Phantom SDK
Cine File Format Manual

Version 13.8.804.6 (March 2023)

100 Dey Rd Wayne, NJ 07470, USA | Tel +1 973 696 4500

Software Release

Some new fields of the SETUP structure may be added in new software releases.

This document is based on software release 804.6

(PhCon.dll, PhFile.dll, PhInt.dll version 13.8.804.6, PCC version 3.8.804.6).

Table of Contents

1. Introduction 5

2. Notation and Formats 7

2.1. Decimal and Hexadecimal 7
2.2. Terminology 7
2.3. Data Types 7
2.4. Simple Data Structures 8
2.5. Image Numbering 10

3. Cine File Structures 12

3.1. The Cine File Header (CINEFILEHEADER structure) 13
3.2. Windows Structure for the Image Header (BITMAPINFOHEADER) 14
3.3. Camera Setup Information (the SETUP Structure) 16
3.3.1 Structure Marker and Size 20
3.3.2 Acquisition Parameters 20
3.3.3 Pixel Value 24
3.3.4 Image Processing 25
3.3.5 Binary and Digital Signal Acquisition, Range Data 33
3.3.6 User and Camera Metadata 34
3.3.7 Calibration Information 37
3.3.8 Continuous Recording 38
3.3.9 Not Used Anymore 38
3.4. The Tagged Information Blocks 39
3.4.1 Analog and Digital Signals Tagged Block 39
3.4.2 Image Time Tagged Block 39
3.4.3 Time Only Block 39
3.4.4 Exposure Only Block 39
3.4.5 Range Data Block 40
3.4.6 BinSig Block 40
3.4.7 AnaSig Block 40
3.4.8 TimeCode Block 40
3.4.9 Array of Image Offsets 40
3.5. The Image Object 41
3.5.1 The Annotation Data 41
3.5.2 Pixel Array 41

4. Pixel Values 44

5. Types of Cine Files 46

5.1. Raw Cine Files 46
5.1.1 Standard Raw Cine Files 46
5.1.2 Packed Raw Cine Files 46
5.1.3 P10 - Packed 10 bit with gamma 47
5.1.4 P12L - Packed 12 bit Linear 47
5.2. Interpolated Cine Files 48
5.3. Compressed Cine Files 48

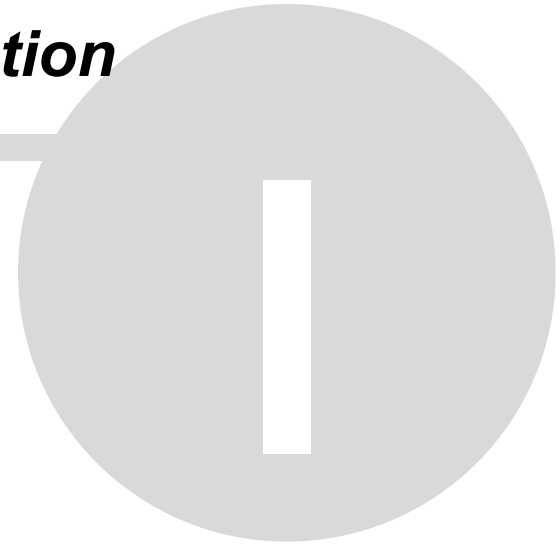
6. Access to Pixels 50

7. Revision Notes 52

8. Appendices 56

8.1. Appendix 1. LUT for Conversion from 10 Bits Packed to the 12 bit Linear 56
8.2. Appendix 2. The SETUP Structure and Substructures 58
8.3. Appendix 3. Cine File Dump 67
8.4. Appendix 4. Fields Offsets in Structures and Cine File 74

Part I Introduction



1. Introduction

This document describes the cine file format used for saving video information and auxiliary data captured from digital, high speed video cameras.

Cine file format was designed for storing and retrieving the recordings made by the Phantom high speed video cameras from Vision Research Inc.

The main goal of the cine file is to store both the image pixels values and additional information like the acquisition parameters, image time, analog signals recorded in parallel with the images, range data, etc. A cine file contains all the data produced at the recording of an event, so you can retrieve later the information, playback the images, analyze them, etc.

The pixel information can be raw (as read from sensor, without the interpolation of the colors or any other image processing) RGB interpolated or compressed. The raw format is preferred because it is fast to save, the file has smaller size and there is not any loss of information at save. The color interpolation is delayed to the moment when the cine is viewed or converted to another RGB format.

Compressed cine files have the .cci default name extension (for Compressed Cine). The uncompressed files (both raw and color interpolated) have the default extension ".cin" before software version 645 (June 2007) and ".cine" after that date.

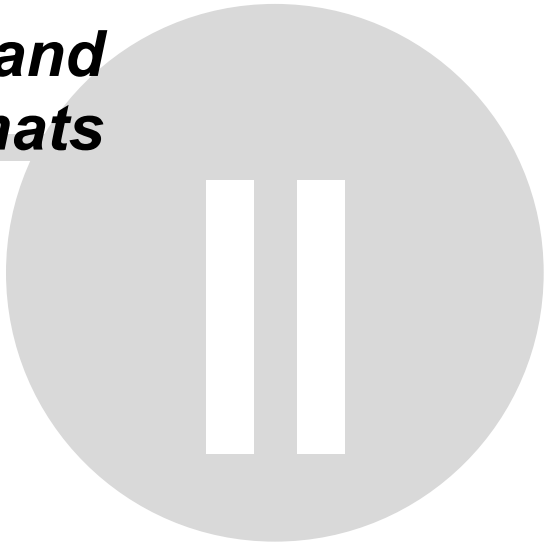
Besides storing all information in one place, another important requirement for a video file format is maintaining the best possible compatibility with video players released during the years. The most important is the backward compatibility of the video players: being able to play all the files recorded in the past (even long time ago) with the most recent version of the players. This is provided by maintaining the initial structure of the file and adding new information in tagged blocks or in expandable data structures.

Since the players are often part of some expensive motion analysis packages that are not upgraded too often, it is also important to be able to open new files in older applications. Those applications will read and use only the known part of the expandable data structures and the known tagged blocks.

The cine files can be read or written using our SDK available for the Windows operating system. The Phantom SDK offers services to read and write the images from cine files and cameras, auxiliary data (image time, exposure, analog and binary signals, range data...) and all the metadata available at the recording moment. This is the preferred approach to access the cine files, it is the simplest to use and provides the best compatibility, both backward and forward. Look into the Phantom SDK to find the documentation and player examples for cine files.

This document is mainly intended for the developers that cannot or do not want to use Phantom SDK. The reason for that is usually writing applications for an operating system different from Windows. Before starting development, you may check on the Internet to find a plug-in or other support software from third parties that can reduce your development effort.

Part II Notation and Formats



2. Notation and Formats

2.1. Decimal and Hexadecimal

Unless otherwise expressed, all numeric values in this document are expressed in decimal. A "0x" is the prefix for hexadecimal values.

2.2. Terminology

Cine is a video file format used for storing a sequence of images, their acquisition parameters, and other auxiliary data.

The *frame rate* represents the number of frames recorded per second acquired during recording. The maximum value of a frame rate depends on the camera model and image resolution. Other terms used for referring to frame rate are: sample rate, picture rate or image rate.

The *file offset* values – *pointers* or *addresses* – are expressed in bytes and are related to the beginning of the file, unless other reference is specified.

The term *size* – or *length* – of memory data or file data refers to the number of bytes of that memory data or file data.

Shutter duration and the *exposure time* have identical meanings.

A *tagged block* refers to a segment of information with variable length. The first field of the tagged block contains the length of the block measured in bytes and provides a simple way to skip the block when analyzing a cine file.

2.3. Data Types

The byte order of the data stored in the cine file is with the least significant byte first (Intel little endian).
The basic data types and their sizes are:

uint8_t	8-bit unsigned integer
char	8-bit signed integer
uint16_t	16-bit (2-byte) unsigned integer
int16_t	16-bit (2-byte) signed integer
bool32_t	32-bit (4-byte) logic value (TRUE = 1, FALSE = 0)
uint32_t	32-bit (4-byte) unsigned integer
int32_t	32-bit (4-byte) signed integer
int64_t	64-bit (8-bytes) signed integer
float	32-bit (4-byte) floating point
double	64-bit (8-byte) floating point
char []	array of chars terminated by a 0 byte

2.4. Simple Data Structures

TIME64

```
typedef uint32_t FRACTIONS, *PFRACTIONS;

typedef struct tagTIME64
{
    FRACTIONS fractions;
    uint32_t  seconds;
}TIME64, *PTIME64;
```

fractions

Fractions of a second.
Stored here multiplied by 2^{32} and rounded to integer.
Least significant 2 bits store information about IRIG synchronization and the camera Event input:

Bit 0 Value	Meaning
0	IRIG synchronized
1	Not synchronized

Bit 1 Value	Meaning
0	Event input = 0 (short to ground)
1	Event input = 1 (open)

seconds

Seconds from January 1, 1970.
Compatible with the C library routines.
The maximum year allowed by the unsigned representation is 2106.

IMFILTER

```
typedef struct tagIMFILTER
{
    int32_t dim;
    int32_t shifts;
    int32_t bias;
    int32_t Coef[5*5];
}IMFILTER, *PIMFILTER;
```

dim

The dimension of the square convolution kernel. It can be 3 or 5.

shifts

Right shifts of Coef (8 shifts mean divide by 256). The shift is applied after the multiply and add operations of the convolution kernel.

bias

Bias to add after convolution.

Coef

Filter coefficients represented as integers.
The actual value of the coefficients is `Coef[i,j]` shifted right `shifts` times. For example;

if `Coef[i,j]` is `1` and `shifts` is `4` the actual float value of the coefficient is `1/16 = 0.0625`. `Coef` can be used both for a `3x3 or a 5x5` filter.

WBGAIN

```
typedef struct tagWBGAIN
{
    float R;
    float B;
}WBGAIN, *PWBGAIN;
```

R White balance coefficient representing gain correction for red channel.

B White balance coefficient representing gain correction for blue channel.

The gain coefficient for green channel is considered `1.0`.

2.5. Image Numbering

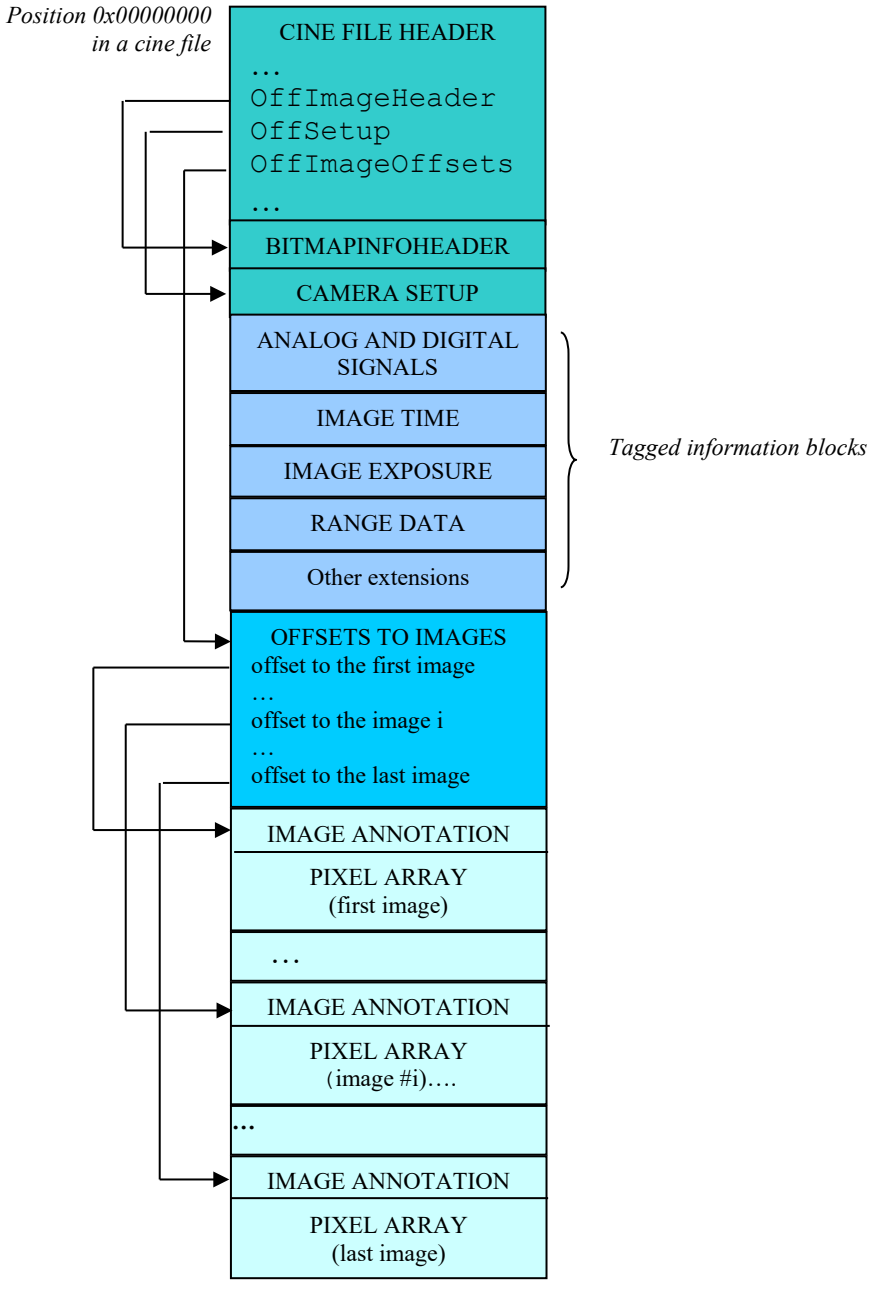
The images are numbered in a growing order using 32 bit signed values.
The images before trigger have negative numbers. The images after the trigger have zero or positive numbers.

Part III Cine File Structures

3. Cine File Structures

At the beginning, the cine file contains a few fixed structures: CINEFILEHEADER, BITMAPINFOHEADER and SETUP. They are followed by an array with the positions of the images in the file that makes the access to the images faster. Between the SETUP structure and the array of the pointers to the images we have a number of tagged blocks with variable length.

Here is what a cine file contains:



The order of structures in the file is according to the figure above.

3.1. The Cine File Header (CINEFILEHEADER structure)

It contains version information, image range, absolute trigger time and offsets to the other structures in the file. This header is also included in the .chd file, created when Phantom software saves the images in a file format other than cine.

Here is the content of the structure:

```
typedef struct tagCINEFILEHEADER
{
    uint16_t Type;
    uint16_t Headersize;
    uint16_t Compression;
    uint16_t Version;
    int32_t  FirstMovieImage;
    uint32_t TotalImageCount;
    int32_t  FirstImageNo;
    uint32_t ImageCount;
    uint32_t OffImageHeader;
    uint32_t OffSetup;
    uint32_t OffImageOffsets;
    TIME64   TriggerTime;
}CINEFILEHEADER;
```

Type
This is the marker of a cine file. It must be "CI" in any cine file.

Headersize
It represents the CINEFILEHEADER structure size as a number of bytes.

Compression

Value		Meaning
CC_RGB	= 0	for gray cines
CC_JPEG	= 1	for a JPEG compressed file (*.cci)
CC_UNINT	= 2	for uninterpolated color (RAW) file

See more details in the CFA field description in SETUP structure.

Version
Version number may increase in time if substantial changes are made to the file format. Version 0 had the array of pointers to images on 32 bits and it was limited to maximum 4GB file size.
Current version is 1 and the format supports files bigger than 4GB. Starting with release 600 all Phantom applications write version 1 cines while still reading version 0 and version 1 cine files.

FirstMovieImage
First recorded image number, relative to trigger.

TotalImageCount
Total count of images recorded in the camera memory.

FirstImageNo
First image saved to this file, relative to trigger.

ImageCount
Count of images saved to this file.

OffImageHeader
Offset of the BITMAPINFOHEADER structure in the cine file.

OffSetup
Offset of the SETUP structure in the cine file.

OffImageOffsets
Offset in the cine file of an array with the positions of each image stored in the file.

TriggerTime
Trigger time is a TIME64 structure having the seconds and fraction of second since Jan 1, 1970 (resolution: approx. 1/4 nanosecond).

3.2. Windows Structure for the Image Header (BITMAPINFOHEADER)

It contains information about the image dimensions and bit depth of the pixels. It is identical to the structure from Windows, but the meaning of some fields has been extended to support images having more than 8 bits per color component.

Here is the content of the structure:

```
typedef struct tagBITMAPINFOHEADER
{
    uint32_t biSize;
    int32_t  biWidth;
    int32_t  biHeight;
    uint16_t biPlanes;
    uint16_t biBitCount;
    uint32_t biCompression;
    uint32_t biSizeImage;
    int32_t  biXPelsPerMeter;
    int32_t  biYPelsPerMeter;
    uint32_t biClrUsed;
    uint32_t biClrImportant;
}BITMAPINFOHEADER;
```

biSize
It specifies the number of bytes required by the structure (without palette).

biWidth
It specifies the width of the bitmap, in pixels.

biHeight
It specifies the height of the bitmap, in pixels.
If **biHeight** is positive, the bitmap is a bottom-up DIB and its origin is the lower-left corner.
If **biHeight** is negative, the bitmap is a top-down DIB and its origin is the upper-left corner.
Phantom specific: the negative biHeight convention is not used. The standard BI_RGB formats (8, 24, 16, 48 bits per pixels) that support image processing are stored bottom-up. The packed BI_PACKED bitmaps are stored top-down as in the camera magazine.

biPlanes
It specifies the number of planes for the target device. This value must be set to 1.

biBitCount
It specifies the number of bits-per-pixel.
The **biBitCount** member of the **BITMAPINFOHEADER** structure determines the number of bits that define each pixel and the maximum number of colors in the bitmap.
Phantom specific: biBitCount can be only 8, 24, 16, 48 bits. 8 and 16 bit DIBs are monochrome, 24 and 48 are RGB color DIBs. The meaning of the 16 bit DIB is different from Windows: it is a 16 bit per pixel gray image. Each pixel value is stored on 16 bits even if the real bit depth produced by the camera is 14, 12 or 10 bits. 48 value of this field corresponds to a color image having 16 bits per color component. Color palette images (8 bpp) are not supported; in the Phantom environment they are converted to 24 bpp after the file load or after the copy from clipboard. The palette is

not written in the cine file but a gray palette is needed to render the monochrome 8 bpp DIBs in Windows.

biCompression

It specifies the type of compression for a compressed bottom-up bitmap (top-down DIBs cannot be compressed).

