and is fully conformed to them, it may be applied to any kind of application or device that supports the relevant standards.

In this paper, EssenceMark is introduced for the first time, not only its generic explanation, but also its technical details. A reader is expected to understand not only what EssenceMark is, but also to grasp the initial idea of how it is implemented based on the standards.

2 What is EssenceMark?

2.1 Introduction

This section describes the generic concept of EssenceMark by using Post-it® for bookmark as a metaphor[2].

2.2 Post-it® for Book

Suppose you have a thick book or a document with many pages. When you initially go through it, you would often *mark* the pages that you want to review later. For that purpose, Post-it is a useful tool. Thanks to Post-it having various colors, you may often classify the marks at the initial reading, based on why you feel it should be reviewed later.

This is a good practice you might often experience. But with only a limited number of colored Post-its, it would often happen that they are not available enough for you to classify as many as you want. Furthermore, if another person needs to review your results, he or she might not know the specific classifications for each. For instance, why was a “blue” Post-it chosen for a particular page marking?

To overcome this limitation, additional textual notes are often written on the Post-it as is shown in Figure 1.

[2] Post-it® is the Registered Trade Mark of 3M Corporation.

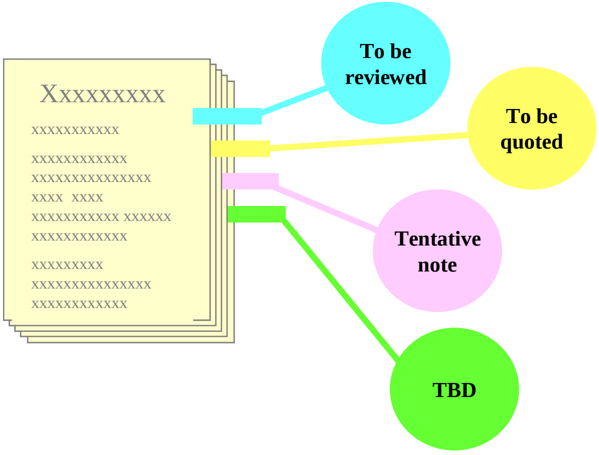


Figure 1: Post-it® for Book

This practice would be workable even though only one kind of Post-it is available at the initial time of reading. In addition, because of the freedom to express anything in text (and in a readable fashion), the subsequent person may obtain more information than just using the colored Post-it as a simple bookmark.

2.3 EssenceMark for Audiovisual Essences

The basic concept of EssenceMark is "Post-it (with textual note) for AV essence". Suppose you have acquired AV clips of a baseball game scene for "Ichiro Suzuki at Seattle Mariners". Figure 2 schematically demonstrates how EssenceMark works for the example. The textual notes, which may be created during acquisition and/or the AV clip review, are directly buried electrically within the AV clips for later use.

It should be noted that, in addition to POI information created by a human being, any marks to the AV essence, which would be automatically created by applications or devices, may be handled in the same way as for POI information, once how to mark in the textual fashion is defined in advance. Figure 2 also shows an example of EssenceMark that is automatically inserted at each recording start by a camcorder, which is textually expressed as "REC Start".

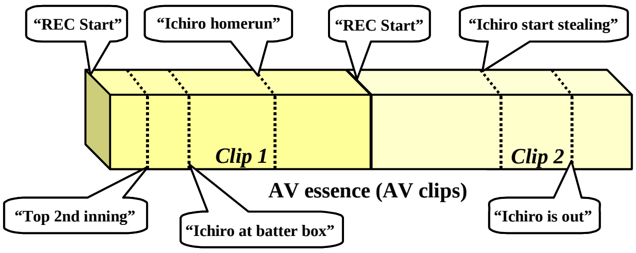


Figure 2: EssenceMark for AV essence

In general, EssenceMark is always handled together with the target AV essence. It is recorded and transferred altogether, regardless of type of recording media or the interface. To achieve interoperability or to allow the existing EssenceMark to be reused in subsequent applications and devices, the importance is placed on the common interface specifications to be implemented by the applications or devices. To achieve this, EssenceMark has been developed as an application of well-known SMPTE KLV metadata.

3 EssenceMark Implementations

3.1 Introduction

This section describes the technical aspect of EssenceMark. Because EssenceMark is an application of the SMPTE KLV metadata, the overview of relevant SMPTE standards is initially introduced. Then, what makes EssenceMark from the technical viewpoint is discussed with its recording, transfer, and additional features that may enhance its functionality.

3.2 SMPTE Key-Length-Value (KLV) coding

The Key-Length-Value (KLV) coding, which constitutes the basis of SMPTE metadata and related technology, is a simple structural encoding rule at any object instantiation (referred to as *KLV packet*, hereafter) [3]. Figure 3 shows a generic structure of the KLV packet.

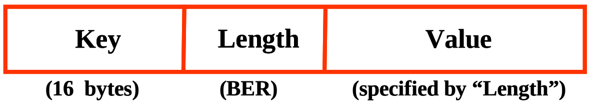


Figure 3: Structure of KLV packet

The KLV packet is composed of three components: “Key”, “Length” and “Value”. A KLV packet begins with a unique “Key” of 16-byte length (often called SMPTE Universal Label (UL)) to identify the packet, which shall start with “06_h $0E_h$ $2B_h$ 34_h …” indicating that this “Key” belongs to SMPTE.

Following the “Key” is the “Length” field to indicate the byte size of the subsequent “Value” field that accommodates actual data.

For those interested in technical details of the KLV coding, more rigorous definitions of “Key”, “Length” and “Value” are provided in Annex A.

3.3 SMPTE Metadata Dictionary

To uniquely assign the 16-byte UL Key to individual metadata items, SMPTE has been developing the SMPTE Metadata Dictionary [4]. Figure 4 shows a part of the Metadata Dictionary.

As is seen in Figure 4, SMPTE Metadata Dictionary is a kind of metadata item lookup table. Each row in the table is allocated to either one particular metadata item or a class (collection of metadata items with common characteristics or attributes) of metadata items, which is composed of the assigned 16-byte UL key, the metadata item name, its definition, its datatype, size when instantiated as “Value”, reference document, and note.

