



*FASTCAM Control
SDK Library "Pcc4VB"
Reference Manual*

*Rev.2.976
English Edition*

*PHOTRON LIMITED
2004-2006*

Table of contents

1. [SDK Introduction](#)
2. [How to Use](#)
 - 2.1. [Who Should Read This Reference Manual](#)
 - 2.2. [Required Operational Environment for This Library](#)
 - 2.3. [Structure of The Library](#)
 - 2.3.1. [Recording Functions](#)
 - 2.3.2. [File Structure](#)
 - 2.4. [Installation in Development Environment](#)
 - 2.4.1. [Setup for Visual Basic Project/
Visual Basic .NETProject](#)
 - 2.4.2. [To run application](#)
3. [Fastcam Control SDK common items](#)
 - 3.1. [Error Codes](#)
 - 3.2. [Camera Model Codes](#)
 - 3.3. [Camera Parameters](#)
 - 3.3.1. [\[CAMERA_PARAMS\]Structure](#)
 - 3.3.2. [\[CAMERA_PARAMS_EX\]Structure](#)
 - 3.4. [Frame Parameters](#)
 - 3.4.1. [Frame Number Management](#)
 - Device-independent Frame
 - Device-dependent Frame
 - 3.4.2. [\[FRAME_PARAMS\]Structure](#)
 - 3.5. [IRIG Timecode](#)
 - 3.5.1. [\[IRIGLIB_TIMECODE\]Structure](#)
4. [CameraControl Functions](#)
 - 4.1. [CameraControl Overview](#)
 - 4.1.1. [About CameraControls](#)
 - 4.1.2. [Extra Commands](#)
 - 4.2. [CameraControl Function List](#)
 - 4.3. [CameraControl Function Specifications](#)
5. [ImageData Funcitons](#)
 - 5.1. [ImageData Overview](#)
 - 5.1.1. [About ImageData](#)
 - 5.2. [ImageData Function List](#)
 - 5.3. [ImageData Function Specifications](#)

6. Compatible with Old Version SDK (Unused)
/Compatible with Old Version SDK

7. File Formats

- 7.1. BMP File (*.bmp)
- 7.2. TIFF File (*.tif)
- 7.3. JPEG File (*.jpg)
- 7.4. PNM File (*.ppm /*.pgm)
- 7.5. RAW File (*.raw/* raww)
- 7.6. PNG File (*.png)

8. Look Up Table(LUT)

- 8.1. Example of application: Negative inversion of image

1. SDK Introduction

This FASTCAM Control SDK Library is a Windows-based class library that controls, from the PC screen, all operational functions – setting parameters, recording and downloading of image data - of the Photron FASTCAM series high-speed video cameras. Its main functions are as follows:

■Control Functions of High-Speed Video Cameras from PC

- Connects to PC via the PCI, IEEE1394, Optical or Ethernet interface.
- Sets camera parameters for recording.
- Selects a trigger mode to start recording.

■Download of Recorded Image Data

- Downloads image data from the camera memory to PC for subsequent processing.
- Corrects image data using LUT (look up table).
- Stores image data in BMP, TIFF, JPEG or RAW image format.

2. How to Use the SDK Library

This section discusses how to install the FASTCAM Control SDK Library in software development environment for actual use.

To use this Library, you need to have your high-speed video camera set up for recording and your PC and necessary hardware devices connected and installed. See the relevant instruction manuals for connection and installation.

Installation of hardware driver varies by the camera model, and each hardware driver must be installed before using this Library. See the driver installation manual of each hardware device.

- 2.1. [Who Should Read This Reference Manual](#)
- 2.2. [Required Operational Environment for This Library](#)
- 2.3. [Structure of The Library](#)
 - 2.3.1. [Recording Functions](#)
 - 2.3.2. [File Structure](#)
- 2.4. [Installation in Development Environment](#)
 - 2.4.1. [Setup for Visual Basic Project/
Visual Basic .NETProject](#)
 - 2.4.2. [To run application](#)

2.1. Who Should Read This Reference Manual

This reference manual has been prepared for users of Photron high-speed video cameras who wish to develop PC control application software programs. Such readers should have basic knowledge of and expertise in computer languages such as the Visual Basic6.0, and should be able to understand the programming words used in this book.

2.2. Required Operational Environment for This Library

The following environment is required to run this Library:

PC :	PC/AT Compatible
OS:	Microsoft Windows98、98SE、Me、2000 Professional、XP Professional、XP Home ※1 ※2
CPU :	Intel Pentium or equivalent PentiumIII 1GHz or higher recommended ※2
Memory :	64MB minimum 256MB or more recommended* *Required for multiple cameras or high-resolution cameras
HDD :	5MB or more needed for installing Library 500MB or more recommended for application development works
Development Environment :	Microsoft VisualBasic6.0 or later ※2
Other requirements :	Large capacity HDD or removable recording media recommended for image data storage. CD-ROM drive required for installaion.

※1 Note: The combination of camera and PC may require a particular OS. See the instruction manual of the camera

※2 Note: Trademarks or registered trademarks belong to their respective owners.

2.3. Structure of The Library

The classes contained in this library and the structure of library files needed for application development are as follows:

2.3.1. Recording Functions

CameraControl (Camera Control Functions)

- Setting up camera status
- Getting camera status
- Starting a recording
- Downloading recorded image data

ImageData Class (Still Image Data Management Functions)

- Input and output of still image data file(in BMP, TIFF, JPEG, RAW, PNM, PNG or RAWW format)
- Converting still image data by lookup table

2.3.2 File Structure

The file structures contained in the SDK Library is as follows:

Dll¥

PccLib.dll	PccLib library main DLL
Pcc4VB.dll	Pcc4VB library main DLL (VisualBasic6.0/VisualBasic.NET Only)
FcamPCI.dll	FASTCAM-PCI device DLL
FcamPCI2.dll	FASTCAM-PCI R2 device DLL
FcamPLMV.dll	FASTCAM -X 1280PCI device DLL
Fcam1394.dll	IEEE1394I/F-compatible FASTCAM series device DLL
FcamEth.dll	Ethernet-compatible FASTCAM series device DLL
FcamOpt.dll	Optical I/F-compatible FASTCAM series device DLL
FcamNPCi.dll	FASTCAM-512PCI device DLL
FcamPPCI.dll	FASTCAM-1024PCI device DLL
FcamGEth.dll	1000Base-T compatible FASTCAM series device DLL
GEthLib.dll	1000Base-T compatible FASTCAM series communication DLL
Iljl15.dll	JPEG Library DLL
IrigLib.dll	IRIG library DLL

Modules¥

FastcamDllDriver.bas(VisualBaic6.0)
FastcamDllConstantValues.bas(VisualBaic6.0)
Pcc4VB library definition File

FastcamDllDriver.vb(VisualBaic.NET)
FastcamDllConstantValues.vb(VisualBaic.NET)
Pcc4VB library definition File

2.4. Installation in Development Environment

This section shows the procedure for installing the Library in an application development environment (VisualBasic 6.0/Visual Basic.NET)

2.4.1. Setup for Visual Baisc Project/ Visual Basic .NETProject

1) Setup for Visual Baisc 6.0 Project

Add the Library to Visual Baisc 6.0 in the following manner:

1. Start up Visual Baisc 6.0 and open a project in which this Library is to be installed.
2. Select "Project" from the Visual Baisc 6.0 menu and open the "Add Files" dialog.
3. add two definition files for libraries to a project
(**FastcamDIIDriver.bas/FastcamDIIConstantValues.bas**)

2) Setup for VisualBasic.NET Project

Add the Library to VisualBasic.NET in the following manner:

1. Start up VisualBasic.NET and open a project in which this Library is to be installed.
2. Select "Project" from the VisualBasic.NET menu and open the "Add Existing Item" dialog.
3. add two definition files for libraries to a project
(**FastcamDIIDriver.bas/FastcamDIIConstantValues.bas**)

2.4.2. To run applications

When you run applications with this Library incorporated, you need to place relevant DLL files in the same folders as the execution files or in the folders in the system where a path has been set.

Note: DLL files may be copied in the Windows system folder. In such a case, however, the specification of each file must usually be updated at software revision. If the DLL files are left without being updated, the revised SDK may erroneously refer to those old DLL files.

3. Fastcam Control SDK common items

This section presents information on the details of the SDK Library.

Items discussed in this section:

- 3.1 [Error Codes](#)
- 3.2 [Camera Model Codes](#)
- 3.3 [Camera Parameters](#)
 - 3.3.1 [\[CAMERA_PARAMS\]Structure](#)
 - 3.3.2 [\[CAMERA_PARAMS_EX\]Structure](#)
- 3.4 [Frame Parameters](#)
 - 3.4.1 [Frame Number Management](#)
 - Device-independent Frame
 - Device-dependent Frame
 - 3.4.2 [\[FRAME_PARAMS\]Structure](#)
- 3.5 [IRIG Timecode](#)
 - 3.5.1 [\[IRIGLIB_TIMECODE\]Structure](#)

3.1. Error Codes

With the Fastcam Control SDK, the returned values are error codes.

Error codes are shown below:

PCC_ERROR_NOERROR	1	Normal
PCC_ERROR_UNINITIALIZE	-1	Not initialized
PCC_ERROR_FUNCTION_FAILED	-2	Command failed
PCC_ERROR_FUNCTION_TIMEOUT	-3	Command time out
PCC_ERROR_FUNCTION_DISABLE	-4	Disabled command or no such function
PCC_ERROR_NO_DEVICE	-5	Device not existent
PCC_ERROR_NO_DATA	-6	Image data not existent
PCC_ERROR_UNKNOWN_FRAME	-7	Frame number not existent or incorrect
PCC_ERROR_CAMERAMODE	-8	Incorrect camera mode
PCC_ERROR_NO_ENDLESS	-9	Trigger mode not endless
PCC_ERROR_FILEREAD_FAILED	-10	File reading failed
PCC_ERROR_FILEWRITE_FAILED	-11	File writing failed
PCC_ERROR_IMAGE_SIZEOVER	-12	Image size error
PCC_ERROR_FRAME_AREAOVER	-13	Frame area error (exceeding)
PCC_ERROR_PLAYMODE	-14	Playback mode error
PCC_ERROR_NO_CAMERA_LIB	-15	Unknown camera library (not found)
PCC_ERROR_NO_MCDL	-16	MCDL data not existent in camera
PCC_ERROR_NO_IRIG	-17	IRIG data not existent in camera
PCC_ERROR_NOT_SUPPORTED	-18	Specified command not supported

3.2. Camera Model Codes

[GetDeviceType] function of the CameraControl Functions gets the model code of cameras being used.

The model codes and the corresponding model names are as follows:

PCAM_TYPE_FASTCAM_NET_COLOR	FASTCAM-Super (color model)
PCAM_TYPE_ULTIMA_1024	FASTCAM-Ultima1024 (monochrome model)
PCAM_TYPE_ULTIMA_1024_ZERO	FASTCAM-Ultima1024 (monochrome model)
PCAM_TYPE_ULTIMA_1024_COLOR	FASTCAM-Ultima1024 (color model)
PCAM_TYPE_ULTIMA_1024_COLOR_ZERO	FASTCAM-Ultima1024 (color model)
PCAM_TYPE_ULTIMA_SE	FASTCAM-UltimaSE (monochrome model)
PCAM_TYPE_ULTIMA_SE_COLOR	FASTCAM-UltimaSE (color model)
PCAM_TYPE_FASTCAM_PCI	FASTCAM-PCI (monochrome model)
PCAM_TYPE_FASTCAM_PCI_COLOR	FASTCAM-PCI (color model)
PCAM_TYPE_FASTCAM_1280PCI	FASTCAM-X1280PCI (monochrome model)
PCAM_TYPE_FASTCAM_1280PCI_COLOR	FASTCAM-X1280PCI (color model)
PCAM_TYPE_FASTCAM_MAX	FASTCAM-UltimaAPX (monochrome model)
PCAM_TYPE_FASTCAM_MAX_COLOR	FASTCAM-UltimaAPX (color model)
PCAM_TYPE_FASTCAM_NEO	FASTCAM-Ultima512 (monochrome model)
PCAM_TYPE_FASTCAM_NEO_COLOR	FASTCAM-Ultima512 (color model)
PCAM_TYPE_FASTCAM_PCI2	FASTCAM-PCI R2 (monochrome model)
PCAM_TYPE_FASTCAM_PCI2_COLOR	FASTCAM-PCI R2 (color model)
PCAM_TYPE_FASTCAM_512PCI	FASTCAM-512PCI (monochrome model)
PCAM_TYPE_FASTCAM_512PCI_COLOR	FASTCAM-512PCI (color model)
PCAM_TYPE_FASTCAM_APX_RS	FASTCAM-APX RS (monochrome model)
PCAM_TYPE_FASTCAM_APX_RS_COLOR	FASTCAM-APX RS (color model)
PCAM_TYPE_FASTCAM_1024PCI	FASTCAM-1024PCI (monochrome model)
PCAM_TYPE_FASTCAM_1024PCI_COLOR	FASTCAM-1024PCI (color model)

3.3. Camera Parameters

All the information pertaining to the camera is included in the CAMERA_PARAMS structure. It is possible to change recording rate and other parameters by directly resetting the CMARA_PARAMS structure, but it may take much time to reset all the items with some of the camera models. So, it is advisable to use CMARA_PARAMS for data reference only, and use individual setting functions for camera setup. Model-dependant information or information on extra commands are found in the CAMERA_PARAM_EX structure.

For parameters common with CAMERA_PARAMS and CAMERA_PARAMS_EX, it is advisable to use the CAMERA_PARAM_EX structure as much as possible.

※This SDK uses 32-bit variables for the VisualBasic6.0 throughout.

When you wish to use this SDK with the VisualBasic.NET, you need to convert the variables to VisualBasic.NET 32-bit variables to interpret them.

※Before you use your user-defined structure with the VisualBasic.NET, you must initialize the structure.

3.3.1[CAMERA_PARAMS] Structure

Note: LMN stands for LIST_MAX_NUMBER

type CAMERA_PARAMS

device_name(63) As Byte	Camera name
device_type As Long	Camera model code
device_version As Long	Camera hardware version
color As Long	Color mode (0:MONO 1:COLOR)
session_number As Long	Recording session number
max_frame_num As Long	Number of frames
camera_mode As Long	Camera mode
is_live As Long	Display mode (0:PLAY 1:LIVE)
is_record_ready As Long	Record ready (0:OFF 1:ON)
Record_rate As Long	Recording rate
record_rate_num As Long	Number of recording rates (list)
record_rate_list(LMN - 1) As Long	Recording rate list
resolution As Long	Resolution
resolution_num As Long	Number of resolutions (list)
resolution_list(LMN - 1) As Long	Resolution list
shutter_speed As Long	Shutter speed
shutter_num As Long	Number of shutter speeds (list)
shutter_list(LMN - 1) As Long	Shutter speed list
trigger_mode As Long	Trigger mode
trigger_frame As Long	Trigger frame
manual_trigger_frame As Long	Manual trigger frame
random_trigger_times As Long	Random trigger time
random_manual_frame As Long	Random manual frame
random_trigger_num As Long	Random trigger frame number
trigger_mode_num As Long	Number of trigger modes (list)
trigger_mode_list(LMN - 1) As Long	Trigger mode list
random_number_num As Long	Number of triggers for random trigger
random_number_list(LMN - 1) As Long	Number of frames for random trigger
enable_random_trigger As Long	Enable random trigger Yes/No
enable_manual_trigger As Long	Enable manual trigger Yes/No
enable_random_manual As Long	Enable random man trigger Yes/No
gamma_value As Long	Gamma value

enable_gamma As Long	Enable gamma Yes/No
gamma_value_num As Long	Number of gamma lists
gamma_value_list(LMN - 1) As Long	Gamma list
gain_level As Long	Gain level
enable_gain As Long	Enable gain Yes/No
gain_level_num As Long	Number of gain lists
gain_level_list(LMN - 1) As Long	Gain list
external_sync As Long	External sync
external_sync_neg As Long	External sync neg signal mode
external_sync_in As Long	External sync in
external_sync_out As Long	External sync out
enable_external_sync As Long	Enable external sync Yes/No
enable_external_sync_in As Long	Enable external sync in Yes/No
enable_external_sync_out As Long	Enable external sync out Yes/No
enable_external_in As Long	General purpose ext input Yes/No
enable_external_out As Long	General purpose ext output Yes/No
external_in_num As Long	Number of gen purpose ext input
external_out_num As Long	Number of gen purpose ext output
external_in(LMN - 1) As Long	Gen purpose ext input list
external_out(LMN - 1) As Long	Gen purpose ext output list
color_temp_mode As Long	Color temperature mode
color_temp_mode_num As Long	Number of color temperature modes
color_temp_mode_list(LMN - 1) As Long	Color temperature mode list
color_temp_user_num As Long	Number of user-set modes
color_temp_user(LMN - 1, 2) As Long	User-set color temperature (RGB)
enable_color_temp As Long	Enable color temperature Yes/No
external_out(LMN - 1) As Long	External signal output mode(Port No.)
external_out_num As Long	Number of external signal output ports
enable_external_out As Long	Enable external signal output Yes/No
monitor_play_rate As Long	External monitor playback rate
monitor_out_mode As Long	External monitor output mode
monitor_edge_mode As Long	Ext monitor edge enhancement mode
monitor_zoom_on As Long	External monitor output zoom mode
monitor_block_on As Long	Ext monitor output block playback mode
monitor_block_start As Long	Block playback start position
monitor_block_end As Long	Block playback end position
enable_monitor_out As Long	Enable external monitor output Yes/No
enable_monitor_out_mode As Long	Enabel ext monitor output setting Yes/No
enable_monitor_edge As Long	Enable ext monitor output edge enhancement Yes/No
enable_monitor_zoom As Long	Enable ext monitor output zoom Yes/No
mcdl_on As Long	MCDL function ON/OFF setting
enable_mcdl As Long	Enable MCDL function Yes/No
irig_on As Long	IRIG function ON/OFF setting
enable_irig As Long	Enable IRIG function Yes/No
restriction_time_on As Long	Rec time restriction ON/OFF setting
enable_restriction_time As Long	Enable rec time restriction Yes/No
camera_type(63) As Byte	Camera type
exist_variable_func As Long	Availability Variable mode
variable_channel As Long	Using Variable channel no.

End Type

3.3.2. [CAMERA_PARAMS_EX] Structure

Note: PMN stands for PARTITION_MAX_NUMBER.

type CAMERA_PARAMS_EX

exist_mcdl As Long	Availability MCDL function Yes/No
exist_monitor_out As Long	Availability ext monitor output Yes/No
exist_monitor_out_mode As Long	Availability ext monitor output mode Yes/No
exist_monitor_edge As Long	Availability ext monitor output edge enhancemnt Yes/No
exist_monitor_edge As Long	Availability ext monitor output zoom Yes/No
exist_irig As Long	Availability IRIG function Yes/No
exist_restriction_time As Long	Availability rec time restriction Yes/No
exist_device_id As Long	Availability camera ID acquisition Yes/No
exist_reset_trigger As Long	Availability reset trigger Yes/No
exist_sensor_bitshift As Long	Availability sensor bitshift correction Yes/No
exist_partition As Long	Availability partitioning Yes/No
exist_shading_compensation As Long	Availability shading calibration Yes/No
exist_live_resolution As Long	Availability live resolution change Yes/No
exist_lut_mode As Long	Availability LUT change Yes/No
exist_user_lut As Long	Availability user-set LUT Yes/No
exist_chroma_mode As Long	Availability chroma change Yes/No
exist_ds_shutter As Long	Availability dynamic range change Yes/No
monitor_block_on As Long	Ext monitor block playback mode ON
monitor_block_start As Long	Ext monitor block playback start frame
monitor_block_end As Long	Ext monitor block playback end frame
monitor_play_rate As Long	Ext monitro playback rate
monitor_out_mode As Long	Ext monitor output mode
monitor_edge_mode As Long	Ext monitor output edge enhancement mode
monitor_zoom_on As Long	Ext monitor output zoom
mcdl_on As Long	MCDL function ON/OFF setting
irig_on As Long	IRIG function ON/OFF setting
restriction_time_on As Long	Rec time restriction ON/OFF setting
device_id As Long	Camera ID
reset_trigger_on As Long	Reset trigger ON/OFF setting
sensor_bitshift As Long	Grayscale (bitshift) change
max_partition_num As Long	Max number of partitions
max_partition_block_num As Long	Number of blocks in max partition
partition_frame_per_block As Long	Number of frames per block
current_partition As Long	Current partition in use
partition_num As Long	Number of partitions
partition_size_list(255) As Long	Partition block list
shading_compensation_mode As Long	Shading calibration mode
live_resolution_mode As Long	Live resolution change
lut_mode As Long	LUT
chroma_mode As Long	Chroma
ds_shutter As Long	Dynamic range expansion
irig_offset As Long	IRIG offset value
exist_irig_offset As Long	Availability IRIG offset Yes/No

hard_partition As Long	Hardware partition
exist_hard_partition As Long	Availability hardware partition Yes/No
shutter_mode As Long	Shutter mode
exist_shutter_mode As Long	Availability shutter mode Yes/No
II_POWER_ON As Long	I.I. power
II_gain As Long	I.I. gain
II_gate_select As Long	I.I. gate mode
II_gate_cycle As Long	I.I. cycle value
II_gate_width As Long	I.I. width value
II_gate_times As Long	I.I. times value
II_gate_delay As Long	I.I. delay value
II_gain_limit As Long	I.I. gain's under limit value
exist_ii As Long	Availability I.I.
exist_event_frame As Long	Availability event frame Yes/No
head_total_num As Long	Total Number of exchanging head
current_head As Long	Using head no.
exist_edge_enhancement As Long	Availability Edge Enhancement
edge_enhancement As Long	Edge Enhancement
exist_shutter_type2 As Long	Availability Shutter speed type2
shutter_type2_list(127) As Long	Shutter speed type2
shutter_speed_type2 As Long	Number of Shutter speed type2
exist_auto_exposure As Long	Availability auto exposure Yes/No
auto_exposure As Long	Auto Exposure
auto_exposure_reso As Long	Auto Exposure resolution
auto_exposure_value As Long	Auto Exposure value
auto_exposure_range As Long	Auto Exposure range
auto_exposure_maxshutter As Long	Auto Exposure max exposure time
exist_ii_protection As Long	Availability I.I. Protection
II_protection As Long	I.I. Hardware protection level

End Type

3.4. Frame Parameters

3.4.1. Frame Number Management

In this Class Library, the following two types of frame number systems are used to manage movie image data.