Phantom specific: Only BI_RGB=0, BI_PACKED=256 and BI_PACKED12L=1024 are supported.

BI_PACKED: Describes a 10 bit packed array of pixels. Four pixels are stored in five bytes (40 bits). Packing the 10 bits pixels reduce the memory or file size by a factor of 10/16.

BI_PACKED12L: Describes a 12 bit packed array of pixels. Two pixels are stored in three bytes (24 bits).

The BI_PACKED and BI_PACKED12L bitmaps remain with top-down row order in the packed cine file, the same as in the camera memory.

biSizeImage

It specifies the image size in bytes.

biXPelsPerMeter

It specifies the horizontal resolution, in pixels-per-meter, of the target device for the bitmap. An application can use this value to select a bitmap from a resource group that best matches the characteristics of the current device.

Phantom specific: biXPelsPerMeter, biYPelsPerMeter are the resolutions computed at the level of the camera sensor. To get the resolution in the scene, you must multiply these values by the distance from the camera to the scene divided by the focal length of the lenses.

biYPelsPerMeter

Vertical resolution in pixels per meter – in the sensor plane.

biClrUsed

Specifies the number of color indexes in the color table that are actually used by the bitmap. If this value is zero, the bitmap uses the maximum number of colors corresponding to the value of the **biBitCount** member for the compression mode specified by **biCompression**.

biClrImportant

It specifies the number of color indexes that are required for displaying the bitmap. If this value is zero, all colors are required.

Phantom specific: If biBitCount is 16 or 48 biClrImportant should be the maximum sample value + 1, that is the maximum number of the sample levels. biClrImportant has to be 16384, 4096 or 1024, for the images having the bit depth 14, 12 or 10 bits.

3.3. Camera Setup Information (the SETUP Structure)

It contains the acquisition parameters used during the recording of the cine.

Here is the content of the structure:

```
typedef struct tagSETUP
{
    uint16_t FrameRate16;
    uint16_t Shutter16;
    uint16_t PostTrigger16;
    uint16_t FrameDelay16;
    uint16_t AspectRatio;
    uint16_t Res7;
    uint16_t Res8;
    uint8_t  Res9;
    uint8_t  Res10;
    uint8_t  Res11;
    uint8_t  TrigFrame;
    uint8_t  Res12;
    char     DescriptionOld[MAXLENDESCRIPTION_OLD];
    uint16_t Mark;
    uint16_t Length;
    uint16_t Res13;
    uint16_t SigOption;
    int16_t  BinChannels;
    uint8_t  SamplesPerImage;
    char     BinName[8][11];
    uint16_t AnaOption;
    int16_t  AnaChannels;
    uint8_t  Res6;
    uint8_t  AnaBoard;
    int16_t  ChOption[8];
    float    AnaGain[8];
    char     AnaUnit[8][6];
    char     AnaName[8][11];
    int32_t  lFirstImage;
    uint32_t dwImageCount;
    int16_t  nQFactor;
    uint16_t wCineFileType;
    char     szCinePath[4][OLDMAXFILENAME];
    uint16_t Res14;
    uint8_t  Res15;
    uint8_t  Res16;
    uint16_t Res17;
    double   Res18;
    double   Res19;
    uint16_t Res20;
    int32_t  Res1;
    int32_t  Res2;
    int32_t  Res3;
    uint16_t ImWidth;
    uint16_t ImHeight;
    uint16_t EDRShutter16;
    uint32_t Serial;
    int32_t  Saturation;
    uint8_t  Res5;
    uint32_t AutoExposure;
    bool32_t bFlipH;
```

```
    bool32_t bFlipV;
    uint32_t Grid;
    uint32_t FrameRate;
    uint32_t Shutter;
    uint32_t EDRShutter;
    uint32_t PostTrigger;
    uint32_t FrameDelay;
    bool32_t bEnableColor;
    uint32_t CameraVersion;
    uint32_t FirmwareVersion;
    uint32_t SoftwareVersion;
    int32_t  RecordingTimeZone;
    uint32_t CFA;
    int32_t  Bright;
    int32_t  Contrast;
    int32_t  Gamma;
    uint32_t Res21;
    uint32_t AutoExpLevel;
    uint32_t AutoExpSpeed;
    RECT     AutoExpRect;
    WBGAIN   WBGain[4]
    int32_t  Rotate;
    WBGAIN   WBView;
    uint32_t RealBPP;
    uint32_t Conv8Min;
    uint32_t Conv8Max;
    int32_t  FilterCode;
    int32_t  FilterParam;
    IMFILTER UF;
    uint32_t BlackCalSVer;
    uint32_t WhiteCalSVer;
    uint32_t GrayCalSVer;
    bool32_t bStampTime;
    uint32_t SoundDest;
    uint32_t FRPSteps;
    int32_t  FRPImgNr[16];
    uint32_t FRPRate[16];
    uint32_t FRPExp[16];
    int32_t  MCCnt;
    float    MCPercent[64];
    uint32_t CICalib;
    uint32_t CalibWidth;
    uint32_t CalibHeight;
    uint32_t CalibRate;
    uint32_t CalibExp;
    uint32_t CalibEDR;
    uint32_t CalibTemp;
    uint32_t HeadSerial[4];
    uint32_t RangeCode;
    uint32_t RangeSize;
    uint32_t Decimation;
    uint32_t MasterSerial;
    uint32_t Sensor;
    uint32_t ShutterNs;
    uint32_t EDRShutterNs;
    uint32_t FrameDelayNs;
    uint32_t ImPosXAcq;
    uint32_t ImPosYAcq;
    uint32_t ImWidthAcq;
    uint32_t ImHeightAcq;
    char     Description[MAXLENDESCRIPTION];
```

```
    bool32_t RisingEdge;
    uint32_t FilterTime;
    bool32_t LongReady;
    bool32_t ShutterOff;
    uint8_t  Res4[16];
    bool32_t bMetaWB;
    int32_t  Hue;
    int32_t  BlackLevel;
    int32_t  WhiteLevel;
    char     LensDescription[256];
    float    LensAperture;
    float    LensFocusDistance;
    float    LensFocalLength;
    float    fOffset;
    float    fGain;
    float    fSaturation;
    float    fHue;
    float    fGamma;
    float    fGammaR;
    float    fGammaB;
    float    fFlare;
    float    fPedestalR;
    float    fPedestalG;
    float    fPedestalB;
    float    fChroma;
    char     ToneLabel[256];
    int32_t  TonePoints;
    float    fTone[32*2];
    char     UserMatrixLabel[256];
    bool32_t EnableMatrices;
    float    cmUser[9];
    bool32_t EnableCrop;
    RECT     CropRect;
    bool32_t EnableResample;
    uint32_t ResampleWidth;
    uint32_t ResampleHeight;
    float    fGain16_8;
    uint32_t FRPShape[16];
    TC       TrigTC;
    Float    fPbRate;
    float    fTcRate;
    char     CineName[256];
    float    fGainR;
    float    fGainG;
    float    fGainB;
    float    cmCalib[9];
    float    fWBTemp;
    float    fWBCc;
    char     CalibrationInfo[1024];
    char     OpticalFilter[1024];
    char     GpsInfo[MAXSTDSTRSZ];
    char     Uuid[MAXSTDSTRSZ];
    char     CreatedBy[MAXSTDSTRSZ];
    uint32_t RecBPP;
    uint16_t LowestFormatBPP;
    uint16_t LowestFormatQ;
    float    fToe;
    uint32_t LogMode;
    char     CameraModel[MAXSTDSTRSZ];
    uint32_t WBType;
    float    fDecimation;
```

```
    uint32_t MagSerial;
    uint32_t CSSerial;
    double   dFrameRate;
    uint32_t SensorMode;
    uint32_t UndecFirst;
    bool32_t SupportsBinning;
    bool32_t UvSensor;
    char     AnaDaqDescription[128];
    char     BinDaqDescription[128];
    bool32_t DaqOptions;

}SETUP;
```

Introduction to the expandable SETUP structure

The SETUP structure contains the acquisition parameters, the image processing settings and all other metadata collected or used during the recording of the cine.

The camera, software or other external equipment collect a lot of data synchronous with the image acquisition such as image time, exposure, analog and binary signals and range data. The range data contains information such as camera orientation, distance to the subject and others.

Inevitably such a structure changes in time: new fields are added, others are not used anymore or require a different representation.

Having a C language structure as a repository for this information provides a simple, fast and direct access to every member but complicates the changes. To maintain the compatibility between versions, the following rules are followed:

- any structure change will not modify existing fields location
- once a new field is added it will never be deleted
- new fields are always added at the end of the structure
- old fields replaced by new ones are marked as updated
- old fields not used anymore are marked as deprecated or “to be ignored”

If a field is updated or a new field added later for a better representation (like extending our 32 bit initial representation of the FrameRate to a double dFrameRate), a reader of the file should use the latest representation of each parameter it knows about. If a cine file reader wants to be able to read older versions of the cine file, it should build an internal SETUP structure according to the latest specification it knows about. If the cine file to be read is newer and has a larger SETUP, only the known part will be read.

More common is to have an older cine file; in this case the new fields from the memory SETUP structure will be initialized in an update function, based on the existing information in the old structure from file, or with default values for the new members. How to find that a SETUP structure field is present or not in the information from an old cine file? You can do that very easy by comparing the SETUP size from the file (stored in the field SETUP.Length) with the offset of your field in the structure obtained with the macro offsetof() or FIELD_OFFSET in Windows.

The cine file writer will write the last version of the information stored in the structure. However, for backward compatibility with older readers the preferred approach is to update the older fields from the SETUP structure too. The cine files written in our applications provide this backward-forward compatibility.

The SETUP structure is detailed in Appendix 2. The SETUP Structure and Substructures. Please note the packing of the structure fields at 1 byte is different from the default setting of today compilers.

The main components of the Setup structure are:

1. Structure marker and size
2. Acquisition parameters. These fields include a subset of the camera settings at the time of recording, directly related to the image acquisition: frame rate, exposure, count of post trigger frames
3. Pixel values description. These fields refer to the bit depth of the pixel value, its representation using a BlackLevel and a WhiteLevel, pixel color based on the CFA and the option to convert higher bit depths to 8 bits.
4. Image processing options at the time of recording. These are active metadata in the RAW cine files: they can be changed and affect the displayed image although the pixels from the file remain untouched. These fields include adjustments like Brightness, Contrast, Gamma, Saturation, Hue and area filtering, cropping, border.
5. Auxiliary information stored in the cine file: time, exposure, signals, range data.
6. Other passive metadata set by the user or stored by the camera, information about the calibration, software settings for continuous recording.
7. Fields that are not used anymore but still remain in the structure for compatibility.

The meaning of the old [-100, 100] values of Bright, Contrast, Gamma

New versions of Phantom software (>693) introduced a new representation for the image processing parameters as float instead of integer values. For example, brightness is now represented as a float field with values from -1.0 to 1.0. The limit values mean a subtraction or addition with the maximum possible pixel value for that cine image. When dealing with cines written in older versions of software, integer values of these parameters should be remapped to float values. The conversion rules for brightness, contrast and gamma between the old and the new representation are illustrated below:

```
float PhfGammaI2F (int nGamma)
{
if(nGamma==34)
      return (float)(1/0.45); //gamma = 2.22
return (float)pow(10.0, nGamma/100.0);
}

int PhfGammaF2I (float fGamma)
{
if ((fGamma >= 2.215) && (fGamma < 2.225))
      return 34;
return iround(100.0*log10((double)fGamma));
}


//Contrast -100 100 is remapped to a gain= 0.1 .... 10
if (Contrast < 0)
      dGain = 1.0 + Contrast * 0.9 / 100;
else
      dGain = 1.0 + Contrast * 9.0 / 100;

//Brightness
//-100 ... 100 means adding -MaxPix/10 ... MaxPix/10
```

3.3.1 Structure Marker and Size

Mark

Will be "ST" = 0x5453 – marker for setup structure.

Length

Length of the current version of setup.

3.3.2 Acquisition Parameters

dFrameRate

High precision acquisition frame rate in frames per seconds. It is the requested frame rate represented as a double value and the camera will realize the closest possible frame rate, depending on its internal clock frequency. This parameter is normally constant during the recording. After the trigger, on certain cameras, it is possible to change dynamically the frame rate using the frame rate profile: see the fields FRPCount, FRPRate.
The actual frame rate can be computed exactly from the difference of two successive Image Times. See the time tagged block in the cine file.
If the synchronization mode is External the parameter is not used, and the image acquisitions are started by the pulses at a camera input.

Replaces: FrameRate – 32 bit integer (renamed to FrameRateInt)
FrameRate16 – Frame rate on 16 bits. It is limited to 65535 fps. If dFrameRate field is not available you have to look at the value of older fields: FrameRate, FrameRate16 in cine file. This is true for all 16 fields (Shutter16, EDRShutter16, PostTrigger16, FrameDelay16)

ShutterOff

This BOOL setting puts the shutter off to force the maximum exposure of the camera. This is useful for PIV where the camera is expected to have the shortest interval between successive exposures.

ShutterNs

Image exposure duration, in nanoseconds.
It is the requested value, camera will do the closest possible value. The exposure can change dynamically during the recording because of an active autoexposure mode. The actual exposure durations for every image acquired can be found in the exposure tagged block.

Replaces: Shutter – a value in microseconds on 32 bits
Shutter16 – a value in microseconds on 16 bits.

EDRShutterNs

EDR (Extreme Dynamic Range) is a camera operation parameter that allows the change of the relation between the light and the pixel level to non-linear. The allowed values for EDRShutterNs are from 0 (disabled) to ShutterNs - the value of exposure.

Replaces: EDRShutter - the value in microseconds
EDRShutter16 - the value in microseconds, on 16 bits

PostTrigger

Number of post trigger frames - the number of frames that the camera records after a hardware or software trigger occurs. If the partition capacity is larger than PostTrigger the recording will include a few frames recorded before the trigger. If PostTrigger is larger than capacity, all the frames recorded are posttrigger and we have a delay between the trigger and the first recorded image.

Replaces: PostTrigger16 - the value on 16 bits

FrameDelayNs

When in the external Sync, the FrameDelayNs allows to set a delay (in nanoseconds) between the external pulse and the exposure of the image.

Replaces: FrameDelay - the value in microseconds
FrameDelay16 - the value in microseconds

TrigFrame

The Phantom cameras can start the exposure for a new image based on the internal master clock of the camera, after the elapse of the period = 1/FrameRate from the start of the previous exposure. This is the normal mode with "internal" synchronization.
The camera can operate also in external sync mode where the exposure starts when an external pulse is applied to the camera Sync input. The image acquisition can also be synchronized from the IRIG time signal applied to the camera.

Values for the TrigFrame and camera operating modes:

Value	Synchronization
0	Internal
1	External
2	Locktoirig

MasterSerial

When a few cameras operate in sync external mode, it is enough to switch their sync mode to external and provide pulses at their Sync input, either from another camera or from a different source of pulses. If the pulses are absent or finish too early the camera will not acquire images and will not answer to the image requests from the computer.
MasterSerial is a software variable that tells the application this camera is synchronized externally with pulses from another Phantom camera having the specified serial. The software can inherit some slave acquisition parameters from the master camera and order the controls sent to the camera like capture and trigger to avoid the lock of the slaves in the absence of pulses.

MasterSerial value:

Value	Meaning
0	This camera is not the slave of another camera
any other value	The serial of the master - this camera is a slave and its master has the specified serial

FRPSteps

Supplementary steps in frame rate profile:

Value	Meaning
0	no frame rate profile
any other value	number of the frame rate change points

The frame rate can be changed only after the trigger and the change can be a step to another value that remains constant (flat) or can be continuously variable from a value to another value (ramp). New cameras can make a profile with up to 16 ramps while old cameras (firmware < 565) can make up to 4 flat steps. The actual frame rate can be computed during the play from the image time of the successive images.

FRPImgNr

Image number where to change the rate and/or exposure, allocated for 16 points.

FRPRate

New value for frame rate (fps).

FRPExp

New value for exposure (nanoseconds, not implemented in cameras).

FRPShape

Value	Meaning
0	flat
1	ramp

All the steps share the same shape: ramp in the cameras that support it and flat in the older cameras

AutoExposure

AutoExposure means adjusting automatically the duration of the exposure to get the specified average pixel level in the acquired images. This automatic adjustment can be programmed to stop and lock at the trigger moment
Autoexposure modes:

bit 0	Meaning
0	Disable auto exposure
1	Enable auto exposure

bit 1	Meaning
0	Lock at trigger
1	Active after trigger

bit 3	bit 2	Meaning
0	0	Only for V1 auto exposure
0	1	Average
1	0	Spot
1	1	Center weighted

AutoExpLevel

The target of the average level for autoexposure control.

AutoExpSpeed

Speed for autoexposure control algorithm.

AutoExpRect

Rectangle for autoexposure control. Makes the autoexposure sensitive only to this area.

3.3.3 Pixel Value

RealBPP

Real number of bits per pixel for this cine:

Camera	Values
8 bit cameras (v3, 4, 5, 6, 42, 43, 51, 62, 72, 9)	8
Phantom v7	8 or 12
14 bit cameras	8, 10, 12, 14
12 bit cameras	8, 10, 12 or only 12

Pixels will be stored on 8 or 16 bits in files. If RealBPP > 8 the storage will be on 16 bits. An exception to the padding to 16 bits is the packed RAW formats where 4 pixels are stored in 5 bytes (10 bits per pixel) or 2 pixels are stored in 3 bytes (12 bits per pixel).