An example of metadata item registered to the Dictionary is shown below:

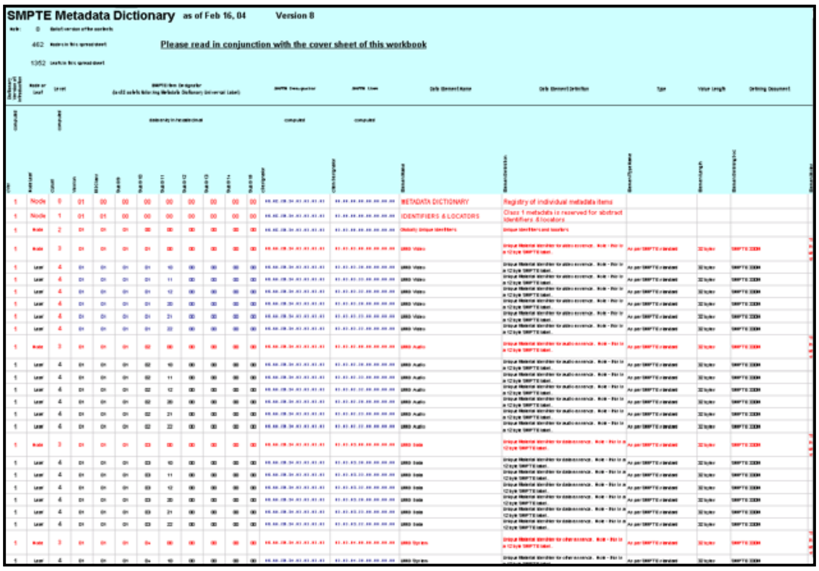
16 byte UL Key: 06_h $0E_h$ $2B_h$ 34_h 01_h 01_h 01_h 01_h 01_h 05_h 02_h 00_h 00_h 00_h 00_h 00_h

Name: “Main title”

Definition: The main title

Type: ISO/IEC 646:1991 - ISO 7-Bit Coded Character Set

Size: 127 bytes max



SMPTE Metadata Dictionary as of Feb 16, 04 Version 8

462

1352

Please read in conjunction with the cover sheet of this workbook

Figure 4: SMPTE Metadata Dictionary

At the time of this writing, more than 1800 metadata items are registered in the Dictionary. It should be noted that the Metadata Dictionary is still evolving with new metadata items being identified according to finding new requirements, applications, and so on. Based on the evolution rule that only addition of metadata items or classes is permitted, however, the backward compatibility with its previous versions is always guaranteed.

The technical details of the Metadata Dictionary such as the dictionary architecture, its design policy and relevant standards are described in Annex B for those interested in them.

3.4 What makes it EssenceMark?

EssenceMark is an application of SMPTE KLV metadata. To accommodate a short textual note as a marker of the AV essence, a metadata item named “Term Value”, which is defined as “*the value of the parameter as a string*”, is used. According to the SMPTE Metadata Dictionary [4], the 16-byte UL Key for “Term Value” and thus EssenceMark is specified as:

$$\texttt{06}_h\ \texttt{0E}_h\ \texttt{2B}_h\ \texttt{34}_h\ \texttt{01}_h\ \texttt{01}_h\ \texttt{01}_h\ \texttt{05}_h\quad \texttt{03}_h\ \texttt{01}_h\ \texttt{02}_h\ \texttt{0A}_h\ \texttt{02}_h\ \texttt{00}_h\ \texttt{00}_h\ \texttt{00}_h$$

for its datatype as ASCII (ISO/IEC 646:1991 - ISO 7-Bit Coded Character Set), and

$$\texttt{06}_h\ \texttt{0E}_h\ \texttt{2B}_h\ \texttt{34}_h\ \texttt{01}_h\ \texttt{01}_h\ \texttt{01}_h\ \texttt{05}_h\quad \texttt{03}_h\ \texttt{01}_h\ \texttt{02}_h\ \texttt{0A}_h\ \texttt{02}_h\ \underline{\mathbf{01}_h}\ \texttt{00}_h\ \texttt{00}_h$$

for Unicode (UTF-16 Unicode string).

In addition, to obtain a better interoperability for a wide range of devices and applications, the following constraints are further applied to the “Term Value” as EssenceMark:

- The size of “Value” field in the KLV packet is limited within 32-byte long. This means up to 32 characters may be accommodated for the EssenceMark of ASCII datatype, and 16 characters for the Unicode,
- In the case of EssenceMark of Unicode, the character code shall be fixed to 2-byte big endian without BOM (Byte Order Mark).

Because EssenceMark is a KLV metadata, the format is the same as shown in Figure 3. For example, if EssenceMark is encoded with its value “REC Start”, it is represented as a byte string shown in Figure 5.

Key (Term Value)	Length (9 byte)	Value (“REC Start”)
$06_h0E_h2B_h34_h01_h01_h01_h05_h03_h01_h02_h0A_h02_h00_h00_h00_h$	09_h	$52_h45_h43_h20_h53_h74_h61_h72_h74_h$

Figure 5: Example of EssenceMark

It should be noted that although 1-byte length is encouraged for the “Length” field according to SMPTE 336M [3], EssenceMark by itself specifies no such constraint, i.e, the “Length” having “83_h 00_h 00_h 09_h” is permitted for the EssenceMark having 9-byte ASCII characters when desired.

3.5 EssenceMark Implementations

3.5.1 EssenceMark recording

An EssenceMark may be recorded along with its target frame (picture) data (together with associated audio data), or separately recorded in the EssenceMark table fashion, in which each EssenceMark may be associated with a particular point of AV essence via, for example, its time code.

In the case of VTR, the EssenceMark information may be recorded at a placeholder associated with each frame, such as video auxiliary sync block in the D-10 tape format [5]. This style of recording, which may be regarded as a direct implementation of the schematic shown in Figure 2, would benefit because the relation between the EssenceMark and its associated frame is unambiguous. Furthermore, because both the AV essence and its EssenceMarks are recorded on the same recording media, the chance of loosing the EssenceMark information is very little.

