Developer Guide

Phantom SDK Demo

Version 13.4.787.0 (June 2019)

Software release

Some new fields of the SETUP structure may be added in new software releases. This document is based on software release 787 (PhCon.dll, PhFile.dll, PhInt.dll version 13.4.787.0, PCC version 3.4.787.0).

VISION RESEARCH

AMETEK®
MATERIALS ANALYSIS DIVISION

100 Dey Road
Wayne, NJ 07470 USA
+1.973.696.4500
phantom@visionresearch.com

www.phantomhighspeed.com

Contents

1. Introduction

The demo programs for Phantom SDK are examples on how to use the functions to control the Phantom high speed Cameras.

The functions available in SDK allow you to set the camera parameters, record a cine (movie) in the camera Frame Buffer Memory (FBM) and transfer the images to the client computer using the available connection.

The demo has two versions – one written in C++ (PhDemoCPP) and the other in C Sharp (PhDemoCS). It offers you an example of sources for an application built using the functions in Phantom libraries. It contains the basic controls you need to get images from a camera, to set and get acquisition parameters and image parameters, to record and save cines to files and to playback the recordings either from camera or from file.

The two versions of the applications are similar and the documentation is unique.

2. Main classes

2.1. Camera class

The Camera class is used for controlling the camera, setting camera options or parameters and reading camera info. It also keeps a reference to a live cine and a selected cine.

2.2. Cine class

The cine class is used to read cine acquisition parameters, read or set the image processing options and provide functions to read cine images and save a cine to file. It is built around the encapsulated cine handle structure.

2.3. CineSaver class

The CineSaver links the Cine class and the save cine user interface. It starts a cine save, stops it and it reports the progress.