BlackLevel

The black level in the raw pixels from camera. The values below this level are "negative" pixels or special codes reserved in the digital video standard.

WhiteLevel

The white level in the raw pixels from camera. The value above WhiteLevel is used to mark the bad pixels

fGain16_8

Allow the change of the apparent camera sensitivity after the recordings with the bitdepth > 8 bits. When converting the images back to 8 bits, for display or convert, the software can take the most significant 8 bits, the least significant 8 bits or something intermediate. To provide a fine adjustment the conversion to 8 bits will be done by multiplying the input pixel value by the factor: fGain16_8 * (2**8 / 2**bitdepth).
If fGain16_8 is 1.0, the maximum value of the input pixel is converted to 255, the maximum value of the 8 bit pixel.

Replaces: Conv8Min – Minimum value when converting to 8 bits. A first degree function is used to convert the values from 16 bits to 8 bits.
Conv8Max – Maximum value when converting to 8 bits.

CFA

The color sensors have every pixel sensitive to one color only. The Color Filter Array describes what color has every pixel of the sensor by telling the colors of the pixels in a small kernel that repeat over all sensor area.
Code for the Color Filter Array of the sensor:

Value		Meaning
CFA_NONE	= 0	gray sensor
CFA_VRI	= 1	gbrg/rggb sensor
CFA_VRIV6	= 2	bggr/grbg sensor
CFA_BAYER	= 3	gb/rg sensor
CFA_BAYERFLIP	= 4	rg/gb sensor
CFA_BAYERFLIPB	= 5	gr/gb sensor
CFA_BAYERFLIPH	= 6	bg/gr sensor

High byte carries information about color/gray heads on v6 and v6.2.

Mask Value	Meaning
0x80000000	Top Left gray
0x40000000	Top Right gray
0x20000000	Bottom Left gray
0x10000000	Bottom Right gray

UvSensor

Indicates the sensor will support UV light detection

3.3.4 Image Processing

3.3.4.1 Point Processing

WBGain

Gain adjust on R, B components for white balance at live, recorded or RAW file images. The gain for the green component is assumed 1.0.
Applying gains below 1.0 to a pixel color component would prevent that pixel to go into saturation, even if the sensor outputs the maximum possible value. To avoid this the WBGain can be normalized so the smallest value will be 1.0 and the other two are over unit.
WBGain is a metadata: the raw pixels remain unchanged and the WB adjustment will be applied at the display of the image or at the conversion to a notRAW file format.

Value	Meaning
1.0	Does nothing.
index 0	Whole image at single head cameras (v4, v5, v7...) and TopLeft head for v6, v6.2 (multihead).
index 1, 2, 3	Top Right, Bottom Left, Bottom Right for multihead.

bMetaWB

When TRUE, the pixel value does not have White Balance applied (or any other processing). When present in SETUP (from software version 671) this field is always TRUE.

WBView

Late white balance to be applied on the color interpolated cines. This field store the last white balance applied. WBView will change the values of the pixels from the interpolated file. Ignore this field in the raw file and use only WBGain.

fOffset

Applying the offset image processing will add to every pixel of the image the value fOffset multiplied by the maximum possible pixel value.

Range	[-1.0, 1.0]
Neutral	0.0
Value 1.0	Means add the maximum pixel value.

Replaces: Bright - similar but have the adjustment parameter integer and with conventional values from -100 to +100. The value 100 corresponds to an adjustment with 10% of the maximum pixel value.

fGain

Applying the gain image processing will multiply every pixel of the image by the fGain.

Range used	[0.1, 10.0]
Neutral	1.0

Replaces: Contrast - similar but have the adjustment parameter integer and with conventional values from -100 to +100. The value -100 corresponds to a gain 0.1 the value 0 means gain neutral 1.0 and the value +100 corresponds to a gain of 10.0. Between these three points the conversion is linear

fGamma

Apply a global gamma function (or green gamma).

Range	[0.1, 10.0]
Neutral	1.0

Replaces: Gamma - similar but have the adjustment parameter integer and with conventional values from -100 to +100. The value -100 corresponds to a gamma 0.1 the value 0 means gamma neutral 1.0 and the value +100 corresponds to a gamma of 10.0. Between these three points the conversion is logarithmic.

fGammaR

Per red component gamma (to be added to overall gamma stored in fGamma field).

Neutral	0

fGammaB

Per blue component gamma (to be added to overall gamma stored in fGamma field).

Neutral	0

fSaturation

Range	[0.0, 2.0]
Neutral	1.0

Replaces: Saturation - similar but has the adjustment parameter integer and with conventional values from -100 to +100. The value -100 corresponds to a complete remove of the color (conversion to gray) the value 0 means saturation neutral 1.0 and the value +100 corresponds to a maximum saturation.

fHue

Degrees and fractions of degree to rotate the color hue of every pixel of the image.

Range	[-180.0, 180.0]
Neutral	0.0

Replaces: Hue - similar but have the adjustment parameter integer

fFlare

Offset to be applied before the White Balance.

Range	[-1.0, 1.0]
Neutral	0.0
Value 1.0	Means adjust by the maximum pixel value.

fPedestalR

Offset, separate on the red component, to be applied after the Gamma.

Range	[-1.0, 1.0]
Neutral	0.0
Value 1.0	Means adjust by the maximum pixel value.

fPedestalG

Offset, separate on the green component, to be applied after the Gamma.

Range	[-1.0, 1.0]
Neutral	0.0
Value 1.0	Means adjust by the maximum pixel value.

fPedestalB

Offset, separate on the blue component, to be applied after the Gamma.

Range	[-1.0, 1.0]
Neutral	0.0
Value 1.0	Means adjust by the maximum pixel value.

fChroma

Chrominance adjustment (after gamma).

Range	[0.0, 2.0]
Neutral	1.0

ToneLabel

TonePoints

fTone

Tone curves processing builds a Look Up Table (LUT) from up to 32 x-y points and interpolating the intervals using spline curves. Beside the count and the array of points we have a label that helps identify the curve. The label is a string terminated by a byte 0. Every point is described by a set of two x, y coordinates both having values between 0.0 and 1.0.

TonePoints	[0,32]
fTone Range	[0.0, 0.0; 1.0,1.0]

UserMatrixLabel

EnableMatrices

cmUser

Point processing on the three components of a color pixel by mutiplying them by an RGB 3 x 3 matrix. The processing is enabled by the BOOL EnableMatrices and it may have a label for identification. The label is a string terminated by a byte 0.

fGainR

fGainG

fGainB

Per color channel gain adjustment.

Range	[0.0, 10.0]
Neutral	1.0

cmCalib

RGB color calibration matrix bringing camera pixels to rec 709. It includes the white balance set into the ph16 cameras using fWBTemp and fWBCc and the original factory calibration. The cine player should decompose this matrix in two components: a diagonal one with the white balance to be applied before interpolation and a normalized one to be applied after interpolation.

fWBTemp

fWBCc

White balance based on color temperature and color compensation index. Its effect is included in the cmCalib matrix.

CalibrationInfo

A string describing the original factory calibration matrix of the camera.

OpticalFilter

Optical filter matrix used to calculate cmCalib.

fToe

Controls the gamma curve in the blacks.
Neutral is 1.0.
Decreasing fToe lifts the blacks, while increasing it compresses them

LogMode

Configures the log mode.
0 - log mode disabled.
1, 2, etc. - log mode enabled.
If log mode enabled, gain, gamma, the pedestals, r/g/b gains, offset and flare are inactive.

WBType

For raw color cines, it describes how meta WB is stored.
If bit 0 is set - wb is stored in WBGain field.
If bit 1 is set - wb is stored in fWBTemp, fWBCc fields.

3.3.4.2 Area Processing

FilterCode

Image filtering is an area processing: the output pixel value depends both on its value and the values of the neighbors. It allows doing processings like sharpening, smoothing, edge detection and many others.
Predefined filter codes and kernels:

PREWITT_3x3_V = 1: A gradient filter (vertical Prewitt operator) that uses the kernel:

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

PREWITT_3x3_H = 2 A gradient filter (horizontal Prewitt operator) that uses the kernel:

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

SOBEL_3x3_V = 3 *A gradient filter (vertical Sobel operator) that uses the kernel:*

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

SOBEL_3x3_H = 4 A gradient filter (horizontal Sobel operator) that uses the kernel:

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

LAPLACIAN_3x3 = 5 A 3x3 Laplacian highpass filter that uses the kernel:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

LAPLACIAN_5x5 = 6 A 5x5 Laplacian highpass filter that uses the kernel:

$$\begin{bmatrix} -1 & -3 & -4 & -3 & -1 \\ -3 & 0 & 6 & 0 & -3 \\ -4 & 6 & 20 & 6 & -4 \\ -3 & 0 & 6 & 0 & -3 \\ -1 & -3 & -4 & -3 & -1 \end{bmatrix}$$

GAUSSIAN_3x3 = 7 A 3x3 Gaussian lowpass filter that uses the kernel A/16 where:

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

These filter coefficients correspond to a 2-dimensional Gaussian distribution with standard deviation 0.85.

GAUSSIAN_5x5 = 8 A 5x5 Gaussian lowpass filter. This filter uses the kernel A/571, where:

$$A = \begin{bmatrix} 2 & 7 & 12 & 7 & 2 \\ 7 & 31 & 52 & 31 & 7 \\ 12 & 52 & 127 & 52 & 12 \\ 7 & 31 & 52 & 31 & 7 \\ 2 & 7 & 12 & 7 & 2 \end{bmatrix}$$

These filter coefficients correspond to a 2-dimensional Gaussian distribution with standard deviation 1.0.

HIPASS_3x3 = 9 A 3x3 highpass filter that uses the kernel:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

HIPASS_5x5 = 10 A 5x5 highpass filter that uses the kernel:

$$\begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

SHARPEN_3x3 = 11 A 3x3 sharpening filter that uses the kernel:

$$(1/8) \times \begin{bmatrix} -1 & -1 & -1 \\ -1 & 16 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

USERFILTER = -1 The filter to be applied has the matrix in the UF member of this structure.

FilterParam

ImageProcessing: optional parameter, not used for common processings.

UF

User filter: a 3x3 or 5x5 user convolution filter. The structure includes the convolution matrix dimensions (3x3 or 5x5), a shift value that allows the division of the result by 2**shift, the bias value that is added at the end and the matrix itself. All the fields are 32 bit integers.

bFlipH

Flips the image horizontally.

bFlipV

Flips the image vertically.

Rotate

Rotates the image.

Value	Meaning
0	Does nothing.
+90	Counterclockwise.
-90	Clockwise.

Grid

Displays a crosshair or a grid over the image.

Value	Meaning
0	No grid.
2	Cross hair.
8	Grid with 8 intervals.

3.3.4.3 Geometry, Border, Crop

ImWidth
Image width, should be identical with BITMAPINFOHEADER.biWidth

ImHeight
Image height, should be identical with BITMAPINFOHEADER.biHeight

ImPosXAcq
Acquired image horizontal offset in sidestamped image.
When you stamp text information outside acquired image these four fields show the position of acquired image from the camera in the sidestamped image stored in file.

ImPosYAcq
Acquired image vertical offset in sidestamped image.

ImWidthAcq
Acquired image width (different value from ImWidth only in sidestamped file).

ImHeightAcq
Acquired image height (different value from ImHeight only in sidestamped file).

EnableCrop

CropRect
Enables cropping: the output image will contain only a rectangle portion of the input image. The rectangle is specified in the CropRect field.

EnableResample

ResampleWidth

ResampleHeight
Enables resampling. Resamples image to a desired output resolution specified in the fields ResampleWidth, ResampleHeight.

3.3.4.4 Image Processing History

RecBPP
It represents the bit depth of the camera when the cine was acquired. Possible values are: 8, 10, 12, and 14.
It is not necessary the same with the current bit depth used for storing the cine images. For example, a cine recorded on 12 bits and later converted to an 8bit cine file, would have this field set to 12. See also the examples below.

LowestFormatBPP

LowestFormatQ
The two fields describe the minimum/lowest format out of all the formats the images of this cine have been represented on since the moment of acquisition. This format is described by specifying a bit depth and a quality indicator.
Possible values for LowestFormatBPP are: 8, 10, 12, and 14.
Possible values for LowestFormatQ are:

Value	Meaning
0	Unpacked.
256	Packed.

Here are a few examples:

Cine Image Processing History	Lowest Format BPP	Lowest Format Q	Rec BPP
Cine recorded on 12 bits. Cine saved from camera to 12bit cine file.	12	0	12

Cine recorded on 12 bits. Cine saved from camera to 12bit cine file. Cine file converted to 8bit cine file.	8	0	12
Cine recorded on 12 bits. Cine saved to CineMag. Cine saved from CineMag to packed cine file.	10	256	12
Cine recorded on 12 bits. Cine saved to CineMag. Cine saved from CineMag to 12bit unpacked cine file.	10	256	12
Cine recorded on 8 bits. Cine saved to CineMag. Cine saved from CineMag to packed cine file.	8	0	8

3.3.5 Binary and Digital Signal Acquisition, Range Data

SigOption

Global signals options to record the maximum number of signal samples during the acquisition of an image.

Bit 0 Value	Meaning
0	It records the specified SamplesPerImage signal samples.
1	It records the maximum possible number of signal samples.

BinChannels

Number of binary channels acquired in the SAM (Signal Acquisition Module).

SamplesPerImage

Number of samples acquired per image, both binary and analog.

BinName

Names for the first 8 binary signals having maximum 10 chars/name; each string is terminated by a zero byte.

BinDaqDescription

A user defined string describing binary channel DAQ control

AnaOption

Global analog options.

bit 3	bit 2	bit 1	bit 0	Meaning
x	x	x	1	Bipolar signal.
x	x	1	x	Differential signal.
0	0	x	x	12 bit data
0	1	x	x	16 bit data
1	0	x	x	64 bit data
1	1	x	x	reserved
(All other values are undefined)				

AnaChannels

Number of analog channels used (each channel sample is stored 16 bit 2's complement).

AnaBoard

Board type.

Value	Meaning
0	none
1	DSP system kit (DSK)
2	DSP system kit plus SAM
3	Data Translation DT9802

4	Data Translation DT3010

ChOption
Per channel analog options; now bits 0...3 are used for analog gain (possible values 1, 2, 4, 8).

AnaGain
Per channel user gain correction for conversion from voltage to real measurement units.

AnaUnit
Measurement unit for analog channels: max 5 chars/name, each terminated by a zero byte.

AnaName
Channel name for the first 8 analog channels: max 10 chars/name each terminated by a zero byte.

AnaDaqDescription
A user defined string describing analog channel DAQ control

RangeCode
Range data code: select the range data format.
0 - No range data
1 - 128 bits formatted as requested by a special customer
2 - 128 bits formatted as requested by another special customer

RangeSize
Range data size per image in bytes. Range data is stored in a separate RangeData tagged block

DaqOptions
Bit Field used to define additional DAQ configuration/Status information

3.3.6 User and Camera Metadata

This section contains user comments and information about the camera configuration at the recording time.

Description
User description or comments (size: 4096 characters). String terminated by a byte 0.

Replaces: DescriptionOld allocated for only 120 characters

CineName
Cine name (or reel name). String terminated by a byte 0.

LensDescription
Text mentioning the producer, the model, the focal range, etc. String terminated by a byte 0.

LensAperture
Aperture f number.

LensFocusDistance
Distance in meters to which the objects are in focus. Not available from Canon motorized lens.

LensFocalLength
Current focal length (zoom factor).

MCCnt
Camera memory partition (segment) count. Maximum count is 64

MCPercent

Percentage of memory reserved for every partition.

Decimation
Factor to reduce the frame rate when sending the images to Image Cube external memory by fiber.

Sensor
Camera sensor code.

SoundDest
Sound device.

Value	Meaning
0	None.
1	Speaker.
2	Sound board.

Rising Edge
The active edge of the trigger signal.

Value	Meaning
TRUE	Rising.
FALSE	Falling.

FilterTime
The time constant to filter the spurious trigger pulses.

LongReady
If TRUE the Ready signal is 1 from the start to the end of recording. Otherwise the Ready falls to 0 at Trigger. It is needed for signal acquisition.

CameraVersion
The version of camera hardware

FirmwareVersion
Firmware version.

SoftwareVersion
Phantom software version.

Serial
Camera serial number. For Firewire cameras you have a translated value here: `factory serial + 0x58000`

HeadSerial
Head serials for Ethernet multihead cameras (v6.2). When multiple heads are saved in a file, the serials for existing heads are not zero. When one head is saved in a file its serial is in `HeadSerial[0]` and the other head serials are `0xFFFFFFFF`.

GpsInfo
Position and status info as received from a GPS receiver at the time the cine recording ended.

RecordingTimeZone
The time zone active during the recording of the cine (in seconds, negative in Eastern hemisphere, positive in the Western hemisphere). All absolute time stored in a cine file is UTC (or GMT). To get the local time at the recording place you should subtract this value from the TIME64.seconds.

TrigTC
Trigger frame SMPTE time code and user bits.

fPbRate
Video playback rate (fps) active when the cine was captured.

fTcRate
Playback rate (fps) used for generating SMPTE time code.

Uuid
Unique cine identifier generated by the camera this cine has been recorded with.

CreatedBy
The name of the application that created this cine.

