SMPTE ENGINEERING GUIDELINE

EG 19-1988

Device Control Elements

1. *General*

1.1 Scope. This guideline is to be used as supplementary information for SMPTE Recommended Practice RP 146-1987, Transfer of Edit Decision Lists. It describes the data format of the device control element, thus adding it to the set of valid elements in an edit decision list. The information in a device control element allows a device specific message to be included in the EDL, thus allowing complex audio/video effects to be reproduced on compatible editing systems. The transfer medium is not specified by this document.

1.2 Defined Terms:

ASCII: American National Standard Code for Information Interchange—7-Bit Coded Character Set, ANSI X3.4-1986, which describes the character encoding used by this guideline.

Edit Decision List: One or more edits.

Edit: One or more interrelated elements treated as a unit.

Element: A single line in an edit decision list containing information specifying audio/video content decisions in an audio/video product.

Field: A syntactical unit within an element that contains a functional grouping of data. Fields are delineated within an element by separators.

Hexadecimal: A notation for representing numbers in base 16. Each 8-bit byte is represented by two ASCII characters. The first character represents the 4 high-order bits, and the second character represents the 4 low-order bits. The range of values of each of the 4-bit nibbles is 0 to 15, which is represented by the ASCII characters 0 through 9 for values 0 through 9, and ASCII characters A through F for values 10 through 15.

Separator: Character(s) used to delineate fields within an element.

Source: A reference to video or audio program material.

Virtual Edit: Specification of an effect that is not to be recorded directly, but can be referenced later as a source.

1.2.1 Notation. The syntax of the device control element format is specified using a notation commonly used in the software industry called BNF (Backus-Naur Form). The use of this notation allows precise, yet easy to understand, syntactic specification of the device control element format.

A BNF grammar consists of a list of rules which define grammatical terms. The first term defined is typically the highest level construct in the grammar, in this case, a DEVICE_CONTROL_ELEMENT. As there are many variations of the notation known as BNF, the following is a formal definition of the notation used in this guideline:

Grammatical terms are capitalized, with an underscore (_) separating words.

Grammatical rules consist of:

the name of the grammatical term being defined,
followed by a colon (:),
followed by a list of grammatical terms,
BNF operators, and/or literals,
and terminated with a semi-colon (;).

Literal strings are enclosed within double-quotes (").

Literal characters are enclosed by single-quotes (').

Non-printable characters are represented by their name as shown in Appendix A1, ASCII Character Values.

Comments begin with "/*" and end with "*/".

The logical OR operation is specified with a vertical bar (|).

Use of square brackets, ([and]), indicates that 0 or more occurrences of the items in the brackets are allowed.

Lower-case alphabetic characters are considered equivalent to upper-case alphabetic characters in the grammar. For the sake of clarity, grammatical rules are specified using only upper-case alphabetic characters.

The complete grammar is defined in Appendix A2.

Copyright © 1988 by the
SOCIETY OF MOTION PICTURE AND TELEVISION ENGINEERS
595 West Hartsdale Avenue, White Plains, NY 10607, (914) 761-1100

Approved October 26, 1988

2. *Device Control Element Structure*

2.1 General Information. The information in a DEVICE_CONTROL_ELEMENT consists of 8-bit data bytes. The low-order 7 bits are interpreted as characters of the ASCII character set. The high-order bit is ignored. No distinction is made between upper- and lower-case alphabetic characters when interpreting device control elements.

2.2 Format of Device Control Elements. A DEVICE_CONTROL_ELEMENT consists of a PRINTABLE_STRING, as defined by the grammar in Appendix A2, followed by a TERMINATOR (ETB). The basic unit of information is the field, consisting of one or more characters. Fields are delineated by one or more separators, which can be any combination of space (SP), tab (HT), line-feed (LF), or carriage return (CR).

2.3 Edit Number General. The first field of an ELEMENT consists of up to four sub-fields, concatenated without separators:

EDIT_NUMBER_SUBFIELD	(required)
VIRTUAL_EDIT_INDICATOR_SUBFIELD	(optional)
RECORDED_INDICATOR_SUBFIELD	(optional)
ELEMENT_TYPE_IDENTIFIER_SUBFIELD	(optional)

The EDIT_NUMBER_FIELD may be preceded by a separator, but embedded separators are not permitted.

2.3.1 Edit Number Subfield. The EDIT_NUMBER_SUBFIELD consists of from one to six numeric characters, followed optionally by an alphabetic character. The range of valid edit numbers is from 1 to 999999Z. The number zero (0) is not a valid edit number. Leading zeroes are insignificant in differentiating edit numbers (i.e., 0001 is equivalent to 1).