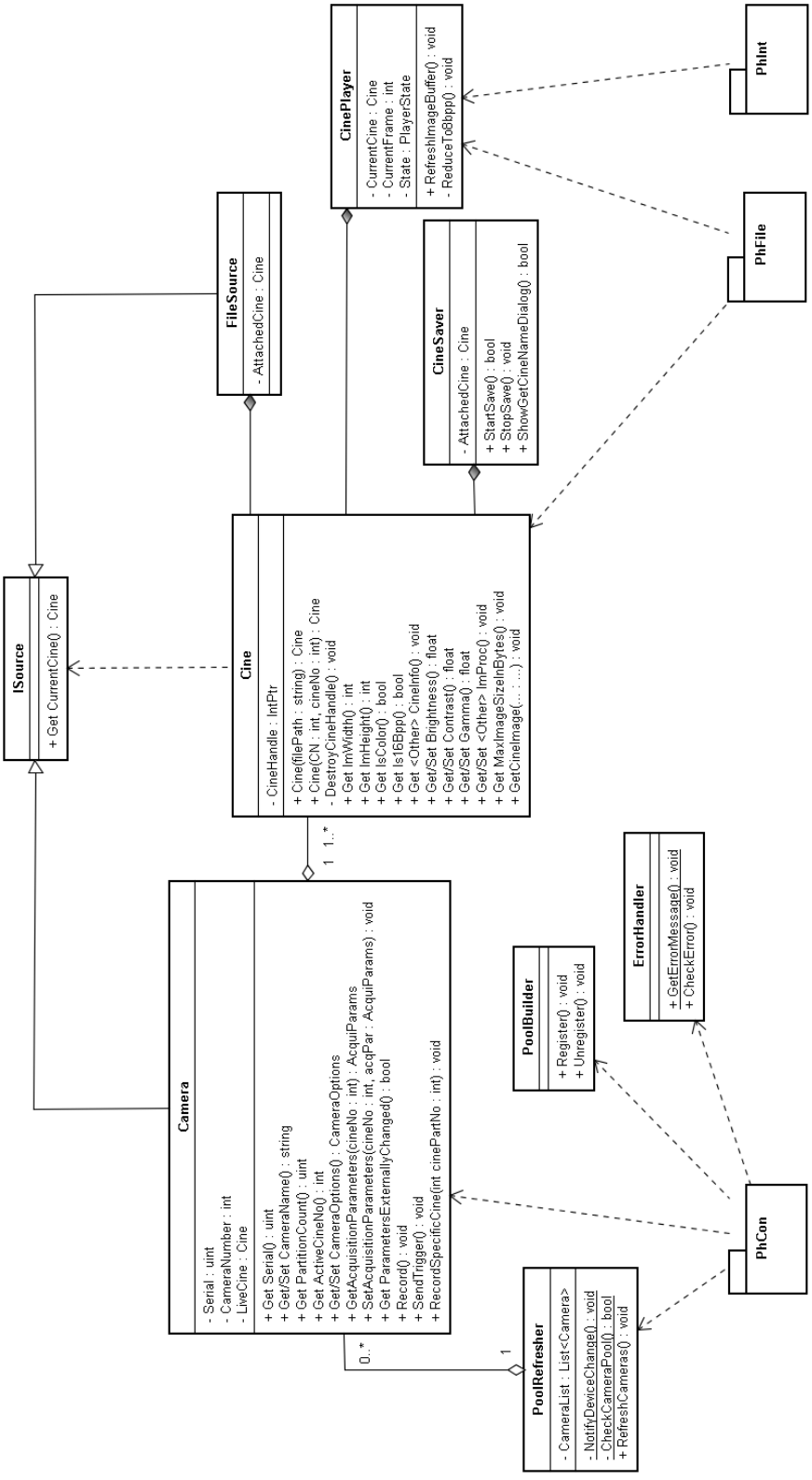
2.4. CinePlayer class

The CinePlayer class is a link between a Cine class and the cine playing interface. It reads cine images, converts them to 8bpp in order to be displayed in a picture box and updates and sets the current image to/from the user interface.

2.5. ErrorHandler class

The ErrorHandler is used to translate phantom dlls error codes to a human readable message and to dispatch them as errors.

3. Simplified Class Diagram



4. Start PhConDemo

You can run the demo connected to a real camera or the demo will simulate an HD camera which shows a software generated image. By pressing CTRL+SHIFT during image displaying, a menu to change the simulated image will appear. You can change the simulated camera model by changing the parameters passed to PhAddSimulatedCamera, in the source code.

When you start the demo, if no settings file (.stg) is available, it reads the settings from the camera and it creates an stg file in your *\Application Data\Phantom\[PhVer]* folder for each connected camera. If any error occurs a dialog with the error text is shown.

SDK functions:

- PhRegisterClientEx
- PhGetCameraCount
- PhAddSimulatedCamera

Classes:

- PoolBuilder
- PoolRefresher

5. Refresh Camera Pool

The *Refresh Camera Pool* button will refresh the list of available cameras. If there are no online cameras a simulated camera will be added by default. A live cine is automatically created for every camera in the online list. Available cameras are shown in the *Source* list. You may choose one from this list. If you disconnect a camera, the list will be updated by pressing *Refresh Camera Pool* button.

SDK functions:

- PhCheckCameraPool
- PhNotifyDeviceChangeCB
- PhGetCameraCount
- PhAddSimulatedCamera
- PhGetCameraID
- PhGetCineLive

Classes:

- PoolRefresher
- Camera
- Cine

6. Camera Info

By pressing the button called *Camera Info* it will display information on the selected camera. You may change here the camera name.

SDK functions:

- PhGetCameraID
- PhSetCameraName
- PhGet
- PhGetVersion

Classes:

- Camera

7. The Cine List

Every connected camera has a list of available RAM cine partitions and flash cines. The list of cine numbers are placed in the cine combo, where the flash cines are prefixed with F. If a stored cine is selected a player interface is shown allowing user to play and seek available images, otherwise the live images are continuously taken from camera and the recording is started in the selected partition. The *Preview* cine cannot be recorded.

The list may contain a cine file too, if a cine file has been opened and *File* item is selected in the *Source* combo.

You may check the cine acquisition and image processing parameters on the right panel. The number of camera partitions can be set in Camera Settings group -> partitions combo.