CameraModel

Camera model string

MagSerial
The serial of the magazine where the cine was recorded or stored (if any, null otherwise)

CSSerial
Cine Save serial: The serial of the device (camera, cine station) used to save the cine to file

fDecimation
Decimation coefficient employed when this cine was saved to file

SupportsBinning
Indicates camera supports binning (not the current state of binning)

SensorMode
Sensor configuration (used for Flex 4K, VEO 4K, v2640 and TMX, cameras only)

Flex 4K / VEO 4K	
bit 4	Meaning
0	Rolling shutter
1	Global shutter
(All other bits are don't care)	

v2640 / v1840, TMX				
bit 3	bit 2	bit 1	bit 0	Meaning
0	0	0	0	Standard
0	0	1	0	Standard Binning
0	1	0	1	High Speed
0	1	1	1	High Speed Binning
1	1	0	1	Bright Field
(All other values are undefined)				

UndecFirst
First frame number selected by user in the save dialog, before decimation was applied the image content will remain first in the decimated cine too.

3.3.7 Calibration Information

This section provides information about the last calibration done on the camera. It is intended for debugging or internal use in Vision Research.

BlackCalSVer
Software Version used for Black Reference.

WhiteCalSVer
Software Version used for White Calibration.

GrayCalSVer
Software Version used for Gray Calibration.

CICalib
Last calibration was done on the resolution of Current Image (CSR, current session reference, not on full sensor).
This cine or this stg is the result of a current image calibration. The last calibration was:

Mask Value	Meaning
1	BlackRef
2	WhiteCalib
4	GrayCheck

Last Calibration was done with these 6 acquisition parameters:

CalibWidth
Width.

CalibHeight
Height.

CalibRate
Frame rate

CalibExp
Exposure duration

CalibEDR
EDR

CalibTemp
Sensor temperature.

3.3.8 Continuous Recording

This section provides a persistent storage of the parameters for the software Continuous Recording mode. The values are not meaningful for the cine player.

lFirstImage
Range of images for continuous recording: first image.

dwImageCount
Range of images for continuous recording: image count; used also for signal recording.

nQFactor
Quality factor – for saving compressed files in continuous recording.

Range	[2, 255]

wCineFileType
Cine file type – for continuous recording.

szCinePath
4 paths to save cine files – for continuous recording. After upgrading to Win32 these strings still remained 65 bytes long each. `GetShortPathName` is used for the filenames saved here. Strings terminated by a byte 0.

bStampTime
The time to be stamped on images in continuous recording.

Value	Meaning
1	Absolute time.
3	Time from trigger.

3.3.9 Not Used Anymore

Fields named **Res**... are old fields no longer used**.** Their names were modified in order to be easier ignored.

3.4. The Tagged Information Blocks

The fixed structures have direct access to the fields and are very easy and fast to use. To allow future expansions or optional data we added tagged blocks to the cine file format.

The tagged blocks are placed between the SETUP structure and the array of pointers to images. They are present in the file if

(`OffSetup` + `SETUP.Length`) < `OffImageOffsets.`

The size of the SETUP structure is in the *Length* field.

The structure of a tagged block is:

```
    uint32_t BlockSize;
    uint16_t Type;
    uint16_t Reserved;
    uint8_t  Data[BlockSize-8];
```

3.4.1 Analog and Digital Signals Tagged Block

Type = 1000 (0x3e8)

The signals are stored for all recorded image (including those not saved in the file). This is outdated and it is not used anymore. It was used with the Phantom v3 camera and SAM 1, SAM 2 signal acquisition modules. Use the new BinSig and AnaSig blocks.

3.4.2 Image Time Tagged Block

Type = 1001 (0x3e9)

Every element of the array is a TIME64 structure (32.32). The time is stored for each recorded image, the count of time items is *TotalImageCount* (even if you saved only a smaller range of images: *ImageCount*). If *BlockSize* is bigger than the size of this time array (Phantom version 477 or more recently) it also contains the exposure length for every image, stored as an array of DWORDs of fractions of second (similar to fractions field in TIME64 structure). This block is outdated and should not be used. Use “Time only block” and “Exposure only block” instead.

3.4.3 Time Only Block

Type = 1002 (0x3ea)

Every element of the array is a TIME64 structure (32.32). The time is stored only for the images saved in this file; the count of time items is *ImageCount* (even if you recorded in camera a larger range – *TotalImageCount*).

3.4.4 Exposure Only Block

Type = 1003 (0x3eb).

This block is needed because the exposure length can be different for different images (for example when using Autoexposure). Every element of the array is a uint32_t that represents a fixed point 0.32 number. You have to divide it by 2**32 to get the real exposure in seconds. The exposures are stored only for the images saved in this file; the count of exposure items is *ImageCount*.

3.4.5 Range Data Block

Type = 1004 (0x3ec).

This block will contain information about camera orientation and distance to the subject. There are SETUP.RangeSize bytes per image and their meaning is described by SETUP.RangeCode and is customer dependent. The standard cine viewer should skip this block.

3.4.6 BinSig Block

Type = 1005 (0x3ed).

It stores binary signals acquired with the SAM3 module, multichannel and multisample per image. The signals are stored 8 samples per byte; only for the images stored in the file.

Information about the number of channels and samples per image are stored in the SETUP structure.

3.4.7 AnaSig Block

Type = 1006 (0x3ee).

It stores analog signals acquired with the SAM3 module, multichannel and multisample per image. The signals are stored at 16 bits per sample; only for the images stored in the file.

Information about the number of channels and samples per image are stored in the SETUP structure.

3.4.8 TimeCode Block

Type = 1007 (0x3ef).

It stores the time code for every image based on the trigger time code and the time code frequency, both read from the camera. The format to store the time code for an image is similar to SMPTE 12M - 1999 and is described in the TC structure.

3.4.9 Array of Image Offsets

It contains the offsets to each image of the recorded cine.

```
int64_t pImage[ImageCount];
```

pImage

It used to be uint32_t in Cine v0.

It represents the position in file of every saved image related to the beginning of file. This provides direct access to the image pixels. The size of these offsets (32 or 64 bits) is the main difference between version 0 and 1 of the cine file format. If you want to support both version 0 and 1 you have to identify the format version in CINEFILEHEADER structure – Version field – and expand the 32 bits to 64 bits, after you read them from file.

Here is an example of an array of offsets for a 25 image file (see the details in Appendix 3). The order of bytes is little endian and there are 8 bytes for each value. Here, the first image starts at offset 0X189C and the last image starts at offset 0X15FA95C.

```
000017D0                9C 18 00 00  00 00 00 00  A4 BE 0E 00
000017E0  00 00 00 00   AC 64 1D 00  00 00 00 00  B4 0A 2C 00
000017F0  00 00 00 00   BC B0 3A 00  00 00 00 00  C4 56 49 00
00001800  00 00 00 00   CC FC 57 00  00 00 00 00  D4 A2 66 00
00001810  00 00 00 00   DC 48 75 00  00 00 00 00  E4 EE 83 00
00001820  00 00 00 00   EC 94 92 00  00 00 00 00  F4 3A A1 00
00001830  00 00 00 00   FC E0 AF 00  00 00 00 00  04 87 BE 00
```

```
00001840  00 00 00 00  0C 2D CD 00  00 00 00 00  14 D3 DB 00
00001850  00 00 00 00  1C 79 EA 00  00 00 00 00  24 1F F9 00
00001860  00 00 00 00  2C C5 07 01  00 00 00 00  34 6B 16 01
00001870  00 00 00 00  3C 11 25 01  00 00 00 00  44 B7 33 01
00001880  00 00 00 00  4C 5D 42 01  00 00 00 00  54 03 51 01
00001890  00 00 00 00  5C A9 5F 01  00 00 00 00
```

3.5. The Image Object

It contains two types of information: the annotation data and the image pixels values.
The annotation data contains at least the annotation size and the image size. If the image is not compressed its size can be estimated exactly from the header information but if the image is compressed the size cannot be computed so it is useful to have image size information for allocation purposes.

3.5.1 The Annotation Data

```
uint32_t AnnotationSize;
uint8_t  Annotation[AnnotationSize - 8];
uint32_t ImageSize;
```

AnnotationSize
Total length of the annotation (*AnnotationSize* included).

Annotation
The use and the structure of this field is not defined, except the last uint32_t that should be always the image size. It can be used to store any auxiliar information related to the image or other annotations created during the processing. The Annotation information can be absent, but the *AnnotationSize* and the *ImageSize* are always present.

ImageSize
The final uint32_t of the annotation is always the pixel array size.

Here is an example of void annotation bytes from the file described in the appendix. It uses the Intel little endian order of bytes:

```
00001890                                          08 00 00 00
000018A0  00 A6 0E 00
```

Annotations include only the image size. Annotation size is 8 and image size is 0XEA600 that is 960 000. The image resolution is 800x600 and we have 2 bytes per pixel; 800*600*2 =960000. The file is RAW (does not have the colors interpolated); this is the reason for the 2 bytes per pixel. An interpolated 16 bit cine would have 6 bytes / pixel.

3.5.2 Pixel Array

The images produced by the Phantom cameras have the Width multiple of 16 pixels so there is no padding needed.
Uncompressed gray images contain the actual gray level as a pixel value. Uncompressed color images have the color component stored: one per pixel in the uninterpolated file and the B, G, R set in the interpolated file.
The values can be represented on 8 bit (values from 0 to 255) or on 16 bit. In the case of 16 bit files we may have the real depth of 10, 12, 14 bits and the corresponding value ranges: [0, 1023], [0, 4095], [0, 16383]. The values stored in the cine files are not left aligned, they are integer values stored as read from the sensor.

```
uint8_t or uint16_t pixels [biWidth * biHeight];
or
uint8_t or uint16_t pixels [3 * biWidth * biHeight]
```

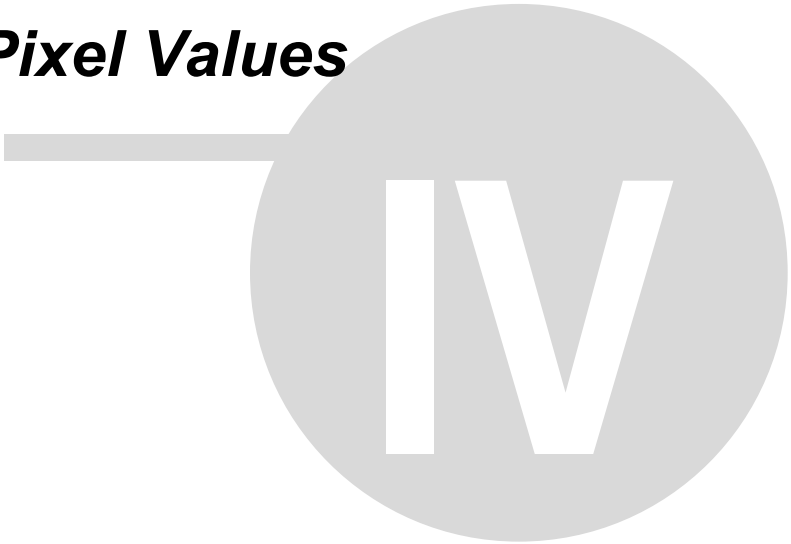
pixels

Uncompressed gray or RAW files contain the value of the pixels on 8 or 16 bits (uint8_t or uint16_t).
If biBitCount is 24 or 48 (color image) the array is three times larger and the values stored are the color components of each pixel.
The order of components for the color pixels (interpolated or from a color system with 3 cameras) is BGR, compatible with the BITMAP from Windows. Below is an example of Pixels from the cine file presented in the Appendix.

```
000018A0                  C0 36 80 2F  C0 36 80 2F  C0 36 80 2F
000018B0  C0 36 80 2F  C0 36 80 2F  00 37 C0 2F  00 37 C0 2F
```

The pixels are acquired on 14 bits, the values are from 0 to 16383 (0X3FFF). The first pixel is the bottom left pixel of the image and has the value 0x36C0 (14016) that is 85.55% from the maximum value. The CFA for the sensor is BAYER (gb / rg), this is a red pixel. The next pixel is green and has the value 0X2F80; this set of r, g pixels repeats on this bottom row of the image; we have a uniform, almost red area in the image.

Part IV Pixel Values



4. Pixel Values

The usual "linear" representation of the pixel values is with the full black pixels at zero and with the full white or saturated pixels at the maximum pixel value.
In some applications it is useful to be able to represent pixel having small negative values; to do that it is assumed the black pixels are represented by a small positive value which is stored in our SETUP structure in the field SETUP.BlackLevel. The values below the BlackLevel can be used to represent the "negative" pixels or other information. We added a SETUP.WhiteLevel too that is a little bit below the maximum integer for the selected bit depth of the camera. Values above WhiteLevel are reserved for marking bad pixels.

If your SETUP structure has a non-zero BlackLevel or a WhiteLevel different from the $2^{BitDepth} - 1$ you should normalize the value of the pixels before doing any image processing like applying a gain. This normalization is a linear transform that brings the pixels having the value BlackLevel or below to 0 and the value WhiteLevel or above to the MaximumValue at the bit depth.

The expected values for the BlackLevel in the Phantom RAW cine files are 4, 16, 64, 256 (for 8, 10, 12, 14 bit depths) The WhiteLevel in the same files could be 254, 1016, 4064, 16256.

On the packed 10 bits, because of applying the gamma function, the levels are at 64 and 1015.

Part V Types of Cine Files

5. Types of Cine Files

5.1. Raw Cine Files

The fastest save of the recordings from the camera can be done using the raw pixels read from sensor. The size of a RAW (uninterpolated color) image is identical to the size of the gray image with the same resolution and the interpolation of the colors will be delayed to the view time. The information about the sensor Color Filter Array (CFA) is needed and has to be stored in the cine file; it is available in the CFA field of the SETUP structure.

An uninterpolated (RAW) cine has the Compression field of the CINEFILEHEADER equal to 2. The CFA code stored in the SETUP structure can have the following values:

Value		Meaning
CFA_NONE	= 0	gray sensor
CFA_VRI	= 1	gbrg/rggb sensor
CFA_VRIV6	= 2	bggr/grbg sensor
CFA_BAYER	= 3	gb/rg sensor
CFA_BAYERFLIP	= 4	rg/gb sensor
CFA_BAYERFLIPB	= 5	gr/gb sensor
CFA_BAYERFLIPH	= 6	bg/gr sensor

A Bayer CFA means the top left pixel is sensitive only to the green color, the second on horizontal is sensitive to blue and the next row starts with a red and green pixel. So, the image is a mosaic of R, G and B pixels like this:

G B G B G B G B G B G B .... G B G B
R G R G R G R G R G R G .... R G R G
G B G B G B G B G B G B .... G B G B
R G R G R G R G R G R G .... R G R G
G B G B G B G B G B G B .... G B G B
R G R G R G R G R G R G .... R G R G
……
G B G B G B G B G B G B .... G B G B
R G R G R G R G R G R G .... R G R G

The above description is for the image displayed on the screen; the pixels stored in the memory or cine file begin with the left of bottom row. It is the task of the cine file reader to produce the standard RGB image from this information and optionally apply the image processing that were preferred at recording. To interpolate the colors or “dimoraic” the images you can use an algorithm or a library from the internet or use the functions from the PhInt library from the Phantom SDK.

5.1.1 Standard Raw Cine Files

The pixels are stored as read from the sensor; in case of color cameras we have two color components missing. The bad pixels are not repaired in the current Phantom cameras; it is a task for the software that writes the unpacked raw file to repair the bad pixels.

If the bit depth is larger than 8, (10, 12 or 14 bits) the values are stored in the file as 16 bits uint16_t, little endian and are padded to the left (most significant bits) with 0.

5.1.2 Packed Raw Cine Files

In the standard approach any pixel with a bit depth larger than 8 is stored in a separate 16 bit WORD.

5.1.3 P10 - Packed 10 bit with gamma

In case of 10 bit pixel, the information can be stored more efficiently by giving up to the padding to 16 bits and storing 4 pixels in 5 bytes.

Byte 0:	P0.9	P0.8	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2
Byte 1:	P0.1	P0.0	P1.9	P1.8	P1.7	P1.6	P1.5	P1.4
Byte 2:	P1.3	P1.2	P1.1	P1.0	P2.9	P2.8	P2.7	P2.6
Byte 3:	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	P3.9	P3.8
Byte 4:	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0

Four packed pixels P0, P1, P2 and P3 are stored in 5 bytes: Byte0, Byte1, Byte2, Byte3 and Byte4 (Byte0 is at the lowest memory address and P0.9 is bit 9 of pixel 0, the most significant bit of pixel 0).
Packed camera pixels are used when the cine is stored in the CineMag or when reading images through the 10g Ethernet adaptors.

Reducing the bit depth from 14 or 12 bit RAW linear as read from sensor to 10 bits means losing some dynamic range. To reduce the noise in the result image a compandor - expander scheme was applied. The first function applied is the gamma function with the adjustments from the Rec. 709 (gamma=2.2). It has the advantage that the data stored in the CineMag can be streamed directly on the HD-SDI digital video channel.

The inverse function should be applied at the linearization. Lookup table for this conversion is provided in Appendix 1.

To allow the fastest possible save of the files after the recording, the pixel stream is not touched at all during the save. This means the bad pixels are not repaired and the order of the packed pixels starts in the top left corner.

5.1.4 P12L - Packed 12 bit Linear

P12L is a lossless packing for recording bit depths up to 12 bit. Pixel values are stored as read from the analog to digital converters.

Two pixels P0 and P1 are stored in 3 bytes B0 and B1 (B0 is the lowest memory address). In the following representation, P0.11 is bit 11 of pixel 0 – the most significant bit of pixel 0.

Byte 0:	P0.11	P0.10	P0.9	P0.8	P0.7	P0.6	P0.5	P0.4
Byte 1:	P0.3	P0.2	P0.1	P0.0	P1.11	P1.10	P1.9	P1.8
Byte 2:	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0

5.2. Interpolated Cine Files

The color interpolated files have all the information needed to display the images already prepared in the file. The bad pixels were repaired, the image processing was applied, and every pixel has the RGB information stored in the file. In the absence of image processing and color interpolation, the display of the image can be faster, but the stream of data and the size of the file is three times larger.

If the SETUP.CFA = 0 we have a monochrome (gray) file that does not need color interpolation. The monochrome files can be on 8 or 16 bits depending on BITMAPINFOHEADER.biBitCount.

