

GVG/EDITWARE

EDIT DECISION LIST (EDL) FORMAT

Last Update: 04-Jan-2001

This document covers the GVG/Editware EDL format currently supported by Editware's Super Edit software. This specification is subject to change at any time without notice.

EDL File Structure

The general form of an EDL file is as follows:

```
<CRLF> <TITLE> <CRLF> <TITLE2> <CRLF> <FRAME CODE TYPE>  
<CRLF> <CRLF> <EVENT> <NOTE> .....<CRLF> <CTRL-Z> <0xFF>.
```

The file is then generally padded out to an even 512-byte boundary with nulls. Each of these parts is important and must be included. They are discussed further. An EDL file is readable with most ASCII based text editors.

1. An EDL file begins with a carriage return / line feed pair (CRLF).
2. Every line of the EDL ends with a CRLF.
3. The EDL title string is the first full line in the file, after the initial CRLF. It is a valid note line. It must start with a non-numeric character in the range of 0x3A to 0x7E. It must end with a CRLF. It must have a maximum length of 79 characters. If a Super Edit user enters a title that begins with a numeric character, Super Edit will add a * (0x2A) character to the beginning of the title string.
4. The second title, noted previously as TITLE2, is the title of the Super Edit software that produced the EDL file. It generally has the following form: "SUPER EDIT V1.2A RELEASE DPE-551 S12345 ACME POST ROOM C (2200) NTSC". Its purpose is for tracking and readability of the resulting EDL file. Super Edit currently discards this line during EDL input, but it should exist in the file as a valid EDL note line for compatibility.
5. The Frame Code Type is a single line indicating the type of Time Code contained in the EDL. Its purpose is to document the time code type of the resulting EDL file. Super Edit currently discards this line during EDL input, but it should exist in the file as a valid EDL note line for compatibility. Current possible types are:
 - a. SMPTE FRAME CODE (Can be a mixture of DF and NDF)
 - b. DROP FRAME CODE
 - c. NDROP FRAME CODE

In the future, additional types may be included and Super Edit may parse them. The strings used to indicate these future types are proposed to be:

- a. EW FRAME CODE: DROP (NTSC, DF (0-29 using standard drop algorithm))
 - b. EW FRAME CODE: NONDROP (NTSC, NDF (0-29 uninterrupted count))
 - c. EW FRAME CODE: SMPTE (NTSC - mixed counts signaled by timecode string separator, “:” for NDF, and “;” for DF)
 - d. EW FRAME CODE: PAL (0-24 uninterrupted count (25fps, nominal))
 - e. EW FRAME CODE: FILM24P (0-23 uninterrupted count (24fps, nominal))
6. After the Frame Code Type and its associated CRLF, there may be one additional line. This line is optional. When an EDL is output, Super Edit currently adds an additional CRLF at this location, which is also optional. Upon EDL input, it ignores this CRLF. If a valid note line did exist here, Super Edit would currently ignore it as well. Editware reserves the right to use and parse this line in the future. Current plans are to use it for version information. If a version note is present, it will currently be in the following form: <EDL V8.70> <sp> <sp> <SE V2.1AR>.
 7. An EDL file ends with <CR> <LF> <CTRL-Z> <0xFF>, which translates to: 0x0D, 0x0A, 0x1A, 0xFF. Upon EDL output, Super Edit will include these bytes verbatim. Upon EDL input, Super Edit will honor either 0x1A or 0xFF as an end of file marker.
 8. Versions of Super Edit that run under the RT-11 operating system (all GVG/Editware VPE series editors) require that the EDL file be padded out with nulls after the 0xFF character, to an even 512 byte boundary. This was a requirement of RT-11. Editware DPE series editors, which currently run under Windows NT, no longer require the file be padded out with nulls. Editware recommends the file be padded out for compatibility.