①Device-independent Frame

In this Class Library, the sequence of movie image data is managed in the way where the first frame is numbered "0" (zero) and the last frame "the total number of frames minus 1".

When a frame number is used to specify a member function argument, this numbering system, Device-independent Frame, is used.

The frame number of a trigger frame varies by the trigger mode as shown below:

Example: When the total number of frames is 100:

Trigger Mode	Trigger Frame
START	0
CENTER	50
END	99
MANUAL	ny number between 0 and 99

②Device-dependent Frame

The frame number of the start and end frames changes by the trigger mode. The particular frame where a trigger is sent is named Frame 1, regardless of the trigger mode. There is no Frame "0", and the frame right before Frame 1 is Frame -1.

For working out applications, it is necessary to take the factor into consideration, that is, a Device-independent frame number must be converted to a Device-dependent frame number, and vice versa, as necessary.

3.4.2[FRAME_PARAMS] Structure

Parameters for frame number management are included in the FRAME_PARAMS structure.

※This SDK uses 32-bit variables for the VisualBasic6.0 throughout.

When you wish to use this SDK with the VisualBasic.NET, you need to convert the variables to VisualBasic.NET 32-bit variables to interpret them.

type FRAME_PARAMS

frame_start As Long	Start frame
frame_end As Long	End frame
frame_total As Long	Total number of frames
area_start As Long	Start frame of an area
area_total As Long	End frame of an area
area_total As Long	Total number of frames in an area
current As Long	Current frame
trigger As Long	Trigger frame
key As Long	Key frame

End Type

3.5. IRIG Timecode

3.5.1.[IRIGLIB_TIMECODE] Structure

IRIG parameters are included in the IRIGLIB_TIMECODE Structure.

※This SDK uses 32-bit variables for the VisualBasic6.0 throughout.
When you wish to use this SDK with the VisualBasic.NET, you need to convert the variables to VisualBasic.NET 32-bit variables to interpret them.

```
type IRIG_TIMECODE
    year As Long           Year
    month As Byte          Month
    day As Byte            Day
    day_of_year As Long    Day 1 (Jan. 1) to Day 365 or 366 (Dec. 31)
    hour As Byte           Hours
    minute As Byte         Minutes
    second As Byte         Seconds
    below_second As Long   Micro seconds
End Type
```

Note: Normally, year, month and day are not used.

4.CameraControl

This section describes the features and member function specifications the CameraControl Functions in the following subsections:

- 4.1 [CameraControl Overview](#)**
- 4.2 [CameraControl Function List](#)**
- 4.3 [CameraControl Function Specifications](#)**

4.1. CameraControl Overview

4.1.1. About CameraControl

The CameraControl Functions extends overall control on camera parameter setting and operation including recording. It also controls playback and displays recorded image on the monitor screen while downloading the image data to the PC memory.

Note: There may be functions that cannot be used with a camera having no monitor output.

Each camera requires its own CameraControl. An ID number is assigned to the camera from an **Init** function after a CameraControl instance has been generated. The assigned camera will be controlled by the instance until **ExitCameraControl** function is called out.

■CameraControl Functions

- Initialization Function
- Camera Information Get Function
- Camera Setup Function
- Camera Setup Get Function
- Camera Mode Function
- Recording Function
- Playback Function
- Playback Setup Function
- Image Data Get Function
- Other Special Functions

4.1.2. Extra Commands

[Extra commands](#) are issued for each of camera models using the CameraControl [\[ExtraCommand\] functions](#).

4.2. CameraControl Function List

List of Functions of CameraControl

Construction	
CameraControl	Construct CameraControl
Initialization	
GetNumberOfDevice	Get the number of cameras connected
Init	Initialize specified cameras
ExitCameraControl	End processing
IsInit	Confirm initialization status
SetCameraNumber	Set Operation camera number
GetCameraNumber	Get Operation camera number
Get Camera Information	
GetDeviceName	Get the camera name
GetDeviceType	Get the camera model code
GetDeviceVersion	Get the camera hardware version information
IsColor	Identify the camera is color or monochrome
HaveGammaCorrection	Gamma correction feature Yes/No
HaveGainLevel	Gain level setup feature Yes/No
HaveColorTemperature	Color temperature setup feature Yes/No
HaveExternalSyncMode	External sync drive feature Yes/No
HaveExternalSyncInMode	External sync-in feature Yes/No
HaveExternalSyncOutMode	External sync-out feature Yes/No
HaveExternalInput	External general purpose signal-in feature Yes/No
HaveExternalOutput	External general purpose signal-out feature Yes/No
HaveEdgeEnhancement	Edge enhancement feature Yes/No (on monitor output)
HaveZoomMode	Zoom display feature Yes/No (on monitor output)
HaveBlockPlay	Block (range) playback feature Yes/No (on monitor output)
HaveMonitorOutMode	External monitor output feature Yes/No
HaveMCDL	MCDL feature Yes/No
HaveIRIG	IRIG feature Yes/No

Camera Setups	
<u>SetCameraParams</u>	Set camera parameters
<u>SetCameraParamsEx</u>	Set camera parameters simultaneously
<u>SetRecordRate</u>	Set frame rate
<u>SetShutterSpeed</u>	Set shutter speed
<u>SetTriggerMode</u>	Set trigger mode
<u>SetRandomTriggerNumber</u>	Set random trigger frame number
<u>SetManualTriggerPosition</u>	Set manual trigger frame position
<u>SetRandomManualPosition</u>	Set random-manual trigger frame position
<u>SetResolution</u>	Set resolution
<u>SetGammaCorrection</u>	Set gamma
<u>SetGainLevel</u>	Set gain level
<u>SetColorTemperature</u>	Set color temperature
<u>SetColorTempUser</u>	Set user color temperature
<u>SetExternalSyncMode</u>	Set external sync drive (Master/Slave)
<u>SetExternalSyncInMode</u>	Set external sync-in drive
<u>SetExternalSyncModeEx</u>	Set external sync drive
<u>SetExternalSyncOutMode</u>	Set external sync-out drive
<u>SetExternalInMode</u>	Set external general purpose signal-in
<u>SetExternalOutMode</u>	Set external general purpose signal-out
<u>SetEdgeEnhancementMode</u>	Set edge enhancement (on monitor output)
<u>SetZoomMode</u>	Set zoom mode On/Off (on monitor output)
<u>SetMonitorOutMode</u>	Set external monitor output setting
<u>SetEnableMCDL</u>	Set MCDL feature On/Off
<u>SetEnableIRIG</u>	Set IRIG feature On/Off
<u>SetPartition</u>	Set partition No.
<u>SetPartitionBlockList</u>	Set partition blocks
<u>SetExtraCommand</u>	Execute extended command
<u>SetVariableSetting</u>	Set Variable mode channel parameter
<u>SetGeometricConvert</u>	Set output image rotation and mirroring
<u>SeAutoExposure</u>	Set values for Auto Exposure function
Get Camera Setups	
<u>GetMaxFrame</u>	Get max frame count to record
<u>GetCameraParams</u>	Get camera parameters
<u>GetCameraParamsEx</u>	Get extended camera parameters
<u>UpdateCameraParams</u>	Get current camera parameters
<u>UpdateCameraParamsEx</u>	Get current extended camera parameters

<u>GetRecordRate</u>	Get recording rate
<u>GetShutterSpeed</u>	Get shutter speed
<u>GetTriggerMode</u>	Get trigger mode
<u>GetRandomTriggerNumber</u>	Get random trigger frame number
<u>GetManualTriggerPosition</u>	Get manual trigger position
<u>GetRandomManualPosition</u>	Get random-manual trigger position
<u>GetResolution</u>	Get resolution
<u>GetGammaCorrection</u>	Get gamma correction value
<u>GetGainLevel</u>	Get gain level
<u>GetColorTemperature</u>	Get color temperature
<u>GetColorTempUser</u>	Get user color temperature
<u>GetExternalSyncMode</u>	Get external sync drive setting
<u>GetExternalSyncInMode</u>	Get external sync-in drive setting
<u>GetExternalSyncModeEx</u>	Get external sync drive ex setting
<u>GetExternalSyncOutMode</u>	Get external sync-out drive setting
<u>GetExternalInMode</u>	Get external general purpose signal-in setting
<u>GetExternalOutMode</u>	Get external general purpose signal-out setting
<u>GetEdgeEnhancementMode</u>	Get edgenhancement setting (on monitor output)
<u>GetZoomMode</u>	Get zoom display On/Off (on monitor output)
<u>GetMonitorOutMode</u>	Get external monitor output setting
<u>GetEnableMCDL</u>	Get MCDL feature enable On/Off
<u>GetEnableIRIG</u>	Get IRIG feature enable On/Off
<u>GetRecordRateList</u>	Get recording rate list
<u>GetShutterSpeedList</u>	Get shutter speed list
<u>GetTriggerModeList</u>	Get trigger mode list
<u>GetResolutionList</u>	Get resolution list
<u>GetRandomTriggerNumberList</u>	Get random trigger frame number list
<u>GetColorTemperatureList</u>	Get color temperature mode list
<u>GetGammaCorrectionList</u>	Get gamma correction value list
<u>GetGainLevelList</u>	Get gain level list
<u>GetPartitionInfo</u>	Get partition information
<u>GetPartition</u>	Get current partition ID number
<u>GetPartitionBlockList</u>	Get partition block list
<u>GetExtraCommand</u>	Execute extended command
<u>GetFrameParams</u>	Get frame management information
<u>GetShutterSpeedType2List</u>	Get shutter speed type2 list
<u>GetShutterSpeedType2</u>	Get shutter speed type2

GetHeadName	Get camera head name
GetVariableSetting	Get Variable mode channel parameter
GetVariableCamMode	Get Variable mode
GetGeometricConvert	Get status of rotation and mirroring of output image
GetAutoExposureArea	Get target area of Auto Exposure function
GetAutoExposureParam	Get image output level of Auto Exposure function
GetShutterSpeedAEList	Get available exposure periods for Auto Exposure
Camera Mode	
OnLive	Switch display Live/Playback
IsLive	Get display status Live/Playback
GetCameraMode	Get camera status
Recording	
OnRecordReady	Set record ready On/Off
OnRecord	Recording Start/Stop
IsRecordReady	Get record ready status On/Off
IsRecord	Get status recording/pause
TriggerIn	Software trigger input (other than START mode)
IsEndlessRec	Get endless recording status
playback	
PlayMonitor	Play (on monitor output)
PauseMonitor	Pause (on monitor output)
StopMonitor	Stop (on monitor output)
StepFowardMonitor	Jog forward (on monitor output)
StepBackMonitor	Jog backward (on monitor output)
GoAnyFrameMonitor	Go to specified frame (on monitor output)
GoStartMonitor	Go to start frame (on monitor output)
GoEndMonitor	Go to end frame (on monitor output)
GoTriggerMonitor	Go to trigger frame (on monitor output)
Playback Setting	
SetPlayRate	Set playback rate (on monitor output)
SetBlockArea	Set block (range) (on monitor output)
SetBlockMode	Set block (range) playback On/Off (on monitor output)
GetPlayRate	Get playback rate (on monitor output)
GetBlockArea	Get block (range) playback (on monitor output)

GetBlockMode	Get block (range) playback On/Off (on monitor output)
Getting Image Data	
TransferFrame	Get image data of specified frame
TransferRawBayer	Get Bayer image data of specified frame
Transfer16BitFrame	Get 16-bit image data of specified frame
Transfer16BitRawBayer	Get Bayer image data of specifies frame
Other Special Items	
TransferMCDL	Get MCDL data of specified frame
TransferIRIG	Get IRIG data of specified frame
GetTimeCodeFromFrame	Get IRIG timecode from specified frame number
ShadingCompensation	Correct shading
GetDateOfRecording	Get date of recording
GetTimeOfRecording	Get time of recording
GetIICycleRange	Get I.I. CYCLE value range
GetIIVidthRange	Get I.I. WIDTH value range
GetIITimesRange	Get I.I. TIMES value range
GetIIDelayRange	Get I.I. DELAY value range
GetVariableRateList	Get Variable mode record rate list
GetVariableMaxResolution	Get Variable mode max square resolution
GetVariableMaxWidth	Get Variable mode max width
GetVariableMaxHeight	Get Variable mode max height
GetVariableMaxFrameRate	Get Variable mode max record rate
SetGEtherConfig	Set connection conditions for Giga-bit Ether I/F
GetGEtherConfig	Get connection conditions for Giga-bit Ether I/F
Extra Commands	
EXTRA_SET_ENABLE_RESTRICTION_TIME	Set rec time restriction function
EXTRA_GET_ENABLE_RESTRICTION_TIME	Get rec time restriction function
EXTRA_GET_CAMERA_ID	Get camera ID
EXTRA_GET_ENABLE_RESET_TRIGGER	Get reset trigger status
EXTRA_SET_ENABLE_RESET_TRIGGER	Set reset trigger status
EXTRA_GET_SENSOR_BITSHIFT	Get grayscale setting
EXTRA_SET_SENSOR_BITSHIFT	Set grayscale setting
EXTRA_SET_LIVE_RESOLUTION	Set live resolution change
EXTRA_GET_LIVE_RESOLUTION	Get live resolution setting
EXTRA_SET_LUT_MODE	Set LUT status

<u>EXTRA_GET_LUT_MODE</u>	Get LUT status
<u>EXTRA_SET_USER_LUT_PARAMS</u>	Set LUT parameter change
<u>EXTRA_GET_USER_LUT_PARAMS</u>	Get LUT parameters
<u>EXTRA_UPLOAD_LUT_DATA</u>	Upload LUT setting to camera
<u>EXTRA_SET_CHROMA_MODE</u>	Set chroma setting change
<u>EXTRA_GET_CHROMA_MODE</u>	Get chroma setting
<u>EXTRA_SET_DS_SHUTTER_MODE</u>	Set dynamic range expansion mode change
<u>EXTRA_GET_DS_SHUTTER_MODE</u>	Get dynamic range expansion setting
<u>EXTRA_SET_INTERFACE_INFO</u>	Set interface setting change
<u>EXTRA_GET_INTERFACE_INFO</u>	Get interface setting
<u>EXTRA_SET_SHADING_MODE</u>	Set calibration mode
<u>EXTRA_GET_SHADING_MODE</u>	Get calibration mode setting
<u>EXTRA_GET_SHADING_LOCK</u>	Get calibration performance satus
<u>EXTRA_SET_IRIG_OFFSET</u>	Set IRIG offset value
<u>EXTRA_GET_IRIG_OFFSET</u>	Get IRIG offset value
<u>EXTRA_SET_HARD_PARTITION</u>	Set hard partition mode
<u>EXTRA_GET_HARD_PARTITION</u>	Get hard partition mode
<u>EXTRA_SET_SHUTTER_MODE</u>	Set shutter mode
<u>EXTRA_GET_SHUTTER_MODE</u>	Get shutter mode
<u>EXTRA_SET_II_POWER</u>	Set I.I. power mode
<u>EXTRA_GET_II_POWER</u>	Get I.I. power mode
<u>EXTRA_SET_II_GAIN</u>	Set I.I. gain value
<u>EXTRA_GET_II_GAIN</u>	Get I.I. gain value
<u>EXTRA_SET_II_GATE_SELECT</u>	Set I.I. gate mode
<u>EXTRA_GET_II_GATE_SELECT</u>	Get I.I. gate mode
<u>EXTRA_SET_II_GATE_CYCLE</u>	Set I.I. cycle value
<u>EXTRA_GET_II_GATE_CYCLE</u>	Get I.I. cycle value
<u>EXTRA_SET_II_GATE_WIDTH</u>	Set I.I. width value
<u>EXTRA_GET_II_GATE_WIDTH</u>	Get I.I. width value
<u>EXTRA_SET_II_GATE_TIMES</u>	Set I.I. times value
<u>EXTRA_GET_II_GATE_TIMES</u>	Get I.I. times value
<u>EXTRA_SET_II_GATE_DELAY</u>	Set I.I. delay value
<u>EXTRA_GET_II_GATE_DELAY</u>	Get I.I. delay value
<u>EXTRA_GET_II_GAIN_LIMIT</u>	Get I.I. Gain's under limit value
<u>EXTRA_GET_EVENT_FRAME</u>	Get event frame
<u>EXTRA_SET_VARIABLE_LOAD</u>	Load Variable mode channel
<u>EXTRA_SET_VARIABLE_ERASE</u>	Erase Variable mode channel

<u>EXTRA_GET_II_GATE_WIDTH_LIMIT</u>	Get I.I. width minimum value
<u>EXTRA_SET_EDGE_ENHANCEMENT</u>	Set Edge enhancement mode
<u>EXTRA_GET_EDGE_ENHANCEMENT</u>	Get Edge enhancement mode
<u>EXTRA_SET_AUTO_EXPOSURE</u>	Set status of Auto Exposure function
<u>EXTRA_GET_AUTO_EXPOSURE</u>	Get status of Auto Exposure function
<u>EXTRA_SET_AUTO_EXPOSURE_MAXSHUTTER</u>	Set max exposure period for Auto Exposure
<u>EXTRA_GET_AUTO_EXPOSURE_MAXSHUTTER</u>	Get max exposure period for Auto Exposure
<u>EXTRA_SET_II_PROTECTION</u>	Set protection level of I.I. hardware
<u>EXTRA_GET_II_PROTECTION</u>	Get protection level of I.I. hardware

4.3. CameraControl Function Specifications

This section shows the feature of functions of the CameraControl .

