X16 Video File Format Draft #2

This is a specification for an Video file format for the Commander X16 platform.   It is not meant for general use on other platforms . Tools will be provided on those Platforms (Linux/BSD/Windows/Mac) to convert standard file formats such as .mp4 & .avi to the X16 format.

DESIGN GOALS

- A format optimized for the X16 that is easily programmed for
- Takes full advantage of all capabilities of the VERA video hardware.
- Portability for software developed for X16

DETAILED FILE SPECIFICATION

FILE ELEMENT	SIZE IN BYTES	PURPOSE	Comment
File ID (*char*)	4	To identify this file as an X16 Video File	Always “**SPRV**” $53,$50,$52,$56
V_TYPE (*byte*)	1	Type of Video Frame	***Described in Addendum 1***
Bit Depth (*byte*)	1	What is the color depth of this video ?	1, 2, 4 or 8 (Only 4 or 8 for Type 1 or 2)
FPS (*byte*)		Frames per Second	Desired Playback rate
Sprite Width (*Byte*)	1		Sprite Width for Sprite Based Videos
Sprite Height (*Byte*)	1		Sprite Height for Sprite Based Videos
Frame Width *(word little endian)*	2		Frame Resolution Width
Frame Height (word little endian)	2		Frame Resolution Height
Number of Frames (Dword little endian)	4		Total number of Frames in this Video. (also used with **FPS** to calculate the **VERA** Audio rate)
SCREEN HINT (Char)	1	N – Normal D – Vertical Dbled. T – Tall (TikTok)	Not exactly the aspect Ratio. But a Hint as to how to Display the Video
EXTRA/RESERVED	14		
VIDEO DATA	UNLIMITED		The ***Frames*** start here.

QB64PE Type Definition

```
Type SVIDHeaderType
   ID As String * 4
   VTYPE As _Unsigned _Byte
   BPP As _Unsigned _Byte
   FPS As _Unsigned _Byte
   SpriteWidth As _Unsigned _Byte
   SpriteHeight As _Unsigned _Byte
   FrameWidth As _Unsigned Integer
   FrameHeight As _Unsigned Integer
   NumFrames As _Unsigned Long
   SHint As String * 1
   EXTRA As String * 14
End Type
```

C struct Definition

```
struct SVIDHeaderType {
   char ID[4];
   uint8_t VTYPE;
   uint8_t BPP;
   uint8_t FPS;
   uint8_t SpriteWidth;
   uint8_t SpriteHeight;
   uint16_t FrameWidth;
   uint16_t FrameHeight;
   uint32_t NumFrames;
   char SHint[1];
   char EXTRA[14];
};
```

ADDENDUM #1 *(Frame Types)*

TYPE 1: Sprite Based Video with full Palette following each frame, Palette is 512 bytes in size for 8 bpp or 32 bytes for 4bpp. Video frame is made up of a *Sprite Grid.* The sprite size is specified in Sprite Width & Sprite Height. Frame data is immediately followed by Palette data for the frame.

TYPE 2: Not implemented (Raw sprites NO palette data)
(for either grayscale 4 bpp or VERA Default Palette 8 bpp)

TYPE 3: Bitmap Based video with full Palette following each frame. Either 640 or 320 width, though the converter disallows 640 for 4bpp vids (just to big). Bit depths 1, 2 or 4. Sprite size is insignificant for this vid type. (Now implemented) **MAXIMUM FRAME BUFFER:** ***38400 bytes***

TYPE 4: Not implemented (bitmap frames no palette)

TYPE 5: TBD

TYPE 6: Similiar to Type 1 but doesn't use a separate Audio file. Interleaved 8 bit single channel Audio. First two bytes of **EXTRA** will be ***Audio bytes per Frame.*** *(Next to implement)****. Not yet implemented as of this writing.***

TYPE 7,8: TBD

TYPE 9: *Similiar to Type 3 but doesn't use a separate Audio file. Interleaved 8 bit single channel Audio. First two bytes of* **EXTRA** *will be* **Audio bytes per Frame.** *(Next to implement)****. Not yet implemented as of this writing.***