EDL Lines in General

1. There are only two valid EDL line types, notes and events. Event lines begin with a numeric character, 0x30 to 0x39. All other lines are considered notes. A valid note line can contain any number of characters from 1 to 255. A note line can be empty, in which case it must contain only <CRLF>.
2. A note line is not currently allowed at the beginning of the EDL. If one exists, it will be discarded upon EDL input. However, Editware reserves the right to change that in the future.
3. Note lines are considered to belong to the event line preceding it in the file.
4. Event lines contain information about an edit event, such as reel names, in times and out times. All event lines begin with a unique event number. They

may or may not be stored in numerical order. Event lines can currently be made up of one or two lines. Each line of a two-line event will have the same event number. Every event line is made up of various fields that contain data or information. The fields are the same for both lines of a two-line event, but they may contain different data, and the data may mean different things.

5. There are two distinct types of note lines. Regular note lines are user input ASCII strings used for readability and documentation. They begin with a * character. System note lines are generated and used by the Super Edit system and generally do not begin with a * character (there are some exceptions). They generally contain information and/or data that will not fit into a standard event line.

EDL Event Lines

An event line is any line that starts with a numeric character. A single line event will always be a CUT event. Two-line events are used to describe a transition between sources, split edits and keys. Split Edits and Keys are described later. For transitions, the first line describes the FROM source and the second line describes the TO source, along with the transition information. Each event line contains a number of fields, each separated by one or more space characters (0x20). The fields are identical for one and two line events. Examples of valid event lines are shown later in figure 1. The fields are enumerated as shown.

1111 222222 333333 4444 555 666666666666 77777777777 888888888888 99999999999 R

1. A unique, 4-digit event number in the range of 0001 to 9999. It is an index number, not a sequence number. Events with consecutive event numbers will not necessarily be recorded in that sequence.
2. A 6 character alphanumeric reel ID or name. All characters are significant. No embedded spaces are allowed. The reel name is padded out with spaces to always be 6 characters long. The size of this field may increase in the future.
3. A 6 character field that identifies the record audio and video channels. If it takes less than six chars to identify the channels, it is padded out to 6 characters with spaces. If any audio channels are involved in the edit, the field will begin with the letter "A" followed by the active audio channels. Currently, up to 4 channels of audio are possible. Examples would be A1, A12, A123, A1234, A4, A34, etc. If there is video recorded in the event the letter "V" will follow the audio channel string. Examples would be A12V, A1234V, A34V etc. If the edit is a split audio or video edit, it will be a two line event. The first line will contain the original audio/video channels and the second line will contain the audio/video channels after the split point.
4. A four character field beginning with a single character indicating the transition type for the event, followed by one or more characters containing information about the transition. The current legal transition types are:
 - a. Cut = <C> <sp> <sp> <sp>. A CUT event is the only legal single line event. For a Wipe or a Dissolve, the first line is a CUT which describes

the FROM source. For a Split Edit, the first line is a CUT which describes the original audio/video channels.

- b. Dissolve = <D> <sp> <sp> <sp>. A Dissolve is always a two-line event. The first line describes the FROM source using a CUT line. The second line, which contains the “D”, describes the TO source.
 - c. Wipe = <W> <0> <1> <0>. A “W” followed by a 3 digit wipe code. A Wipe is always a two-line event. The first line describes the FROM source using a CUT line. The second line, which contains the “W”, describes the TO source.
 - d. Key = <K> <sp> <KSub> <sp>. A “K”, followed by 1 space, followed by a key sub field, followed by 1 space. A KEY is always a two-line event. A delayed KEY will have a CUT line followed by a two-line KEY event with an event number 1 higher than the CUT line. The first line of a KEY describes the Background source. The second line describes the Foreground source. The KEY sub field is described as follows:
 - i. The first line of a KEY will contain a “B” for Background, <K> <sp> <sp>.
 - ii. The second line of a KEY will contain all spaces if it is a KEY IN, <K> <sp> <sp> <sp>, or it will contain an “O” if it is a KEY OUT, <K> <sp> <O> <sp>.
5. A 3 character field that has three possibilities.
- a. If the event line represents a Cut transition, it will contain 3 spaces.
 - b. If the event line represents a Wipe or Dissolve, it will contain a three-digit transition duration, in frames. Transitions durations are always in the second line of a two-line event.
 - c. If the event represents a Key, it will describe the transition as follows:
 - i. The first line of a KEY will contain three spaces, unless it is a FADE in/out, in which case it will contain (F) as <(<F> <)>.
 - ii. The second line of a KEY will contain a three-digit transition duration.
6. An 11 character field that contains the Source In Time as a time code number in the form <hours> <.:> <minutes> <.:> <seconds> <.:> <frames>. For example, 00:01:05:22. The delimiter between the seconds and frames conforms to the SMPTE standard and will be a “:” if the timecode is a non-drop frame number, or a “;” if it is a drop frame number. This is also true of the time code numbers in fields 7, 8 and 9 below. This will be the first frame of the source that was recorded on the record machine or timeline.
7. An 11 character field that contains the Source Out Time as a time code number in the form <hours> <.:> <minutes> <.:> <seconds> <.:> <frames>. This will be the frame AFTER the last frame of the source that was recorded on the record machine or timeline.