VisualBasic6.0

CameraControl (ByVal device_select As Long) As Long

VisualBasic.NET

CameraControl (ByVal device_select As Integer) As Integer

Functions	Generates a CameraControl instance.	
Arguments	device_select	
	Specifies the camera type.	
	DEVICE_SELECT_AUTO	Automatic recognition (default)
	DEVICE_SELECT_1394	FASTCAM UltimaSE & FASTCAM Ultima1024 & FASTCAM UltimaAPX & FASTCAM Ultima512 & FASTCAM APX RS (IEEE1394)
	DEVICE_SELECT_ETHER	FASTCAM Ultima APX & FASTCAM Ultima512 (100Base-TX)
	DEVICE_SELECT_G_ETHER	FASTCAM APX RS (1000Base-T)
	DEVICE_SELECT_OPT	FASTCAM Ultima APX & FASTCAM Ultima512 & FASTCAM APX RS (Optical/F)
	DEVICE_SELECT_PCI	FASTCAM PCI
	DEVICE_SELECT_PCI2	FASTCAM PCI R2
	DEVICE_SELECT_PLMV	FASTCAM-X 1280PCI
	DEVICE_SELECT_512PCI	FASTCAM 512PCI
	DEVICE_SELECT_1024PCI	FASTCAM 1024PCI
Returned values	None	
Remarks	<p>One CameraControl controls one camera.</p> <p>When multiple cameras with different types of interface are present in a system and when DEVICE_SELECT_AUTO is selected, DEVICE_SELECT_1394, DEVICE_SELECT_OPT, DEVICE_SELECT_PCI, , DEVICE_SELECT_PCI2, DEVICE_SELECT_PLMV, DEVICE_SELECT_512PCI and DEVICE_SELECT_1024PCI are recognized in this order.</p> <p>Note: DEVICE_SELECT_ETHER and DEVICE_SELECT_G_ETHER are not recognized when DEVICE_SELECT_AUTO has been selected.</p>	

VisualBasic6.0**GetNumberOfDevice (ByRef number As Long) As Long****VisualBasic.NET****GetNumberOfDevice (ByRef number As Integer) As Integer**

Functions	Gets the maximum number of connected cameras
Arguements	number Specifies an 32Bit that represents the maximum number of cameras.
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	One CameraControl controls one camera. See "On Error Codes" for details of error codes On Error Codes

VisualBasic6.0

Init (ByVal camera_num As Long) As Long

VisualBasic.NET

Init (ByVal camera_num As Integer) As Integer

Functions	Initializes CameraControl and specifies the cameras to be controlled.
Arguments	camera_num Specifies the camera number to be controlled.
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	One CameraControl controls one camera. The camera number begins with 1 and goes up to the total number of cameras connected to the system. See "On Error Codes" for details of error codes On Error Codes

VisualBasic6.0

ExitCameraControl () As Long

VisualBasic.NET

ExitCameraControl () As Integer

Functions	Gets a CameraControl initialization status.
Arguements	None
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>This function is normally called out from the default destructor: no need to call it directly.</p> <p>See "On Error Codes" for details of error codes On Error Codes.</p>

VisualBasic6.0

SetCameraNumber (ByVal number As Long) As Long

VisualBasic.NET

SetCameraNumber (ByVal number As Integer) As Integer

Functions	Sets a Operation camera number
Arguements	number The management camera number in a library
Returned values	A [1] is returned when the process is success, and a [0] when in error.
Remarks	

VisualBasic6.0

GetCameraNumber () As Long

VisualBasic.NET

GetCameraNumber () As Integer

Functions	Gets a Operation camera number
Arguments	None
Returned values	The management camera number in a library which can be operated returns.
Remarks	

VisualBasic6.0

IsInit () As Long

VisualBasic.NET

IsInit () As Integer

Functions Confirms the initialization status of CameraControl.

Arguments None

Returned Values [1]: Initialized
[0]: Not initialized

Remarks

VisualBasic6.0

GetDeviceName (ByVal name As String) As Long

VisualBasic.NET

GetDeviceName (ByVal name As String) As Integer

Functions	Gets the name of a camera to control.		
Arguments	<table><tr><td>name</td><td>Specifies the address of a LPTSTR type buffer that gets the letter string of the camera name.</td></tr></table>	name	Specifies the address of a LPTSTR type buffer that gets the letter string of the camera name.
name	Specifies the address of a LPTSTR type buffer that gets the letter string of the camera name.		
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.		
Remarks	See "On Error Codes" for details of error codes On Error Codes .		

VisualBasic6.0

GetDeviceType (ByRef type As Long) As Long

VisualBasic.NET

GetDeviceType (ByRef type As Integer) As Integer

Functions	Gets the camera type code of the camera to control.		
Arguments	<table><tr><td>type</td><td>Specifies the 32bit argument that gets the camera type code.</td></tr></table>	type	Specifies the 32bit argument that gets the camera type code.
type	Specifies the 32bit argument that gets the camera type code.		
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.		
Remarks	See "On Error Codes" for details of error codes On Error Codes		

VisualBasic6.0

GetDeviceVersion (ByRef version As Long) As Long

VisualBasic.NET

GetDeviceVersion (ByRef version As Integer) As Integer

Functions	Gets the hardware version number of the camera to control.
Arguments	<p>version</p> <p>Specifies the 32bit parameter that gets the hardware version number of the camera.</p>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	See "On Error Codes" for details of error codes On Error Codes

VisualBasic6.0

IsColor (ByRef color As Long) As Long

VisualBasic.NET

IsColor (ByRef color As Integer) As Integer

Functions	Gets information whether the camera is a color or monochrome model.
Arguments	<p>color</p> <p>Specifies the 32Bit type parameter that receives the result of judging.</p> <p>[1]: Color</p> <p>[0]: Monochrome</p>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	See "On Error Codes" for details of error codes On Error Codes

VisualBasic6.0**HaveGammaCorrection (ByRef have As Long) As Long****VisualBasic.NET****HaveGammaCorrection (ByRef have As Integer) As Integer**

Functions	Tells whether or not the camera has a gamma setting function.	
Arguments	have	Specifies the 32bit type parameter that receives the result of judgement.
	[1]	Yes
	[0]	No
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.	
Remarks	See "On Error Codes" for details of error codes On Error Codes	

VisualBasic6.0**HaveGainLevel (ByRef have As Long) As Long****VisualBasic.NET****HaveGainLevel (ByRef have As Integer) As Integer**

Functions	Tells whether or not the camera has a gain level control function.	
Arguments	have	Specifies the 32bit type parameter that receives the result of judgment.
	[1]	Yes
	[0]	No
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.	
Remarks	See "On Error Codes" for details of error codes On Error Codes	

VisualBasic6.0**HaveColorTemperature (ByRef have As Long) As Long****VisualBasic.NET****HaveColorTemperature (ByRef have As Integer) As Integer**

Functions	Tells whether or not the camera has a color temperature setting function.
Arguments	<div>have Specifies a 32bit type parameter that receives the result of judgment. [1] Yes [0] No</div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	See "On Error Codes" for details of error codes On Error Codes

VisualBasic6.0

HaveExternalSyncMode (ByRef have As Long) As Long

VisualBasic.NET

HaveExternalSyncMode (ByRef have As Integer) As Integer

Functions	Tells whether or not the camera has an external clock sync function.	
Arguments	have	Specifies a 32bit type parameter that receives the result of judgment.
	[1]	Yes
	[0]	No
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.	
Remarks	See "On Error Codes" for details of error codes On Error Codes	

VisualBasic6.0

HaveExternalSyncInMode (ByRef have As Long) As Long

VisualBasic.NET

HaveExternalSyncInMode (ByRef have As Integer) As Integer

Functions	Tells whether or not the camera has an external clock sync input functin.
Arguments	<div><div>have</div><div>Specifies a 32bit type parameter that receives the result of judgment.</div><div>[1] Yes</div><div>[0] No</div></div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	See "On Error Codes" for details of error codes On Error Codes

VisualBasic6.0

HaveExternalSyncOutMode (ByRef have As Long) As Long

VisualBasic.NET

HaveExternalSyncOutMode (ByRef have As Integer) As Integer

Functions	Tells whether or not the camera has an external clock sync output functin.
Arguments	<div>have Specifies a 32bit type parameter that receives the result of judgment. [1] Yes [0] No</div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	See "On Error Codes" for details of error codes On Error Codes

VisualBasic6.0**HaveExternallInput (ByRef have As Long) As Long****VisualBasic.NET****HaveExternallInput (ByRef have As Integer) As Integer**

Functions	Tells whether or not the camera has an external general purpose signal input functin.
Arguments	<div>have Specifies a 32bit type parameter that receives the result of judgment. [1] Yes [0] No</div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	See "On Error Codes" for details of error codes On Error Codes

VisualBasic6.0**HaveExternalOutput (ByRef have As Long) As Long****VisualBasic.NET****HaveExternalOutput (ByRef have As Integer) As Integer**

Functions	Tells whether or not the camera has an external general purpose signal output function.
Arguments	<div>have Specifies a 32bit type parameter that receives the result of judgment. [1] Yes [0] No</div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	See "On Error Codes" for details of error codes On Error Codes

VisualBasic6.0**HaveEdgeEnhancement (ByRef have As Long) As Long****VisualBasic.NET****HaveEdgeEnhancement (ByRef have As Integer) As Integer**

Functions	Tells whether or not the camera has an edge enhancement function.						
Arguments	<table><tr><td>have</td><td>Specifies a 32bit type parameter that receives the result of judging.</td></tr><tr><td>[1]</td><td>Yes</td></tr><tr><td>[0]</td><td>No</td></tr></table>	have	Specifies a 32bit type parameter that receives the result of judging.	[1]	Yes	[0]	No
have	Specifies a 32bit type parameter that receives the result of judging.						
[1]	Yes						
[0]	No						
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.						
Remarks	Has effect on monitor output signal only. See "On Error Codes" for details of error codes On Error Codes						

Note: This function is not used with present camera models. This is only included here to match the older version of SDK manual.

VisualBasic6.0**HaveZoomMode (ByRef have As Long) As Long****VisualBasic.NET****HaveZoomMode (ByRef have As Integer) As Integer**

Functions	Tells whether or not the camera has a zooming display function.
Arguments	<div>have Specifies a 32bit type parameter that receives the result of judging. [1] Yes [0] No</div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<div>Has effect on monitor output signal only. See "On Error Codes" for details of error codes On Error Codes</div>

Note: This function is not used with present camera models. This is only included here to match the older version of SDK manual.

VisualBasic6.0**HaveBlockPlay (ByRef have As Long) As Long****VisualBasic.NET****HaveBlockPlay (ByRef have As Integer) As Integer**

Functions	Tells whether or not the camera has a block playback function.
Arguments	<div>have Specifies a 32bit type parameter that receives the result of judging. [1] Yes [0] No</div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	Has effect on monitor output signal only. See "On Error Codes" for details of error codes On Error Codes

Note: This function has been combined with ExtraCommand functions.
This is only included here to match the older version of SDK manual.
Note: This member function is not supported by FASTCAM Ultima 1024 or later models.

VisualBasic6.0**HaveMonitorOutMode (ByRef have As Long) As Long****VisualBasic.NET****HaveMonitorOutMode (ByRef have As Integer) As Integer**

Functions	Tells whether the camera has an external monitor selecting function.						
Arguments	<table><tr><td>have</td><td>Specifies a 32bit type parameter that receives the result of judging.</td></tr><tr><td>[1]</td><td>Yes</td></tr><tr><td>[0]</td><td>No</td></tr></table>	have	Specifies a 32bit type parameter that receives the result of judging.	[1]	Yes	[0]	No
have	Specifies a 32bit type parameter that receives the result of judging.						
[1]	Yes						
[0]	No						
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.						
Remarks	See "On Error Codes" for details of error codes On Error Codes						

Note: This function has been combined with ExtraCommand fuctions.
This is only included here to match the older version of SDK manual.

VisualBasic6.0

HaveMCDL (ByRef have As Long) As Long

VisualBasic.NET

HaveMCDL (ByRef have As Integer) As Integer

Functions	Tells whether or not the camera has an MCDL function.
Arguments	<div><div>have</div><div>Specifies a 32bit type parameter that receives the result of judging.</div><div>[1] Yes</div><div>[0] No</div></div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	See "On Error Codes" for details of error codes On Error Codes

Note: This function has been combined with ExtraCommand fuctions.This is only included here to match the older version of SDK manual.

VisualBasic6.0

HavelIRIG (ByRef have As Long) As Long

VisualBasic.NET

HavelIRIG (ByRef have As Integer) As Integer

Functions	Tells whether or not the camera has an IRIG function.	
Arguments	have	Specifies a 32bit type parameter that receives the result of judging.
	[1]	Yes
	[0]	No
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.	
Remarks	See "On Error Codes" for details of error codes On Error Codes	
	Note: This function has been combined with ExtraCommand fuction. This is only included here to match the older version of SDK manual.	

VisualBasic6.0

SetCameraParams (ByRef params As CAMERA_PARAMS)
As Long

VisualBasic.NET

SetCameraParams (ByRef params As CAMERA_PARAMS)
As Integer

Functions	Sets camera parameters.
Arguments	<p>params</p> <p>Specifies the CAMERA_PARAMS structure that has the camera parameters.</p>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>Values of unchangeable items are neglected.</p> <p>All the changeable parameters are reset. It may take some time to reset depending on the camera type. It is advisable to use a different function on each camera.</p> <p>See "CAMERA_PARAMS Structure" for details CAMARA_PARAMS Structure</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0

**SetCameraParamsEx(ByRef params As CAMERA_PARAMS_EX)
As Long**

VisualBasic.NET

**SetCameraParamsEx(ByRef params As CAMERA_PARAMS_EX)
As Integer**

Functions	Sets extra camera parameters.
Arguments	<p>params</p> <p>Specifies a CAMERA_PARAMS structure that has extra camera parameters.</p>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>Values of unchangeable items are neglected.</p> <p>All the changeable parameters are reset. It may take some time to reset depending on the camera type. It is advisable to use a different function on each camera.</p> <p>See "CAMERA_PARAMS Structure" for details CAMARA_PARAMS_EX Structure</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0

SetRecordRate (ByVal rate As Long) As Long

VisualBasic.NET

SetRecordRate (ByVal rate As Integer) As Integer

Functions	Sets a recording rate.
Arguments	<div>rate Specifies a recording rate (FPS) to be set.</div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>Resettable rates are received from the GetRecordRateList function.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0

SetShutterSpeed (ByVal speed As Long) As Long

VisualBasic.NET

SetShutterSpeed (ByVal speed As Integer) As Integer

Functions	Sets a shutter speed.
Arguments	<p>speed</p> <p>Specifies a shutter speed (1/speed) to set</p>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR
Remarks	<p>Settable speeds are received from GetShutterSpeedList function.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0

SetTriggerMode (ByVal mode As Long) As Long

VisualBasic.NET

SetTriggerMode (ByVal mode As Integer) As Integer

Functions Sets the trigger mode.

Arguments **mode**

Specifies the trigger mode to be set.

TRIGGER_START	Start Trigger
TRIGGER_CENTER	Center Trigger
TRIGGER_END	End Trigger
TRIGGER_RANDOM	Random Trigger
TRIGGER_MANUAL	Manual Trigger
TRIGGER_RANDOMRESET	Random Reset Trigger
TRIGGER_RANDOMCENTER	Random Center Trigger
TRIGGER_RANDOMMANUAL	Random Manual Trigger
TRIGGER_TWOSTAGE	Dual-speed Trigger

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR

Remarks Settable values are received from the [GetTriggerModeList](#) function.

See "On Error Codes" for details of error codes [On Error Codes](#)

VisualBasic6.0

SetRandomTriggerNumber (ByVal number As Long) As Long

VisualBasic.NET

SetRandomTriggerNumber(ByVal number As Integer)As Integer

Functions	Sets the number of frames to record per one trigger pulse in random trigger mode.
Arguments	<p>number</p> <p>Specifies the number of frames to be set.</p>
Returned values	Returns and error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>Settable numbers are received from the GetRandomTriggerNumberList function.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0

SetManualTriggerPosition (ByVal position As Long) As Long

VisualBasic.NET

SetManualTriggerPosition(ByVal position As Integer) As Integer

Functions	Sets the position of the trigger in manual trigger mode.
Arguments	<p>position</p> <p>Specifies the trigger position (frame number) to be set.</p>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>See "Frame Number" for details of frame number Frame Parameters</p> <p>See the hardware manual of each camera for details of trigger modes.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0

SetRandomManualPosition (ByVal position As Long) As Long

VisualBasic.NET

**SetRandomManualPosition(ByVal position As Integer)
As Integer**

Functions	Sets the position of the trigger in random manual trigger mode.
Arguments	<p>position</p> <p>Specifies the trigger position (frame number) to be set.</p>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>See "Frame Number" for details of frame number Frame Parameters</p> <p>See the hardware manual of each camera for details of trigger modes.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0**SetResolution (ByVal resolution As Long) As Long****VisualBasic.NET****SetResolution (ByVal resolution As Integer) As Integer**

Functions Sets the resolution

Arguments **resolution**
 Specifies the value of resolution to be set:
 32bit Upper 16 bits for Width
 32bit Lower 16 bits for Height

Returned values Returns an error code. When normal, returns
 a PCC_ERROR_NOERROR.

Remarks Settable values are dependand on the recording rate.

 Settable numbers are received from [GetResolutionList](#) function.

 See "On Error Codes" for details of error codes [On Error Codes](#)

VisualBasic6.0**SetGammaCorrection (ByVal mode As Long) As Long****VisualBasic.NET****SetGammaCorrection (ByVal mode As Integer) As Integer**

Functions Sets gamma correction value.

Arguments **mode**

Specifies a gamma correction mode.

GAMMA_1_0 $\gamma = 1.0$

GAMMA_0_9 $\gamma = 0.9$

GAMMA_0_8 $\gamma = 0.8$

GAMMA_0_7 $\gamma = 0.7$

GAMMA_0_6 $\gamma = 0.6$

GAMMA_0_5 $\gamma = 0.5$

GAMMA_0_4 $\gamma = 0.4$

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
With a camera without gamma correction function, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code.

Remarks Settable numbers are received from [GetGammaCorrectionList](#) function.

See "On Error Codes" for details of error codes [On Error Codes](#)

VisualBasic6.0

SetGainLevel (ByVal mode As Long) As Long

VisualBasic.NET

SetGainLevel (ByVal mode As Integer) As Integer

Functions Sets the gain level mode.

Arguments **mode**

Specifies the gain level mode.

GAIN_0DB	0dB
GAIN_3DB	+3dB
GAIN_6DB	+6dB
GAIN_12DB	+12dB
GAIN_18DB	+18dB
GAIN_24DB	+24dB
GAIN_30DB	+30dB
GAIN_36DB	+36dB

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
On a camera without gain level setting function, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code.

Remarks Settable numbers are received from [GetGainLevelList](#) function.

See "On Error Codes" for details of error codes [On Error Codes](#)

VisualBasic6.0

SetColorTemperature (ByVal mode As Long) As Long

VisualBasic.NET

SetColorTemperature (ByVal mode As Integer) As Integer

Functions	Sets color temperature mode (white balance).										
Arguments	<p>mode</p> <p>Specifies a color temperature (white balance) mode.</p> <table> <tr> <td>COLORTEMPMODE_DEF1</td><td>Default setting 1</td></tr> <tr> <td>COLORTEMPMODE_DEF2</td><td>Default setting 2</td></tr> <tr> <td>COLORTEMPMODE_USER1</td><td>User setting 1</td></tr> <tr> <td>COLORTEMPMODE_USER2</td><td>User setting 2</td></tr> <tr> <td>COLORTEMPMODE_USER3</td><td>User setting 3</td></tr> </table>	COLORTEMPMODE_DEF1	Default setting 1	COLORTEMPMODE_DEF2	Default setting 2	COLORTEMPMODE_USER1	User setting 1	COLORTEMPMODE_USER2	User setting 2	COLORTEMPMODE_USER3	User setting 3
COLORTEMPMODE_DEF1	Default setting 1										
COLORTEMPMODE_DEF2	Default setting 2										
COLORTEMPMODE_USER1	User setting 1										
COLORTEMPMODE_USER2	User setting 2										
COLORTEMPMODE_USER3	User setting 3										
Returned values	<p>Returns an error code. When normal, returns a PCC_ERROR_NOERROR.</p> <p>On cameras (color or monochrome) without color temperature mode setting function, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code.</p>										
Remarks	<p>Settable values are received from 「GetColorTemperatureList」 function.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p> <p>Note: COLORTEMPMODE_USER2 and COLORTEMPMODE_USER3 are not currently used. COLORTEMPMODE_DEF1 and COLORTEMPMODE_DEF2 are set to 5100K and 3100K, respectively.</p>										

VisualBasic6.0

**SetColorTempUser (ByVal mode As Long, ByVal r As Long,
ByVal g As Long, ByVal b As Long) As Long**

VisualBasic.NET

**SetColorTempUser (ByVal mode As Integer, ByVal r As Integer,
ByVal g As Integer, ByVal b As Integer) As Integer**

Functions	Sets white balance in user-set color temperature mode.		
Arguments	mode	Specifies the user-set color temperature mode.	
		COLORTEMPMODE_USER1	user setting 1
		COLORTEMPMODE_USER2	user setting 2
		COLORTEMPMODE_USER3	user setting 3
	r	0 to 63	Red level (default 16)
	g	0 to 63	Green level (default 16)
	b	0 to 63	Blue level (default 16)
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.		
	On cameras (color or monochrome) without color temperature setting function,		
	a PCC_ERROR_FUNCTION_DISABLE is returned as an error code.		
Remarks	Settable values are received from GetColorTemperatureList .		
	See "On Error Codes" for details of error codes On Error Codes		
	Note: COLORTEMPMODE_USER2 and COLORTEMPMODE_USER3 are not currently used.		