2.3.2 Virtual Edit Indicator Subfield. A VIRTUAL_EDIT is indicated by the greater-than character (>) immediately following the EDIT_NUMBER_SUBFIELD. All elements of a VIRTUAL_EDIT must have this indicator. VIRTUAL_EDITs are defined in RP 146-1987.

2.3.3 Recorded Indicator Subfield. The RECORDED_INDICATOR_SUBFIELD indicates that this element has been recorded. The RECORDED_INDICATOR_SUBFIELD consists of an exclamation mark (!) immediately following the VIRTUAL_EDIT_INDICATOR_SUBFIELD, which may be empty.

2.3.4 Element Type Identifier Subfield. The ELEMENT_TYPE_IDENTIFIER_SUBFIELD indicates the type of ELEMENT. Element type identifiers are as follows:

'*'—COMMENT_ELEMENT
'?'—USER_ELEMENT
'#'—TRIGGER_ELEMENT
'%'—DEVICE_CONTROL_ELEMENT
' '—SOURCE_ELEMENT

The absence of this subfield indicates a SOURCE_ELEMENT.

2.4 Device Control Element. The DEVICE_CONTROL_ELEMENT defines message strings and their execution times. It consists of the following fields:

EDIT_NUMBER_FIELD
DEVICE_ID_FIELD
TIME_CODE_SRC_IDENTIFIER_FIELD
TIME_CODE_TYPE_FIELD
MESSAGE_TIME_FIELD
MESSAGE_COUNT_FIELD
MESSAGE_FIELD
TERMINATOR

2.4.1 Device ID Field. This field contains an identifier which specifies the device to which this message should be sent. The device identifier is a symbolic name assigned by the user.

2.4.2 Time Code SRC Identifier Field. The TIME_CODE_SRC_IDENTIFIER_FIELD is an optional field which changes the MESSAGE_TIME_FIELD from a time code in the audio/video product to a time code in the specified source material. If the TIME_CODE_SRC_IDENTIFIER_FIELD is present, the specified source must appear in a SOURCE_ELEMENT in the current edit.

2.4.3 Time Code Type Field. The TIME_CODE_TYPE_FIELD is an optional field which specifies that the MESSAGE_TIME_FIELD is an offset relative to the entry or exit time code of the audio/video product (or the specified source). If the TIME_CODE_TYPE_FIELD is not used, the MESSAGE_TIME_FIELD specifies an absolute time code in the audio/video product or an absolute time code in the specified source material. The valid options for TIME_CODE_TYPE_FIELD are:

ENTRY+ /* Entry time code plus offset */
ENTRY— /* Entry time code minus offset */
EXIT+ /* Exit time code plus offset */
EXIT— /* Exit time code minus offset */

If a TIME_CODE_TYPE_FIELD is specified, the current edit must include at least one SOURCE_ELEMENT.

2.4.4 Message Time Field. This field contains either a time code or the keyword "SETUP." The "SETUP" keyword indicates that this message should be sent prior to execution of the edit.

If this field contains a time code, it specifies when the message should be sent to the device.

Table 1 summarizes the options for the modifiers to the MESSAGE_TIME_FIELD.

Table 1
Message Time Field Modifiers

TIME_CODE_SRC_IDENTIFIER	TIME_CODE_TYPE	Message Time
		Program Absolute
	ENTRY +	Program Relative Entry Plus
	ENTRY —	Program Relative Entry Minus
	EXIT +	Program Relative Exit Plus
	EXIT —	Program Relative Exit Minus
SRC_ID		Source Absolute
SRC_ID	ENTRY +	Source Relative Entry Plus
SRC_ID	ENTRY —	Source Relative Entry Minus
SRC_ID	EXIT +	Source Relative Exit Plus
SRC_ID	EXIT —	Source Relative Exit Minus

2.4.5 Message Count Field. This field contains the length, in bytes, of the message string which will be sent. The message count is expressed as a hexadecimal number.

2.4.6 Message Field. This field defines the actual message string which will be sent. Each byte in the actual message is represented by encoding it into ASCII using two hexadecimal digits per byte. Separators within the message field will be ignored.

3. *Time Code Format*

Time code is expressed in terms of four two-digit groups representing hours (HH), minutes (MM), seconds (SS), and frames (FF), separated by punctuation marks (p), in the form HHpMMpSSpFF. (See American National Standard for Television —Time and Control Code—Video and Audio Tape for 525-Line/60-Field Systems, ANSI/SMPTE 12M-1986.) The separator between the seconds and frames groups is significant in that it is used to denote drop-frame/non-drop-frame time code and to indicate field 1 or field 2, as follows:

. (period)	Non-drop-frame code, field 1.
, (comma)	Drop-frame code, field 1.
: (colon)	Non-drop-frame code, field 2.
; (semicolon)	Drop-frame code, field 2.