8. An 11 character field that contains the Record In Time as a time code number in the form <hours> <:> <minutes> <:> <seconds> <:> <frames>. This will be the first frame recorded on the record machine or timeline.
9. An 11 character field that contains the Record Out Time as a time code number in the form <hours> <:> <minutes> <:> <seconds> <:> <frames>. This will be the frame AFTER the last frame recorded on the record machine or timeline.
10. A 1 character field that will contain an R-mark, <R>, if the event has been recorded, or a space if not.

EDL Event Line Timecode Numbers

The timecode numbers for the Source/Record In Times and Out Times indicate different things depending upon the type of EDL line it is. The duration of an EDL line is defined as the absolute value of the Source In Time subtracted from the Source Out Time. Note that the Source duration and the Record duration will always be equal in an event line.

1. For a standard Wipe or Dissolve, the CUT lines In Times and Out Times will be equal indicating a duration of zero. In other words there is no delay before the transition.
2. For a delayed Wipe or Dissolve, the duration of the first line is the delay time before the transition start. The In Time of the second line is the frame the transition starts on.
3. For Split Edits, the duration of the first line is the full duration of the edit. The In Time of the second line will be the A/V split point.
4. For all Key Edits, the duration of the first line is the full duration of the edit.
5. For a Key In, if the duration of the 2nd line is zero, the Key is on for the duration of the edit. If it is not zero, the Key Foreground ends at the Out Time. If the EDL line has a transition duration, the Transition On starts at the In Time and the transition Off starts at the Out Time.
6. A Key Out is the same as a Key In except the Transition Out occurs at the Out Time noted in the second line.

EDL System Note Lines

EDL note lines are used by the system to hold data that helps describe the event it is associated with. They may contain data or describe specific actions.

1. PEGS notes. PEGS notes start with the string "PEG". PEGS are a way of describing actions in a timeline fashion during an event. They have the following form (from V7.0 and up):

<P> <E> <G> <sp> <Function> <sp> <Code> <sp> <TC Offset> <sp> <sp> <Reel ID>

Note that Super Edit versions prior to V7.0 did not include the Reel ID field.

- a. The single character Function and Code field may contain the following:
 - i. “A, B, C, D, E or F”. These correspond to programmed motion commands to a particular source, such as A-VTR, B-VTR, etc. This value is present but ignored with Super Edit software V7.0 and later. Instead, the Reel ID field is used to determine which source this PEG refers to. The 4-character code field will be a three-digit number, plus sign, corresponding to the percent of play speed command this device was given, for example 030 or -120.
 - ii. “G” corresponds to a GPI (General Purpose Interface) command, which would be a contact closure pulse. The Code field will be a space plus a 3 digit number that corresponds to which GPI contact to pulse.
 - iii. “V” corresponds to a command sent to a Video Switcher. The Code field corresponds to the command to execute, such as an EMEM recall.
 - iv. “M” corresponds to a command sent to an Audio Mixer. The code field corresponds to the command to execute such as an AMEM recall to a GVG AMX-170.
 - v. “T” corresponds to a command sent to a Dubner Character Generator. The code field corresponds to the command to execute such as a trigger to a Dubner. A string is sent, not a contact closure.
 - vi. “K” corresponds to a command sent to a GVG Kaleidoscope. The code field corresponds to the command to execute such as an effect trigger. A string is sent, not a contact closure.
 - vii. “P” corresponds to a command sent to a Peripheral device. The code field corresponds to the command to execute such as a TBC setting transfer sent to a Zaxcom TBC controller.
 - viii. “R” corresponds to a Pre-Read command sent to a record device. The code field corresponds to the command to execute such as ON or OFF.
 - ix. “X” corresponds to a command sent to a GVG switcher to perform a transition command. The code field corresponds to the command to execute.
 - x. “Q” corresponds to a command sent to a device to store or recall its TBC settings. The code field corresponds to the command to execute.
 - xi. “W” corresponds to special wipe commands sent to certain Sony switchers. The code field corresponds to the command to execute.
- b. The eight-character Time Code Offset (TC Offset) field corresponds to when the PEG was triggered or sent, and is an offset, in frames, from the