VisualBasic6.0

SetExternalSyncInMode (ByRef mode As Long) As Long

VisualBasic.NET

SetExternalSyncInMode (ByRef mode As Integer) As Integer

Functions Sets external sync mode.

Arguements **mode**

Specifies an external sync mode from:

EXTSYNC_NONE	OFF (Master mode)
EXTSYNC_POSI	ON (Slave mode with positive signal input)
EXTSYNC_NEGA	ON (Slave mode with negative signal input)
EXTSYNC_OTHERS_POSI	ON (Positive signal)
EXTSYNC_OTHERS_NEGA	ON (Negative signal)

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR.

Remarks With cameras without Pos/Neg setting of input signal, switching between ON and OFF is only effective.

See "On Error Codes" for details of error codes [On Error Codes](#)

Note: This function has been combined with the SetExternalSyncModeEx function. This is only included here to match the old version of the SDK manual.

VisualBasic6.0**SetExternalSyncModeEx (ByVal mode As Long) As Long****VisualBasic.NET****SetExternalSyncModeEx (ByVal mode As Integer) As Integer**

Functions	Sets external sync mode.												
Arguments	<table><tr><td>mode</td><td>Specifies an external sync mode from:</td></tr><tr><td>EXTSYNC_NONE</td><td>OFF (Master mode)</td></tr><tr><td>EXTSYNC_POSI</td><td>ON (Slave mode with positive signal input)</td></tr><tr><td>EXTSYNC_NEGA</td><td>ON (Slave mode with negative signal input)</td></tr><tr><td>EXTSYNC_OTHERS_POSI</td><td>ON (Positive signal)</td></tr><tr><td>EXTSYNC_OTHERS_NEGA</td><td>ON (Negative signal)</td></tr></table>	mode	Specifies an external sync mode from:	EXTSYNC_NONE	OFF (Master mode)	EXTSYNC_POSI	ON (Slave mode with positive signal input)	EXTSYNC_NEGA	ON (Slave mode with negative signal input)	EXTSYNC_OTHERS_POSI	ON (Positive signal)	EXTSYNC_OTHERS_NEGA	ON (Negative signal)
mode	Specifies an external sync mode from:												
EXTSYNC_NONE	OFF (Master mode)												
EXTSYNC_POSI	ON (Slave mode with positive signal input)												
EXTSYNC_NEGA	ON (Slave mode with negative signal input)												
EXTSYNC_OTHERS_POSI	ON (Positive signal)												
EXTSYNC_OTHERS_NEGA	ON (Negative signal)												
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.												
Remarks	<p>With cameras without Pos/Neg setting of input signal, switching between ON and OFF is only effective.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>												

VisualBasic6.0

SetExternalSyncOutMode (ByVal mode As Long) As Long

VisualBasic.NET

SetExternalSyncOutMode (ByVal mode As Integer) As Integer

Functions	Sets external sync output mode.				
Arguments	<p>mode</p> <p>Specifies an external sync output mode from:</p> <table><tr><td>EXTOUT_VSYNC_POSI</td><td>VSYNC signal (Positive going)</td></tr><tr><td>EXTOUT_VSYNC_NEGA</td><td>VSYNC signal (Negative going)</td></tr></table>	EXTOUT_VSYNC_POSI	VSYNC signal (Positive going)	EXTOUT_VSYNC_NEGA	VSYNC signal (Negative going)
EXTOUT_VSYNC_POSI	VSYNC signal (Positive going)				
EXTOUT_VSYNC_NEGA	VSYNC signal (Negative going)				
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.				
Remarks	See "On Error Codes" for details of error codes On Error Codes				

VisualBasic6.0

**SetExternallnMode (ByVal port As Long, ByVal mode As Long)
As Long**

VisualBasic.NET

**SetExternallnMode (ByVal port As Integer,
ByVal mode As Integer) As Integer**

Functions	Sets external general purpose signal input mode.												
Arguements	<p>port Specifies an output port number beginning with a "0".</p> <p>mode Specifies an output general purpose signal input mode from:</p> <table> <tr> <td>EXTIN_EVENT_POSI</td><td>Event signal (Positive going)</td></tr> <tr> <td>EXTIN_EVENT_NEGA</td><td>Event signal (Negative going)</td></tr> <tr> <td>EXTIN_TRIGGER_POSI</td><td>Trigger signal (Positive going)</td></tr> <tr> <td>EXTIN_TRIGGER_NEGA</td><td>Trigger signal (Negative going)</td></tr> <tr> <td>EXTIN_READY_POSI</td><td>Ready signal (Positive going)</td></tr> <tr> <td>EXTIN_READY_NEGA</td><td>Ready signal (Negative going)</td></tr> </table>	EXTIN_EVENT_POSI	Event signal (Positive going)	EXTIN_EVENT_NEGA	Event signal (Negative going)	EXTIN_TRIGGER_POSI	Trigger signal (Positive going)	EXTIN_TRIGGER_NEGA	Trigger signal (Negative going)	EXTIN_READY_POSI	Ready signal (Positive going)	EXTIN_READY_NEGA	Ready signal (Negative going)
EXTIN_EVENT_POSI	Event signal (Positive going)												
EXTIN_EVENT_NEGA	Event signal (Negative going)												
EXTIN_TRIGGER_POSI	Trigger signal (Positive going)												
EXTIN_TRIGGER_NEGA	Trigger signal (Negative going)												
EXTIN_READY_POSI	Ready signal (Positive going)												
EXTIN_READY_NEGA	Ready signal (Negative going)												
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.												
Remarks	See "On Error Codes" for details of error codes On Error Codes												

VisualBasic6.0

**SetExternalOutMode (ByVal port As Long,
ByVal mode As Long) As Long**

VisualBasic.NET

**SetExternalOutMode (ByVal port As Integer,
ByVal mode As Integer) As Integer**

Functions Sets the external signal output mode.

Arguments

port

Specifies the output port number beginning with a "0".

mode

Specifies the external general purpose signal input mode from:

EXTOUT_RECORD_POSI	Recording period signal (Positive going)
EXTOUT_RECORD_NEGA	Recording period signal (Negative going)
EXTOUT_TRIGGER_POSI	Trigger signal (Positive going)
EXTOUT_TRIGGER_NEGA	Trigger signal (Negative going)
EXTOUT_EXPOSE_POSI	Exposure period signal (Positive going)
EXTOUT_EXPOSE_NEGA	Exposure period signal (Negative going)
EXTOUT_READY_POSI	Ready signal (Positive going)
EXTOUT_READY_NEGA	Ready signal (Negative going)
EXTOUT_IRIG_RESET_POSI	IRIG reset signal (Positive going)
EXTOUT_IRIG_RESET_NEGA	IRIG reset signal (Negative going)

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR.

Remarks See "On Error Codes" for details of error codes [On Error Codes](#)

VisualBasic6.0**SetEdgeEnhancementMode (ByVal mode As Long) As Long****VisualBasic.NET****SetEdgeEnhancementMode (ByVal mode As Integer) As Integer**

Functions Sets edge enhancement mode of monitor output signal

Arguments **mode**

Specifies the edge enhancement mode from:

EDGEMODE_NONE	No edge enhancement
EDGEMODE_1	Edge enhancement 1
EDGEMODE_2	Edge enhancement 2
EDGEMODE_3	Edge enhancement 3

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
With cameras without monitor output or edge enhancement function, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code.

Remarks Effective only on the monitor output video signal.
See "On Error Codes" for details of error codes [On Error Codes](#)

Note: This member function is not used with currently supported cameras.
This is only included here to match the old version of SDK manual.

VisualBasic6.0**SetZoomMode (ByVal On As Long) As Long****VisualBasic.NET****SetZoomMode (ByVal On As Integer) As Integer**

Functions	Sets zooming display mode of monitor output signal.						
Arguments	<table><tr><td>on</td><td>Specifies ON/OFF of zooming display.</td></tr><tr><td>[1]</td><td>ON</td></tr><tr><td>[0]</td><td>OFF</td></tr></table>	on	Specifies ON/OFF of zooming display.	[1]	ON	[0]	OFF
on	Specifies ON/OFF of zooming display.						
[1]	ON						
[0]	OFF						
Returned values	<p>Returns an error code. When normal, returns a PCC_ERROR_NOERROR.</p> <p>On cameras without zooming display function, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code.</p>						
Remarks	<p>Effective only on the monitor output video signal.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p> <p>Note: This member function is not used with currently supported cameras. This is only included here to match the old version of SDK manual.</p>						

VisualBasic6.0**SetMonitorOutMode (ByVal mode As Long) As Long****VisualBasic.NET****SetMonitorOutMode (ByVal mode As Integer) As Integer**

Functions Sets external monitor output mode.

Arguments **mode**

 Specifies an external monitor output mode from:

MONITOROUT_VGA	VGA monitor output
MONITOROUT_NTSC	NTSC monitor output

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
On cameras without external monitor output setting function, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code.

Remarks See "On Error Codes" for details of error codes [On Error Codes](#)

Note: This function has been combined with ExtraCommand function. This is only included here to match the old version of SDK manual.

VisualBasic6.0**SetEnableMCDL (ByVal On As Long) As Long****VisualBasic.NET****SetEnableMCDL (ByVal On As Integer) As Integer**

Functions Sets ON/OFF of MCDL function.

Arguments **on**

 Specifies ON/OFF of MCDL function.

 [1] MCDL function ON

 [0] MCDL function OFF

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
On cameras without MCDL function, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code.

Remarks See "On Error Codes" for details of error codes [On Error Codes](#)

Note: This function has been combined with ExtraCommand function. This is only included here to match the old version of SDK manual.

VisualBasic6.0**SetEnableIRIG (ByVal On As Long) As Long****VisualBasic.NET****SetEnableIRIG (ByVal On As Integer) As Integer**

Functions Sets ON/OFF of IRIG function.

Arguments **on**

Specifies ON/OFF of IRIG function.

[1] IRIG function ON

[0] IRIG function OFF

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
On cameras without IRIG function, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code.

Remarks See "On Error Codes" for details of error codes [On Error Codes](#)

Note: This function has been combined with ExtraCommand function. This is only included here to match the old version of SDK manual.

VisualBasic6.0

SetPartition (ByVal partition As Long) As Long

VisualBasic.NET

SetPartition (ByVal partition As Integer) As Integer

Functions	Sets partition numbers to use.
Arguments	partition Partition number beginning with "1".
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	See "On Error Codes" for details of error codes On Error Codes

VisualBasic6.0

**SetPartitionBlockList (ByRef list As Long,
ByVal listSize As Long) As Long**

VisualBasic.NET

**SetPartitionBlockList (ByRef list As Integer,
ByVal listSize As Integer) As Integer**

Functions	Sets a list of partition blocks.
Arguements	<div><div>list</div><div>Specifies a address for 32bit type memory array containing the number of blocks in each partition.</div><div>listsize</div><div>Specifies the size of list memory array (number of partitions).</div></div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>When the frame_per_block value indicated by GetPartitionInfo is 1, you cannot set the number of blocks within each partition as you desire. In such a case, first decide on the number of partitions to set and then set [the total number of blocks] divided by [the number of partitions] as the number of blocks per partition.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0

**SetExtraCommand (ByVal command As Long,
ByVal param1 As Long, ByVal param2 As Long,
ByVal param3 As Long, ByVal param4 As Long)
As Long**

VisualBasic.NET

**SetExtraCommand (ByVal command As Integer,
ByVal param1 As Integer, ByVal param2 As Integer,
ByVal param3 As Integer, ByVal param4 As Integer)
As Integer**

Functions	Sets Executes extra commands on each of cameras
Arguements	<p>command Specifies an extra command</p> <p>param1 First argument of extra command (cannot be omitted)</p> <p>param2 Second argument of extra command (set 0 when omitted)</p> <p>param3 Third argument of extra command (set 0 when omitted)</p> <p>param4 Fouth argument of extra command (set 0 when omitted)</p>
Returned values	<p>Returns an error code. When normal, returns a PCC_ERROR_NOERROR.</p> <p>On cameras without the specified command, a PCC_ERROR_NOT_SUPPORTED is returned as an error code.</p>
Remarks	<p>See "On Extra Commands" for details of extra commands On Extra Commands</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0

**SetVariableSetting(ByVal nChannel As Long,
ByVal nFrameRate As Long, ByVal nWidth As Long,
ByVal nHeight As Long, ByVal nXPos As Long,
ByVal nYPos As Long) As Long**

VisualBasic.NET

**SetVariableSetting(ByVal nChannel As Integer,
ByVal nFrameRate As Integer, ByVal nWidth As Integer,
ByVal nHeight As Integer, ByVal nXPos As Integer,
ByVal nYPos As Integer) As Integer**

Functions Set the setting value to a specification channel by variable setup.

Arguments

nChannel	Specifies setting channel (1-20)
nFrameRate	Specifies frame rate
nWidth	Specifies width of resolution
nHeight	Specifies height of resolution
nXPos	Specifies upper left X coordinates of rectangle
nYPos	Specifies upper left Y coordinates of rectangle

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR.

Remarks Each specification value sets up equivalent to the execution result of a GetVariableMaxXXX function, or a low speed and low resolution. Moreover, fulfill the following conditions.

nWidth	128 multiples
nHeight	16 multiples
nXPos	64 multiples (0 is included)
nYPos	8 multiples (0 is included)

In addition, nWidth+nXPos or nHeight+nYPos should not exceed the highest resolution (if it is FASTCAM-APX RS 1024x1024) of a camera.

VisualBasic6.0**SetGeometricConvert (ByVal mode As Long) As Long****VisualBasic.NET****SetGeometricConvert (ByVal mode As Integer) As Integer**

Functions Sets rotation and mirroring of output image.

Arguments **mode** Sets rotation and mirroring modes.

IMAGE_CONV_DEFAULT	No rotation
IMAGE_CONV_ROTATE90	90-degree rotation to right
IMAGE_CONV_ROTATE180	180-degree rotation to right
IMAGE_CONV_ROTATE270	270-degree rotation to right
IMAGE_CONV_MIRROR_H	Horizontal mirroring
IMAGE_CONV_MIRROR_V	Vertical mirroring
Combination of rotation (any of 0, 90, 180 or 270 degrees) and mirroring possible	

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR.

Remarks

VisualBasic6.0

**SetAutoExposure (ByVal nWidth As Long,
ByVal nHeight As Long, ByVal nXPos As Long,
ByVal nYPos As Long, ByVal nValue As Long,
ByVal nRange As Long) As Long**

VisualBasic.NET

**SetAutoExposure (ByVal nWidth As Integer,
ByVal nHeight As Integer, ByVal nXPos As Integer,
ByVal nYPos As Integer, ByVal nValue As Integer,
ByVal nRange As Integer) As Integer**

Functions	Sets values for the Auto Exposure function.
Arguments	<p>nWidth Sets the width for a target area.</p> <p>nHeight Sets the height for a target area.</p> <p>nXPos Sets the x-coordinate of the upper left corner of a target area.</p> <p>nYPos Sets the Y-coordinate of the upper left corner of a target area.</p> <p>nValue Sets the image output level (0 to 255).</p> <p>nRange Sets the range for image output level (0 to 255).</p>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>Each of setting values in the above should meet the following condition:</p> <p>nWidth 128 multiples nHeight 16 multiples nXPos 64 multiples (0 is included) nYPos 8 multiples (0 is included)</p> <p>Note 1: nWidth + nXPos or nHeight + nYPos should not exceed the resolution of the camera.</p> <p>Note 2: nXPos and nYPos represent the coordinate of the maximum resolution of the camera (with FASTCAM-APX-RS, for example, 1024 x 1024). As an example, for a resolution of 512 x 512 around an optical axis, the coordinate of the origin in the upper left corner is X = 256 and Y = 256.</p>

VisualBasic6.0**GetMaxFrame (ByRef frame As Long) As Long****VisualBasic.NET****GetMaxFrame (ByRef frame As Integer) As Integer**

Functions	Gets the maximum number of frames in the current mode (recording rate and resolution).
Arguments	frame Specifies an 32bit type parameter that receives the maximum number of frames.
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>The maximum number is different by the mode the camera is in: LIVE or PLAY. The maximum number of frames available to record is given in the LIVE mode and the maximum number of recorded frames in the PLAY mode.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0

GetCameraParams (ByRef params As CAMERA_PARAMS)
As Long

VisualBasic.NET

GetCameraParams (ByRef params As CAMERA_PARAMS)
As Integer

Functions	Gets camera parameters.
Arguments	<p>params</p> <p>Specifies a CAMERA_PARAMS structure that receives camera parameters.</p>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>The acquired structure is a copy. So, you must use setting functions such as SetCameraParams or SetRecordRate to change the parameters like the recording rate. Likewise, when you change settings directly on the camera, you must call out the UpdateCameraParams function before getting parameters by this function.</p> <p>See "CAMERA_PARAMS Structure" for details of structure CAMARA_PARAMS Structure</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0

**GetCameraParamsEx(ByRef params As CAMERA_PARAMS_EX)
As Long**

VisualBasic.NET

**GetCameraParamsEx(ByRef params As CAMERA_PARAMS_EX)
As Integer**

Functions	Gets extra camera parameters.
Arguments	<p>params</p> <p>Specifies a CAMERA_PARAMS_EX structure that receives extra camera parameters.</p>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>The acquired structure is a copy. So, you must use setting functions such as SetCameraParamsEx to change the parameters like the recording rate. Likewise, when you change settings directly on the camera, you must call out the UpdateCameraParamsEx function before getting parameters by this function.</p> <p>See "CAMERA_PARAMS_EX Structure" for details of structure CAMARA_PARAMS_EX Structure</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0

UpdateCameraParams () As Long

VisualBasic.NET

UpdateCameraParams () As Integer

Functions	Gets parameters from cameras.
Arguments	None
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	The function updates itself with information on current settings of the cameras.

VisualBasic6.0

UpdateCameraParamsEx () As Long

VisualBasic.NET

UpdateCameraParamsEx () As Integer

Functions	Gets extra parameters from cameras.
Arguments	None
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	The function updates itself with information on current settings of the cameras.

VisualBasic6.0**GetRecordRate (ByRef rate As Long) As Long****VisualBasic.NET****GetRecordRate (ByRef rate As Integer) As Integer**

Functions	Gets the current recording rate.
Arguments	<div>rate Specifies the 32bit type parameter that receives a recording rate (FPS).</div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>The number the function gets is different by the mode the camera is in: LIVE or PLAY. The current recording rate is given in the LIVE mode and the recording rate of playback data is given in the PLAY mode.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0**GetShutterSpeed (ByRef speed As Long) As Long****VisualBasic.NET****GetShutterSpeed (ByRef speed As Integer) As Integer**

Functions	Gets the current shutter speed.
Arguments	speed Specifies the 32bit type parameter that receives a shutter speed (1/speed).
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>The number the function gets is different by the mode the camera is in: LIVE or PLAY. The current recording rate is given in the LIVE mode and the recording rate of playback data is given in the PLAY mode.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0

GetTriggerMode (ByRef mode As Long) As Long

VisualBasic.NET

GetTriggerMode (ByRef mode As Integer) As Integer

Functions	Gets the current trigger mode.	
Arguments	mode	Specifies the 32bit type parameter that receives a trigger mode:
	TRIGGER_START	START trigger
	TRIGGER_CENTER	CENTER trigger
	TRIGGER_END	END trigger
	TRIGGER_RANDOM	RANDOM trigger
	TRIGGER_MANUAL	MANUAL trigger
	TRIGGER_RANDOMRESET	RANDOM RESET trigger
	TRIGGER_RANDOMCENTER	RANDOM CENTER trigger
	TRIGGER_RANDOMMANUAL	RANDOM MANUAL trigger
	TRIGGER_TWOSTAGE	DUAL SPEED trigger
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.	
Remarks	<p>The number the function gets is different by the mode the camera is in: LIVE or PLAY. The current recording rate is given in the LIVE mode and the recording rate of playback data is given in the PLAY mode.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>	

VisualBasic6.0

GetRandomTriggerNumber (ByRef number As Long) As Long

VisualBasic.NET

**GetRandomTriggerNumber (ByRef number As Integer)
As Integer**

Functions	Gets the number of frames to record per one trigger in RANDOM trigger mode.
Arguments	<p>number</p> <p>Specifies the 32bit type parameter that receives the number of frames to record.</p>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>The number the function gets is different by the mode the camera is in: LIVE or PLAY. The number of frames to be recorded per currently set trigger is given in the LIVE mode and the number of frames recorded per trigger of playback data is given in the PLAY mode.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0

GetManualTriggerPosition (ByRef position As Long) As Long

VisualBasic.NET

**GetManualTriggerPosition (ByRef position As Integer)
As Integer**

Functions	Gets the position of trigger frame in MANUAL trigger mode.
Arguments	<p>position</p> <p>Specifies 32bit type parameters that receive a trigger frame position.</p>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>The number the function gets is different by the mode the camera is in: LIVE or PLAY. The current trigger setting position to start recording is given in the LIVE mode and the trigger setting position of playback data is given in the PLAY mode.</p> <p>See the hardware manual of each camera for details of trigger modes. See On Frame Parameters for details of frame numbers. See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0

GetRandomManualPosition (ByRef position As Long) As Long

VisualBasic.NET

**GetRandomManualPosition (ByRef position As Integer)
As Integer**

Functions	Gets the trigger position for RANDOM MANUAL trigger mode.
Arguments	<p>position</p> <p>Specifies 32bit type parameters that receive a trigger position (frame number).</p>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>The number the function gets is different by the mode the camera is in: LIVE or PLAY. The current trigger setting position to start recording is given in the LIVE mode and the trigger setting position of playback data is given in the PLAY mode.</p> <p>See the hardware manual of each camera for details of trigger modes. See On Frame Parameters for details of frame numbers. See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0

GetResolution (ByRef resolution As Long) As Long

VisualBasic.NET

GetResolution (ByRef resolution As Integer) As Integer

Functions	Gets the current resolution.
Arguments	<p>resolution</p> <p>Specifies a 32bit type parameter that receives the resolution.</p> <p>Upper 16 bits: Width of resolution</p> <p>Lower 16 bits: Height of resolution</p>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>The number the function gets is different by the mode the camera is in: LIVE or PLAY. The current resolution setting is given in the LIVE mode and the resolution setting of playback data is given in the PLAY mode.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0**GetGammaCorrection (ByRef mode As Long) As Long****VisualBasic.NET****GetGammaCorrection (ByRef mode As Integer) As Integer**

Functions Gets the current gamma correction mode.