Allowance is made for the expression of time code in compact form by the suppression of leading zeroes and associated punctuators, up to but not including the seconds-frames punctuator.

Appendix A1
ASCII Character Values

ASCII Character Set
(American National Standard Code for Information Interchange)

Character Values in Octal

000 NUL	020 DLE	040 SP	060 0	100 @	120 P	140 `	160 p
001 SOH	021 DC1	041 !	061 1	101 A	121 Q	141 a	161 q
002 STX	022 DC2	042 "	062 2	102 B	122 R	142 b	162 r
003 ETX	023 DC3	043 #	063 3	103 C	123 S	143 c	163 s
004 EOT	024 DC4	044 $	064 4	104 D	124 T	144 d	164 t
005 ENQ	025 NAK	045 %	065 5	105 E	125 U	145 e	165 u
006 ACK	026 SYN	046 &	066 6	106 F	126 V	146 f	166 v
007 BEL	027 ETB	047 '	067 7	107 G	127 W	147 g	167 w
010 BS	030 CAN	050 (	070 8	110 H	130 X	150 h	170 x
011 HT	031 EM	051)	071 9	111 I	131 Y	151 i	171 y
012 NL	032 SUB	052 *	072 :	112 J	132 Z	152 j	172 z
013 VT	033 ESC	053 +	073 ;	113 K	133 [	153 k	173 {
014 NP	034 FS	054 ,	074 <	114 L	134 \	154 l	174 |
015 CR	035 GS	055 —	075 =	115 M	135]	155 m	175 }
016 SO	036 RS	056 .	076 >	116 N	136 ^	156 n	176 ~
017 SI	037 US	057 /	077 ?	117 O	137 _	157 o	177 DEL

Character Values in Hex

00 NUL	10 DLE	20 SP	30 0	40 @	50 P	60 `	70 p
01 SOH	11 DC1	21 !	31 1	41 A	51 Q	61 a	71 q
02 STX	12 DC2	22 "	32 2	42 B	52 R	62 b	72 r
03 ETX	13 DC3	23 #	33 3	43 C	53 S	63 c	73 s
04 EOT	14 DC4	24 $	34 4	44 D	54 T	64 d	74 t
05 ENQ	15 NAK	25 %	35 5	45 E	55 U	65 e	75 u
06 ACK	16 SYN	26 &	36 6	46 F	56 V	66 f	76 v
07 BEL	17 ETB	27 '	37 7	47 G	57 W	67 g	77 w
08 BS	18 CAN	28 (	38 8	48 H	58 X	68 h	78 x
09 HT	19 EM	29)	39 9	49 I	59 Y	69 i	79 y
0A NL	1A SUB	2A *	3A :	4A J	5A Z	6A j	7A z
0B VT	1B ESC	2B +	3B ;	4B K	5B [	6B k	7B {
0C NP	1C FS	2C ,	3C <	4C L	5C \	6C l	7C |
0D CR	1D GS	2D —	3D =	4D M	5D]	6D m	7D }
0E SO	1E RS	2E .	3E >	4E N	5E ^	6E n	7E ~
0F SI	1F US	2F /	3F ?	4F O	5F _	6F o	7F DEL