Record In Time of the event associated with this PEG (the event immediately prior). It only shows minutes, seconds and frames. Note the TC Offset may change in the future to an 11 character TC number which would show hours, minutes, seconds and frames.

- c. The six-character Reel ID is only present with Super Edit software V7.0 and later, and only with a programmed motion PEG. It corresponds to the six-character Reel Name of the source a PEG refers to. Note the length of the Reel ID may change in the future to accommodate reel names of 24 characters or more.
2. Slave notes. Slave notes start with the string "SLV". They have the following form: <SLV> <sp> <Reel ID> <sp> <Time Code Offset> <sp> <Slave Set> <sp> <Channel>. Slave notes describe additional sources that were involved in the edit beyond those included in the event lines.
 - a. Reel ID is a six-character field corresponding to the six-character Reel ID of the source a Slave Note refers to.
 - b. Time Code Offset is an 11-character field, which contains a SMPTE standard time code value that represents the slave devices In Time as an offset to the Record In Time of the event.
 - c. Slave Set refers to a single character field corresponding to the slave set this slave belongs to. If it contains an "R", it is a Record Slave, but may or may not be a Recording Slave, as determined by the Channel field.
 - d. Channel is a variable length field corresponding to the active channels of a Recording Slave. If it is not present, the slave was not a Recording Slave, even though it may be slaved to the master recorder. If any audio channels were active on this device, the field will begin with the letter "A" followed by the active audio channels. Examples would be A1, A12, A123, A1234, A4, A34, etc. If there was video recorded on this device, the letter "V" will follow the audio channel string. Examples would be A12V, A1234V, A34V etc.
 3. 8 Channel Audio notes. In order to allow support for 8 channel audio while remaining compatible with the existing EDL format, audio channels beyond the original 4 are designated in an audio note. In other words, audio channels 1,2,3 and 4 are specified in the event proper and channels 5,6,7 and 8 are specified in a system note. The note has the form: <AUDX> <sp> <sp> <A5678> <sp> <A5678> <sp> <sp>.
 - a. The note starts with the string "AUDX" followed by two spaces.
 - b. The next field is a 5 character field that will be all spaces if this is not a split audio edit. If it is, it will contain the extended audio channels active before the split point. If it takes less than 5 characters to describe the audio channels, it is padded out with spaces. It always starts with an "A", with the channels immediately following. Examples are A56, A5678, A78.

- c. After a space comes the second audio field. If this event is not a split audio edit, this field will describe the events extended audio channels. If it is a split audio event, it will describe the extended audio channels active after the split point. The format is identical to the previous field.
 - d. The last field is followed by two spaces.
4. EMEM notes. EMEM notes hold effects memory and other data from a switcher. The data is stored with an event in EMEM blocks that consist of an ASCII header followed by ASCII Hex data. Each EMEM block can hold up to 255 bytes of data. An EMEM note can have multiple EMEM blocks. An event can have multiple EMEM notes. They have the following general form:
- ```
<EMEM> <sp> <sp> <BANK/REG#> <0x03> <BYTCNT> <DATA.....>
<EMEM> <*> <sp> <1101> <0x03> <BYTCNT>
<DATA.....>.....
```
- a. The note starts off with ASCII “EMEM” to indicate it is an EMEM note.
  - b. The “EMEM” is followed by two spaces if this is the first block of EMEM data; otherwise it is followed by an ampersand (&) and one space. The ampersand indicates this is an EMEM continuation block, and not the beginning of a new EMEM note. There can be many continuation blocks. And there can be many EMEM notes associated with one event.
  - c. The next field is a 4-digit EMEM register number. This indicates the switcher register number and effects bank that held this EMEM data. It will have one of two possible forms:
 