Arguments **mode**

Specifies 32bit type parameters that receive a gamma correction value mode.

GAMMA_1_0 $\gamma = 1.0$

GAMMA_0_9 $\gamma = 0.9$

GAMMA_0_8 $\gamma = 0.8$

GAMMA_0_7 $\gamma = 0.7$

GAMMA_0_6 $\gamma = 0.6$

GAMMA_0_5 $\gamma = 0.5$

GAMMA_0_4 $\gamma = 0.4$

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
On cameras without gamma correction function, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code.

Remarks The number the function gets is different by the mode the camera is in: LIVE or PLAY. The current gamma correction setting is given in the LIVE mode and the gamma correction setting of playback data is given in the PLAY mode.

See "On Error Codes" for details of error codes [On Error Codes](#)

VisualBasic6.0

GetGainLevel (ByRef mode As Long) As Long

VisualBasic.NET

GetGainLevel (ByRef mode As Integer) As Integer

Functions Gets the current gain level mode.

Arguments **mode**

Specifies 32bit type parameters that receive a gain level value mode.

GAIN_0DB	0dB
GAIN_3DB	3dB
GAIN_6DB	6dB
GAIN_12DB	12dB
GAIN_18DB	18dB
GAIN_24DB	24dB
GAIN_30DB	30dB
GAIN_36DB	36dB

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
On cameras (color or monochrome) without gain level setting function, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code.

Remarks The number the function gets is different by the mode the camera is in: LIVE or PLAY. The current gain level setting is given in the LIVE mode and the gain level setting of playback data is given in the PLAY mode.

See "On Error Codes" for details of error codes [On Error Codes](#)

VisualBasic6.0

GetColorTemperature (ByRef mode As Long) As Long

VisualBasic.NET

GetColorTemperature (ByRef mode As Integer) As Integer

Functions	Gets the current color temperature mode.										
Arguments	<p>mode</p> <p>Specifies 32bit type parameters that receive a color temperature mode.</p> <table> <tr> <td>COLORTEMPMODE_DEF1</td><td>Default setting 1</td></tr> <tr> <td>COLORTEMPMODE_DEF2</td><td>Default setting 2</td></tr> <tr> <td>COLORTEMPMODE_USER1</td><td>User setting 1</td></tr> <tr> <td>COLORTEMPMODE_USER2</td><td>User setting 2</td></tr> <tr> <td>COLORTEMPMODE_USER3</td><td>User setting 3</td></tr> </table>	COLORTEMPMODE_DEF1	Default setting 1	COLORTEMPMODE_DEF2	Default setting 2	COLORTEMPMODE_USER1	User setting 1	COLORTEMPMODE_USER2	User setting 2	COLORTEMPMODE_USER3	User setting 3
COLORTEMPMODE_DEF1	Default setting 1										
COLORTEMPMODE_DEF2	Default setting 2										
COLORTEMPMODE_USER1	User setting 1										
COLORTEMPMODE_USER2	User setting 2										
COLORTEMPMODE_USER3	User setting 3										
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.										
Remarks	<p>On cameras (color or monochrome) without color temperature setting function,</p> <p>a PCC_ERROR_FUNCTION_DISABLE is returned as an error code. The number the function gets is different by the mode the camera is in: LIVE or PLAY. The current color temperature mode setting is given in the LIVE mode and the color temperature mode setting of playback data is given in the PLAY mode.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>										

Note: COLORTEMPMODE_USER2 and COLORTEMPMODE_USER3 are not currently used. COLORTEMPMODE_DEF1 and COLORTEMPMODE_DEF2 are set 5100K and 3100K, respectively.

VisualBasic6.0

**GetColorTempUser (ByVal mode As Long, ByRef r As Long,
ByRef g As Long, ByRef b As Long) As Long**

VisualBasic.NET

**GetColorTempUser (ByVal mode As Integer, ByRef r As Integer,
ByRef g As Integer, ByRef b As Integer) As Integer**

Functions	Gets the value of a specified user-setting color temperature.		
Arguments	mode	Specifies a user-setting color temperature mode to get white balance value.	
		COLORTEMPMODE_USER1	user setting 1
		COLORTEMPMODE_USER2	uset setting 2
		COLORTEMPMODE_USER3	user setting 3
	r	Specifies the 32bit type parameter that receives Red level.	
	g	Specifies the 32bit type parameter that receives Green level.	
	b	Specifies the 32bit type parameter that receives Blue lebel.。	
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR. On cameras (color or monochrome) without color temperature setting function, a PCC_ERROR_FUNCTION_DISABLE is returened as an error code.		
Remarks	The number the function gets is different by the mode the camera is in: LIVE or PLAY. The current white balance value setting is given in the LIVE mode and the white balance value of playback data is given in the PLAY mode. See “On Error Codes” for details of error codes On Error Codes Note: COLORTEMPMODE_USER2, and COLORTEMPMODE_USER3 are not currently used.		

VisualBasic6.0

GetExternalSyncMode (ByRef On As Long) As Long

VisualBasic.NET

GetExternalSyncMode (ByRef On As Integer) As Integer

Functions	Gets ON or OFF information of the current external sync mode.
Arguments	<div><div>on</div><div>Specifies 32bit type parameter that receives ON/OFF status of external sync setting.</div><div>[0]: OFF (Master mode)</div><div>[1]: ON (Slave mode)</div></div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	See "On Error Codes" for details of error codes On Error Codes

Note: This function has been combined with GetExternalSyncEx function and is only included here to match the old version SDK manual.

VisualBasic6.0**GetExternalSyncInMode (ByRef mode As Long) As Long****VisualBasic.NET****GetExternalSyncInMode (ByRef mode As Integer) As Integer**

Functions	Gets the current external sync mode status.	
Arguments	mode	Specifies 32bit type parameter that receives an external sync mode.
	EXTSYNC_NONE	OFF (Master mode)
	EXTSYNC_POSI	ON (Slave mode with positive signal input)
	EXTSYNC_NEGA	ON (Slave mode with negative signal input)
	EXTSYNC_OTHERS_POSI	ON (Positive signal)
	EXTSYNC_OTHERS_NEGA	ON (Negative signal)
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.	
Remarks	On cameras without Pos/Neg input signal setting, the function gets an ON or OFF only.	
	See "On Error Codes" for details of error codes On Error Codes	

VisualBasic6.0**GetExternalSyncModeEx (ByRef mode As Long) As Long****VisualBasic.NET****GetExternalSyncModeEx (ByRef mode As Integer) As Integer**

Functions Gets the current external sync mode setting.

Arguments **mode**

Specifies 32bit type parameter that receives external sync mode status.

EXTSYNC_NONE	OFF (Master mode)
EXTSYNC_POSI	ON (Slave mode with positive signal input)
EXTSYNC_NEGA	ON (Slave mode with negative signal input)
EXTSYNC_OTHERS_POSI	ON (Positive signal)
EXTSYNC_OTHERS_NEGA	ON (Negative signal)

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR.

Remarks On cameras without Pos/Neg input signal setting, the function gets an ON or OFF only.

See "On Error Codes" for details of error codes [On Error Codes](#)

VisualBasic6.0

GetExternalSyncOutMode (ByRef mode As Long) As Long

VisualBasic.NET

GetExternalSyncOutMode (ByRef mode As Integer) As Integer

Functions	Gets the current external sync mode setting.						
Arguments	<table><tr><td>mode</td><td>Specifies 32bit type parameter that receives external sync output mode status.</td></tr><tr><td>EXTOUT_VSYNC_POSI</td><td>VSYNC signal (Positive going)</td></tr><tr><td>EXTOUT_VSYNC_NEGA</td><td>VSYNC signal (Negative going)</td></tr></table>	mode	Specifies 32bit type parameter that receives external sync output mode status.	EXTOUT_VSYNC_POSI	VSYNC signal (Positive going)	EXTOUT_VSYNC_NEGA	VSYNC signal (Negative going)
mode	Specifies 32bit type parameter that receives external sync output mode status.						
EXTOUT_VSYNC_POSI	VSYNC signal (Positive going)						
EXTOUT_VSYNC_NEGA	VSYNC signal (Negative going)						
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.						
Remarks	See "On Error Codes" for details of error codes On Error Codes						

VisualBasic6.0

**GetExternallnMode (ByVal port As Long, ByRef mode As Long)
As Long**

VisualBasic.NET

**GetExternallnMode (ByVal port As Integer,
ByRef mode As Integer) As Integer**

Functions	Gets the current external general purpose signal input mode setting.	
Arguments	port	Specifies an output port number beginning with a "0".
	mode	Specifies 32bit type parameter that receives external general purpose signal input status.
	EXTIN_EVENT_POSI	Event signal (Positive going)
	EXTIN_EVENT_NEGA	Event signal (Negative going)
	EXTIN_TRIGGER_POSI	Trigger signal (Positive going)
	EXTIN_TRIGGER_NEGA	Trigger signal (Negative going)
	EXTIN_READY_POSI	Ready signal (Positive going)
	EXTIN_READY_NEGA	Ready signal (Negative going)
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.	
Remarks		

VisualBasic6.0

**GetExternalOutMode (ByVal port As Long,
ByRef mode As Long) As Long**

VisualBasic.NET

**GetExternalOutMode (ByVal port As Integer,
ByRef mode As Integer) As Integer**

Functions Gets the current external general purpose signal output mode setting.

Arguments

port	Specifies an output port number beginning with a "0".
mode	Specifies 32bit type parameter that receives the external signal output mode.
EXTOUT_RECORD_POSI	Recording period signal (Positive going)
EXTOUT_RECORD_NEGA	Recording period signal (Negative going)
EXTOUT_TRIGGER_POSI	Trigger signal (Positive going)
EXTOUT_TRIGGER_NEGA	Trigger signal (Negative going)
EXTOUT_EXPOSE_POSI	Exposure period signal (Positive going)
EXTOUT_EXPOSE_NEGA	Exposure period signal (Negative going)
EXTOUT_READY_POSI	Ready signal (Positive going)
EXTOUT_READY_NEGA	Ready signal (Negative going)
EXTOUT_IRIG_RESET_POSI	IRIG reset signal (Positive going)
EXTOUT_IRIG_RESET_NEGA	IRIG reset signal (Negative going)

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
On cameras (color or monochrome) without external signal output function, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code.

Remarks

The number designation of settable ports varies by the camera model.

The kind of settable output modes varies by the camera model and port number.

See "On Error Codes" for details of error codes [On Error Codes](#)

VisualBasic6.0**GetEdgeEnhancementMode (ByRef mode As Long) As Long****VisualBasic.NET****GetEdgeEnhancementMode (ByRef mode As Integer) As Integer**

Functions	Gets the current edge enhancement mode on the monitor output signal.										
Arguments	<table><tr><td>mode</td><td>Specifies 32bit type parameter that receives the current edge enhancement mode status.</td></tr><tr><td>EDGEMODE_NONE</td><td>No edge enhancement in effect</td></tr><tr><td>EDGEMODE_1</td><td>Edge enhancement level 1</td></tr><tr><td>EDGEMODE_2</td><td>Edge enhancement level 2</td></tr><tr><td>EDGEMODE_3</td><td>Edge enhancement level 3</td></tr></table>	mode	Specifies 32bit type parameter that receives the current edge enhancement mode status.	EDGEMODE_NONE	No edge enhancement in effect	EDGEMODE_1	Edge enhancement level 1	EDGEMODE_2	Edge enhancement level 2	EDGEMODE_3	Edge enhancement level 3
mode	Specifies 32bit type parameter that receives the current edge enhancement mode status.										
EDGEMODE_NONE	No edge enhancement in effect										
EDGEMODE_1	Edge enhancement level 1										
EDGEMODE_2	Edge enhancement level 2										
EDGEMODE_3	Edge enhancement level 3										
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR. On cameras without edge enhancement function, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code.										
Remarks	Effective on monitor output only. See "On Error Codes" for details of error codes On Error Codes										

Note: This function is not used on currently supported cameras, but is only included here to match the old version SDK manual.

VisualBasic6.0**GetZoomMode (ByRef On As Long) As Long****VisualBasic.NET****GetZoomMode (ByRef On As Integer) As Integer**

Functions	Gets the current zooming display mode of monitor output signal.						
Arguments	<table><tr><td>on</td><td>Specifies the 32bit type parameter that receives the status of ON/OFF of zooming display.</td></tr><tr><td>[1]</td><td>ON</td></tr><tr><td>[0]</td><td>OFF</td></tr></table>	on	Specifies the 32bit type parameter that receives the status of ON/OFF of zooming display.	[1]	ON	[0]	OFF
on	Specifies the 32bit type parameter that receives the status of ON/OFF of zooming display.						
[1]	ON						
[0]	OFF						
Returned values	Returns an error code. Whenk normal, returns a PCC_ERROR_NOERROR. On cameras without zooming display function or monitor output, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code.						
Remarks	Effective on the monitor output signal only. See "On Error Codes" for details of error codes On Error Codes						

Note: This function is not used on currently supported cameras, but is only included here to match the old version SDK manual.

VisualBasic6.0**GetMonitorOutMode (ByRef mode As Long) As Long****VisualBasic.NET****GetMonitorOutMode (ByRef mode As Integer) As Integer**

Functions	Gets the external monitor output mode.
Arguments	mode Specifies 32bit type parameter that receives the extenal monitor output mode. MONITOROUT_VGA VGA monitor output MONITOROUT_NTSC NTSC monitor output
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR. On cameras without error code function, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code.
Remarks	See "On Error Codes" for details of error codes On Error Codes Note: This function has been combined with ExtraCommand function. It is only included here to match the old version of SDK manual.

VisualBasic6.0**GetEnableMCDL (ByRef On As Long) As Long****VisualBasic.NET****GetEnableMCDL (ByRef On As Integer) As Integer**

Functions	Gets ON/OFF status of the MCDL function.						
Arguments	<table><tr><td>on</td><td>Specifies 32bit type parameter that receives ON/OFF status of the MCDL status.</td></tr><tr><td>[1]</td><td>MCDL function ON</td></tr><tr><td>[0]</td><td>MCDL function OFF</td></tr></table>	on	Specifies 32bit type parameter that receives ON/OFF status of the MCDL status.	[1]	MCDL function ON	[0]	MCDL function OFF
on	Specifies 32bit type parameter that receives ON/OFF status of the MCDL status.						
[1]	MCDL function ON						
[0]	MCDL function OFF						
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR. With a camera without MCDL function, a PCC_ERROR_FUNCTION_DISABLE is returned.						
Remarks	See the hardware manual of each camera for details of MCDL function. See "On Error Codes" for details of error codes On Error Codes .						

Note: This function has been combined with ExtraCommand function, and is only included here to match the old version of SDK manual.

VisualBasic6.0**GetEnableIRIG (ByRef On As Long) As Long****VisualBasic.NET****GetEnableIRIG (ByRef On As Integer) As Integer**

Functions	Gets ON/OFF status of the IRIG function.
Arguments	<div><div>on</div><div>Specifies 32bit type parameter that receives the ON/OFF status of the IRIG function.</div><div><div>[1]</div><div>MCDL function ON</div></div><div><div>[0]</div><div>MCDL function OFF</div></div></div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR. With a camera without IRIG function, a PCC_ERROR_FUNCTION_DISABLE is returned.
Remarks	<div>See the hardware manual of each camera for details of IRIG function.</div> <div>See "On Error Codes" for details of error codes On Error Codes.</div> <div>Note: This function has been combined with ExtraCommand function, and is only included here to match the old version of SDK manual.</div>

VisualBasic6.0

**GetRecordRateList (ByRef list As Long, ByVal listSize As Long)
As Long**

VisualBasic.NET

**GetRecordRateList (ByRef list As Integer,
ByVal listSize As Integer) As Integer**

Functions	Gets a list of recording rates that can be set on the camera.
Arguments	<div><div>list</div><div>Specifies the 32bit type address of memory array that receives the recording rate list</div><div>listsize</div><div>Specifies the size of the list memory array. The default size is LIST_MAX_NUMBER.</div></div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<div>The size of list memory array is normally determined by a LIST_MAX_NUMBER.</div> <div>See "On Error Codes" for details of error codes On Error Codes</div>

VisualBasic6.0

**GetShutterSpeedList (ByRef list As Long,
ByVal listSize As Long) As Long**

VisualBasic.NET

**GetShutterSpeedList (ByRef list As Integer,
ByVal listSize As Integer) As Integer**

Functions	Gets a list of shutter speeds that are currently available to set.								
Arguments	<p>list Specifies 32bit type address of memory array that receives the shutter speed list.</p> <p>listsize Specifies the size of the list memory array. The default size is LIST_MAX_NUMBER.</p>								
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.								
Remarks	<p>For list memory array, secure the size of LIST_MAX_NUMBER unless there is any problem.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p> <p>If a "1" exists in a shutter speed list, the camera is compatible with special shutter speeds (eg. FASTCAM-ultima APX camera). A camera compatible with special shutter speeds can get information on the following:</p> <table> <tr> <td>list[n] = 1</td><td>A camera compatible with special shutter speeds</td></tr> <tr> <td>list[n+1]</td><td>Shutter speed variable unit</td></tr> <tr> <td>list[n+2]</td><td>Minimum shutter speed</td></tr> <tr> <td>list[n+3]</td><td>Maximum shutter speed</td></tr> </table> <p>With a camera compatible with special shutter speeds, any shutter speed between the minimum and maximum shutter speeds can be set with an interval of the above variable unit.</p> <p>Also, a shutter speed that is the same as the one currently used can be set.</p>	list[n] = 1	A camera compatible with special shutter speeds	list[n+1]	Shutter speed variable unit	list[n+2]	Minimum shutter speed	list[n+3]	Maximum shutter speed
list[n] = 1	A camera compatible with special shutter speeds								
list[n+1]	Shutter speed variable unit								
list[n+2]	Minimum shutter speed								
list[n+3]	Maximum shutter speed								

VisualBasic6.0

**GetTriggerModeList (ByRef list As Long,
ByVal listSize As Long) As Long**

VisualBasic.NET

**GetTriggerModeList (ByRef list As Integer,
ByVal listSize As Integer) As Integer**

Functions	Gets a list of trigger mode that is available with the camera.
Arguments	<div><div>list</div><div>Specifies 32bit type address of memory array that receives the trigger mode list.</div></div> <div><div>listsize</div><div>Specifies the size of memory array . The default size is LIST_MAX_NUMBER.</div></div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>The size of list memory array is nusually determined, and secured, by a LIST_MAX_NUMBER.</p> <p>See the hardware manual of each camera for details of trigger modes.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0

**GetResolutionList (ByRef list As Long, ByVal listSize As Long)
As Long**

VisualBasic.NET

**GetResolutionList (ByRef list As Integer,
ByVal listSize As Integer) As Integer**

Functions	Gets a list of resolutions that are currently available for setting.		
Arguments	list	Specifies 32bit type address of memory array that receives the resolution list.	
		Upper 16 bits	Width
		Lower 16 bits	Height
	listsize	Specifies the size of list memory array. The default size is LIST_MAX_NUMBER.	
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.		
Remarks	The size of list memory array is usually determined, and secured, by a LIST_MAX_NUMBER.		
	Because the resolutions available for setting varies by the current recording rate, it may be necessary to get another list when changing the recording rate.		
	See “On Error Codes” for details of error codes On Error Codes		

VisualBasic6.0

**GetRandomTriggerNumberList (ByRef list As Long,
ByVal listSize As Long) As Long**

VisualBasic.NET

**GetRandomTriggerNumberList (ByRef list As Integer,
ByVal listSize As Integer) As Integer**

Functions Gets a list of the numbers of frames that can be set on the camera for recording in RANDOM trigger mode.