This style of EssenceMark recording, as described above, has obvious limitations, with only a sequential access permitted to reach a particular point of AV essence via EssenceMark. To directly access to the desired points, they should be recorded such that all EssenceMark information, including their pointers to the points of AV essence, are collected and recorded separately from the AV essence. By using XML (eXtensible Markup Language) [6], an example of such collected EssenceMarks would be described in a table fashion as shown in Figure 6.

However, note that in this case, the benefit of the aforementioned EssenceMark recording, such as unambiguous frame association, would decline. There is no best way for the EssenceMark recording; therefore, a decision should be made by carefully considering its applications and use cases. It should be also noted that the EssenceMark recording may be proprietary, as long as it is fully concealed from the standard interfaces. In other words, it is the transfer format that makes a device ready to participate in the open standard environment.

```
<EssenceMarkTable targetMedia="Clip 1">
   <EssenceMark value="REC Start" timecode="00:00:00:00"/>
   <EssenceMark value="Top 2nd inning" timecode="00:02:25:04"/>
   <EssenceMark value="Ichiro at batter box" timecode="00:05:12:13"/>
   <EssenceMark value="Ichiro homerun" timecode="00:11:23:05"/>
                  :
</EssenceMarkTable>
```

Figure 6: Example of EssenceMark Table[3]

3.5.2 EssenceMark transfer

Compared with EssenceMark recording, the EssenceMark transfer should be fully compliant with the relevant standards in order to achieve interoperability. Because EssenceMark is an application of the SMPTE KLV metadata, the standard transfer of the KLV metadata is also directly applied to the

[3] This is provided for an example purpose only.

EssenceMark transfer.

In the case of SDI (Serial Digital Interface), EssenceMark, as a KLV metadata item, is packed into the SMPTE 291M ancillary data packets [7]. Figure 7 shows the structure of the data packet (Type 2) for KLV metadata. Note that for the ancillary data packet to accommodate the KLV metadata, an additional standard [8] needs to apply, i.e., the User Data Words field of the data packet should start with MID (Message ID), PSC (Packet Sequence Count), and then the KLV packet for EssenceMark follows[4].

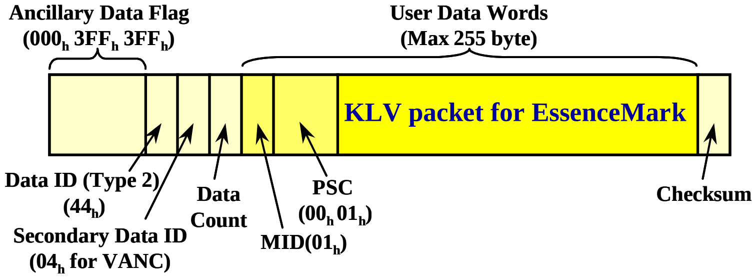


Figure 7: Ancillary data packet for KLV metadata

It should be noted that although the ancillary data packet may be located in either the horizontal blanking area of SDI (known as H-ANC) or the vertical blanking lines (known as V-ANC), the latter is made as a convention for the data packet of EssenceMark.

In the SDTI-CP (Serial Data Transfer Interface – Content Package [9]), the SMPTE standard [10] specifies that a KLV packet may be packed either in the System Item with the Type Value of 88_h or in the Auxiliary Item with the Type Value of 21_h (indicating the SMPTE 291M ancillary data packet). Because the KLV packet for EssenceMark is small, and not many commercially available products fully support the SDTI-CP Auxiliary Item, a conventional rule is made that the KLV packet for EsseceMark should be packed in the System Item.

Material Exchange Format (MXF) [11], the emergent technology in the coming IT-based production environment for the industry, has also specified, as a part of its document family, how to accommodate a KLV metadata in its file body (together with AV essence). While there are variations, some of which are still under standardization process at the time of this writing, a standard document defining the file body structure [12] indicates that the KLV packet for EssenceMark may be packed in the same way as is done for SDTI-CP.

So far, the discussion focuses on EssenceMark transfer, along with its associated frame. But there might be cases in which the EssenceMark Table, such as shown in Figure 6, should be transferred along with its target AV essence. Although it is hard to achieve in the conventional synchronous environment[5], it will be dealt with in the MXF-based asynchronous environment.

One method would be to simply embed a KLV packet containing such a table on the MXF file header. According to the MXF specifications [11], an arbitrary KLV packet is permitted to insert into the MXF file header as "dark metadata". It should be noted that although such a KLV packet insertion may be effective among those sharing the table scheme in advance in such a case as its local exchange, it generally prevents interoperability among a wide range of applications. Hence the use of standard metadata scheme needs to be carefully considered to achieve interoperability[6].

[4] Because the size of "Length" field is not fixed, the size of a KLV packet for EssenceMak might take 176 bytes (=16 bytes for UL Key + max. 128 byte for "Length" + max 32 byte for "Value") at its maximum, though the 1 byte length for "Length" field is strongly recommended.

[5] It would be theoretically possible when the EssenceMark Table is wrapped to form a KLV packet. But in reality most commercially available products cannot handle such a long KLV packet.

[6] The mapping of EssenceMark Table to DMS-1 [13], the default descriptive metadata scheme specified as a part of the MXF standard family, is under investigation.

3.6 EssenceMark Additional Features

3.6.1 Controlled terms for EssenceMark

The value of EssenceMark by nature is an arbitrary text string up to 32-byte long. However, it is often more convenient that the value is restricted to choose only from a predefined value list, which is called *controlled term*. An example of such a term would be “Good point”, for its EssenceMark to be inserted when one feels particular interest. Without such a term shared among relevant people in advance, one might insert “Cool point” EssenceMark, instead of the “Good point”, to the point of AV essence at which he or she feels so, but this cannot be retrieved by others because it is unknown to them.

In a practical situation, it is likely that plausible terms for the EssenceMark value be preregistered to a list for a given purpose and be chosen at its operational stage, rather than arbitrary text being inserted on the fly. For example, if EssenceMark is applied for a baseball game acquisition such as shown in Figure 2, it is expected that terms such as players’ name (e.g., “Ichiro”), typical action (e.g., “homerun”) and result (e.g., “out”) are registered to the list(s) in advance, and the realtime logging with EssenceMark is performed during the baseball game where appropriate terms are chosen from the list(s) as needed to create a particular EssenceMark value such as “Ichiro homerun”.