5.3. Compressed Cine Files

The compressed cine files have the extension .cci and the Compression field of the CINEFILEHEADER equal to 1. The compression is based on a proprietary algorithm from Leadtools so the cci files decompression can be done only by using our SDK and Leadtools libraries. You need to buy a license from Leadtools to provide support in your code for the cci files.

Part VI Access to Pixels

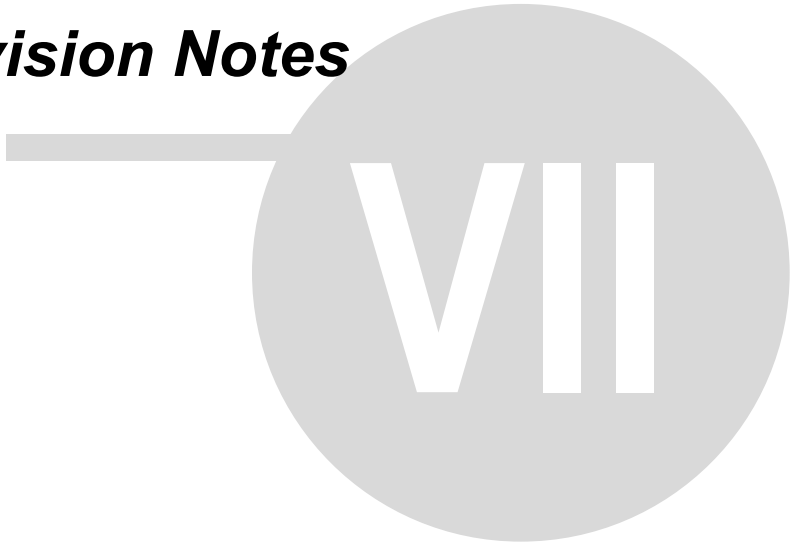
6. Access to Pixels

Here is an example of how you can get access to the images contained in a gray cine file.

1. Check the file type (the "CI" marker).
2. Use *OffImageHeader* and look for the fields *biWidth*, *biHeight* of the BITMAPINFOHEADER structure to get the image dimensions.
3. Use *OffImageOffsets* to access the table with pointers to the images.
4. The first pointer (64 bits) in the table corresponds to the first image stored in this file (*FirstImageNo*). Select the pointer to the image you want in *pImage[YourImage-FirstImageNo]*.
5. Access the image object and skip the annotation using its length stored in the first uint32_t.
6. You are now at the beginning of the pixel array for image *YourImage*.

Repeat the steps 3-6 to access other images in the file.

Part VII Revision Notes



7. Revision Notes

1992...2003

The cine format was continuously extended by adding new fields to the SETUP structure, in which all the parameters of a recording are stored. The compatibility was maintained, both forward and backward between different versions of Phantom cameras. The Version field of the header remained 0 all this time.

November 1, 1997 (Phantom software version 235)

This version includes the first interface to an IRIG board. The *TriggerTime* and *TriggerTimeExt* from the CINEFILEHEADER were replaced by a TIME64 structure. The cines recorded with Phantom software with versions older than 235 had a reversed order of time components (32 bits for seconds followed by 32 bits for fractions of second) and the *TriggerTimeExt* was always 0. If you read *TriggerTime.seconds* = 0 from an old cine file, you have to read *TriggerTime.fractions* as being the number of seconds starting from Jan 1, 1970.

IRIG time for every image of the cine is stored in a tagged block if the selected time annotation was an IRIG board.

1997

Color cines were created with the same structure.
The BITMAPINFOHEADER field *biBitCount* = 24 bits per pixel and there is no pallete.

November 18, 1998 (Phantom software version 301)

Phantom application was upgraded to 32 bits. SETUP fields remained unchanged except for a few data types: INT became INT16, BOOL became WORD etc. The structure members alignment should be set to 1, at least for the SETUP structure.

February 3, 2000 (Phantom software version 424)

For the color v4 camera a new format is available: uninterpolated color cine file. The Compression field in the CINEFILEHEADER is 2, *biBitCount* = 8, and the palette is absent. A dll library is available to interpolate the color.

July 3, 2000 (Phantom software version 459)

Some of the 16 bits fields were upgraded to 32 bits. The old fields were renamed, getting a "*16"* termination (e.g. FrameRate16) and they still carry the information if possible (value less than 65536). You can use the new fields if the SETUP.Length is greater than FIELD_OFFSET(FrameRate). This means the cine was saved by a Phantom software version that wrote the new fields upgraded to 32 bits.

December 15, 2000 (Phantom software version 477)

When saved or converted to a set of image files or to an avi format, a cine header file having the extension .chd is automatically written. It contains an exact copy of the cine file header, BITMAPINFOHEADER and the setup and tagged blocks described above.

The time block contains exposure duration information for each image. The size of the time block tells you whether the exposure information is present or not.

April 22, 2003 (Phantom software version 600)

This represents version 1 of cine file format on 64 bits.

Starting from version 600, Phantom software is able to write and read files bigger than 4 GB. The operating system has to support files bigger than 4GB – when using Windows, the file system must be NTFS.

The main change is the extension of the image pointers to 64 bits. All other file pointers remained at 32 bits. This means the *array of pointers to images* should be declared as:

```
int64_t pImage[ImageCount];   //position in file of every image
```

The Version field in the CINEFILEHEADER is 1. (It was 0 before version 600). When reading an old cine file having Version 0, the image pointers have to be expanded to 64 bits.

The palette of the BITMAPINFO is not stored in the header of the cine v1 file. Only BITMAPINFOHEADER is stored even for gray images. A cine reader has to add its own gray palette if needed.

In the v1 cine the information from tagged blocks is stored only for the range of images that are saved in the file, not for the full range of acquired images.

New tagged blocks were added: TimeOnlyBlock and ExposureOnlyBlock.

August 28, 2003 (Phantom software version 603)

Support for 16 bpp monochrome images and 48 bpp color images. The values 16 and 48 of the *biBitCount* field in the image header describe these types of cine files. The real bit depth of the camera can be between 8 and 16 bits, e.g. Phantom v7 has 12 bits. The pixel value is stored “as it is”; it is not left aligned to 16 bits. This means the pixels from a v7 camera configured to record on 16 bits are stored as 16 bit integers with values from 0 to 4095. The *RealBPP* field of the SETUP structure has to be used to find the real bit depth and the maximum value of the pixels. If the *Length* field of SETUP is smaller than the offset of this field, the value of *RealBPP* should be considered 8. The byte order is little endian (Intel – least significant byte first) and the color order is BGR.

The value *biBitCount* = 16 has a different meaning in Windows (color image 5:6:5) but this is not usually a problem since the bitmap has anyway to be converted to 8 or 24 bits before being displayed. The PhInt library provides the support for color interpolation and image processing for all bit depths.

June 1, 2007 (Phantom software version 645)

The default extension for the uncompressed cine files changed from .cin to .cine

January 26, 2009 (Phantom software version 668)

The pixel value in the RAW file does not include anymore the application of the White Balance. The White Balance coefficients from SETUP.WBGain are now active metadata and should be applied at view or convert.

May 22, 2009 (Phantom software version 671)
Our application started to write optionally a new variant of the RAW file: "10 bits Packed RAW" when the source images from the camera are packed. This file format is written faster and occupies less space on the disc.
BlackLevel and WhiteLevel information was added to the SETUP structure.

April 8, 2011 (Phantom software version 701)
We started to provide the time code stamping of the images (SMPTE 12M - 1999). The time code is stored in a new tagged block: TimeCodeBlock.

June 2011 (Phantom software version 705)

September 2011 (Phantom software version 709)

March 9, 2012 (Phantom software version 719)

March 16, 2012 (Phantom software version 720)

January 24, 2013 (Phantom software version 731)

November 19, 2013 (Phantom software version 742)

April 29, 2014 (Phantom software version 745)

July 5, 2015 (Phantom software version 751)
Added (double) dFrameRate to replace (int32)FrameRate

December 2017 (Phantom software version 771)
Added SensorMode for the Phantom 2640, VEO 4K and Flex4k Global shutter cameras

February 2021 (Phantom software version 800)
Added UndecFirst

October 2022 (Phantom software version 803)
Added SupportsBinning
Added UvSensor
Added AnaDaqDescription
Added BinDaqDescription

January 2023 (Phantom software version 804)
Added DaqOptions

Part VIII Appendices

8. Appendices

8.1. Appendix 1. LUT for Conversion from 10 Bits Packed to the 12 bit Linear

```
int LinLUT[1024] =
{
   2,   5,   6,   7,   8,   9,  10,  11,  12,  13,  14,  15,  16,  17,  17,  18,
  19,  20,  21,  22,  23,  24,  25,  26,  27,  28,  29,  30,  31,  32,  33,  33,
  34,  35,  36,  37,  38,  39,  40,  41,  42,  43,  44,  45,  46,  47,  48,  48,
  49,  50,  51,  52,  53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  63,
  64,  65,  66,  67,  68,  69,  70,  71,  72,  73,  74,  75,  76,  77,  78,  79,
  79,  80,  81,  82,  83,  84,  85,  86,  87,  88,  89,  90,  91,  92,  93,  94,
  94,  95,  96,  97,  98,  99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109,
 110, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124,
 125, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 137, 138,
 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154,
 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 167, 168, 169, 170, 171, 172,
 173, 175, 176, 177, 178, 179, 181, 182, 183, 184, 186, 187, 188, 189, 191, 192,
 193, 194, 196, 197, 198, 200, 201, 202, 204, 205, 206, 208, 209, 210, 212, 213,
 215, 216, 217, 219, 220, 222, 223, 225, 226, 227, 229, 230, 232, 233, 235, 236,
 238, 239, 241, 242, 244, 245, 247, 249, 250, 252, 253, 255, 257, 258, 260, 261,
 263, 265, 266, 268, 270, 271, 273, 275, 276, 278, 280, 281, 283, 285, 287, 288,
 290, 292, 294, 295, 297, 299, 301, 302, 304, 306, 308, 310, 312, 313, 315, 317,
 319, 321, 323, 325, 327, 328, 330, 332, 334, 336, 338, 340, 342, 344, 346, 348,
 350, 352, 354, 356, 358, 360, 362, 364, 366, 368, 370, 372, 374, 377, 379, 381,
 383, 385, 387, 389, 391, 394, 396, 398, 400, 402, 404, 407, 409, 411, 413, 416,
 418, 420, 422, 425, 427, 429, 431, 434, 436, 438, 441, 443, 445, 448, 450, 452,
 455, 457, 459, 462, 464, 467, 469, 472, 474, 476, 479, 481, 484, 486, 489, 491,
 494, 496, 499, 501, 504, 506, 509, 511, 514, 517, 519, 522, 524, 527, 529, 532,
 535, 537, 540, 543, 545, 548, 551, 553, 556, 559, 561, 564, 567, 570, 572, 575,
 578, 581, 583, 586, 589, 592, 594, 597, 600, 603, 606, 609, 611, 614, 617, 620,
 623, 626, 629, 632, 635, 637, 640, 643, 646, 649, 652, 655, 658, 661, 664, 667,
 670, 673, 676, 679, 682, 685, 688, 691, 694, 698, 701, 704, 707, 710, 713, 716,
 719, 722, 726, 729, 732, 735, 738, 742, 745, 748, 751, 754, 758, 761, 764, 767,
 771, 774, 777, 781, 784, 787, 790, 794, 797, 800, 804, 807, 811, 814, 817, 821,
 824, 828, 831, 834, 838, 841, 845, 848, 852, 855, 859, 862, 866, 869, 873, 876,
 880, 883, 887, 890, 894, 898, 901, 905, 908, 912, 916, 919, 923, 927, 930, 934,
 938, 941, 945, 949, 952, 956, 960, 964, 967, 971, 975, 979, 982, 986, 990, 994,
 998,1001,1005,1009,1013,1017,1021,1025,1028,1032,1036,1040,1044,1048,1052,1056,
1060,1064,1068,1072,1076,1080,1084,1088,1092,1096,1100,1104,1108,1112,1116,1120,
1124,1128,1132,1137,1141,1145,1149,1153,1157,1162,1166,1170,1174,1178,1183,1187,
1191,1195,1200,1204,1208,1212,1217,1221,1225,1230,1234,1238,1243,1247,1251,1256,
1260,1264,1269,1273,1278,1282,1287,1291,1295,1300,1304,1309,1313,1318,1322,1327,
1331,1336,1340,1345,1350,1354,1359,1363,1368,1372,1377,1382,1386,1391,1396,1400,
1405,1410,1414,1419,1424,1428,1433,1438,1443,1447,1452,1457,1462,1466,1471,1476,
1481,1486,1490,1495,1500,1505,1510,1515,1520,1524,1529,1534,1539,1544,1549,1554,
1559,1564,1569,1574,1579,1584,1589,1594,1599,1604,1609,1614,1619,1624,1629,1634,
1639,1644,1649,1655,1660,1665,1670,1675,1680,1685,1691,1696,1701,1706,1711,1717,
1722,1727,1732,1738,1743,1748,1753,1759,1764,1769,1775,1780,1785,1791,1796,1801,
1807,1812,1818,1823,1828,1834,1839,1845,1850,1856,1861,1867,1872,1878,1883,1889,
1894,1900,1905,1911,1916,1922,1927,1933,1939,1944,1950,1956,1961,1967,1972,1978,
1984,1989,1995,2001,2007,2012,2018,2024,2030,2035,2041,2047,2053,2058,2064,2070,
2076,2082,2087,2093,2099,2105,2111,2117,2123,2129,2135,2140,2146,2152,2158,2164,
2170,2176,2182,2188,2194,2200,2206,2212,2218,2224,2231,2237,2243,2249,2255,2261,
2267,2273,2279,2286,2292,2298,2304,2310,2317,2323,2329,2335,2341,2348,2354,2360,
2366,2373,2379,2385,2392,2398,2404,2411,2417,2423,2430,2436,2443,2449,2455,2462,
2468,2475,2481,2488,2494,2501,2507,2514,2520,2527,2533,2540,2546,2553,2559,2566,
2572,2579,2586,2592,2599,2605,2612,2619,2625,2632,2639,2645,2652,2659,2666,2672,
2679,2686,2693,2699,2706,2713,2720,2726,2733,2740,2747,2754,2761,2767,2774,2781,
2788,2795,2802,2809,2816,2823,2830,2837,2844,2850,2857,2864,2871,2878,2885,2893,
2900,2907,2914,2921,2928,2935,2942,2949,2956,2963,2970,2978,2985,2992,2999,3006,
3013,3021,3028,3035,3042,3049,3057,3064,3071,3078,3086,3093,3100,3108,3115,3122,
3130,3137,3144,3152,3159,3166,3174,3181,3189,3196,3204,3211,3218,3226,3233,3241,
3248,3256,3263,3271,3278,3286,3294,3301,3309,3316,3324,3331,3339,3347,3354,3362,
3370,3377,3385,3393,3400,3408,3416,3423,3431,3439,3447,3454,3462,3470,3478,3486,
3493,3501,3509,3517,3525,3533,3540,3548,3556,3564,3572,3580,3588,3596,3604,3612,
3620,3628,3636,3644,3652,3660,3668,3676,3684,3692,3700,3708,3716,3724,3732,3740,
3749,3757,3765,3773,3781,3789,3798,3806,3814,3822,3830,3839,3847,3855,3863,3872,
3880,3888,3897,3905,3913,3922,3930,3938,3947,3955,3963,3972,3980,3989,3997,4006,
```

```
4014,4022,4031,4039,4048,4056,4064,4095,4095,4095,4095,4095,4095,4095,4095,4095
};
```

You can copy - paste this table to your application. Black level is at 64 and white level at 1014 in the 10 bits packed representation. In the 12 bits representation the levels are 64 and 4064.