Arguments

list Specifies 32bit type address of memory array that receives the list of recording frames.

listsize Specifies the size of list memory array.
The default size is LIST_MAX_NUMBER.

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR.

Remarks The size of list memory array is usually determined, and secured, by a LIST_MAX_NUMBER.

See the hardware manual of each camera for details of trigger modes.

See "On Error Codes" for details of error codes [On Error Codes](#).

VisualBasic6.0

**GetColorTemperatureList (ByRef list As Long,
ByVal listSize As Long) As Long**

VisualBasic.NET

**GetColorTemperatureList (ByRef list As Integer,
ByVal listSize As Integer) As Integer**

Functions	Gets a list of color temperature (white balance) mode available for setting on cameras.										
Arguments	<p>list</p> <p>list</p> <p>Specifies 32bit type memory array address that receives the of color temperature (white balance).</p> <table> <tr> <td>COLORTEMP_MODE_DEF1</td><td>Default setting 1</td></tr> <tr> <td>COLORTEMP_MODE_DEF2</td><td>Default setting 2</td></tr> <tr> <td>COLORTEMP_MODE_USER1</td><td>User setting 1</td></tr> <tr> <td>COLORTEMP_MODE_USER2</td><td>User setting 2</td></tr> <tr> <td>COLORTEMP_MODE_USER3</td><td>User setting 3</td></tr> </table> <p>listsize</p> <p>Specifies the size of list memory array. The default size is a LIST_MAX_NUMBER.</p>	COLORTEMP_MODE_DEF1	Default setting 1	COLORTEMP_MODE_DEF2	Default setting 2	COLORTEMP_MODE_USER1	User setting 1	COLORTEMP_MODE_USER2	User setting 2	COLORTEMP_MODE_USER3	User setting 3
COLORTEMP_MODE_DEF1	Default setting 1										
COLORTEMP_MODE_DEF2	Default setting 2										
COLORTEMP_MODE_USER1	User setting 1										
COLORTEMP_MODE_USER2	User setting 2										
COLORTEMP_MODE_USER3	User setting 3										
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR. With a camera without color temperature setting function, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code.										
Remarks	<p>The size of list memory array is usually determined, and secured, by LIST_MAX_NUMBER.</p> <p>See "On Error Codes" for details of error codes On Error Codes.</p>										

Note: COLORTEMP_MODE_USER2 and COLORTEMP_MODE_USER3 are not currently used. COLORTEMP_MODE_DEF1 and COLORTEMP_MODE_DEF2 are set 5100K and 3100K, respectively.

VisualBasic6.0

**GetGammaCorrectionList (ByRef list As Long,
ByVal listSize As Long) As Long**

VisualBasic.NET

**GetGammaCorrectionList (ByRef list As Integer,
ByVal listSize As Integer) As Integer**

Funtions	Gets a list of gamma correction values available to be set on cameras.
Arguments	<div><div>list</div><div>Specifies 32bit type memory array address that receives the gamma correction mode list.</div></div> <div><div>listsize</div><div>Specifies the size of memory array. The default size is LIST_MAX_NUMBER.</div></div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR. With a camera without gamma correction function, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code.
Remarks	<p>The size of list memory array is usually determined, and secured, by a LIST_MAX_NUMBER.</p> <p>See "On Error Codes" for details of error codes On Error Codes.</p>

VisualBasic6.0

**GetGainLevelList (ByRef list As Long, ByVal listSize As Long)
As Long**

VisualBasic.NET

**GetGainLevelList (ByRef list As Integer,
ByVal listSize As Integer) As Integer**

Functions	Gets a list of gain level modes that can be set on cameras.
Arguments	<div><div>list</div><div>Specifies an address for 32bit type memory array that receives the mode list of gain levels.</div></div> <div><div>listsize</div><div>Specifies the size of the list memory array. The default size is LIST_MAX_NUMBER.</div></div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR. With a camera without gainsetting function, a PCC_ERROR_FUNCTION_DISABLE is returned.
Remarks	The size of list memory array is usually determined, and secured, by a LIST_MAX_NUMBER. See "On Error Codes" for details of error codes On Error Codes .

VisualBasic6.0

**GetPartitionInfo (ByRef max_partition As Long,
ByRef max_partition_block As Long,
ByRef frame_per_block As Long) As Long**

VisualBasic.NET

**GetPartitionInfo (ByRef max_partition As Integer,
ByRef max_partition_block As Integer,
ByRef frame_per_block As Integer) As Integer**

Functions	Gets partition information.
Arguments	<p>max_partition The maximum number of available partitions.</p> <p>max_partition_block The maximum number of available blocks.</p> <p>frame_per_block The number of frames per block.</p>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>Arguments [max_partition_block] and [frame_per_block] vary by the current status of respective cameras.</p> <p>When the value for [frame_per_block] is [-1] on a camera, you cannot decide the number of blocks per partition for it.</p> <p>See "On Error Codes" for details of error codes On Error Codes.</p>

VisualBasic6.0

GetPartition (ByRef partition As Long) As Long

VisualBasic.NET

GetPartition (ByRef partition As Integer) As Integer

Functions	Gets a number for the partition being used.
Arguments	<p>partition</p> <p>Specifies 32bit type parameter that gets a partition number beginning with "1".</p>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>The size of list memory array is usually determined, and secured, by a PARTITIONT_MAX_NUMBER.</p> <p>See "On Error Codes" for details of error codes On Error Codes.</p>

VisualBasic6.0

**GetPartitionBlockList (ByRef list As Long,
ByVal listSize As Long) As Long**

VisualBasic.NET

**GetPartitionBlockList (ByRef list As Integer,
ByVal listSize As Integer) As Integer**

Functions	Gets a list of partitions.
Arguements	<div><div>list</div><div>Specifies a address for 32bit type memory array containing the number of blocks in each partition.</div><div>listsize</div><div>Specifies the size of list memory array. The default size is determined by a PARTITION_MAX_NUMBER.</div></div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	See "On Error Codes" for details of error codes On Error Codes .

VisualBasic6.0

**GetExtraCommand (ByVal command As Long,
ByRef param1 As Long, ByRef param2 As Long,
ByRef param3 As Long, ByRef param4 As Long)
As Long**

VisualBasic.NET

**GetExtraCommand (ByVal command As Integer,
ByRef param1 As Integer, ByRef param2 As Integer,
ByRef param3 As Integer, ByRef param4 As Integer)
As Integer**

Functions	Gets Executes extra commands on each of cameras
Arguements	<p>command Specifies an extra command</p> <p>param1 First argument of extra command (cannot be omitted)</p> <p>param2 Second argument of extra command (set 0 when omitted)</p> <p>param3 Third argument of extra command (set 0 when omitted)</p> <p>param4 Fourth argument of extra command (set 0 when omitted)</p>
Returned values	<p>Returns an error code. When normal, returns a PCC_ERROR_NOERROR.</p> <p>On cameras without the specified command, a PCC_ERROR_NOT_SUPPORTED is returned as an error code.</p>
Remarks	<p>See "On Extra Commands" for details of extra commands On Extra Commands</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0

GetFrameParams (ByRef params As FRAME_PARAMS) As Long

VisualBasic.NET

GetFrameParams (ByRef params As FRAME_PARAMS)

As Integer

Functions	Gets frame management information.
Arguments	<p>params</p> <p>Specifies a address of FRAME_PARAMS structure to store.</p>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	See "On Error Codes" for details of error codes On Error Codes .

VisualBasic6.0

**GetShutterSpeedType2List(ByRef list As Long,
ByVal listSize As Long) As Long**

VisualBasic.NET

**GetShutterSpeedType2List(ByRef list As Integer,
ByVal listSize As Integer) As Integer**

Functions	Gets a list of shutter speeds that are currently available to set.
Arguments	<div><div>list</div><div>Specifies 32bit type address of memory array that receives the shutter speed list(usec).</div></div> <div><div>listsize</div><div>Specifies the size of the list memory array. The default size is LIST_MAX_NUMBER.</div></div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>For list memory array, secure the size of LIST_MAX_NUMBER unless there is any problem.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0

GetShutterSpeedType2(ByRef speed As Long) As Long

VisualBasic.NET

GetShutterSpeedType2(ByRef speed As Integer) As Integer

Functions	Gets the current shutter speed.
Arguments	<p>speed</p> <p>Specifies a 32bit type parameter that receives a shutter speed (usec).</p>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>The number the function gets is different by the mode the camera is in: LIVE or PLAY. The current recording rate is given in the LIVE mode and the recording rate of playback data is given in the PLAY mode.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0

**GetHeadName(ByVal name As String, ByVal no As Long)
As Long**

VisualBasic.NET

**GetHeadName(ByVal name As String, ByVal no As Integer)
As Integer**

Functions	Gets the camera head name				
Arguments	<table><tr><td>name</td><td>Specifies the address of LPTSTR type character sequence buffer which receives a camera head name</td></tr><tr><td>no</td><td>Specifies camera head number to acquire (1-3)</td></tr></table>	name	Specifies the address of LPTSTR type character sequence buffer which receives a camera head name	no	Specifies camera head number to acquire (1-3)
name	Specifies the address of LPTSTR type character sequence buffer which receives a camera head name				
no	Specifies camera head number to acquire (1-3)				
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.				
Remarks	Use only with the dissociated type camera corresponding to the head exchange function is possible.				

VisualBasic6.0

GetVariableSetting(ByVal nChannel As Long,
ByRef nFrameRate As Long, ByRef nWidth As Long,
ByRef nHeight As Long, ByRef nXPos As Long,
ByRef nYPos As Long) As Long

VisualBasic.NET

GetVariableSetting(ByVal nChannel As Integer,
ByRef nFrameRate As Integer, ByRef nWidth As Integer,
ByRef nHeight As Integer, ByRef nXPos As Integer,
ByRef nYPos As Integer) As Integer

Functions	Gets the setting value of a specification channel by variable setting.
Arguments	<p>nChannel Specifies acquisition channel (1-20)</p> <p>nFrameRate Integer parameter that frame rate is received.</p> <p>nWidth Integer parameter that receives the width of resolution</p> <p>nHeight Integer parameter that receives the height of resolution</p> <p>nXPos Integer parameter that receives upper left X coordinates of rectangle</p> <p>nYPos Integer parameter that receives upper left Y coordinates of rectangle</p>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	

VisualBasic6.0

**GetVariableCamMode(ByRef nMode As Long,
ByRef nChannel As Long) As Long**

VisualBasic.NET

**GetVariableCamMode(ByRef nMode As Integer,
ByRef nChannel As Integer) As Integer**

Functions Gets the current variable mode.

Arguments **nMode**

Integer parameter that variable mode is received.

CAMMODE_DEFAULT Default mode

CAMMODE_VARIABLE Variable mode

CAMMODE_EXTERNAL External (Signal input) mode

nChannel

Integer parameter that receives current variable channel.

Returned values Returns an error code. When normal, returns a
PCC_ERROR_NOERROR.

Remarks This function is used on with a "Variable function" cameras

VisualBasic6.0**GetGeometricConvert (ByRef mode As Long) As Long****VisualBasic.NET****GetGeometricConvert (ByRef mode As Integer) As Integer**

Functions Gets a status of rotation and mirroring of the output image.

Arguments **mode** Gets rotation and mirroring mode.。

IMAGE_CONV_DEFAULT	No rotation
IMAGE_CONV_ROTATE90	90-degree rotation to right
IMAGE_CONV_ROTATE180	180-degree rotation to right
IMAGE_CONV_ROTATE270	270-degree rotation to right
IMAGE_CONV_MIRROR_H	Horizontal mirroring
IMAGE_CONV_MIRROR_V	Vertical mirroring

Gets a combined value.

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR.

Remarks

VisualBasic6.0

**GetAutoExposureArea (ByRef nWidth As Long,
ByRef nHeight As Long, ByRef nXPos As Long,
ByRef nYPos As Long) As Long**

VisualBasic.NET

**GetAutoExposureArea(ByRef nWidth As Integer,
ByRef nHeight As Integer, ByRef nXPos As Integer,
ByRef nYPos As Integer) As Integer**

Functions	Gets a target area for Auto Exposure function.
Arguments	<p>nWidth Gets the width for a target area.</p> <p>nHeight Sets the height for a target area.</p> <p>nXPos Sets the X-coordinate of the upper left corner of a target area.</p> <p>nYPos Sets the Y-coordinate of the upper left corner of a target area.</p>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	XPos and nYPos represent the coordinate of the maximum resolution of the camera (with FASTCAM-APX-RS, for example, 1024 x 1024). As an example, for a resolution of 512 x 512 around an optical axis, the coordinate of the origin in the upper left corner is X = 256 and Y = 256.

VisualBasic6.0

**GetAutoExposureParam (ByRef nValue As Long,
ByRef nRange As Long) As Long**

VisualBasic.NET

**GetAutoExposureParam (ByRef nValue As Integer,
ByRef nRange As Integer) As Integer**

Functions Gets the image output level for the Auto Exposure function.

Arguments

nValue	Image output level (0 to 255)
nRange	Range of image output level (0 to 255)

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR.

Remarks

VisualBasic6.0

**GetShutterSpeedAEList (ByRef list As Long,
ByVal listsize As Long, ByRef listnum As Long) As Long**

VisualBasic.NET

**GetShutterSpeedAEList (ByRef list As Integer,
ByVal listsize As Integer, ByRef listnum As Integer)
As Integer**

Functions	Gets an exposure period (shutter speed) that can be set for the Auto Exposure function.
Arguments	<p>list Specifies the address of an 32bit-type memory array that receives an exposure period (1/shutter speed) that can be set at the current frame rate and resolution.</p> <p>listsize Specifies the size of a list memory array. The default size is LIST_MAX_NUMBER_EX</p> <p>listnum The number of lists to get.</p>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	For the size of a list memory array, secure a size that is the same as that for LIST_MAX_NUMBER_EX (not LIST_MAX_NUMBER) unless there is a specific problem.

VisualBasic6.0

OnLive (ByVal On As Long) As Long

VisualBasic.NET

OnLive (ByVal On As Integer) As Integer

Functions Switches between LIVE and PLAY modes.

Arguments **on** Specifies a camera mode.
[1]: LIVE mode
[0]: PLAY mode

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR.

Remarks See "On Error Codes" for details of error codes [On Error Codes](#)

VisualBasic6.0**IsLive (ByRef On As Long) As Long****VisualBasic.NET****IsLive (ByRef On As Integer) As Integer**

Functions	Gets the current camera mode.						
Arguments	<table><tr><td>on</td><td>Specifies a 32bit type parameter that receives the camera mode.</td></tr><tr><td>[1]</td><td>LIVE mode</td></tr><tr><td>[0]</td><td>PLAY mode</td></tr></table>	on	Specifies a 32bit type parameter that receives the camera mode.	[1]	LIVE mode	[0]	PLAY mode
on	Specifies a 32bit type parameter that receives the camera mode.						
[1]	LIVE mode						
[0]	PLAY mode						
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.						
Remarks	See "On Error Codes" for details of error codes On Error Codes						

VisualBasic6.0**GetCameraMode (ByRef mode As Long) As Long****VisualBasic.NET****GetCameraMode (ByRef mode As Integer) As Integer**

Functions Gets the current camera mode.

Arguments **mode**

Gets a camera mode as a flag.

CAMERA_MODE_PLAY	PLAY
CAMERA_MODE_LIVE	LIVE
CAMERA_MODE_RECREADY	REC READY
CAMERA_MODE_REC	REC
CAMERA_MODE_ENDLESS	ENDLESS
CAMERA_MODE_RECORDED	RECORDED

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR.

Remarks See "On Error Codes" for details of error codes [On Error Codes](#).

VisualBasic6.0**OnRecordReady (ByVal On As Long) As Long****VisualBasic.NET****OnRecordReady (ByVal On As Integer) As Integer**

Functions	Sets ON or OFF of the ready-to-record (REC READY) mode.						
Arguments	<table><tr><td>on</td><td>Specifies ON or OFF of the ready-to-record status.</td></tr><tr><td>[1]</td><td>REC READY status ON</td></tr><tr><td>[0]</td><td>REC READY status OFF (LIVE mode)</td></tr></table>	on	Specifies ON or OFF of the ready-to-record status.	[1]	REC READY status ON	[0]	REC READY status OFF (LIVE mode)
on	Specifies ON or OFF of the ready-to-record status.						
[1]	REC READY status ON						
[0]	REC READY status OFF (LIVE mode)						
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR. When the camera is in the PLAY mode, a PCC_ERROR_CAMERAMODE is returned.						
Remarks	This function can only be set when the camera is in the LIVE mode. See [OnLive] function. See "On Error Codes" for details of error codes On Error Codes						

VisualBasic6.0**OnRecord (ByVal On As Long) As Long****VisualBasic.NET****OnRecord (ByVal On As Integer) As Integer**

Functions	Starts and stops recording.						
Arguments	<table><tr><td>on</td><td>Specifies start or stop recording.</td></tr><tr><td>[1]</td><td>Starts recording (ON)</td></tr><tr><td>[0]</td><td>Stops recording (OFF)</td></tr></table>	on	Specifies start or stop recording.	[1]	Starts recording (ON)	[0]	Stops recording (OFF)
on	Specifies start or stop recording.						
[1]	Starts recording (ON)						
[0]	Stops recording (OFF)						
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR. When the camera is in the PLAY mode, a PCC_ERROR_CAMERAMODE is returned.						
Remarks	A start of recording is only possible when the camera is in the REC READY mode. See the [OnRecordReady] function. See "On Error Codes" for details of error codes On Error Codes						

VisualBasic6.0

IsRecordReady (ByRef On As Long) As Long

VisualBasic.NET

IsRecordReady (ByRef On As Integer) As Integer

Functions	Gets information whether or not the camera is in REC READY mode.		
Arguments	on	Specifies a 32bit type parameter that receives REC READY mode.	
	[1]	REC READY Mode ON	
	[0]	REC READY Mode OFF	
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.		
Remarks	See "On Error Codes" for details of error codes On Error Codes		

VisualBasic6.0**IsRecord (ByRef On As Long) As Long****VisualBasic.NET****IsRecord (ByRef On As Integer) As Integer**

Functions	Gets information whether or not the camera is currently recording.	
Arguments	on	Specifies a 32bit type parameter that receives the recording status.
	[1]	Camera is currently recording.
	[0]	Camera is not recording.
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.	
Remarks	See "On Error Codes" for details of error codes On Error Codes	

VisualBasic6.0
TriggerIn () As Long

VisualBasic.NET
TriggerIn () As Integer

Functions	Inputs software triggers
Auguments	None
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR. When the camera is in the PLAY mode, a PCC_ERROR_CAMERAMODE is returned.
Remarks	<p>In the START trigger mode, this function can be used only when the camera is in the REC READY mode. In other trigger modes, it can be used any time the camera is recording. See [OnRecordReady] and [OnRecord] member functions.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p> <p>Note: In a multiple-camera operation with FASTCAM-PCI, PCI-R2 or 1280PCI cameras, the master camera only receives this command. Because slave cameras automatically receive a trigger, this function does not have to be called out for each of the slave cameras.</p>

VisualBasic6.0**IsEndlessRec (ByRef On As Long) As Long****VisualBasic.NET****IsEndlessRec (ByRef On As Integer) As Integer**

Functions	Gets information whether or not the camera is currently recording in the endless mode.						
Arguments	<table><tr><td>on</td><td>Specifies a 32bit type parameter that receives the recording status.</td></tr><tr><td>[1]</td><td>Recording in the endless mode</td></tr><tr><td>[0]</td><td>Not recording in the endless mode</td></tr></table>	on	Specifies a 32bit type parameter that receives the recording status.	[1]	Recording in the endless mode	[0]	Not recording in the endless mode
on	Specifies a 32bit type parameter that receives the recording status.						
[1]	Recording in the endless mode						
[0]	Not recording in the endless mode						
Returned values	Retruns an error code. When normal, returns a PCC_ERROR_NOERROR .						
Remarks	See "On Error Codes" for details of error codes On Error Codes						

VisualBasic6.0

PlayMonitor (ByVal mode As Long) As Long

VisualBasic.NET

PlayMonitor (ByVal mode As Integer) As Integer

Functions Displays playback image data on the monitor.