From a device and application systemization viewpoint, the EssenceMark controlled terms will provide another important feature, that is, to *share the semantics* of the terms among those involved in the system. For example, a camcorder, which can detect the occurrence of flashlight, will notify it by attaching the EssenceMark “Flash” to the frame that captures the flashlight. Another example is a VTR capable of shot cut detection during its playback, which will report the result by inserting the EssenceMark “Cut” at the detected points.

When the terms “Flash” and “Cut” are reserved for the EssenceMark value to represent respective semantics, then devices and applications receiving those EssenceMark values can recognize their semantics and behave accordingly. For example, an application might create a video summary composed only of video segments containing the flash light occurrence (See Section 4.3.2). Another application might disassemble a complete video program into video shots according to the “Cut” points for reusing purpose (See Section 4.3.3).

In Annex C, the controlled terms for EssenceMark, reserved as a part of the EssenceMark specifications in Sony Corporation is further discussed.

3.6.2 EssenceMark list name

As discussed in previous section, it is usually expected in practical situations that EssenceMark value is selected from a list that registers plausible values in advance, rather than arbitrary text being inserted on the fly. Such a list would have its own unique name so that it is identified without ambiguity. This is called *EssenceMark list name*. In many cases, the relationship between the EssenceMark list name and AV essences (to be) acquired would be obvious, i.e., the list name might be directed by the desk as a part of the field assignment; it might be obvious because of a shooting target, or it might be created as a part of the field work. However, there are cases in which the EssenceMark list name is desired to be also recorded on acquired AV essence, especially when the same EssenceMark values, which are registered to different EssenceMark lists, are used simultaneously at the acquisition. For example, in shooting a baseball game, if the game is “New York Mets vs. New York Yankees”, then the EssenceMark value “Matsui homerun” may be used. But because both baseball teams have a player named “Matsui” (known as “Little Matsui” at NY Mets and “Godzilla Matsui” at NY Yankees), it is ambiguous if only the EssenceMark value of “Matsui homerun” is applied.

To address this issue, *EssenceMarkListName* is additionally introduced as a supporting tool for EssenceMark, which is based on the metadata item, “Thesaurus Name”, in the SMPTE Metadata Dictionary [4], and is applied the same constraints as for the EssenceMark.

The 16-byte UL Key for “Thesaurus Name” and thus EssenceMarkListName is specified as:

$$06_h\ 0E_h\ 2B_h\ 34_h\ 01_h\ 01_h\ 01_h\ 01_h\quad 03_h\ 02_h\ 01_h\ 02_h\ 02_h\ 00_h\ 00_h\ 00_h$$

for its datatype as ASCII (ISO/IEC 646:1991 - ISO 7-Bit Coded Character Set), and

$06_h\ 0E_h\ 2B_h\ 34_h\ 01_h\ 01_h\ 01_h\ \underline{\mathbf{04_h}}\quad 03_h\ 02_h\ 01_h\ 02_h\ 02_h\ \underline{\mathbf{01_h}}\ 00_h\ 00_h$

for Unicode (UTF-16 Unicode string).

Figure 8 schematically demonstrates a possible use of EssenceMarkListName. In this example, the EssenceMark list name is declared at an initial frame from which all the attached EssenceMarks belong to the list. The result is that the EssenceMark value “Matsui homerun” on the left-hand side (in Clip 1) unambiguously indicates a homerun by “Little Matsui”, while that on the right-hand side (in Clip 2) by “Godzilla Matsui”.

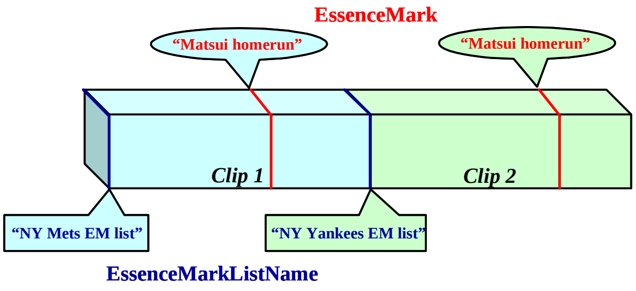


Figure 8: Use of EssenceMarkListName

It would be ideal if both EssenceMark and EssenceMarkListName are always associated with a frame, but in reality, few commercially available products support plural KLV metadata associated with a frame. Although the example in Figure 8 illustrates such a situation, this is not only the solution. In other words, how to use the EssenceMark list name is open from the EssenceMark specification viewpoint, and thus it is left up to EssenceMark users, depending on their system environment, operational convention, or application requirements.

4 EssenceMark Basic Applications

4.1 Introduction

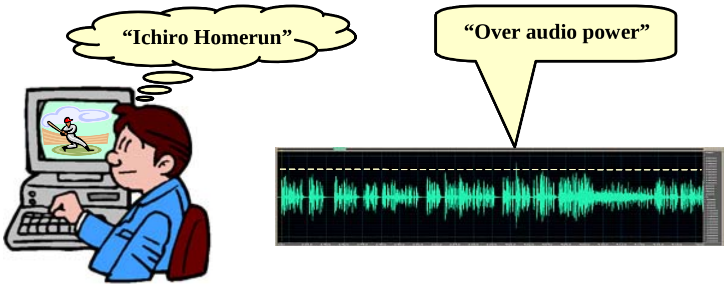
This section describes some basic treatments of EssenceMark in the EssenceMark applications. EssenceMark is fundamental metadata that is applicable to a wide range of applications; however, most may be regarded as a combination of some basic treatments of EssenceMark, which are discussed in this section.

4.2 EssenceMark Creation

Any EssenceMark creation method may be classified into one of two categories (Figure 9): (a) those created by human being (based on his or her judgment, or any other interactions) and (b) those automatically created by devices and/or applications (usually based on audiovisual signal processing).