8.2. Appendix 2. The SETUP Structure and Substructures

```
#ifdef __cplusplus
extern "C"
{
#endif

#ifdef __STDC_VERSION__      // Check C99 support
#if __STDC_VERSION__ >= 199901L
#define _C99SUPPORTED_
#endif
#endif
#ifndef _C99SUPPORTED_
    //define the integer types with known size according to C99 and stdint.h
    typedef               char int8_t;
    typedef unsigned      char uint8_t;
    typedef          short int int16_t;
    typedef unsigned short int uint16_t;
    typedef                int int32_t;
    typedef unsigned       int uint32_t;
    typedef             __int64 int64_t;
    typedef unsigned    __int64 uint64_t;
#endif
typedef int bool32_t;

/*****************************************************************************/
#if !defined(_TIMEDEFINED_)
#define _TIMEDEFINED_

//A format for small intervals of time: [250 picosecond ... 1 second)
//It is fixed point 0.32 or, in other words, the time in seconds is
//stored multiplied by 4Gig i.e.  4294967296.0 and rounded to int.
typedef uint32_t FRACTIONS, *PFRACTIONS;

//The absolute time format used in PC software is TIME64
typedef struct tagTIME64    // A compact format for time 64 bits
                            // fixed point (32.32 seconds)
{
    FRACTIONS fractions;    // Fractions of seconds
                            // (resolution 1/4Gig i.e.  cca. 1/4 ns)
                            // The fractions of the second are stored here
                            // multiplied by 2**32. Least significant 2 bits
                            // store info about IRIG synchronization
                            // bit0 = 0 IRIG synchronized
                            // bit0 = 1 not synchronized
                            // bit1 = 0 Event input=0 (short to ground)
                            // bit1 = 1 Event input=1 (open)
    uint32_t seconds;       // Seconds from Jan 1 1970, compatible with the C
                            // library routines
                            // (max year: 2038 signed, 2106 unsigned)
                            // VS2005 changed the default time_t to 64 bits;
                            // here we have to maintain the 32 bits size to
                            // remain compatible with the stored file format
                            // and the public interfaces
} TIME64, *PTIME64;
/*****************************************************************************/

//Time code according to the standard SMPTE 12M-1999
typedef struct tagTC
{
    uint8_t framesU:4;        // Units of frames
    uint8_t framesT:2;        // Tens of frames
    uint8_t dropFrameFlag:1;  // Dropframe flag
    uint8_t colorFrameFlag:1; // Colorframe flag
    uint8_t secondsU:4;       // Units of seconds
    uint8_t secondsT:3;       // Tens of seconds
    uint8_t flag1:1;          // Flag 1
    uint8_t minutesU:4;       // Units of minutes
    uint8_t minutesT:3;       // Tens of minutes
    uint8_t flag2:1;          // Flag 2
    uint8_t hoursU:4;         // Units of hours
    uint8_t hoursT:2;         // Tens of hours
    uint8_t flag3:1;          // Flag 3
    uint8_t flag4:1;          // Flag 4
    uint32_t userBitData;     // 32 user bits
}TC, *PTC;
```

```
// Unpacked representation of SMPTE 12M-1999 Time Code
typedef struct tagTCU
{
    uint32_t frames;
    uint32_t seconds;
    uint32_t minutes;
    uint32_t hours;
    bool32_t dropFrameFlag;
    bool32_t colorFrameFlag;
    bool32_t flag1;
    bool32_t flag2;
    bool32_t flag3;
    bool32_t flag4;
    uint32_t userBitData;
}TCU, *PTCU;
#endif

/*****************************************************************************/

#if !defined(_WBGAIN_)
#define _WBGAIN_
    //Color channels adjustment
    //intended for the White balance adjustment on color camera
    //by changing the gains of the red and blue channels
    typedef struct tagWBGAIN
    {
        float R;    //White balance, gain correction for red
        float B;    //White balance, gain correction for blue
    }
    WBGAIN, *PWBGAIN;
#endif
/*****************************************************************************/

#if !defined(_WINDOWS)
//Rectangle with well defined fields size
typedef struct tagRECT
{
    int32_t left;
    int32_t top;
    int32_t right;
    int32_t bottom;
} RECT *PRECT;
#endif

#define OLDMAXFILENAME 65       // maximum file path size for the continuous recording
                                // to keep compatibility with old setup files
#define MAXLENDESCRIPTION_OLD 121//maximum length for setup description
                                 //(before Phantom 638)
#define MAXLENDESCRIPTION 4096  // maximum length for new setup description
/*****************************************************************************/

// Image processing: Filtering
typedef struct tagIMFILTER
{
    int32_t dim;        //square kernel dimension 3,5
    int32_t shifts;     //right shifts of Coef (8 shifts means divide by 256)
    int32_t bias;       //bias to add at end
    int32_t Coef[5*5];  //maximum alocation for a 5x5 filter
}
IMFILTER, *PIMFILTER;
/*****************************************************************************/

// SETUP structure - camera setup parameters
// It started to be used in 1992 during the 16 bit compilers era;
// the fields are arranged compact with alignment at 1 byte - this was
// the compiler default at that time. New fields were added, some of them
// replace old fields but a compatibility is maintained with the old versions.

// ---UPDF = Updated Field. This field is maintained for compatibility with old
// versions but a new field was added for that information. The new field can
// be larger or may have a different measurement unit. For example FrameRate16
// was a 16 bit field to specify frame rate up to 65535 fps (frames per second).
// When this was not enough anymore, a new field was added: FrameRate (32 bit
// integer, able to store values up to 4 billion fps). Another example: Shutter
// field (exposure duration) was specified initially in microseconds,
```

```
// later the field ShutterNs was added to store the value in nanoseconds.
// The UF can be considered outdated and deprecated; they are updated in the
// Phantom libraries but the users of the SDK can ignore them.
//
// ---TBI - to be ignored, not used anymore
//
// Use the definition from stdint.h with known size for the integer types
//
#pragma pack(1)
typedef struct tagSETUP
{
    uint16_t FrameRate16;   // ---UPDF replaced by FrameRate
    uint16_t Shutter16;     // ---UPDF replaced by ShutterNs
    uint16_t PostTrigger16; // ---UPDF replaced by PostTrigger
    uint16_t FrameDelay16;  // ---UPDF replaced by FrameDelayNs
    uint16_t AspectRatio;   // ---UPDF replaced by ImWidth, ImHeight
    uint16_t Res7;          // ---TBI Contrast16
                            // (analog controls, not available after
                            // Phantom v3)
    uint16_t Res8;          // ---TBI Bright16
    uint8_t Res9;           // ---TBI Rotate16
    uint8_t Res10;          // ---TBI TimeAnnotation
                            // (time always comes from camera )
    uint8_t Res11;          // ---TBI TrigCine (all cines are triggered)
    uint8_t TrigFrame;      // Sync imaging mode:
                            // 0, 1, 2 = internal, external, locktoirig
    uint8_t Res12;          // ---TBI ShutterOn (the shutter is always on)
    char DescriptionOld[MAXLENDESCRIPTION_OLD];
                            // ---UPDF replaced by larger Description able to
                            // store 4k of user comments

    uint16_t Mark;          // "ST"  -  marker for setup file
    uint16_t Length;        // Length of the current version of setup
    uint16_t Res13;         // ---TBI Binning (binning factor)

    uint16_t SigOption;     // Global signals options:
                            // MAXSAMPLES = records the max possible samples
    int16_t BinChannels;    // Number of binary channels read from the
                            // SAM (Signal Acquisition Module)
    uint8_t SamplesPerImage;// Number of samples acquired per image, both
                            // binary and analog;
    char BinName[8][11];    // Names for the first 8 binary signals having
                            // maximum 10 chars/name; each string ended by a
                            // byte = 0
    uint16_t AnaOption;     // Global analog options single ended,  bipolar
    int16_t AnaChannels;    // Number of analog channels used (16 bit 2's
                            // complement per channel)
    uint8_t Res6;           // ---TBI (reserved)
    uint8_t AnaBoard;       // Board type 0=none, 1=dsk (DSP system kit),
                            // 2 dsk+SAM
                            // 3 Data Translation DT9802
                            // 4 Data Translation DT3010

    int16_t ChOption[8];    // Per channel analog options;
                            // now:bit 0...3 analog gain (1,2,4,8)
    float AnaGain[8];       // User gain correction for conversion from voltage
                            // to real units ,   per channel
    char AnaUnit[8][6];     // Measurement unit for analog channels: max 5
                            // chars/name ended each by a byte = 0
    char AnaName[8][11];    // Channel name for the first 8 analog channels:
                            // max 10 chars/name ended each by a byte = 0

    int32_t lFirstImage;    // Range of images for continuous recording:
                            // first image
    uint32_t dwImageCount;  // Image count for continuous recording;
                            // used also for signal recording
    int16_t nQFactor;       // Quality  -  for saving to compressed file at
                            // continuous recording; range 2...255
    uint16_t wCineFileType; // Cine file type  -  for continuous recording
    char szCinePath[4][OLDMAXFILENAME]; //4 paths to save cine files - for
                            // continuous recording. After upgrading to Win32
                            // this still remained 65 bytes long each
                            // GetShortPathName is used for the filenames
                            // saved here

    uint16_t Res14;         // ---TBI bMainsFreq (Mains frequency:
```

```
                        // TRUE = 60Hz USA,  FALSE = 50Hz
                        // Europe, for signal view in DSP)

                        // Time board - settings for PC104 irig board
                        // used in Phantom v3 not used anymore after v3
uint8_t Res15;          // ---TBI  bTimeCode;
                        // Time code: IRIG_B, NASA36, IRIG-A
uint8_t Res16;          // ---TBI bPriority
                        // Time code has priority over PPS
uint16_t Res17;         // ---TBI wLeapSecDY
                        // Next day of year with leap second
double Res18;           // ---TBI dDelayTC Propagation delay for time code
double Res19;           // ---TBI dDelayPPS Propagation delay for PPS

uint16_t Res20;         // ---TBI  GenBits
int32_t Res1;           // ---TBI
int32_t Res2;           // ---TBI
int32_t Res3;           // ---TBI

uint16_t ImWidth;       // Image dimensions in v4 and newer cameras: Width
uint16_t ImHeight;      // Image height

uint16_t EDRShutter16;  // ---UPDF replaced by EDRShutterNs
uint32_t Serial;        // Camera serial number. For firewire cameras you
                        // have a translated value here:
                        // factory serial + 0x58000
int32_t Saturation;     // ---UPDF replaced by float fSaturation
                        // Color saturation adjustmment [-100,   100] neutral 0

uint8_t Res5;           // ---TBI
uint32_t AutoExposure;  // Autoexposure enable 0=disable, 1=lock at trigger,
                        // 3=active after trigger
bool32_t bFlipH;        // Flips image horizontally
bool32_t bFlipV;        // Flips image vertically;
uint32_t Grid;          // Displays a crosshair or a grid in setup, 0=no grid
                        // 2=cross hair, 8= grid with 8 intervals

uint32_t FrameRateInt;     // Frame rate in frames per seconds
uint32_t Shutter;       // ---UPDF replaced by ShutterNs
                        // (here the value is in microseconds)
uint32_t EDRShutter;    // ---UPDF replaced by EDRShutterNs
                        // (here the value is in microseconds)
uint32_t PostTrigger;   // Post trigger frames, measured in frames
uint32_t FrameDelay;    // ---UPDF replaced by FrameDelayNs
                        // (here the value is in microseconds)

bool32_t bEnableColor;  // User option: when 0 forces gray images from
                        // color cameras

uint32_t CameraVersion; // The version of camera hardware (without decimal
                        // point). Examples of cameras produced after the
                        // year 2000
                        // Firewire: 4, 5, 6
                        // Ethernet: 42 43 51 7 72 73 9 91 10
                        // 650 (p65) 660 (hd)  ....
uint32_t FirmwareVersion;// Firmware version
uint32_t SoftwareVersion;// Phantom software version
                        // End of SETUP in software version 551 (May 2001)

int32_t RecordingTimeZone;// The time zone active during the recording of
                        // the cine
                        // End of SETUP in software version 552 (May 2001)

uint32_t CFA;               // Code for the Color Filter Array of the sensor
                        // CFA_NONE=0,(gray) CFA_VRI=1(gbrg/rggb),
                        // CFA_VRIV6=2(bggr/grbg), CFA_BAYER=3(gb/rg)
                        // CFA_BAYERFLIP=4 (rg/gb)
                        // high byte carries info about color/gray heads at
                        // v6 and v6.2
                        // Masks: 0x80000000: TLgray 0x40000000: TRgray
                        // 0x20000000: BLgray 0x10000000: BRgray

//Final adjustments after image processing:
int32_t Bright;         // ---UPDF replaced by fOffset
                        // Brightness -100...100 neutral:0
int32_t Contrast;       // ---UPDF replaced by fGain
```

```
                        // -100...100 neutral:0
int32_t Gamma;          // ---UPDF replaced by fGamma
                        // -100...100 neutral:0

uint32_t Res21;         // ---TBI

uint32_t AutoExpLevel;  // Level for autoexposure control
uint32_t AutoExpSpeed;  // Speed for autoexposure control
RECT AutoExpRect;       // Rectangle for autoexposure control

WBGAIN WBGain[4];       // Gain adjust on R,B components, for white balance,
                        // at Recording
                        // 1.0 = do nothing,
                        // index 0: all image for v4,5,7...
                        // and TL head for v6, v6.2 (multihead)
                        // index 1, 2, 3 :   TR, BL, BR for multihead
int32_t Rotate;         // Rotate the image 0=do nothing
                        // +90=counterclockwise -90=clockwise
                        // End of SETUP in software version 578 (Nov 2002)

WBGAIN WBView;          // White balance to apply on color interpolated Cines

uint32_t RealBPP;       // Real number of bits per pixel for this cine
                        // 8 on 8 bit cameras
                        // (v3, 4, 5, 6, 42, 43, 51, 62, 72, 9)
                        // Phantom v7:  8 or 12
                        // 14 bit cameras  8, 10, 12, 14
                        // Pixels will be stored on 8 or 16 bit in files
                        // and in PC memory
                        // (if RealBPP>8 the storage will be on 16 bits)

//First degree function to convert the 16 bits pixels to 8 bit
//(for display or file convert)
uint32_t Conv8Min;      // ---TBI
                        // Minimum value when converting to 8 bits
uint32_t Conv8Max;      // ---UPDF replaced by fGain16_8
                        // Max value when converting to 8 bits

int32_t FilterCode;     // ImageProcessing: area processing code
int32_t FilterParam;    // ImageProcessing: optional parameter
IMFILTER UF;            // User filter: a 3x3 or 5x5 user convolution filter

uint32_t BlackCalSVer;  // Software Version used for Black Reference
uint32_t WhiteCalSVer;  // Software Version used for White Calibration
uint32_t GrayCalSVer;   // Software Version used for Gray Calibration
bool32_t bStampTime;    // Stamp time (in continuous recording)
                        // 1 = absolute time, 3 = from trigger
                        // End of SETUP in software version 605 (Nov 2003)

uint32_t SoundDest;     // Sound device 0: none, 1: Speaker, 2: sound board

//Frame rate profile
uint32_t FRPSteps;      // Suplimentary steps in frame rate profile
                        // 0 means no frame rate profile
int32_t FRPImgNr[16];   // Image number where to change the rate and/or
                        // exposure allocated for 16 points (4 available
                        // in v7)
uint32_t FRPRate[16];   // New value for frame rate (fps)
uint32_t FRPExp[16];    // New value for exposure
                        // (nanoseconds, not implemented in cameras)

//Multicine partition
int32_t MCCnt;          // Partition count (= cine count - 1)
                        // Preview cine does not need a partition
float MCPercent[64];    // Percent of memory used for partitions
                        // Allocated for 64 partitions, 15 used in the
                        // current cameras
                        // End of SETUP in software version 606 (May 2004)

// CALIBration on Current Image (CSR, current session reference)
uint32_t CICalib;       // This cine or this stg is the result of
                        // a current image calibration
                        // masks: 1 BlackRef, 2 WhiteCalib, 4 GrayCheck
                        // Last cicalib done at the acqui params:
uint32_t CalibWidth;    // Image dimensions
uint32_t CalibHeight;
```

```
uint32_t CalibRate;      // Frame rate (frames per second)
uint32_t CalibExp;       // Exposure duration (nanoseconds)
uint32_t CalibEDR;       // EDR (nanoseconds)
uint32_t CalibTemp;      // Sensor Temperature

uint32_t HeadSerial[4]; // Head serials for ethernet multihead cameras
                        // (v6.2) When multiple heads are saved in a file,
                        // the serials for existing heads are not zero
                        // When one head is saved in a file its serial is
                        // in HeadSerial[0] and the other head serials
                        // are 0xFFffFFff
                        // End of SETUP in software version 607 (Oct 2004)
uint32_t RangeCode;      // Range data code: describes the range data format
uint32_t RangeSize;      // Range data,  per image size

uint32_t Decimation;     // Factor to reduce the frame rate when sending
                        //the images to i3 external memory by fiber
                        // End of SETUP in software version 614 (Feb 2005)
uint32_t MasterSerial;   // Master camera Serial for external sync. 0 means
                        // none (this camera is not a slave of another
                        // camera)
                        // End of SETUP in software version 624 (Jun 2005)

uint32_t Sensor;         // Camera sensor code
                        // End of SETUP in software version 625 (Jul 2005)

//Acquisition parameters in nanoseconds
uint32_t ShutterNs;      // Exposure, in nanoseconds
uint32_t EDRShutterNs;   // EDRExp, in nanoseconds
uint32_t FrameDelayNs;   // FrameDelay, in nanoseconds
                        // End of SETUP in software version 631 (Oct 2005)

//Stamp outside the acquired image
//(this increases the image size by adding a border with text information)
uint32_t ImPosXAcq;      // Acquired image horizontal offset in
                        // sideStamped image
uint32_t ImPosYAcq;      // Acquired image vertical offset in sideStamped
                        // image
uint32_t ImWidthAcq;     // Acquired image width  (different value from
                        // ImWidth if sideStamped file)
uint32_t ImHeightAcq;    // Acquired image height (different value from
                        // ImHeight if sideStamped file)

char Description[MAXLENDESCRIPTION];//User description or comments
                                    //(enlarged to 4096 characters)
                        // End of SETUP in software version 637 (Jul 2006)

bool32_t RisingEdge;     // TRUE rising, FALSE falling
uint32_t FilterTime;     // time constant
bool32_t LongReady;      // If TRUE the Ready is 1 from the start
                        // to the end of recording (needed for signal
                        // acquisition)
bool32_t ShutterOff;     // Shutter off - to force maximum exposure for PIV
                        // End of SETUP in software version 658 (Mar 2008)

uint8_t Res4[16];        // ---TBI
                        // End of SETUP in software version 663 (May 2008)

bool32_t bMetaWB;        // pixels value does not have WB applied
                        // (or any other processing)
int32_t Hue;             // ---UPDF replaced by float fHue
                        // hue corection (degrees: -180 ...180)
                        // End of SETUP in software version 671 (May 2009)

int32_t BlackLevel;      // Black level in the raw pixels
int32_t WhiteLevel;      // White level in the raw pixels

char  LensDescription[256];// text with the producer, model,
                           // focal range etc ...
float LensAperture;      // aperture f number
float LensFocusDistance;// distance where the objects are in  focus in
                        // meters, not available from Canon motorized lens
float LensFocalLength;   // current focal length; (zoom factor)
                        // End of SETUP in software version 691 (Jul 2010)

// Image adjustment
```

```
float fOffset;          // [-1.0, 1.0], neutral 0.0;
                        // 1.0 means shift by the maximum pixel value
float fGain;            // [0.0, Max], neutral 1.0;
float fSaturation;      // [0.0, Max], neutral 1.0;

float fHue;             // [-180.0, 180.0] neutral 0;
                        // degrees and fractions of degree to rotate the hue

float fGamma;           // [0.0, Max], neutral 1.0;  global gamma
                        // (or green gamma)
float fGammaR;          // per component gammma (to be added to the field
                        // Gamma)
                        // 0 means neutral
float fGammaB;

float fFlare;           // [-1.0, 1.0], neutral 0.0;
                        // 1.0 means shift by the maximum pixel value
                        // pre White Balance offset

float fPedestalR;       // [-1.0, 1.0], neutral 0.0;
                        // 1.0 means shift by the maximum pixel value
float fPedestalG;       // after gamma offset
float fPedestalB;
float fChroma;          // [0.0, Max], neutral 1.0;
                        // chrominance adjustment (after gamma)
char  ToneLabel[256];
int32_t   TonePoints;
float fTone[32*2];      // up to 32  points  + 0.0,0.0  1.0,1.0
                        // defining a LUT using spline curves

char  UserMatrixLabel[256];
bool32_t  EnableMatrices;
float cmUser[9];         // RGB color matrix

bool32_t  EnableCrop;   // The Output image will contains only a rectangle
                        // portion of the input image
RECT  CropRect;
bool32_t  EnableResample;// Resample image to a desired output Resolution
uint32_t  ResampleWidth;
uint32_t  ResampleHeight;

float fGain16_8;        // Gain coefficient used when converting to 8bps
                        // Input pixels (bitdepth>8) are multiplied by
                        // the factor:  fGain16_8 * (2**8 / 2**bitdepth)
                        // End of SETUP in software version 693 (Oct 2010)

uint32_t FRPShape[16];  // 0: flat,  1 ramp
TC TrigTC;              // Trigger frame SMPTE time code and user bits
float fPbRate;          // Video playback rate (fps) active when the cine
                        // was captured
float fTcRate;          // Playback rate (fps) used for generating SMPTE
                        // time code
                        // End of SETUP in software version 701 (Apr 2011)

char CineName[256];     // Cine name
                        // End of SETUP in software version 702 (May 2011)

float fGainR;           // Per component gain - user adjustment
float fGainG;           // [0.0, Max], neutral 1.0;
float fGainB;

float cmCalib[9];       // RGB color calibration matrix bringing camera pixels to
                        // rec 709. It includes the white balance set into the
                        // ph16 cameras using fWBTemp and fWBCc and the original
                        // factory calibration. The cine player should decompose
                        // this matrix in two components: a diagonal one with the
                        // white balance to be applied before interpolation and a
                        // normalized one to be applied after interpolation.

float fWBTemp;          // White balance based on color temperature and color
float fWBCc;            // temperature and color compensation index.
                        // Its effect is included in the cmCalib.

char CalibrationInfo[1024];// Original calibration matrices: used to calculate
                        // cmCalib in the camera
```

```
char OpticalFilter[1024];// Optical filter matrix: used to calculate cmCalib in
                         // the camera
                         // End of SETUP in software version 709 (September 2011)

char GpsInfo[MAXSTDSTRSZ]; //Current position and status info as received from a
                         // GPS receiver

char Uuid[MAXSTDSTRSZ]; // Unique cine identifier
                         // End of SETUP in software version 719 (March 2012)

char CreatedBy[MAXSTDSTRSZ]; // The name of the application that created this cine
                         // End of SETUP in software version 720 (March 2012)
uint32_t RecBPP;         // Acquisition bit depth. It can be 8, 10, 12 or 14

uint16_t LowestFormatBPP; // Description of the minimum format out of all formats
uint16_t LowestFormatQ;   //the images of this cine have been represented on
                          //since the moment of acquisition.

                          // End of SETUP in software version 731 (February 2013)

float fToe;               // Controls the gamma curve in the blacks.
                          // Neutral is 1.0f.
                          // Decreasing fToe lifts the blacks, while
                          // increasing it compresses them.
uint32_t LogMode;         // Configures the log mode.
                          // 0 - log mode disabled.
                          // 1, 2, etc - log mode enabled.
                          // If log mode enabled, gain, gamma, the pedestals,
                          // r/g/b gains, offset and flare are inactive.
char CameraModel[MAXSTDSTRSZ];    // Camera model string.

                          // End of SETUP in software version 742

uint32_t WBType;          // For raw color cines, it describes how meta WB is
                          // stored.
                          // If bit 0 is set - wb is stored in WBGain field.
                          // If bit 1 is set - wb is stored in fWBTemp, fWBCc
                          // fields.
float fDecimation;        // Decimation coefficient employed when this cine was
                          // saved to file.
                          // End of SETUP in software version 745

uint32_t MagSerial;       // The serial of the magazine where the cine was
                          // recorded or stored (if any, null otherwise)
uint32_t CSSerial;        // Cine Save serial: The serial of the device (camera,
                          // cine station) used to save the cine to file
double dFrameRate;        // High precision acquisition frame rate, replace
                          // uint32_t FrameRate

                          // End of SETUP in software version 751
uint32_t SensorMode;      // Camera sensor Mode used to define what mode the
                          // Sensor is in
                          // This is currently valid for the Flex4K, VEO 4K and
                          // VIRGO V2640
                          // it is undefined for all other camera (should be 0)
                          //
                          //  Flex4K / VEO 4K
                          //
                          // bits 4  is for rolling vs Global Shutter
                          // 0x0 = Rolling
                          // 0x1 = Global
                          //
                          //  Virgo (V2640)
                          //
                          // bits 3, 2, 1, 0 bits for acquisition mode
                          // This is only valid for Virgo cameras currently
                          //(NA = not used currently)
                          // 0x0000 = Standard
                          // 0x0001 = NA
                          // 0x0010 = Standard_Binned
                          // 0x0011 = NA
                          // 0x0100 = NA
                          // 0x0101 = HS
                          // 0x0110 = NA
                          // 0x0111 = HS_Binned
                          // all other values currently undefined
```

```
                                //
                                // End of SETUP in software version 771 (Dec 2017)
    int32_t UndecFirst;         // First frame number selected by user in the save
                                dialog, before decimation was applied
                                // The image content will remain first in the decimated
                                cine too.
                                // End of SETUP in software version 800 (Feb 2021)

    bool32_t SupportsBinning;   // Used to indicate the camera supports binning bin2
                                // does not indicate the camera is in binning,
                                // only that it supports it.
                                // see SensorMode for current binning state
    bool32_t UvSensor;          // Used to indicate the sensor supports UV light
    CHAR AnaDaqDescription[MAXSTDSTRSZ_128]; // used to provide DAQ operation
                                             description
    CHAR BinDaqDescription[MAXSTDSTRSZ_128]; // used to provide DAQ operation
                                             description

                                // End of SETUP in software version 803 (Oct 2022)
    //
    ////////////////////
    //
    uint32_t DaqOptions;        // Bit Field used to defines additional DAQ
                                configuration/Status information
                                // End of SETUP in software version 804 (Jan 2023)

    //VRI internal note: Size checked structure.
    //Update oldcomp.c if new fields are added
    //-----------------------------------------------------------------------
} SETUP, *PSETUP;
/*****************************************************************************/

#pragma pack()

#ifdef __cplusplus
}
#endif
```