Arguments **mode**

 Specifies a playback mode.

PLAYMODE_PLAY	Playback
PLAYMODE_PLAYBACK	Reverse playback
PLAYMODE_FASTPLAY	Fast forward
PLAYMODE_FASTPLAYBACK	Fast reverse

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR.

 With a camera that has no monitor output, a PCC_ERROR_FUNCTION_DISABLE is returned.

 When the camera is in the LIVE mode, a PCC_ERROR_CAMERAMODE is returned.

Remarks This function is effective on monitor output only. It can only be set when the camera is in the PLAY mode. See [OnLive] function.

See "On Error Codes" for details of error codes [On Error Codes](#)

Note: This function has been combined with [ExtraCommand] functions. It is only included here to match the old version of SDK manual.

Note 2: This function is not supported by cameras that are newer than the FASTCAM ultima 1024.

VisualBasic6.0

PauseMonitor (ByVal On As Long) As Long

VisualBasic.NET

PauseMonitor (ByVal On As Integer) As Integer

Functions	Sets ON/OFF of pause on playback monitor output.
Arguments	<div>on Specifies ON or OFF of pause. [1]: Pause ON [0]: Pause OFF</div>
Returned values	<p>Returns an error code. When normal, returns a PCC_ERROR_NOERROR.</p> <p>On cameras without monitor output, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code.</p> <p>When the camera is in the LIVE mode, a PCC_ERROR_CAMERAMODE is returned.</p>
Remarks	<p>This function is effective on monitor output only. It can only be used when the camera is in the PLAY mode. See [OnLive] function.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p> <p>Note: This function has been combined with [ExtraCommand] functions. It is only included here to match the old version of SDK manual.</p> <p>Note 2: This function is not supported by cameras that are newer than the FASTCAM ultima 1024.</p>

VisualBasic6.0
StopMonitor () As Long

VisualBasic.NET
StopMonitor () As Integer

Functions	Stops the monitor output.
Arguments	None
Returned values	Returns and error code. When normal, returns a PCC_ERROR_NOERROR. On cameras without monitor output, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code. When the camera is in the LIVE mode, a PCC_ERROR_CAMERAMODE is returned.
Remarks	<p>This function is effective on monitor outpur only. It can only be used when the camera is in the PLAY mode. See [OnLive] function.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p> <p>Note: This function has been combined with [ExtraCommand] functions. It is only included here to match the old version of SDK manual.</p> <p>Note 2: This function is not supported by cameras that are newer than the FASTCAM ultima 1024.</p>

VisualBasic6.0
StepFowardMonitor () As Long

VisualBasic.NET
StepFowardMonitor () As Integer

Functions	Steps the displayed image one frame forward.
Arguments	None.
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR On cameras without monitor output, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code. When the camera is in the LIVE mode, a PCC_ERROR_CAMERAMODE is returned.
Remarks	<p>This function is effective on the monitor output only. It can only be used when the camera is in the PLAY mode. See [OnLive] function.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p> <p>Note: This function has been combined with [ExtraCommand] functions. It is only included here to match the old version of SDK manual.</p> <p>Note 2: This function is not supported by cameras that are newer than the FASTCAM ultima 1024.</p>

VisualBasic6.0
StepBackMonitor () As Long

VisualBasic.NET
StepBackMonitor () As Integer

Functions	Steps the displayed image one frame backward.
Arguments	None.
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR On cameras without monitor output, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code. When the camera is in the LIVE mode, a PCC_ERROR_CAMERAMODE is returned.
Remarks	<p>This function is effective on the monitor output only. It can only be used when the camera is in the PLAY mode. See [OnLive] function.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p> <p>Note: This function has been combined with [ExtraCommand] functions. It is only included here to match the old version of SDK manual.</p> <p>Note 2: This function is not supported by cameras that are newer than the FASTCAM ultima 1024.</p>

VisualBasic6.0**GoAnyFrameMonitor (ByVal frame As Long) As Long****VisualBasic.NET****GoAnyFrameMonitor (ByVal frame As Integer) As Integer**

Functions	Moves the displayed image to any specified frame.
Arguments	frame Specifies the number of frames to move.
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR On cameras without monitor output, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code. When the camera is in the LIVE mode, a PCC_ERROR_CAMERAMODE is returned.
Remarks	<p>This function is effective on the monitor output only. It can only be used when the camera is in the PLAY mode. See [OnLive] member function. The system automatically comes into a temporary stop status after the specified frame has been displayed.</p> <p>See "Details of Frame Parameters" for details of frame numbers.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p> <p>Note: This function has been combined with [ExtraCommand] functions. It is only included here to match the old version of SDK manual.</p> <p>Note 2: This function is not supported by cameras that are newer than the FASTCAM ultima 1024.</p>

VisualBasic6.0

GoStartMonitor () As Long

VisualBasic.NET

GoStartMonitor () As Integer

Functions	Jumps to the first frame of a recording.
Arguments	None.
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR On cameras without monitor output, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code. When the camera is in the LIVE mode, a PCC_ERROR_CAMERAMODE is returned.
Remarks	<p>This function is effective on the monitor output only. It can only be used when the camera is in the PLAY mode. See [OnLive] function.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p> <p>Note: This function has been combined with [ExtraCommand] functions. It is only included here to match the old version of SDK manual.</p> <p>Note 2: This function is not supported by cameras that are newer than the FASTCAM ultima 1024.</p>

VisualBasic6.0**GoEndMonitor () As Long****VisualBasic.NET****GoEndMonitor () As Integer**

Functions	Jumps to the last frame of a recording. Jumps to the last frame of a block in the Block playback mode.
Arguments	None.
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR On cameras without monitor output, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code. When the camera is in the LIVE mode, a PCC_ERROR_CAMERAMODE is returned.
Remarks	This function is effective on the monitor output only. It can only be used when the camera is in the PLAY mode. See [OnLive] function. See "On Error Codes" for details of error codes On Error Codes

Note: This function has been combined with [ExtraCommand] functions. It is only included here to match the old version of SDK manual.

Note 2: This function is not supported by cameras that are newer than the FASTCAM ultima 1024.

VisualBasic6.0**GoTriggerMonitor () As Long****VisualBasic.NET****GoTriggerMonitor () As Integer**

Functions	Jumps to a trigger frame
Arguments	None.
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR On cameras without monitor output, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code. When the camera is in the LIVE mode, a PCC_ERROR_CAMERAMODE is returned.
Remarks	This function is effective on the monitor output only. It can only be used when the camera is in the PLAY mode. See [OnLive] function.

See "On Error Codes" for details of error codes [On Error Codes](#)

Note: This function has been combined with [ExtraCommand] functions. It is only included here to match the old version of SDK manual.

Note 2: This function is not supported by cameras that are newer than the FASTCAM ultima 1024.

VisualBasic6.0**SetPlayRate (ByVal rate As Long) As Long****VisualBasic.NET****SetPlayRate (ByVal rate As Integer) As Integer**

Functions	Sets playback rate of the monitor output.
Arguments	<div>rate Specifies a playback rate (FPS). Available playback rates: 30, 15, 10, 5, 2, 1</div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR On cameras without monitor output, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code. When the camera is in the LIVE mode, a PCC_ERROR_CAMERAMODE is returned.
Remarks	<p>This function is effective on the monitor output only. It can only be used when the camera is in the PLAY mode. See [OnLive] function.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p> <p>Note: This function has been combined with [ExtraCommand] functions. It is only included here to match the old version of SDK manual.</p> <p>Note 2: This function is not supported by cameras that are newer than the FASTCAM ultima 1024.</p>

VisualBasic6.0

**SetBlockArea (ByVal start As Long, ByVal end As Long)
As Long**

VisualBasic.NET

**SetBlockArea (ByVal start As Integer,
ByVal end As Integer) As Integer**

Functions	Sets a range (block) of frames for playback on monitor output.
Arguments	<p>start Specifies the first frame of a range (block).</p> <p>end Specifies the last frame of a range (block).</p>
Returned values	<p>Returns an error code. When normal, returns a PCC_ERROR_NOERROR</p> <p>On cameras without monitor output, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code.</p> <p>When the camera is in the LIVE mode, a PCC_ERROR_CAMERAMODE is returned.</p>
Remarks	<p>This function is effective on the monitor output only. It can only be used when the camera is in the PLAY mode. See [OnLive] function. See "Details of Frame Parameters" for details of frame numbers.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p> <p>Note: This function has been combined with [ExtraCommand] functions. It is only included here to match the old version of SDK manual.</p> <p>Note 2: This function is not supported by cameras that are newer than the FASTCAM ultima 1024.</p>

VisualBasic6.0

SetBlockMode (ByVal On As Long) As Long

VisualBasic.NET

SetBlockMode (ByVal On As Integer) As Integer

Functions	Sets ON/OFF of block playback on monitor output.						
Arguments	<table><tr><td>on</td><td>Specifies ON/OFF of block playback.</td></tr><tr><td>[1]</td><td>Block playback ON</td></tr><tr><td>[0]</td><td>Block playback OFF</td></tr></table>	on	Specifies ON/OFF of block playback.	[1]	Block playback ON	[0]	Block playback OFF
on	Specifies ON/OFF of block playback.						
[1]	Block playback ON						
[0]	Block playback OFF						
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR On cameras without monitor output, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code. When the camera is in the LIVE mode, a PCC_ERROR_CAMERAMODE is returned.						
Remarks	<p>This function is effective on the monitor output only. It can only be used when the camera is in the PLAY mode. See [OnLive] function.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p> <p>Note: This function has been combined with [ExtraCommand] functions. It is only included here to match the old version of SDK manual.</p> <p>Note 2: This function is not supported by cameras that are newer than the FASTCAM ultima 1024.</p>						

VisualBasic6.0**GetPlayRate (ByRef rate As Long) As Long****VisualBasic.NET****GetPlayRate (ByRef rate As Integer) As Integer**

Functions	Gets a playback rate on monitor output.
Arguments	rate Specifies 32bit type parameter that receives the display rate of playback (FPS)
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR On cameras without monitor output, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code. When the camera is in the LIVE mode, a PCC_ERROR_CAMERAMODE is returned.
Remarks	This function is effective on the monitor output only. See "On Error Codes" for details of error codes On Error Codes

Note: This function has been combined with [ExtraCommand] functions. It is only included here to match the old version of SDK manual.

Note 2: This function is not supported by cameras that are newer than the FASTCAM ultima 1024.

VisualBasic6.0

**GetBlockArea (ByRef start As Long, ByRef end As Long)
As Long**

VisualBasic.NET

**GetBlockArea (ByRef start As Integer,
ByRef end As Integer) As Integer**

Functions	Gets a range (block) of block playback on monitor output.
Arguments	<div><div>start</div><div>Specifies 32bit type parameter that receives the first frame of the range (block) .</div></div> <div><div>end</div><div>Specifies 32bit type parameter that receives the last frame of the range (block).</div></div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR On cameras without monitor output, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code. When the camera is in the LIVE mode, a PCC_ERROR_CAMERAMODE is returned.
Remarks	This function is effective on the monitor output only. See "Details of Frame Parameters" for details of frame numbers. See "On Error Codes" for details of error codes On Error Codes

Note: This function has been combined with [ExtraCommand] functions. It is only included here to match the old version of SDK manual.

Note 2: This function is not supported by cameras that are newer than the FASTCAM ultima 1024.

VisualBasic6.0**GetBlockMode (ByRef On As Long) As Long****VisualBasic.NET****GetBlockMode (ByRef On As Integer) As Integer**

Functions	Gets ON/OFF status of monitor output in Block playback.
Arguments	<div><div>on</div><div>Specifies 32bit type parameter that receives the status of Block playback.</div><div>[1] Block playback ON</div><div>[0] Block playback OFF</div></div>
Returned values	<div>Returns an error code. When normal, returns a PCC_ERROR_NOERROR</div> <div>On cameras without monitor output, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code.</div> <div>When the camera is in the LIVE mode, a PCC_ERROR_CAMERAMODE is returned.</div>
Remarks	<div>This function is effective on the monitor output only.</div> <div>See "On Error Codes" for details of error codes On Error Codes</div> <div>Note: This function has been combined with [ExtraCommand] functions. It is only included here to match the old version of SDK manual.</div> <div>Note 2: This function is not supported by cameras that are newer than the FASTCAM ultima 1024.</div>

VisualBasic6.0

**TransferFrame (ByVal frame As Long, ByRef buff As Byte)
As Long**

VisualBasic.NET

**TransferFrame (ByVal frame As Integer, ByRef buff As Byte)
As Integer**

Functions	Gets live and memory image data from the camera.
Arguments	<div><div>frame</div><div>Specifies the frame number to get recorded image. Specifies -1 to get LIVE image.</div></div> <div><div>buff</div><div>Specifies address of the BYTE type memory array that stores the image data.</div></div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>Stored image data has its origin at the upper left corner. Color image data is interleaved RGBRGBRGB, etc.</p> <p>See [Frame Parameters] for details of frame numbers.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0

**TransferRawBayer (ByVal frame As Long, ByRef buff As Byte)
As Long**

VisualBasic.NET

**TransferRawBayer (ByVal frame As Integer, ByRef buff As Byte)
As Integer**

unctions Gets Bayer image data from the camera.

Arguments **frame**

Specifies a frame number.

buff

Specifies the address of a BYTE-type memory array that stores Bayer image data.

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR.

Remarks See "On Error Codes" for details of error codes [On Error Codes](#)

VisualBasic6.0

**Transfer16BitFrame (ByVal frame As Long, ByRef buff As Byte)
As Long**

VisualBasic.NET

**Transfer16BitFrame (ByVal frame As Integer,
ByRef buff As Byte) As Integer**

Functions	Gets 16-bit recorded image data from the camera.
Arguments	<div><div>frame</div><div>Specifies a frame number.</div><div>buff</div><div>Specifies the address of a WORD-type memory array that stores image data.</div></div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	See "On Error Codes" for details of error codes On Error Codes

VisualBasic6.0

**Transfer16BitRawBayer (ByVal frame As Long,
ByRef buff As Byte) As Long**

VisualBasic.NET

**Transfer16BitRawBayer (ByVal frame As Integer,
ByRef buff As Byte) As Integer**

Functions	Gets 16-bit Bayer image data from the camera.
Arguments	<div><div>frame</div><div>Specifies a frame number.</div><div>buff</div><div>Specifies the address of a WORD-type memory array that stores Bayer image data.</div></div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	See "On Error Codes" for details of error codes On Error Codes

VisualBasic6.0

**TransferMCDL (ByVal frame As Long, ByRef buff As Byte)
As Long**

VisualBasic.NET

**TransferMCDL (ByVal frame As Integer, ByRef buff As Byte)
As Integer**

Functions	Gets MCDL data from the camera.
Arguments	<p>frame</p> <p>Specifies a frame number that receives the MCDL data.</p> <p>buff</p> <p>Specifies the address of a BYTE type memory array that stores the MCDL data. The size of memory array is determined by a SIZE_PF_MCDL_DATA.</p>
Returned values	<p>Returns an error code. When normal, returns a PCC_ERROR_NOERROR.</p> <p>On a camera without MCDL function or a camera with its MCDL function set off, a PCC_ERROR_NO_MCDL is returned as an error code.</p>
Remarks	<p>See [On Frame Parameters] for details of frame numbers.</p> <p>See the hardware manual of each camera for details of MCDL.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p> <p>Note: This function has been combined with [ExtraCommand] functions. It is only included here to match the old version of SDK manual.</p>

VisualBasic6.0

TransferIRIG (ByVal frame As Long, ByRef buff As Byte)
As Long

VisualBasic.NET

TransferIRIG (ByVal frame As Integer, ByRef buff As Byte)
As Integer

unctions	Gets IRIG data from the camera.
Arguments	<p>frame Specifies a frame number that receives the IRIG data.</p> <p>buff Specifies the address of a BYTE type memory array that stores the IRIG data. The size of memory array is determined by a SIZE_PF_MCDL_IRIG.</p>
Returned values	<p>Returns an error code. When normal, returns a PCC_ERROR_NOERROR.</p> <p>On camera without IRIG function or a camera with its IRIG function set off, a PCC_ERROR_NO_IRIG is returned as an error code.</p>
Remarks	<p>See [On Frame Parameters] for details of frame numbers.</p> <p>See the hardware manual of each camera for details of IRIG.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p> <p>Note: This function has been combined with [ExtraCommand] functions. It is only included here to match the old version of SDK manual.</p>

VisualBasic6.0

GetTimeCodeFromFrame

(ByRef time_code As IRIGLIB_TIMECODE,
ByVal frame As Long) As Long

VisualBasic.NET

GetTimeCodeFromFrame

(ByRef timecode As IRIGLIB_TIMECODE,
ByVal frame As Integer) As Integer

Functions	Gets IRIG timecode from a frame number.
Arguments	<p>time_code Specifies IRIGLIB_TIMECODE structure that receives IRIG timecode.</p> <p>frame Specifies a frame number that receives IRIG data.</p>
Returned values	<p>Returns an error code. When normal, returns a PCC_ERROR_NOERROR.</p> <p>On camera without IRIG function or a camera with its IRIG function set off, a PCC_ERROR_NO_IRIG is returned as an error code.</p>
Remarks	<p>See [On Frame Parameters] for details of frame numbers.</p> <p>See the hardware manual of each camera for details of IRIG.</p> <p>See [IRIGLIB_TIMECODE Structure] for the details of IRIGLIB_TIMECODE structure.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0
ShadingCompensation () As Long

VisualBasic.NET
ShadingCompensation () As Integer

Functions	Corrects shading of image sensor.
Arguments	None
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR. On cameras without shading correction, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code.
Remarks	Shading correction must be done with the lens completely covered to shut out incomin light. See the hardware manual of each camera for details of operation. See "On Error Codes" for details of error codes On Error Codes Note: This function has been combined with [ExtraCommand] functions. It is only included here to match the old version of SDK manual.

VisualBasic6.0

**GetDateOfRecording (ByRef year As Long,
ByRef month As Long, ByRef day As Long) As Long**

VisualBasic.NET

**GetDateOfRecording (ByRef year As Integer,
ByRef month As Integer, ByRef day As Integer) As Integer**

Functions Gets the date of recording.

Arguments

year	Specifies 32bit type parameter that shows the year (last two digits).
month	Specifies 32bit type parameter that shows the month.
day	Specifies 32bit type parameter that shows the date.

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
On cameras without shading correction, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code.

