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 Defective Pixel List (*.dpl) File Format

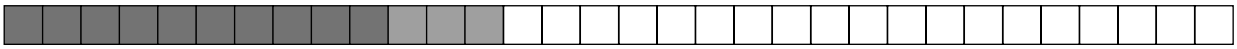
Below is a portion of the source code that contains a list of define's and struct's that make up the Lumitron Defective Pixel List (DPL) file.

The file is written out in the following manner:
(1) **LFileHdr** (All Lumitron files, begins @ 0x0)
(2) **BadPixFileHdr** (begins @ 0x4D)
(3) **Pixel Data** (begins @ 0x64)

Notes:
a. DWORD - unsigned long (4 bytes)
b. WORD - unsigned short (2 bytes)
c. 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_DEFECT_FILE         6       // Defective pixel file

/*
================================================================================================
Below, define the BadPixel File header.  This header is written to the disk after the base
Lumitron header has been written.
================================================================================================
*/
struct _BadPixFileHdr
{
        short   format;                 // Area or Linear
        short   numAcqCards;            // Number of acquisition cards we have data for
        int     chPerCard;
        DWORD   hPix;                   // Horizontal pixels
        DWORD   vPix;                   // Vertical pixels
        DWORD   pixCount;               // Count of defective pixels
        DWORD   dataSize;               // Size of DPL data
};
typedef struct _BadPixFileHdr BadPixFileHdr;
```

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).