10503 Timberwood Circle
Suite 120
Louisville, KY 40223-5318

Voice: 502.423.7225
Fax: 502.425.7064
Web: www.lumitron-ir.com

Thursday, July 06, 2000

Subject: SVS-2000 Mk2 Vector (*.vct/*.rvt) File Format

Below is a portion of the source code that contains a list of define's and struct's that make up the Lumitron Vector or Replacement Vector file format.

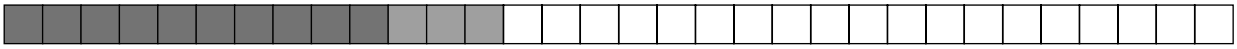
The file is written out in the following manner:
(1) **LFileHdr** (All Lumitron files, begins @ 0x0)
(2) **VectorFileHdr** (begins @ 0x4D)
(3) **Vector Data** (begins @ 0xBC)

Notes:
a. The scanType member of the VectorFileHdr structure may or may not exist in your vector file. Older versions of software had VectorFileHdr.custom[60] and no scanType member. This will not affect the start of the data.
b. DWORD - unsigned long (4 bytes)
c. WORD - unsigned short (2 bytes)
d. Data is typically packed on 8 byte boundaries, therefore if a member type will not fit into the boundary it will be pushed into the next block.

```
struct _LFileHdr
{
        char    text[47];               // Man readable text
        char    eof;                    // End of file character (decimal 26)
        char    signature[6];           // Lumitron signature ("LInc")
        UINT    product;                // Lumitron product code
        char    filetype;               // Product file type
        char    ver[13];                // Version of program that wrote this file
};
typedef struct _LFileHdr LFileHdr;

#define         LUM_SIGNATURE           "LInc"  // For Mk2 v2.00 and above

// Define Lumitron product codes
#define         LUM_SVS2000             4       // Lumitron Standard SVS2000 Product

// Define Product file types
#define         LUM_VECTOR_FILE         4       // Vector file

/*
================================================================================================
Below, define the Vector File header.  This header is written to the disk after the base
Lumitron header has been written.
================================================================================================
*/
struct _VectorFileHdr
{
        int     fpaDimFormat;           // Listbox control index
        int     fpaScanFormat;          // Listbox control index
        WORD    inScanPix;              // Vertical Input Pixels
        WORD    videoOutputs;           // Number Video Outputs
        WORD    videoChannels;          // Number of Channels
        WORD    crossScanPix;           // Horizontal Input Pixels
```

```
        WORD    xferHorizPix;         // Displayed Horizontal Pixels
        WORD    xferVertPix;          // Displayed Vertical Pixels
        int     pattern;              // Pattern type used
        int     flipHoriz;            // Flip horizontal setting
        int     flipVert;             // Flip vertical setting
        int     rotationAngle;        // Rotation angle setting
        char    custom[56];           // Name of DLL defining custom vector
        DWORD   scanType;             // Custom in-scan type 0-Horiz 1-Vert
        WORD    fullFrameHoriz;       // Full Frame Horizontal setting
        WORD    fullFrameVert;        // Full Frame Vertical setting
        DWORD   intraClip;            // Intra clip setting
        DWORD   interClip;            // Inter clip setting
        DWORD   dataSize;             // Size of Vector data
};
typedef _VectorFileHdr VectorFileHdr;
```

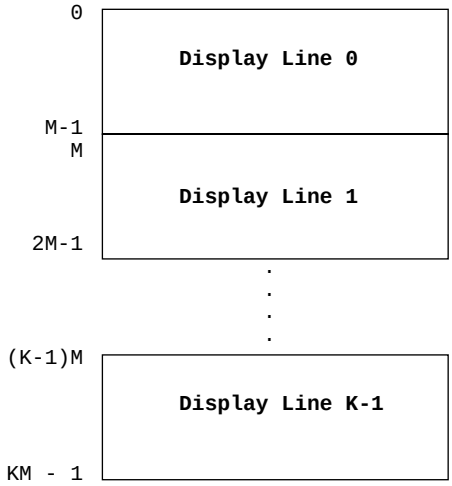
Data Format (32 Bits for each pixel):



32 **22 21** **19 18**
0

Bits 22 - 32: Don’t care.
Bits 19 - 21: Acquisition Card Number -> 0 thru 7.
Bits 0 - 18: Acquisition Card Pixel Addresses -> 0 thru (512k - 1).

There are M x K entries in the vector table, one entry for each display pixel organized as shown below:



Where M = # of pixels per display line,
and K = # of lines per display frame.