Appendix A2
Device Control Element Syntax

The following is a formal definition of the syntax of the DEVICE CONTROL ELEMENT:

```
DEVICE_CONTROL_ELEMENT:
    EDIT_NUMBER_FIELD '%'
        SEPARATOR  DEVICE_ID_FIELD
        SEPARATOR  TIME_CODE_SRC_IDENTIFIER_FIELD
        SEPARATOR  TIME_CODE_TYPE_FIELD
        SEPARATOR  MESSAGE_TIME_FIELD
        SEPARATOR  MESSAGE_COUNT_FIELD
        SEPARATOR  MESSAGE_FIELD
    ;

DEVICE_ID_FIELD:
    IDENTIFIER
    ;

TIME_CODE_SRC_IDENTIFIER_FIELD:
    SOURCE_IDENTIFIER
  | /* empty */
    ;

TIME_CODE_TYPE_FIELD:
    "ENTRY+"
  | "ENTRY—"
  | "EXIT+"
  | "EXIT—"
  | /* empty */
    ;

MESSAGE_TIME_FIELD:
    "SETUP"
  | TIME_CODE_FIELD
    ;

MESSAGE_COUNT_FIELD:
    NUMBER
    ;

MESSAGE_FIELD:
    MESSAGE_BYTE
  | MESSAGE_BYTE MESSAGE_FIELD
  | MESSAGE_BYTE SEPARATOR MESSAGE_FIELD
    ;

MESSAGE_BYTE:
    HEXADECIMAL_DIGIT  HEXADECIMAL_DIGIT
    ;
```

```
IDENTIFIER:
     ALPHA
   | NUMBER
   | IDENTIFIER ALPHA
   | IDENTIFIER NUMBER
   ;

TIME_CODE_FIELD:
     LEADING_RADIX_NUMBER
          TIME_CODE_SEPARATOR RADIX_NUMBER
          TIME_CODE_SEPARATOR RADIX_NUMBER
          TIME_CODE_SEPARATOR RADIX_NUMBER
   | LEADING_RADIX_NUMBER
          TIME_CODE_SEPARATOR RADIX_NUMBER
          TIME_CODE_SEPARATOR RADIX_NUMBER
   | LEADING_RADIX_NUMBER
          TIME_CODE_SEPARATOR RADIX_NUMBER
   | TIME_CODE_SEPARATOR RADIX_NUMBER
   ;

LEADING_RADIX_NUMBER:
     DIGIT
   | DIGIT DIGIT
   ;

RADIX_NUMBER:
   DIGIT DIGIT
   ;

PRINTABLE_STRING:
     PRINTABLE_CHARACTER
   | PRINTABLE_STRING PRINTABLE_CHARACTER
   ;

PRINTABLE_CHARACTER:
     '!' | '"' | '#' | '$' | '%' | '&' | '''
   | '(' | ')' | '*' | '+' | '-' | '/'
   | '<' | '=' | '>' | '?'
   | '@'
   | '[' | '\' | ']' | '^' | '_'
   | '`'
   | '{' | '|' | '}' | '~'
   | ALPHA
   | DIGIT
   | TIME_CODE_SEPARATOR
   | SEPARATOR
   ;

TIME_CODE_SEPARATOR:
     ':'
   | ';'
   | '.'
   | ','
   ;

EDIT_NUMBER_FIELD:
     EDIT_NUMBER_SUBFIELD VIRTUAL_EDIT_INDICATOR_SUBFIELD
          RECORDED_INDICATOR_SUBFIELD
   ;

EDIT_NUMBER_SUBFIELD:
     NUMBER                          /*   Note the limit to 6 digits.   */
   | NUMBER ALPHA
   ;

VIRTUAL_EDIT_INDICATOR_SUBFIELD
     '>'
   |                                 /*   Empty Field                   */
   ;
```



```
RECORDED_INDICATOR_SUBFIELD
     '!'
   |                                 /*   Empty Field                   */
   ;

ALPHA:
     UPPER_ALPHA
   | LOWER_ALPHA
   ;

UPPER_ALPHA:
     'A' | 'B' | 'C' | 'D' | 'E' | 'F' | 'G' | 'H' | 'I' | 'J'
   | 'K' | 'L' | 'M' | 'N' | 'O' | 'P' | 'Q' | 'R' | 'S' | 'T'
   | 'U' | 'V' | 'W' | 'X' | 'Y' | 'Z'
   ;

LOWER_ALPHA:
     'a' | 'b' | 'c' | 'd' | 'e' | 'f' | 'g' | 'h' | 'i' | 'j'
   | 'k' | 'l' | 'm' | 'n' | 'o' | 'p' | 'q' | 'r' | 's' | 't'
   | 'u' | 'v' | 'w' | 'x' | 'y' | 'z'
   ;

REAL_NUMBER:
     INTEGER
   | INTEGER '.' NUMBER
   ;

INTEGER:
     NUMBER
   | '+' NUMBER
   | '-' NUMBER
   ;

NUMBER:
     DIGIT
   | NUMBER DIGIT
   ;

HEXADECIMAL_NUMBER:
     HEXADECIMAL_DIGIT
   | HEXADECIMAL_NUMBER  HEXADECIMAL_DIGIT
   ;

HEXADECIMAL_DIGIT:
     DIGIT
   | 'A' | 'B' | 'C' | 'D' | 'E' | 'F'
   ;

DIGIT:
     '0'
   | NON_ZERO_DIGIT
   ;

NON_ZERO_DIGIT:
   '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'
   ;

SEPARATOR:
     SPACE | 'HT' | 'CR' | 'LF'
   | SEPARATOR SPACE
   | SEPARATOR 'HT'
   | SEPARATOR 'CR'
   | SEPARATOR 'LF'
   ;

SPACE:
     ' '
   ;

TERMINATOR:
     'ETB'
   ;
```