Considering EssenceMark as POI metadata, the category (a) would be best understood, i.e., when one feels interested in a certain point of AV essence and judges worthwhile to be reviewed later, then he or she would insert an EssenceMark such as “Ichiro homerun” to the point.

While such “EssenceMarking” can usually be conducted at a logging stage of given AV essences, it could also be done even during the acquisition of AV essence, provided an appropriate supporting device or function such as a remote EssenceMark inserter (operated by those accompanied by a cameraperson) or a voice commander (by a cameraperson). Furthermore, the physiological measurement and analysis to characterizing human emotional and mental states discussed in the literature [14] could be a promising tool for issuing the EssenceMark without a person’s consciousness in the future.



(a) By human being

(b) By device/application

Figure 9: EssenceMark creation methods

The category (b) methods constitute another important aspect of the EssenceMark creation. In many signal processing-based audiovisual feature extractions, it often encounters a problem on how to represent the extracted results compactly. EssenceMark with predefined appropriate values may work. For example, when a device can detect a point at which a power of audio signal exceeds a given threshold value, it would insert EssenceMark "Over audio power" to the point, as shown in Figure 9 (b). Such a detection often captures unexpected audio events such as a bomb explosion or gunfire.

Another example would be a video-processing device that may detect a point from which the video frame drastically changes and inserts an EssenceMark such as "Frame change". Note that such a point often corresponds to the cut-point of an edited AV clip.

Furthermore, EssenceMark may be use for more elaborate feature extractions. A device capable of the face recognition would insert EssenceMark "Ichiro" to the point of an AV clip about a Seattle Mariners baseball game, when to detect the appearance of Ichiro Suzuki.

It should be noted that there are a couple of EssenceMark interpretations. In principle, EssenceMark associated with a frame just annotates the frame only, i.e., it is the POI information. However, in some applications, it may be interpreted as a starting point from which something happens to remain, such as the recording start point on tape or the cut-point of an edited AV clip.

4.3 EssenceMark Uses

4.3.1 POI Search

The most common use of EssenceMark is for search and access of the POIs of AV essence, according to the associated EssenceMarks. Because the value of EssenceMark is given as text data, any text matching algorithms may be applicable in identifying the desired points. In particular, when all the EssenceMarks and their reference points are summarized in a table fashion, as shown in Figure 6.

Figure 10 shows one possible GUI to represent the result of the POI search, as well as to provide the interface for the POI access.

In this example, all the frames associated with EssenceMark "ShotMark1" in all AV clips on a given recording medium are detected and shown as a thumbnail list of those frames (totally 31 detected frames). Then a person can access one of the POIs by selecting its thumbnail (No.6 in this example), e.g, a double click of the selected thumbnail would start playing back the AV clip (containing the selected frame) from the POI.

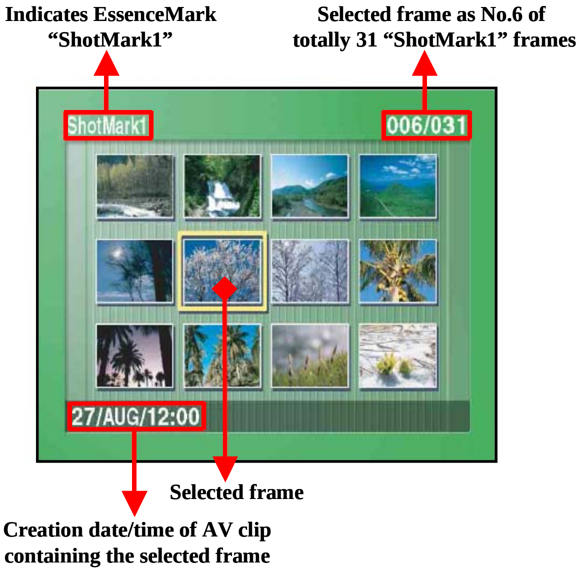

Indicates EssenceMark
"ShotMark1"
Selected frame as No.6 of
totally 31 "ShotMark1" frames
ShotMark1
006/031
27/AUG/12:00
Selected frame
Creation date/time of AV clip
containing the selected frame

Figure 10: GUI for thumbnail-based POI search

4.3.2 Video Summary

Video summary is often desired to quickly grasp the content of AV clip(s). Once AV clips are annotated with EssenceMark according to a given theme, then a video summary based on the theme would be easily created. Figure 11 below schematically demonstrates a video summary creation according to the flashlight occurrence for example.

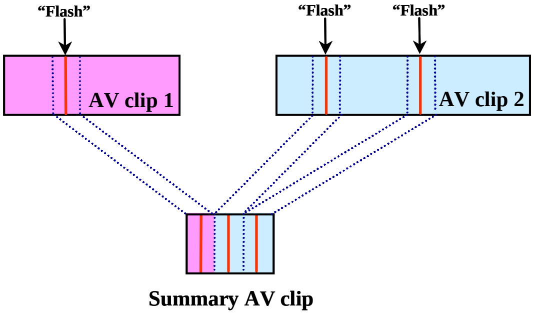


Figure 11: Flashlight occurrence-based video summary

In this example, two AV clips (AV clip1 and AV clip 2), which already contain the EssenceMark "Flash" at their flashlight occurrence points, are provided in advance. The video summary is then created by clipping video segments containing each "Flash" point (which is indicated by the EssenceMark) and serializing them as a single video sequence (Summary AV clip).

Note that because EssenceMark may take any text value, this scenario is applicable to any kind of video summary creation. Another example would be "Today's Ichiro" video summary in a baseball game when AV clips shown in Figure 2 are given and the video segments are extracted according to the EssenceMarks containing a term "Ichiro".

4.3.3 Video Decomposition

In the case of EssenceMark being interpreted as a starting point of something, it may be used as a point to decompose a given AV clip. An example is given by the cut-point detection and program decomposition.

For reuse of the legacy program (the complete package), it would be desirable to decompose a program into shots[7] by detecting the cut-points. Although various tools already exist to detect the cut-points, a common problem would be how to output this result for a subsequent application to efficiently use it. EssenceMark may address it, provided EssenceMark such as “Cut”, the cut-point detection tool can record the result by inserting the EssenceMark into those identified points, as shown on the left in Figure 12.