8.3. Appendix 3. Cine File Dump

An example of a raw cine file contents in hex and the meaning of fields. The file is 14bpp color, 800x600 from a simulated v7.3 and was written in the software version 640.

CINEFILEHEADER

```
00000000  43 49 2C 00  02 00 01 00  FD FD FF FF  D3 09 00 00  BITMAPINFOHEADER
00000010  FD FD FF FF  19 00 00 00  2C 00 00 00  54 00 00 00
00000020  D4 17 00 00  F3 18 DF 62  8F F1 2D 46  28 00 00 00  SETUP
00000030  20 03 00 00  58 02 00 00  01 00 10 00  00 00 00 00
00000040  00 A6 0E 00  8E B1 00 00  8E B1 00 00  00 00 00 00
00000050  00 40 00 00  E8 03 84 03  D0 07 01 00  01 00 00 00
00000060  00 00 00 01  01 00 01 00  00 00 00 00  00 00 00 00
00000070  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000080  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000090  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000000A0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000000B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000000C0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000000D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000000E0  53 54 3C 16  01 00 00 00  00 00 01 00  00 00 00 00
000000F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000100  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000110  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000120  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000130  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000140  00 00 00 00  00 00 00 00  00 01 00 01  00 01 00 01
00000150  00 01 00 01  00 01 00 01  00 00 00 80  3F 00 00 80
00000160  3F 00 00 80  3F 00 00 80  3F 00 00 80  3F 00 00 80
00000170  3F 00 00 80  3F 00 00 80  3F 00 00 00  00 00 00 00
00000180  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000190  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000001A0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000001B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000001C0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000001D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000001E0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000001F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000200  00 9C FF FF  FF E8 03 00  00 02 00 00  80 00 00 00
00000210  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000220  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000230  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000240  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000250  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000260  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000270  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000280  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000290  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000002A0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000002B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000002C0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000002D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000002E0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000002F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000300  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000310  00 01 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000320  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000330  00 00 00 00  00 20 03 58  02 00 00 0A  00 00 00 00
00000340  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000350  00 00 00 00  E8 03 00 00  84 03 00 00  00 00 00 00
00000360  D0 07 00 00  01 00 00 00  01 00 00 00  49 00 00 00
```

```
00000370  00 00 00 00  80 02 00 00  D0 D5 FF FF  03 00 00 00
00000380  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000390  C8 00 00 00  05 00 00 00  80 00 00 00  80 00 00 00
000003A0  80 01 00 00  80 01 00 00  00 00 80 3F  00 00 80 3F
000003B0  00 00 80 3F  00 00 80 3F  00 00 80 3F  00 00 80 3F
000003C0  00 00 80 3F  00 00 80 3F  00 00 00 00  00 00 80 3F
000003D0  00 00 80 3F  0E 00 00 00  00 00 00 00  FF 3F 00 00
000003E0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000003F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000400  00 00 00 00  01 00 00 00  00 00 00 00  00 00 00 00
00000410  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000420  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000430  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000440  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000450  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000460  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000470  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000480  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000490  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000004A0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000004B0  0A 00 00 00  0A 00 00 00  0A 00 00 00  0A 00 00 00
000004C0  E8 03 00 00  E8 03 00 00  E8 03 00 00  E8 03 00 00
000004D0  E8 03 00 00  E8 03 00 00  E8 03 00 00  E8 03 00 00
000004E0  E8 03 00 00  E8 03 00 00  E8 03 00 00  E8 03 00 00
000004F0  D0 07 00 00  D0 07 00 00  D0 07 00 00  D0 07 00 00
00000500  A0 BB 0D 00  A0 BB 0D 00  A0 BB 0D 00  A0 BB 0D 00
00000510  A0 BB 0D 00  A0 BB 0D 00  A0 BB 0D 00  A0 BB 0D 00
00000520  A0 BB 0D 00  A0 BB 0D 00  A0 BB 0D 00  A0 BB 0D 00
00000530  01 00 00 00  00 00 C8 42  00 00 00 00  00 00 00 00
00000540  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000550  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000560  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000570  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000580  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000590  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000005A0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000005B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000005C0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000005D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000005E0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000005F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000600  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000610  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000620  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000630  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000640  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000650  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00000660  00 00 00 00  00 00 00 00  01 00 00 00  00 00 00 00
00000670  00 00 00 00  A0 BB 0D 00  00 00 00 00  E8 03 00 00
00000680  00 00 00 00  00 00 00 00  20 03 00 00  58 02 00 00
00000690  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000006A0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000006B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00

.......................... zero area of the SETUP structure

00001670  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00001680  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00001690  D0 00 00 00  EA 03 01 00  B7 0E 08 DF  8E F1 2D 46
000016A0  EF 97 49 DF  8E F1 2D 46  27 21 8B DF  8E F1 2D 46
000016B0  5B AA CC DF  8E F1 2D 46  93 33 0E E0  8E F1 2D 46
```

Images time

```
000016C0  CB BC 4F E0  8E F1 2D 46  03 46 91 E0  8E F1 2D 46
000016D0  3B CF D2 E0  8E F1 2D 46  73 58 14 E1  8E F1 2D 46
000016E0  A7 E1 55 E1  8E F1 2D 46  DF 6A 97 E1  8E F1 2D 46
000016F0  17 F4 D8 E1  8E F1 2D 46  4F 7D 1A E2  8E F1 2D 46
00001700  87 06 5C E2  8E F1 2D 46  BF 8F 9D E2  8E F1 2D 46
00001710  F3 18 DF E2  8E F1 2D 46  2B A2 20 E3  8E F1 2D 46
00001720  63 2B 62 E3  8E F1 2D 46  9B B4 A3 E3  8E F1 2D 46
00001730  D3 3D E5 E3  8E F1 2D 46  07 C7 26 E4  8E F1 2D 46
00001740  3F 50 68 E4  8E F1 2D 46  77 D9 A9 E4  8E F1 2D 46
00001750  AF 62 EB E4  8E F1 2D 46  E7 EB 2C E5  8E F1 2D 46
00001760  6C 00 00 00  EB 03 01 00  7F FB 3A 00  7F FB 3A 00
00001770  7F FB 3A 00  7F FB 3A 00  7F FB 3A 00  7F FB 3A 00
00001780  7F FB 3A 00  7F FB 3A 00  7F FB 3A 00  7F FB 3A 00
00001790  7F FB 3A 00  7F FB 3A 00  7F FB 3A 00  7F FB 3A 00
000017A0  7F FB 3A 00  7F FB 3A 00  7F FB 3A 00  7F FB 3A 00
000017B0  7F FB 3A 00  7F FB 3A 00  7F FB 3A 00  7F FB 3A 00
000017C0  7F FB 3A 00  7F FB 3A 00  7F FB 3A 00  08 00 00 00
000017D0  EC 03 00 00  9C 18 00 00  00 00 00 00  A4 BE 0E 00
000017E0  00 00 00 00  AC 64 1D 00  00 00 00 00  B4 0A 2C 00
000017F0  00 00 00 00  BC B0 3A 00  00 00 00 00  C4 56 49 00
00001800  00 00 00 00  CC FC 57 00  00 00 00 00  D4 A2 66 00
00001810  00 00 00 00  DC 48 75 00  00 00 00 00  E4 EE 83 00
00001820  00 00 00 00  EC 94 92 00  00 00 00 00  F4 3A A1 00
00001830  00 00 00 00  FC E0 AF 00  00 00 00 00  04 87 BE 00
00001840  00 00 00 00  0C 2D CD 00  00 00 00 00  14 D3 DB 00
00001850  00 00 00 00  1C 79 EA 00  00 00 00 00  24 1F F9 00
00001860  00 00 00 00  2C C5 07 01  00 00 00 00  34 6B 16 01
00001870  00 00 00 00  3C 11 25 01  00 00 00 00  44 B7 33 01
00001880  00 00 00 00  4C 5D 42 01  00 00 00 00  54 03 51 01
00001890  00 00 00 00  5C A9 5F 01  00 00 00 00  08 00 00 00
000018A0  00 A6 0E 00  C0 36 80 2F  C0 36 80 2F  C0 36 80 2F
000018B0  C0 36 80 2F  C0 36 80 2F  00 37 C0 2F  00 37 C0 2F
000018C0  40 37 C0 2F  40 37 00 30  40 37 00 30  80 37 40 30
000018D0  80 37 40 30  80 37 40 30  80 37 40 30  80 37 40 30
000018E0  80 37 40 30  80 37 40 30  80 37 40 30  40 37 00 30
000018F0  40 37 00 30  80 37 00 30  80 37 40 30  40 37 00 30
00001900  00 37 C0 2F  C0 36 80 2F  C0 36 40 2F  00 37 40 2F
00001910  00 37 00 2F  00 37 00 2F  00 37 00 2F  C0 36 C0 2E
00001920  C0 36 C0 2E  C0 36 C0 2E  80 36 C0 2E  C0 36 00 2F
00001930  C0 36 C0 2E  C0 36 C0 2E  C0 36 00 2F  80 36 C0 2E
00001940  80 36 40 2E  40 36 00 2E  40 36 00 2E  00 37 80 2E
00001950  00 37 C0 2E  40 37 C0 2E  00 37 80 2E  40 37 80 2E
00001960  C0 36 C0 2D  80 36 80 2D  C0 36 80 2D  00 37 C0 2C
00001970  80 35 80 2B  C0 35 40 2B  40 35 40 2B  80 35 80 2B
00001980  40 35 C0 2A  40 33 00 29  80 31 00 29  40 33 40 2C
00001990  C0 35 00 2E  80 35 40 2D  80 35 40 2D  80 35 00 2D
000019A0  C0 35 40 2D  C0 35 40 2D  C0 35 40 2D  00 36 80 2D
000019B0  00 36 80 2D  00 36 80 2D  00 36 80 2D  00 36 80 2D
000019C0  00 36 80 2D  00 36 80 2D  00 36 80 2D  C0 35 40 2D
000019D0  C0 35 40 2D  C0 35 40 2D  C0 35 40 2D  C0 35 40 2D
000019E0  C0 35 40 2D  C0 35 40 2D  C0 35 40 2D  80 35 00 2D
000019F0  80 35 00 2D  40 35 C0 2C  00 35 80 2C  00 35 80 2C
00001A00  40 35 C0 2C  80 35 00 2D  80 35 00 2D  80 35 00 2D
00001A10  C0 35 C0 2D  00 36 80 2E  C0 36 80 2F  40 37 C0 30
00001A20  40 38 C0 31  C0 38 C0 32  40 39 00 33  00 39 40 32
00001A30  C0 37 80 30  40 36 80 2E  80 34 C0 2C  40 33 C0 2A
00001A40  80 31 C0 28  00 30 00 27  C0 2E 40 26  00 30 40 26
00001A50  00 30 40 26  00 30 80 26  00 30 80 26  40 30 00 27
00001A60  80 30 40 27  80 30 80 27  C0 30 80 27  00 30 00 27
00001A70  00 30 80 27  80 30 40 28  80 31 40 29  40 32 40 2A
00001A80  00 33 40 2B  C0 33 C0 2B  00 34 80 2C  C0 34 C0 2D
```

Images exposures

Images pointers (64 bits)