<sp> <1 digit bank> <2 digit register #>, or  
 <2 digit bank> <2 digit register #>.
  - d. The next field is a special code to mark the beginning of hex data. All characters after this are stored in hex pairs that represent the EMEM data. It will always be the 11<sup>th</sup> character from the beginning of the EMEM note.
  - e. The next two bytes are a hex pair representing the byte count of the data that follows. You must use this byte count to acquire the EMEM data and/or to advance to the next EMEM or EMEM block. It will have a value from 1 to 255. There is no end of data marker. You must use the byte count to navigate. It will always be the 12<sup>th</sup> character from the beginning of the EMEM note.
  - f. If the EMEM is larger than 255 bytes, there will be multiple EMEM blocks contained in the EMEM note. The second and subsequent EMEM blocks will have a (&) located immediately after the “EMEM” in the header, to distinguish them from the start of a new EMEM note.
5. PMEM notes are identical to EMEM notes except they begin with ASCII “PMEM”. The byte count will always be the 12<sup>th</sup> character from the beginning of the PMEM note.



6. Auto TBC notes are similar to EMEM notes, but they begin with "TBC<sp>". The byte count will always be the 12<sup>th</sup> character from the beginning of the TBC note.
7. QMEM notes are identical to EMEM notes except they begin with ASCII "QMEM". The byte count will always be the 12<sup>th</sup> character from the beginning of the QMEM note.
8. Summary notes. When an event has PEGS and/or SLAVES associated with it, there is a summary note placed after all the notes in an event. It indicates the total number of PEGS and SLAVES associated with that event. It takes the form of: xxxx PEGS, yyyy SLVS<0x04>, where xxxx is the blank padded 4 digit number of PEGS and yyyy is the blank padded 4 digit number of SLAVES.

### **Profile NLE Additions to the EDL**

1. Clip notes identify the full name of the clip and original sources for certain 'trace' functions. Clip names do not include the directory, and are limited to a total of 32 characters including the '=REEL ID'. Clip names cannot have an embedded space. However, spaces between fields are required. Both Super Edit and 409NT can generate clip notes. The general format is:

\*CLIP: <sp> <clip name> <sp> <From source> <sp> <To source>

The From and To source may or may not be present depending on the type of clip. An attempt to make a more formal definition might loosely be:

\*CLIP: <sp> <clip[=reel]> <sp> [<From[=reel]> <sp> [<To[=reel]>]]

Some examples are :

Ex1: a clip named BOB1 that was digitized from a reel of taped named REEL1 would look like this:

\*CLIP: BOB1=REEL1

Ex2: A clip that was made from 2 sources (a dissolve or wipe,etc) with the clip named BOB2 and the From source REEL2 and the To source REEL3 would look like this:

\*CLIP: BOB2=TEMP REEL2 REEL3

Ex3: Similar to ex2, with clip named 'GARY' and the two sources are from 2 other clips, BOB8=REEL1, and BOB9=REEL1 would look like this:

\*CLIP: GARY=TEMP BOB8=REEL1 BOB9=REEL1

The reel ID that follows the equal sign is limited to 6 characters and the following names are reserved for use by Super Edit.

- MOVI1 - represents a Mode 1 movie clip
- MOVI2 - represents a Mode 2 movie clip
- MOVI3 - represents a Mode 3 movie clip
- TEMP - indicates a rendered event located in the temp directory
- \*TMP - pointer to a TEMP clip
- HOT - points to a segment of a clip

2. A work directory note identifies the directory in Profile where the clips are located. Only 1 work directory per EDL is allowed.

The format is:

\*WORK DIR = <directory name>

example:

\*WORK DIR = INT1:/DEFAULT/

3. A timeline start note shows the time at the beginning of the timeline. Only one timeline start time per EDL is allowed.

The format is:

TL START <start time>

Example:

TL START 01:00:00:00