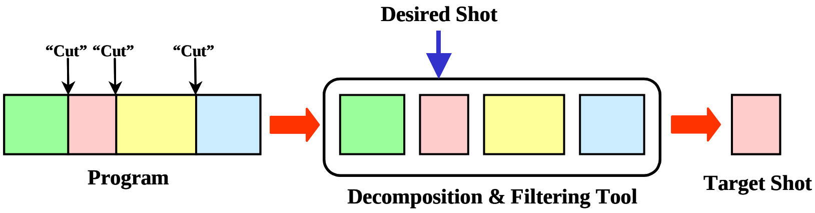


Figure 12: Cut-point based video decomposition and filtering

Once a program is annotated with the EssenceMark “Cut”, a subsequent application may identify each shot by detecting the cut-points with the EssenceMark “Cut”. In Figure 12, a conceptual tool to decompose a program into shots and to filter resulting shots according to its external control is schematically illustrated as a subsequent application example. It is worthwhile to note that the subsequent application is assumed to understand the EssenceMark value “Cut” and to behave accordingly. In other words, the value “Cut” is reserved among the cut-point detection and the subsequent decomposition/filtering tool. As a result, the former tool uses EssenceMark to control the latter, just like predefined commands to externally control it.

5 Discussions

EssenceMark is already in practice to improve the production workflow [15]. However, this does not mean that only EssenceMark is sufficient to achieve the workflow innovation. It should be noted that EssenceMark by itself is not self-completed from the metadata scheme viewpoint. Various standard metadata schemes already exist and such rich and self-completed metadata schemes should be introduced to construct a stable video archive systems.

On the other hand, it often happens that such metadata schemes are too cumbersome to handle just for marking the desired points of AV essence for the later use. A similar situation may be observed in a usual document survey. For example, in the survey, Post-it is often used to temporarily mark the desired pages of documents. Once the survey is completed, however, they are replaced with more established cataloging index to compile the results.

In the case of AV essence, EssenceMark is expected to be used mainly for the Post-it. Therefore, it is not intended to compete with existing metadata standards, but should be complementary, depending on the situation and the process in the innovated workflow.

6 Conclusions

EssenceMark is a simple textual video marker that may be associated with a particular point of interest (POI) in AV essence. Because of its simplicity and fundamentality, EssenceMark would be applicable to a wide range of applications and devices. Furthermore, because EssenceMark is based on the well-known SMPTE standards and is fully compliant with them, interoperability among various applications and devices is easily achieved. EssenceMark promises to be a common basic tool, like “Post-it for AV essence” in the broadcast and professional industry.

[7] Here, the term “shot” is used to represent an AV clip composed of a continuous group of frames only.

Annex A Definitions of “Key”, “Length” and “Value”

In this annex, more rigorous definitions of “Key”, “Length” and “Value” are provided. According to SMPTE 336M [3], those definitions are described in the following sections.

“Key” is a 16-byte Universal Label (UL) according to SMPTE 298M [16] to identify the data in the “Value” field. Each word (e.g. byte) in the SMPTE 298M UL is coded using the Basic Encoding Rules (BER) for Object Identifier coding specified in ISO/IEC 8825-1 [17].

The full UL Key shall consist of a 16-byte field including an Object ID (06_h) and the UL size ($0E_h$ indicating a total UL Key size of 16 bytes) followed by a series of sub-identifiers starting with the UL Code ($2B_h$) and SMPTE Designator (34_h).

The sub-identifiers shall define the UL Designator and Item Designators. The first two sub-identifiers in the UL Designator, which immediately follow the SMPTE Designator, shall have reserved values for the KLV Coding Protocol according to SMPTE 336M [3].

The value of each sub-identifier shall be limited to the range `0x01` to `0x7f`, which is represented by a single byte in the BER Object Identifier coding.

The sub-identifiers shall have left to right significance with the first sub-identifier as the most significant. The leftmost sub-identifier of value 00_h in the UL Key shall define the termination of the label and all sub-identifiers of lower significance shall also be set to 00_h. Sub-identifiers of value 00_h shall have no significance to the meaning of the UL Key.

Note that SMPTE 298M [16] defines only the first four bytes of a UL: the Object ID, UL Size, UL Code and SMPTE Designator. SMPTE 336M [3] specifies the application of SMPTE 298M ULs for the purpose of Key-Length-Value coding and defines the semantics of the remaining sub-identifiers of the UL Designator (bytes 5 to 8). The semantics of the Item Designator (bytes 9-16) are defined by other relevant standards, which together cover all the defined values of the UL Key.

“Length” specifies the length of subsequent “Value” field in byte. The value of the “Length” field shall be encoded using the Basic Encoding Rules (BER) for either the short form or long form encoding of length bytes specified in ISO/IEC 8825-1 [17]. This method of encoding the “Length” field is self-contained and allows for efficient parsing of KLV encoded data. For example, a decoder that does not recognize a “Key” is able to skip over the unknown “Value” and inspect the next Key.

The “Length” field is always coded MSB (most significant byte) first. If bit 7 of the first byte is a ‘0’ then the 7 least significant bits contains the length value (0 .. 127). If bit 7 of the first byte is a ‘1’ then the 7 least significant bits tell you the number of subsequent bytes in the length field. For example, the value ‘83_h’ means that the next 3 bytes contain the actual length value.

The examples below show a length value of 64 coded in the 3 different ways:

- 40_h : short form coded
- 83_h 00_h 00_h 40_h : long form coding using 4 bytes overall
- 87_h 00_h 00_h 00_h 00_h 00_h 00_h 40_h : long form coding using 8 bytes overall

“Value” is a value of actual data. Data values may be either an individual data item or a group of data items. In either case, the data is a byte string whose length is specified by the “Length” field value. The last byte of the “Value” field shall be the terminating byte of the data sequence.