First image annotation

First image pixels (bottom left corner)

```
00001A90  C0 34 C0 2D  00 35 00 2E  40 35 00 2E  40 35 00 2E
00001AA0  00 35 00 2E  C0 34 C0 2D  C0 34 40 2D  40 34 80 2C
00001AB0  80 33 40 2C  40 33 C0 2B  C0 32 00 2B  40 32 80 2A
00001AC0  80 31 00 2A  00 31 C0 29  C0 30 80 29  40 31 80 29
00001AD0  80 31 80 29  80 31 80 29  40 31 40 29  40 31 40 29
00001AE0  40 31 00 29  00 31 40 29  00 31 00 29  80 30 80 28
00001AF0  80 30 80 28  80 30 80 28  80 30 40 28  40 30 80 28
00001B00  40 30 40 28  00 30 00 28  00 30 40 28  40 2F 80 28

..................................  the pixels of all images

016E4D20  00 0D 00 0F  00 0D 40 0F  00 0D 40 0F  C0 0C 80 0F
016E4E10  C0 2C 00 24  00 29 80 22  00 28 C0 22  80 28 00 26
016E4E20  40 2A C0 26  40 27 80 23  80 22 40 1D  C0 15 40 10
016E4E30  40 0C C0 0B  C0 07 80 09  00 07 00 0B  00 09 00 0E
016E4E40  40 0A 00 0F  40 0B 80 10  40 0C 40 12  40 10 80 13
016E4E50  C0 0D 40 0D  00 09 00 0B  00 09 80 0C  40 0A 80 0D
016E4E60  40 0C C0 0F  00 0E C0 0F  80 0C 00 0D  C0 0A 40 0C
016E4E70  C0 0C 40 10  40 13 40 17  80 1C 80 20  80 25 80 26
016E4E80  40 28 80 25  C0 26 80 23  00 27 C0 24  80 28 00 25
016E4E90  40 28 00 24  40 27 C0 22  80 26 80 22  80 26 80 22
016E4EA0  C0 26 C0 22  40 27 C0 22  40 27 C0 22  80 27 00 23
016E4EB0  40 27 40 22  80 26 00 22  40 26 C0 21  C0 25 40 21
016E4EC0  00 25 80 20  80 24 C0 20  00 25 00 22  00 28 80 21
016E4ED0  40 1D C0 0F  C0 12 80 0D  80 11 C0 0A  C0 0A C0 02
016E4EE0  C0 04 80 00  00 05 C0 02  80 07 40 05  80 09 40 07
016E4EF0  C0 0B C0 0C  80 18 40 1D  00 23 80 20  00 24 00 21
016E4F00  C0 26 00 24  40 24 40 1E  40 22 C0 20  00 27 40 25
016E4F10  40 27 C0 20  00 20 C0 15  80 0E 40 02  00 03 00 02
016E4F20  40 0E 80 15  00 23 40 26  C0 28 80 22  00 25 80 20
016E4F30  40 24 C0 20  80 24 80 20  80 24 C0 20  80 24 80 20
016E4F40  80 24 40 20  00 24 00 20  00 24 00 20  00 24 40 20
016E4F50  00 24 40 20  00 24 40 20  00 24 40 20  00 24 40 20
016E4F60  00 24 40 20
```

End of file

The decoded header fields for the above file:

```
[Cine File Header]
Type=CI
HeaderSize=44
Compression=0x2
Version=1
FirstMovieImage=-515
TotalImageCount=2515
FirstImageNo=-515
ImageCount=25
OffImageHeader=0x2c
OffSetup=0x54
OffImageOffsets=0x17d4
TriggerTime=0x462df18f, 0x62df18f3 (Tue Apr 24 2007 15:01:19.386 217)

[Setup]
Mark="ST"
Length=5692
Description=""
TrigFrame=0
lFirstImage=-100
dwImageCount=1000
nQFactor=2
wCineFileType=32768
szCinePath[0]=""
szCinePath[1]=""
szCinePath[2]=""
szCinePath[3]=""
ContrastR(old)=0
ContrastG(old)=0
ContrastB(old)=0
BrightR(old)=0
BrightG(old)=0
BrightB(old)=0
ImWidth=800
ImHeight=600
Serial=10 (10)
Saturation=0
AutoExposure=0
bFlipH=0
bFlipV=0
Grid=0
FrameRate=1000
Shutter=900
EDRShutter=0
PostTrigger=2000
FrameDelay=1
bEnableColor=1
CameraVersion=73
FirmwareVersion=0
SoftwareVersion=640
RecordingTimeZone=-10800
CFA=0x3
Bright=0
Contrast=0
Gamma=0
AutoExpLevel=200
AutoExpSpeed=5
AutoExpRect=128,384,128,384
WBGain[0]=1.000000,1.000000
```

```
WBGain[1]=1.000000,1.000000
WBGain[2]=1.000000,1.000000
WBGain[3]=1.000000,1.000000
Rotate=0
WBView=1.000000,1.000000
RealBPP=14
Conv8Min=0
Conv8Max=16383
FilterCode=0
FilterParam=0
UserFilter:
dim=0
shifts=0
bias=0
Coefficients:
   0    0    0    0    1
   0    0    0    0    0
   0    0    0    0    0
   0    0    0    0    0
   0    0    0    0    0
BlackCalSVer=0
WhiteCalSVer=0
GrayCalSVer=0
bStampTimeConRec=0
SoundDest=0
FRPSteps=0
FRPImgNr= 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
FRPRate= 10  10  10  10  1000  1000  1000  1000  1000  1000  1000
1000  1000  1000  1000  1000
FRPExp= 2000  2000  2000  2000  900000  900000  900000  900000
900000  900000  900000  900000  900000  900000  900000  900000
MCCnt=1
MCPercent= 100.000000  0.000000  0.000000  0.000000  0.000000
0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
0.000000  0.000000  0.000000
CICalib=0
CalibWidth=0
CalibHeight=0
CalibRate=0
CalibExp=0
CalibEDR=0
CalibTemp=0
HeadSerial= 0  0  0  0
AnaBoard=0
AnaChannels=0
BinChannels=0
SamplesPerImage=1
AnaOption=0
RangeCode=0
RangeSize=0
Decimation=1
MasterSerial=0
Sensor=0
ShutterNs=900000
```

```
EDRShutterNs=0
FrameDelayNs=1000
ImWidthAcq=800
ImHeightAcq=600
ImPosXAcq=0
ImPosYAcq=0

[Bitmap Info Header]
biSize=40
biWidth=800
biHeight=600
biPlanes=1
biBitCount=16
biCompression=0
biSizeImage=960000
biXPelsPerMeter=45454
biYPelsPerMeter=45454
biClrUsed=0
biClrImportant=16384

[Image Time]
Frame -515=0x462df18e, 0xdf080eb7 (Tue Apr 24 2007 15:01:18.871 217)
Frame -514=0x462df18e, 0xdf4997ef (Tue Apr 24 2007 15:01:18.872 217)
Frame -513=0x462df18e, 0xdf8b2127 (Tue Apr 24 2007 15:01:18.873 217)
Frame -512=0x462df18e, 0xdfccaa5b (Tue Apr 24 2007 15:01:18.874 217)
Frame -511=0x462df18e, 0xe00e3393 (Tue Apr 24 2007 15:01:18.875 217)
Frame -510=0x462df18e, 0xe04fbccb (Tue Apr 24 2007 15:01:18.876 217)
Frame -509=0x462df18e, 0xe0914603 (Tue Apr 24 2007 15:01:18.877 217)
Frame -508=0x462df18e, 0xe0d2cf3b (Tue Apr 24 2007 15:01:18.878 217)
Frame -507=0x462df18e, 0xe1145873 (Tue Apr 24 2007 15:01:18.879 217)
Frame -506=0x462df18e, 0xe155e1a7 (Tue Apr 24 2007 15:01:18.880 217)
...
Frame -495=0x462df18e, 0xe426c707 (Tue Apr 24 2007 15:01:18.891 217)
Frame -494=0x462df18e, 0xe468503f (Tue Apr 24 2007 15:01:18.892 217)
Frame -493=0x462df18e, 0xe4a9d977 (Tue Apr 24 2007 15:01:18.893 217)
Frame -492=0x462df18e, 0xe4eb62af (Tue Apr 24 2007 15:01:18.894 217)
Frame -491=0x462df18e, 0xe52cebe7 (Tue Apr 24 2007 15:01:18.895 217)

[Image Exposure]
At frame -515, Exposure(microsec.)=900.000
At frame -514, Exposure(microsec.)=900.000
At frame -513, Exposure(microsec.)=900.000
At frame -512, Exposure(microsec.)=900.000
At frame -511, Exposure(microsec.)=900.000
At frame -510, Exposure(microsec.)=900.000
At frame -509, Exposure(microsec.)=900.000
At frame -508, Exposure(microsec.)=900.000
At frame -507, Exposure(microsec.)=900.000
At frame -506, Exposure(microsec.)=900.000
...
At frame -495, Exposure(microsec.)=900.000
At frame -494, Exposure(microsec.)=900.000
At frame -493, Exposure(microsec.)=900.000
At frame -492, Exposure(microsec.)=900.000
At frame -491, Exposure(microsec.)=900.000
```

8.4. Appendix 4. Fields Offsets in Structures and Cine File

These offsets were stable until now and will remain unchanged until one of the first two structures will change size.

Field	Structure offset	File offset
	CINEFILEHEADER	
Type	0x0000	0x0000
HeaderSize	0x0002	0x0002
Compression	0x0004	0x0004
Version	0x0006	0x0006
FirstMovieImage	0x0008	0x0008
TotalImageCount	0x000C	0x000C
FirstImageNo	0x0010	0x0010
ImageCount	0x0014	0x0014
OffImageHeader	0x0018	0x0018
OffSetup	0x001C	0x001C
OffImageOffsets	0x0020	0x0020
TriggerTime	0x0024	0x0024
	BITMAPINFOHEADER	
biSize	0x0000	0x002C
biWidth	0x0004	0x0030
biHeight	0x0008	0x0034
biPlanes	0x000C	0x0038
biBitCount	0x000E	0x003A
biCompression	0x0010	0x003C
biSizeImage	0x0014	0x0040
biXPelsPerMeter	0x0018	0x0044
biYPelsPerMeter	0x001C	0x0048
biClrUsed	0x0020	0x004C
biClrImportant	0x0024	0x0050
	SETUP	
FrameRate16	0x0000	0x0054
Shutter16	0x0002	0x0056
PostTrigger16	0x0004	0x0058
FrameDelay16	0x0006	0x005A
AspectRatio	0x0008	0x005C
Res7	0x000A	0x005E
Res8	0x000C	0x0060
Res9	0x000E	0x0062
Res10	0x000F	0x0063
Res11	0x0010	0x0064
TrigFrame	0x0011	0x0065
Res12	0x0012	0x0066
DescriptionOld	0x0013	0x0067
Mark	0x008C	0x00E0
Length	0x008E	0x00E2
Res13	0x0090	0x00E4
SigOption	0x0092	0x00E6
BinChannels	0x0094	0x00E8
SamplesPerImage	0x0096	0x00EA
BinName	0x0097	0x00EB
AnaOption	0x00EF	0x0143

Field	Structure offset	File offset
AnaChannels	0x00F1	0x0145
Res6	0x00F3	0x0147
AnaBoard	0x00F4	0x0148
ChOption	0x00F5	0x0149
AnaGain	0x0105	0x0159
AnaUnit	0x0125	0x0179
AnaName	0x0155	0x01A9
lFirstImage	0x01AD	0x0201
dwImageCount	0x01B1	0x0205
nQFactor	0x01B5	0x0209
wCineFileType	0x01B7	0x020B
szCinePath	0x01B9	0x020D
Res14	0x02BD	0x0311
Res15	0x02BF	0x0313
Res16	0x02C0	0x0314
Res17	0x02C1	0x0315
Res18	0x02C3	0x0317
Res19	0x02CB	0x031F
Res20	0x02D3	0x0327
Res1	0x02D5	0x0329
Res2	0x02D9	0x032D
Res3	0x02DD	0x0331
ImWidth	0x02E1	0x0335
ImHeight	0x02E3	0x0337
EDRShutter16	0x02E5	0x0339
Serial	0x02E7	0x033B
Saturation	0x02EB	0x033F
Res5	0x02EF	0x0343
AutoExposure	0x02F0	0x0344
bFlipH	0x02F4	0x0348
bFlipV	0x02F8	0x034C
Grid	0x02FC	0x0350
FrameRateInt	0x0300	0x0354
Shutter	0x0304	0x0358
EDRShutter	0x0308	0x035C
PostTrigger	0x030C	0x0360
FrameDelay	0x0310	0x0364
bEnableColor	0x0314	0x0368
CameraVersion	0x0318	0x036C
FirmwareVersion	0x031C	0x0370
SoftwareVersion	0x0320	0x0374
RecordingTimeZone	0x0324	0x0378
CFA	0x0328	0x037C
Bright	0x032C	0x0380
Contrast	0x0330	0x0384
GAMMA	0x0334	0x0388
Res21	0x0338	0x038C
AutoExpLevel	0x033C	0x0390
AutoExpSpeed	0x0340	0x0394
AutoExpRect	0x0344	0x0398
WBGain	0x0354	0x03A8
Rotate	0x0374	0x03C8

Field	Structure offset	File offset
WBView	0x0378	0x03CC
RealBPP	0x0380	0x03D4
Conv8Min	0x0384	0x03D8
Conv8Max	0x0388	0x03DC
FilterCode	0x038C	0x03E0
FilterParam	0x0390	0x03E4
UF	0x0394	0x03E8
BlackCalSVer	0x0404	0x0458
WhiteCalSVer	0x0408	0x045C
GrayCalSVer	0x040C	0x0460
bStampTime	0x0410	0x0464
SoundDest	0x0414	0x0468
FRPSteps	0x0418	0x046C
FRPImgNr	0x041C	0x0470
FRPRate	0x045C	0x04B0
FRPExp	0x049C	0x04F0
MCCnt	0x04DC	0x0530
MCPercent	0x04E0	0x0534
CICalib	0x05E0	0x0634
CalibWidth	0x05E4	0x0638
CalibHeight	0x05E8	0x063C
CalibRate	0x05EC	0x0640
CalibExp	0x05F0	0x0644
CalibEDR	0x05F4	0x0648
CalibTemp	0x05F8	0x064C
HeadSerial	0x05FC	0x0650
RangeCode	0x060C	0x0660
RangeSize	0x0610	0x0664
Decimation	0x0614	0x0668
MasterSerial	0x0618	0x066C
Sensor	0x061C	0x0670
ShutterNs	0x0620	0x0674
EDRShutterNs	0x0624	0x0678
FrameDelayNs	0x0628	0x067C
ImPosXAcq	0x062C	0x0680
ImPosYAcq	0x0630	0x0684
ImWidthAcq	0x0634	0x0688
ImHeightAcq	0x0638	0x068C
Description	0x063C	0x0690
RisingEdge	0x163C	0x1690
FilterTime	0x1640	0x1694
LongReady	0x1644	0x1698
ShutterOff	0x1648	0x169C
Res4	0x164C	0x16A0
bMetaWB	0x165C	0x16B0
Hue	0x1660	0x16B4
BlackLevel	0x1664	0x16B8
WhiteLevel	0x1668	0x16BC
LensDescription	0x166C	0x16C0
LensAperture	0x176C	0x17C0
LensFocusDistance	0x1770	0x17C4
LensFocalLength	0x1774	0x17C8

Field	Structure offset	File offset
fOffset	0x1778	0x17CC
fGain	0x177C	0x17D0
fSaturation	0x1780	0x17D4
fHue	0x1784	0x17D8
fGamma	0x1788	0x17DC
fGammaR	0x178C	0x17E0
fGammaB	0x1790	0x17E4
fFlare	0x1794	0x17E8
fPedestalR	0x1798	0x17EC
fPedestalG	0x179C	0x17F0
fPedestalB	0x17A0	0x17F4
fChroma	0x17A4	0x17F8
ToneLabel	0x17A8	0x17FC
TonePoints	0x18A8	0x18FC
fTone	0x18AC	0x1900
UserMatrixLabel	0x19AC	0x1A00
EnableMatrices	0x1AAC	0x1B00
cmUser	0x1AB0	0x1B04
EnableCrop	0x1AD4	0x1B28
CropRect	0x1AD8	0x1B2C
EnableResample	0x1AE8	0x1B3C
ResampleWidth	0x1AEC	0x1B40
ResampleHeight	0x1AF0	0x1B44
fGain16_8	0x1AF4	0x1B48
FRPShape	0x1AF8	0x1B4C
TrigTC	0x1B38	0x1B8C
fPbRate	0x1B40	0x1B94
fTcRate	0x1B44	0x1B98
CineName	0x1B48	0x1B9C
fGainR	0x1C48	0x1C9C
fGainG	0x1C4C	0x1CA0
fGainB	0x1C50	0x1CA4
cmCalib	0x1C54	0x1CA8
fWBTemp	0x1C78	0x1CCC
fWBCc	0x1C7C	0x1CD0
CalibrationInfo	0x1C80	0x1CD4
OpticalFilter	0x2080	0x20D4
GpsInfo	0x2480	0x24D4
Uuid	0x2580	0x25D4
CreatedBy	0x2680	0x26D4
RecBPP	0x2780	0x27D4
LowestFormatBPP	0x2784	0x27D8
LowestFormatQ	0x2786	0x27DA
fToe	0x2788	0x27DC
LogMode	0x278C	0x27E0
CameraModel	0x2790	0x27E4
WBType	0x2890	0x28E4
fDecimation	0x2894	0x28E8
MagSerial	0x2898	0x28EC
CSSerial	0x289C	0x28F0
dFrameRate	0x28A0	0x28F4
SensorMode	0x28A8	0x28FC

Field	Structure offset	File offset
UndecFirst	0x28AC	0x2900
ends at	0x28B0	0x2904