SDK functions:

- PhGetPartitions
- PhSetPartitions
- PhGetCineCount
- PhGetCineStatus
- PhGetCineLive
- PhNewCineFromCamera

Classes:

- Camera
- Cine

8. The Cine Player

After selecting a cine from camera or a file, images are taken from camera or from the file. If the partition you select does not contain a recorded cine the application will display live images. Otherwise, if the cine is recorded, a play interface shows up and allows you to play and seek cine images.

When the images are displayed, the image processing options you have set are applied. Image bit depths larger than 8bpp are reduced to 8 for display. The demo will fit the image dimensions to the size of the picture box, keeping aspect ratio.

SDK functions:

- PhGetCineImage
- PhProcessImage
- PhGetCineInfo
- PhSetCineInfo

Classes:

- Camera
- Cine
- CinePlayer

9. Setting Acquisition Parameters

On the right panel there is a group box which shows the acquisition parameters associated with the selected partition. If the cine is an empty camera partition you may change these parameters and apply them using the *Set* button. In this demo just a few parameters are exemplified. If you wish more advanced settings you should review the structures ACQUIPARAMS, CAMERAOPTIONS, the cgs... and the gs... selectors.

SDK functions:

- PhGetCineParams
- PhSetCineParams
- PhSetSingleCineParams
- PhGetCameraOptions
- PhSetCameraOptions

Classes:

- Camera

10. Metadata and Image Processing Options

After a cine is recorded in the camera, the recording parameters are stored as "metadata". The metadata provides information about the recording event and essential parameters to decode the pixels like the image width, height and bit depth. Most of the metadata is readonly only the image processing settings can be changed after the recordings to enhance the images.

On the right panel there is a group box which contains image processing options. These settings can always be changed, but there are a few of them which can be applied only on color images. Change the options and use *Set* button to apply them. Please be aware that these changes are not written in the RAW cine file automatically.

SDK functions:

- PhGetCineInfo
- PhSetCineInfo

Classes:

- Cine

11. Recording Commands

The *Capture* and *Trigger* commands are used to start the recording and respectively to trigger the current recording. Select an empty partition number from RAM to start recording in that partition of the camera Frame Buffer Memory, press *Trigger* to stop recording after the post trigger frames. If the partition has a cine stored you can press *Capture* to delete the cine and start a new recording. While in Preview a capture command will delete all cines and start recording in the first partition.

If you want to view the stored images, wait until the camera finishes recording the post trigger frames of the current cine partition. The camera will then return to preview or will start recording in the next empty partition. You may check the camera status at the bottom of the PhDemo window.

When post trigger frames is larger than actual partition capacity, the difference will work as a recording delay, storing only the last images that fit into the partition capacity and overwriting older images from the circular image buffer.

A few Phantom devices like the CineStation do not have acquisition parts and as a result you cannot send recording commands or read live images.

SDK functions:

- PhRecordCine
- PhRecordSpecificCine
- PhSendSoftwareTrigger
- PhGet

Classes:

- Camera

12. Cine Open and Save Actions

The *Open Cine* button can be used to start browsing and open a cine file from a data storage device. After opening a file, the *File* item will be available to be selected in the *Source* combo, indicating that a cine file is opened. The cine list is populated with the cine file name.

Every time a new cine file is opened, the old one is replaced. If a cine file is selected, a player interface is shown and you can also change its image processing options.

When a camera recorded cine is selected the *Save Cine* button can be used to save it into a file. The save progress is shown at the bottom of the PhDemo window. Closing the application while a save is in progress will abort it and the already saved images from the cine file will be usable.

SDK functions:

- PhNewCineFromFile
- PhWriteCineFileAsync
- PhStopWriteCineFileAsync
- PhGetOpenCineName
- PhGetSaveCineName
- PhGetCineInfo
- PhDuplicateCine
- PhSetUseCase

Classes:

- Cine
- CineSaver

13. Closing PhDemo

To properly exit the *PhDemo* application use the *Close* (X) button from the upper right corner of the window. Before exiting the application disposes Phantom dlls memory resources.

SDK functions:

- PhDestroyCine
- PhUnregisterClient