Annex B Technical details of SMPTE Metadata Dictionary

This annex describes technical details of SMPTE Metadata Dictionary. According to SMPTE 298M [16], the SMPTE UL Key starts with two bytes UL Header, which shall be “06$_h$ 0E$_h$” immediately followed by 6-bytes UL Designator whose first two bytes shall be “2B$_h$ 34$_h$”. The initial 8 bytes of SMPTE UL Key for a metadata item, which is thus to be registered to the Metadata Dictionary, is specified by SMPTE 336M [3] as Table B-1 below:

Byte No.	Name	Description	Content/Format
	UL Header		
1	OID	Object Identifier	Always 06$_h$
2	UL Size	16-byte size of the UL	Always 0E$_h$
	UL Designator		
3	UL Code	Concatenated sub-identifiers ISO, ORG	Always 2B$_h$
4	SMPTE Designator	SMPTE sub-identifier	Always 34$_h$
5	Registry Category Designator	Registry Category designator identifying the category of registry described (e.g. Dictionaries)	01$_h$ for Dictionaries
6	Registry Designator	Registry Designator identifying the specific registry in a category (e.g. Metadata Dictionary)	01$_h$ for Metadata Dictionaries
7	Structure Designator	Designator of the structure variant within the given Registry (non-backwards compatible)	01$_h$ for SMPTE Metadata Dictionary (SMPTE RP 210 [4])
8	Version Number	Version of the given Registry which first defines the item specified by the Item Designator (backwards compatible)	Incrementing number

Table B-1: The UL Designator (byte 5-8) for SMPTE metadata item

This 8-byte sequence is then followed by another 8-byte, called Item Designator, with which the full 16-byte UK Key is formed.

The SMPTE Metadata Dictionary is a complete list of metadata items, each of which is uniquely identified by the last 8-byte Item Designator. The Item Designator defines a tree structure with a multiplicity of braches (called *nodes*) representing metadata classes, and the actual metadata items defined as leaves in the structure. It should be noted that only the items defined as leaves may be instantiated, i.e, the items defined as nodes cannot be instantiated but are introduced to the Dictionary in order to effectively classify a bunch of actual metadata items. Whether an item is defined as a node or a leaf is also specified in the Dictionary

As the topmost classes of the Dictionary, the following classes are specified by SMPTE 335M [18],

Class 1: IDENTIFIERS & LOCATORS

Class 2: ADMINISTRATION

Class 3: INTERPRETIVE

Class 4: PARAMETRIC

Class 5: PROCESS

Class 6: RELATIONAL

Class 7: SPATIO-TEMPORAL

Class 13: USER ORGANISATION REGISTERED FOR PUBLIC USE

Class 14: USER ORGANISATION REGISTERED FOR PRIVATE USE

Class 15: EXPERIMENTAL METADATA

whose class number are assigned to the first byte of the Item Designator (byte 9 of the 16 byte UL Key).

Below those primary classes are developed with finer nodes (representing subclasses) and leaves (representing metadata items) to organize the metadata items into a hierarchical structure. This is implemented in the Item Designator by inheriting the (sub-) class number of a parent class into its child classes or metadata items belonging to the class. For example, all classes and metadata items that belong to “IDENTIFIERS & LOCATORS” primary class have its first byte of the Item Designator as $\mathtt{01}_h$. Also, if a parent class has non-zero values for the first *n* bytes of its Item Designator, any child classes or metadata items belonging to the parent class shall have the same non-zero values for the first *n* bytes of their Item Designator. Please see SMPTE EG 37 [19] for more information on the subclass development.

In general, a metadata item that is regarded as a leaf of the hierarchy structure has no children items. But there is one exception when the metadata item has multiple representations, or different datatypes for the “Value” field at its instantiation. In this case, a metadata item having default representation or datatype is used not only as the metadata item to be instantiated on its own but also as a *representative* (a kind of class) of those variations. For example, the metadata item “Main title” with the Unicode datatype has the 16-byte UL Key of $\mathtt{06}_h$ $\mathtt{0E}_h$ $\mathtt{2B}_h$ $\mathtt{34}_h$ $\mathtt{01}_h$ $\mathtt{01}_h$ $\mathtt{01}_h$ $\mathtt{01}_h$ $\mathtt{01}_h$ $\mathtt{05}_h$ $\mathtt{02}_h$ $\mathbf{01_h}$ $\mathtt{00}_h$ $\mathtt{00}_h$ $\mathtt{00}_h$ $\mathtt{00}_h$, which is different from the UL Key of “Main title” with ASCII (ISO/IEC 646:1991 - ISO 7-Bit Coded Character Set) by its byte 12 being $\mathtt{01}_h$ (See Section 3.3).

It should be noted that the representative metadata item, or “Main title” with ASCII in the above case, is still regarded as a *leaf* in the hierarchy structure even though it has its own variations as its child items from the Item Designator viewpoint. Therefore care needs to be taken when to consider whether an item is as a node or a leaf, which is unambiguously specified at the special column of the SMPTE Metadata Dictionary [4].

Annex C Reserved terms for EssenceMark

In this annex, the reserved terms for EssenceMark as a part of the EssenceMark specifications in Sony Corporation is presented.

To minimize collisions with other EssenceMark values used elsewhere, a set of syntax rules is initially established, which shall apply to any reserved terms to be registered, as follows:

- ✓ The term shall be expressed in 7 bit ASCII (ISO 7-Bit Coded Character Set) only,
- ✓ The term shall start with "_" ($5F_h$),
- ✓ If the term is composed of more than one word, then it shall be represented in so called *camel style*, or each word shall start with upper character and be directly connected without any delimiter,
- ✓ The term less than 32 characters shall be terminated with a white space (20_h). Arbitrary text may follow the white space if needed, which is out of scope of this specification.

Table C-1 below shows a list of the reserved terms registered at the time of this writing[8]. Note that this list will continue to evolve without notice when to find new requirements or demands.

Reserved term[a)]	Definition	Target frame[b)]	Note
`_RecStart`	A position from which AV essence recording starts.	A frame around the recoding start position.	Compatible with Sony conventional "REC START MARK".
`_RecEnd`	A position to which AV essence recording ends.	A frame around the recoding end position.	
`_ShotMark1`	Noticeable point 1.	A frame specified by a device, a user, or an application.	Compatible with Sony conventional "SHOT MARK 1".
`_ShotMark2`	Noticeable point 2.	A frame specified by a device, a user, or an application.	Compatible with Sony conventional "SHOT MARK 2".
`_ShotMark3`	Noticeable point 3.	A frame specified by a device, a user, or an application.	
`_ShotMark4`	Noticeable point 4.	A frame specified by a device, a user, or an application.	
`_ShotMark5`	Noticeable point 5.	A frame specified by a device, a user, or an application.	
`_ShotMark6`	Noticeable point 6.	A frame specified by a device, a user, or an application.	
`_ShotMark7`	Noticeable point 7.	A frame specified by a device, a user, or an application.	

[8] Table C-1 has been updated to reflect the information as of April 20, 2009.

_ShotMark8	Noticeable point 8.	A frame specified by a device, a user, or an application.	
_ShotMark9	Noticeable point 9.	A frame specified by a device, a user, or an application.	
_ShotMark0	Noticeable point 0.	A frame specified by a device, a user, or an application.	
_Cut	Cut detected position.	A frame around the cut detected position.	
_Flash	Flash detected position.	A frame around the flash detected position.	
_FilterChange	A position from which the lens filter is exchanged.	A frame around which the lens filter is exchanged.	Only for those occur during AV essence recording.
_ShutterSpeedChange	A position from which the shutter speed is changed.	A frame around which the shutter speed is changed.	Only for those occur during AV essence recording.
_GainChange	A position from which the gain is changed.	A frame around which the gain is changed.	Only for those occur during AV essence recording.
_WhiteBalanceChange	A position from which the white balance is changed.	A frame around which the white balance is changed.	Only for those occur during AV essence recording.
_OverBrightness	A position at which video output level exceeds 100%.	A frame around which video output level exceeds 100%.	
_OverAudioLimiter	A position at which audio output level exceeds its predefined upper limit.	A frame around which audio output level exceeds its predefined upper limit.	
_In-XXX	In point, or the start position from which AV essence is to be clipped.	A frame specified by a device, a user, or an application.	May be used with corresponding _Out-XXX as a pair, where XXX denotes arbitrary printable characters. A frame associated with _In-xxx shall be included in the essence to be clipped. If absent (_Out-xxx exists only), _In-xxx is assumed to be associated with the first frame of AV essence.
_Out-XXX	Out point, or the end position to which AV essence is to be clipped.	A frame specified by a device, a user, or an application.	May be used with corresponding _In-XXX as a pair, where XXX denotes arbitrary printable characters. A frame associated with _Out-xxx shall NOT be included in the essence to be clipped. If absent (_In-xxx exists only), _Out-xxx is assumed to be associated with *the next frame* of the last frame of AV essence.

_KeyFrame	A position of a key frame that represents the AV essence.	A frame specified by a device, a user, or an application.	
_ClipTransit-NN	A divided position when a single AV material is divided into multiple AV clips	The first frame of each resultant AV clip.	NN denotes the number between 02 and 99 representing the sequence order of the AV clips in the original AV material, where AV clip is defined as a content unit in basic operation such as record, playback and delete.

Table C-1: Reserved terms for EssenceMark

Note:

a) If the EssenceMark list name (See Section 3.6.2) for the reserved terms is needed, the value of “Sony EssenceMark” shall be used.

b) For a device or an application capability reason, this specification by itself does not require the frame level accuracy of the EssenceMark association.

References

1 Yoshiaki Shibata, Takumi Yoshida and Mitsutoshi Shinkai, "EssenceMark - SMPTE Standard-based Textual Video Marker -", SMPTE Motion Imaging Journal, Vol.114, No.12, pp.463-473 (Dec 2005)

2 http://www.smpte.org

3 SMPTE 336M-2001: SMPTE Standard for Television – Data Encoding Protocol Using Key-Length-Value

4 SMPTE RP 210-2004 : SMPTE Metadata Dictionary (Version 8)

5 SMPTE 365M-2001: SMPTE Standard for Digital Television Tape Recording – 12.65-mm Type D-10 Format for MPEG-2 Compressed Video 525/60 and 625/50

6 W3C Recommendation, 04 February 2004: Extensible Markup Language (XML) 1.0 (Third Edition) (http://www.w3.org/TR/REC-xml)

7 SMPTE 291M-1998: SMPTE Standard for Television – Ancillary Data Packet and Space Formatting

8 SMPTE RP 214-2002: Packing KLV Encoded Metadata and Data Essence into SMPTE 291M Ancillary Data Packets

9 SMPTE 326M-2000: SMPTE Standard for Television – SDTI Content Package Format (SDTI-CP)

10 SMPTE 331M-2004: SMPTE Standard for Television – Element and Meta-data Definition for the SDTI-CP

11 SMPTE 377M-2004: SMPTE Standard for Television – Material Exchange Format (MXF) Material Exchange Format (MXF) File Format Specification

12 SMPTE 385M-2004: SMPTE Standard for Television – Material Exchange Format (MXF) Mapping SDTI-CP Essence and Metadata into the MXF Generic Container

13 SMPTE 380M-2004: SMPTE Standard for Television – Material Exchange Format (MXF) Descriptive Metadata Scheme – 1

14 For example, see Clause 4.4.9 (Affective description) of ISO/IEC TR 15938-8:2002 Information technology – Multimedia content description interface -- Part 8: Extraction and use of MPEG-7 descriptions

15 Yoshiaki SHIBATA, Takayoshi KAWAMURA, Hideki ANDO and Mitsutoshi SHINKAI, "Introduction to XDCAM Metadata", Proceedings of NAB 2005 Broadcast Engineering Conference, pp.421-424 (2005)

16 SMPTE 298M-1997: SMPTE Standard for Television – Universal Labels for Unique Identification of Digital Data

17 ISO/IEC 8825-1:1995 Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)

18 SMPTE 335M-2001: SMPTE Standard for Television – Metadata Dictionary Structure

19 SMPTE EG 37-2001: Node Structure for the SMPTE Metadata Dictionary