Remarks

VisualBasic6.0

**GetTimeOfRecording (ByRef hour As Long,
ByRef minute As Long, ByRef second As Long) As Long**

VisualBasic.NET

**GetTimeOfRecording (ByRef hour As Integer,
ByRef minute As Integer, ByRef second As Integer) As Integer**

Functions	Gets the time of recording.
Arguments	<div><div>hour</div><div>Specifies 32bit type parameter that receives the hours.</div></div> <div><div>minute</div><div>Specifies 32bit type parameter that receives the minutes.</div></div> <div><div>second</div><div>Specifies 32bit type parameter that receives the seconds.</div></div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR. On cameras without shading correction, a PCC_ERROR_FUNCTION_DISABLE is returned as an error code.
Remarks	

VisualBasic6.0

**GetIICycleRange (ByRef dwMin As Long,
ByRef dwMax As Long)**

VisualBasic.NET

**GetIICycleRange (ByRef dwMin As Integer,
ByRef dwMax As Integer)**

Functions	Gets the CYCLE value of Image Intensifier(I.I.).
Arguments	<div><div>dwMin</div><div>Specifies 32bit type parameter that receives the Minimum CYCLE value.</div></div> <div><div>dwMax</div><div>Specifies 32bit type parameter that receives the Maximum CYCLE value.</div></div>
Returned values	In recording operations using the image intensifier, you must call this function whenever you have change parameters such as CYCLE, WIDTH, TIMES and DELAY.
Remarks	

VisualBasic6.0

**GetIIWidthRange (ByRef dwMin As Long,
ByRef dwMax As Long)**

VisualBasic.NET

**GetIIWidthRange (ByRef dwMin As Integer,
ByRef dwMax As Integer)**

Functions	Gets the WIDTH value of Image Intensifier(I.I.).
Arguments	<div><div>dwMin</div><div>Specifies 32bit type parameter that receives the Minimum WIDTH value.</div></div> <div><div>dwMax</div><div>Specifies 32bit type parameter that receives the Maximum WIDTH value.</div></div>
Returned values	In recording operations using the image intensifier, you must call this function whenever you have changee parameters such as CYCLE, WIDTH, TIMES and DELAY.
Remarks	

VisualBasic6.0

**GetIITimesRange (ByRef dwMin As Long,
ByRef dwMax As Long)**

VisualBasic.NET

**GetIITimesRange (ByRef dwMin As Integer,
ByRef dwMax As Integer)**

Functions	Gets the TIMES value of Image Intensifier(I.I.).
Arguments	<div><div>dwMin</div><div>Specifies 32bit type parameter that receives the Minimum TIMES value.</div></div> <div><div>dwMax</div><div>Specifies 32bit type parameter that receives the Maximum TIMES value.</div></div>
Returned values	In recording operations using the image intensifier, you must call this function whenever you have changee parameters such as CYCLE, WIDTH, TIMES and DELAY.
Remarks	

VisualBasic6.0

**GetIIDelayRange (ByRef dwMin As Long,
ByRef dwMax As Long)**

VisualBasic.NET

**GetIIDelayRange (ByRef dwMin As Integer,
ByRef dwMax As Integer)**

Functions	Gets the DELAY value of Image Intensifier(I.I.).
Arguments	<div><div>dwMin</div><div>Specifies 32bit type parameter that receives the Minimum DELAY value.</div></div> <div><div>dwMax</div><div>Specifies 32bit type parameter that receives the Maximum DELAY value.</div></div>
Returned values	In recording operations using the image intensifier, you must call this function whenever you have changee parameters such as CYCLE, WIDTH, TIMES and DELAY.
Remarks	

VisualBasic6.0

**GetVariableRateList(ByRef list As Long, ByVal listSize As Long)
As Long**

VisualBasic.NET

**GetVariableRateList(ByRef list As Integer,
ByVal listSize As Integer) As Integer**

Functions	Gets the frame rate list, which can be used by variable setting
Arguments	<div><div>list</div><div>Specifies the address of the integer memory arrangement that receives a frame rate list</div></div> <div><div>listsize</div><div>Specifies the size of the list memory array. The default size is LIST_MAX_NUMBER.</div></div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>The size of list memory array is normally determined by a LIST_MAX_NUMBER.</p> <p>See "On Error Codes" for details of error codes On Error Codes</p>

VisualBasic6.0

**GetVariableMaxResolution(ByVal nFrame As Long,
ByRef nHeight As Long, ByRef nWidth As Long) As Long**

VisualBasic.NET

**GetVariableMaxResolution(ByVal nFrame As Integer,
ByRef nHeight As Integer, ByRef nWidth As Integer) As Integer**

Functions	Gets the Maximum Square Resolution at specification frame rate by variable setting
Arguments	<div><div>nFrame</div><div>Specifies frame rate</div></div> <div><div>nWidth</div><div>Integer parameter that receives the width of the Maximum Square Resolution</div></div> <div><div>nHeight</div><div>Integer parameter that receives the height of the Maximum Square Resolution</div></div>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	Maximum square resolution is noticed about the point that is not necessarily the resolution at the time of the maximum area.

VisualBasic6.0

**GetVariableMaxWidth(ByVal nFrame As Long,
ByVal nHeight As Long, ByRef nWidth As Long) As Long**

VisualBasic.NET

**GetVariableMaxWidth(ByVal nFrame As Integer,
ByVal nHeight As Integer, ByRef nWidth As Integer)
As Integer**

Functions Gets the maximum width at specification frame rate and height by variable setting

Arguments **nFrame**
Specifies frame rate

nHeight
Specifies height of image

nWidth
Integer parameter that image maximum width at the time of the above-mentioned specification value is received.

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR.

Remarks

VisualBasic6.0

**GetVariableMaxHeight(ByVal nFrame As Long,
ByVal nWidth As Long, ByRef nHeight As Long) As Long**

VisualBasic.NET

**GetVariableMaxHeight(ByVal nFrame As Integer,
ByVal nWidth As Integer, ByRef nHeight As Integer)
As Integer**

Functions Gets the maximum height at specification frame rate and width by variable setting

Arguments **nFrame**
Specifies frame rate

nWidth
Specifies width of image

nHeight
Integer parameter that image maximum height at the time of the above-mentioned specification value is received.

Returned values Returns an error code. When normal, returns a PCC_ERROR_NOERROR.

Remarks

VisualBasic6.0

**GetVariableMaxFrameRate(ByVal nWidth As Long,
ByVal nHeight As Long, ByRef nRate As Long) As Long**

VisualBasic.NET

**GetVariableMaxFrameRate(ByVal nWidth As Integer,
ByVal nHeight As Integer, ByRef nRate As Integer)
As Integer**

Functions	Gets the maximum frame rate at specification resolution by variable setting
Arguments	
	nWidth Specifies width of image
	nHeight Specifies height of image
	nRate Integer parameter that maximum frame rate at the time of the above-mentioned specification value is received
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	

VisualBasic6.0

**SetGEtherConfig (ByVal PacketSize As Long,
ByVal SendPort As Long, ByVal ReceivePort As Long,
ByVal ConnectMode As Long)
As Long**

VisualBasic.NET

**SetGEtherConfig (ByVal PacketSize As Integer,
ByVal SendPort As Integer, ByVal ReceivePort As Integer,
ByVal ConnectMode As Integer)
As Integer**

Functions	Sets conditions for connection of 1000Base-T Ethernet I/F.
Arguments	<p>PacketSize Packet size (default 0)</p> <p>SendPort Send port (default 0)</p> <p>ReceivePort Receive port (default 0)</p> <p>ConnectMode Connection mode</p> <p>GETHER_CONNECT_NORMAL Normal connection (default)</p> <p>GETHER_CONNECT_LOWSPEED Low-speed connection</p>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	<p>A setting is applied when CCameraControl class is nullified</p> <p>Note 1: For packet size, use a number 58 + (multiple of 8).</p> <p>Note 2: When "0" (zero) is specified, the packet size is automatically readjusted.</p> <p>If and when communication cannot be established at the specified size, the system operates at the default value (1458).</p> <p>For normal use, select GETHER_CONNECT_NORMAL in ConnectMode. Take note that, when GETHER_CONNECT_LOWSPEED is used, PacketSize is forced into 722.</p>

VisualBasic6.0

GetGEtherConfig (ByRef PacketSize As Long,
ByRef SendPort As Long, ByRef ReceivePort As Long,
ByRef ConnectMode As Long)
As Long

VisualBasic.NET

GetGEtherConfig (ByRef PacketSize As Integer,
ByRef SendPort As Integer, ByRef ReceivePort As Integer,
ByRef ConnectMode As Integer)
As Integer

Functions	Gets setting conditions for connection of 1000Base-T Ethernet I/F
Arguments	<p>PacketSize Packet size (default 0)</p> <p>SendPort Send port (default 0)</p> <p>ReceivePort Receive port (default 0)</p> <p>ConnectMode Connection mode GETHER_CONNECT_NORMAL Normal connection (default) GETHER_CONNECT_LOWSPEED Low-speed connection</p>
Returned values	Returns an error code. When normal, returns a PCC_ERROR_NOERROR.
Remarks	When the packet size is "0" (zero), the system operates as it is automatically readjusted.

EXTRA_SET_ENABLE_RESTRICTION_TIME

unctions	Sets recording time restriction function status.	
Argument 2	Valid/Invalid	
	[1]	Valid (default)
	[0]	Invalid
Argument 3	None	
Argument 4	None	
Remarks	This function switches the recording time restriction between Valid and Invalid on FASTCAM-PCI/R2 cameras. Normally, use the function in the Valid position.	

EXTRA_GET_ENABLE_RESTRICTION_TIME

Functions	Gets recording time restriction function status.	
Argument 2	Valid/Invalid	
	[1]	Valid
	[0]	Invalid
Argument 3	None	
Argument 4	None	
Remarks	This function switches the recording time restriction between Valid and Invalid on FASTCAM-PCI/R2 cameras. Normally, use the function in the Valid position.	

EXTRA_GET_CAMERA_ID

Functions	Gets a camera ID.
Argument 2	32bit type address that contains a camera ID
Argument 3	None
Argument 4	None
Remarks	

EXTRA_GET_ENABLE_RESET_TRIGGER

Functions	Gets a reset trigger status.
Argument 2	Pointer that contains Valid/Invalid. [1] Valid [0] Invalid
Argument 3	None
Argument 4	None
Remarks	This function is used on FASTCAM-X1280PCI and FASTCAM-PCI R2 cameras.

EXTRA_SET_ENABLE_RESET_TRIGGER

Functions	Sets a reset trigger status.	
Argument 2	Valid or Invalid.	
	[1]	Valid
	[0]	Invalid
Argument 3	None	
Argument 4	None	
Remarks	This function is used on FASTCAM-X1280PCI and FASTCAM-PCI R2 cameras.	

EXTRA_GET_SENSOR_BITSHIFT

Functions	Gets grayscale setting status obtained by FASTCAM-X1280PCI camera.	
Argument 2	32bit type adress that contains grayscale.	
	0	Upper 8 bits of 10bits (default)
	1	Middle 8 bits of 10bits
	2	Lower 8 bits of 10bits
Argument 3	None	
Argument 4	None	
Remarks	This function gets the current grayscale setting status obtained by a FASTCAM-X1280PCI camera.	

EXTRA_SET_SENSOR_BITSHIFT

Functions	Sets grayscale setting change obtained by FASTCAM-X 1280PCI camera.						
Argument 2	Grayscale. Selects one of the following three: <table><tr><td>0</td><td>Upper 8 bits of 10bits (default)</td></tr><tr><td>1</td><td>Middle 8 bits of 10bits</td></tr><tr><td>2</td><td>Lower 8 bits of 10bits</td></tr></table>	0	Upper 8 bits of 10bits (default)	1	Middle 8 bits of 10bits	2	Lower 8 bits of 10bits
0	Upper 8 bits of 10bits (default)						
1	Middle 8 bits of 10bits						
2	Lower 8 bits of 10bits						
Argument 3	None						
Argument 4	None						
Remarks	This function selects one of three possible settings of grayscale values and sets in a FASTCAM-X1280PCI camera.						

EXTRA_SET_LIVE_RESOLUTION

Functions	Changes resolution setting in the Live mode.	
Argument 2	Resolution settings for live mode	
	LIVE_RESO_FULL	Full resolution (default)
	LIVE_RESO_HALF	1/2
	LIVE_RESO_QQRTER	1/4
	LIVE_RESO_DQARTER	1/8
Argument 3	None	
Argument 4	None	
Remarks	Setting live resolution to any one other than the full resolution may result in a faster transfer rate of image data, but the image quality is sacrificed.	

EXTRA_GET_LIVE_RESOLUTION

Functions	Gets resolution setting in the Live mode.	
Argument 2	32bit type adress that contains resolution settings for the live mode	
	LIVE_RESO_FULL	Full resolution (default)
	LIVE_RESO_HALF	1/2
	LIVE_RESO_QQRTER	1/4
	LIVE_RESO_DQARTER	1/8
Argument 3	None	
Argument 4	None	
Remarks	Setting live resolution to any one other than the full resolution may result in a faster transfer rate of image data, but the image quality is sacrificed.	

EXTRA_SET_LUT_MODE

Funtions	Sets the LUT status.	
Argument 2	Mode setting values	
	LUT_DEFAULT1	Default 1
	LUT_DEFAULT2	Default 2
	LUT_DEFAULT3	Default 3
	LUT_DEFAULT4	Default 4
	LUT_DEFAULT5	Default 5
	LUT_USER1	User-set number value
Argument 3	None	
Argument 4	None	
Remarks	The LUT mentioned in the above is a parameter within the camera and has nothing to do with the LUT shown in the ImageCtrl	

EXTRA_GET_LUT_MODE

Funtions Gets the LUT status.

Argument 2 32bit type adress that contain mode setting values

LUT_DEFAULT1	Default 1
LUT_DEFAULT2	Default 2
LUT_DEFAULT3	Default 3
LUT_DEFAULT4	Default 4
LUT_DEFAULT5	Default 5
LUT_USER1	User-set number value

Argument 3 None

Argument 4 None

Remarks The LUT mentioned in the above is a parameter within the camera and has nothing to do with the LUT shown in the CImageCtrl.

EXTRA_SET_USER_LUT_PARAMS

Argument 2	Mode setting values LUT_USER1 Value set by user
Argument 3	adress for PCAM_LUT_PARAMS structure.
Argument 4	None
Remarks	To make the changes effective on cameras, it is necessary to perform EXTRA_UPLOAD_LUT_DATA separately.

Note 1. This command is not currently supported.

EXTRA_GET_USER_LUT_PARAMS

Functions	Gets LUT parameters.		
Argument 2	Mode setting values		
	LUT_USER1		Value set by user
Argument 3	adress for PCAM_LUT_PARAMS structure structure that contains obtained values.		
Argument 4	None		
Remarks	Note 1. This command is not currently supported.		

EXTRA_UPLOAD_LUT_DATA

Functions	Uploads LUT to a camera.
Argument 2	None
Argument 3	None
Argument 4	None
Remarks	this function uploads the currently-set LUT to a camera. The camera must be set in LUT_USER1 mode.

EXTRA_SET_CHROMA_MODE

Functions	Changes chroma settings.
Argument 2	Chroma setting values: CHROMA_NORMAL (Default) CHROMA_UP01 CHROMA_UP02 CHROMA_UP03 CHROMA_UP04 CHROMA_UP05 CHROMA_UP06 CHROMA_UP07 CHROMA_UP08
Argument 3	None
Argument 4	None
Remarks	

EXTRA_GET_CHROMA_MODE

Functions	Gets chroma settings.
Argument 2	32bit type adress that receives chroma settings: CHROMA_NORMAL (Default) CHROMA_UP01 CHROMA_UP02 CHROMA_UP03 CHROMA_UP04 CHROMA_UP05 CHROMA_UP06 CHROMA_UP07 CHROMA_UP08
Argument 3	None
Argument 4	None
Remarks	

EXTRA_SET_DS_SHUTTER_MODE

Functions	Sets dynamic range expansion mode.	
Argument 2	Dyanamic range expansion mode setting values:	
	DS_SHUTTER_OFF	OFF (Default)
	DS_SHUTTER_MODE1	Exposure control Small
	DS_SHUTTER_MODE2	Exposure control Medium
	DS_SHUTTER_MODE3	Exposure control Large
Argument 3	None	
Argument 4	None	
Remarks	This function sets a dynamic range expansion mode.	

EXTRA_GET_DS_SHUTTER_MODE

Functions	Gets dynamic range expansion mode setting.	
Argument 2	32bit type adress that contains dyanamic range expansion setting values:	
	DS_SHUTTER_OFF	OFF (Default)
	DS_SHUTTER_MODE1	Exposure control Small
	DS_SHUTTER_MODE2	Exposure control Medium
	DS_SHUTTER_MODE3	Exposure control Large
Argument 3	None	
Argument 4	None	
Remarks	This function sets a dynamic range expansion mode.	

EXTRA_SET_INTERFACE_INFO

Functions	Changes interface settings.		
Argument 2	Interface parameters:		
	IEEE1394		
		INTERFACE_IEEE1394	Fixed value
	100Base-TX		
		INTERFACE_ETHER	Fixed value
	1000Base-T		
		INTERFACE_G_ETHER	Fixed value
Argument 3	Interface parameters		
	IEEE1394		
		400	400Mbps operaion
		200	200Mbps operation
		100	100Mbps operation
	100Base-TX		
		(Time for time-out):	
		IP address search time (milliseconds)	
	1000Base-T		
		Base address for automatic IP search	
		IP address is set with a DWORD-type number.	
		Example: For192.168.12.0, the IP address is 0xC0A80A00	

Argument 4	Interface paremeters
	<p>IEEE1394</p> <p>None</p> <p>100Base-TX / 1000Base-T</p> <p>(IP address anutomatic search number and IP address list)</p> <p>Top of the list,</p> <p>Upper 16 bits automatic search number</p> <p> Set a "-1" for Using IP list</p> <p>Lower 16 bits IP address number</p> <p> Set a "-1" for automatic search.</p> <p>Set an IP address by 32bit type</p> <p>Example: Set "0xC0A80101" for IP address 192.168.1.1.</p> <p>(IP address anutomatic search number)</p> <p>Maximum value: 64</p> <p>If a negative number is set, automatic search is turned off.</p>
Remarks	<p>Set a "0" for arguments that are not used.</p> <p>Do not set an interface that is different from the type of the camera.</p> <p>When you do not use automatic search, be sure to set interface before initialization.</p>

EXTRA_GET_INTERFACE_INFO

Functions	Gets interface settings.		
Argument 2	32bit type parameters that receives types of interface:		
	IEEE1394	INTERFACE_IEEE1394	Fixed value
	100Base-TX	INTERFACE_ETHER	Fixed value
	1000Base-T	INTERFACE_G_ETHER	Fixed value
	Optical I/F	INTERFACE_OPTICAL	Fixed value
Argument 3	32bit type parameters that receive interface parameters.		
	IEEE1394		
	400	400Mbps operation	
	200	200Mbps operation	
	100	100Mbps operation	
	100Base-TX		
		Time-out for automatic IP search	
		IP address search time (msec)	
	1000Base-T		
		Base address for automatic IP search	
		IP address is set with a DWORD-type number.	
		Example: For 192.168.12.0, the IP address is 0xC0A80A00	
	Optical I/F		
		None	

Argument 4	<p>32bit type parameters that stores interface paremeters</p> <p>IEEE1394</p> <p>None</p> <p>100Base-TX / 1000Base-T</p> <p>(IP address anutomatic search number and IP address list)</p> <p>Top of the list,</p> <p>Upper 16 bits automatic search number</p> <p> Set a "-1" for Using IP list</p> <p>Lower 16 bits IP address number</p> <p> Set a "-1" for automatic search.</p> <p>Set an IP address by 32bit type</p> <p>Example: Set "0xC0A80101" for IP address 192.168.1.1.</p> <p>(IP address anutomatic search number)</p> <p>Maximum value: 64</p> <p>If a negative number is set, automatic search is turned off.</p> <p>Optical I/F</p> <p>None</p>
Remarks	No parameters are needed.for inteface types that are not used,

EXTRA_SET_SHADING_MODE

Functions	Sets calibration mode.	
Argument 2	Calibration mode setting values:	
	SHADING_OFF	Will not use Calibration results.
	SHADING_ON1	Will use Calibration results.
	SHADING_SAVE	Will store Calibration results.
	SHADING_LOAD	Will read Calibration results.
Argument 3	None	
Argument 4	None	
Remarks	To check the status in SAVE/LOAD, the function [EXTRA_GET_SHADING_LOCK] is used.	

EXTRA_GET_SHADING_MODE

Functions	Gets calibration mode.	
Argument 2	32bit type parameters that receive calibration mode:	
	SHADING_OFF	Will not use Calibration results.
	SHADING_ON1	Will use Calibration results.
	SHADING_SAVE	Will store Calibration results.
	SHADING_LOAD	Will read Calibration results.
Argument 3	None	
Argument 4	None	
Remarks	To check the status in SAVE/LOAD, the function [EXTRA_GET_SHADING_LOCK] is used.	

EXTRA_GET_SHADING_LOCK

Functions	Gets an execution status of calibration function.
Argument 2	32bit type parameters that receive calibration execution status. [1] Locked [0] Not Locked
Argument 3	None
Argument 4	None
Remarks	After storing or reading a calibration, wait until Argument 2 turns to "Not Locked" before starting any other operation.

EXTRA_SET_IRIG_OFFSET

Functions	Sets IRIG offset value.
Argument 2	Offset value -999999 – 999999 (microsecond)
Argument 3	None
Argument 4	None
Remarks	

EXTRA_GET_IRIG_OFFSET

Functions	Gets IRIG offset value.
Argument 2	32bit type parameters that receive calibration execution offset value.
Argument 3	None
Argument 4	None
Remarks	.

EXTRA_SET_HARD_PARTITION

Functions	Sets hard partition mode.
Argument 2	Hard partition mode setting status. [1] Hard partition mode [0] Not hard partition mode (default)
Argument 3	None
Argument 4	None
Remarks	

EXTRA_GET_HARD_PARTITION

Functions	Gets an execution status of hard partition mode.
Argument 2	32bit type parameters that receive hard partition mode status. [1] Hard partition mode [0] Not hard partition mode(default)
Argument 3	None
Argument 4	None
Remarks	

EXTRA_SET_SHUTTER_MODE

Functions	Sets shutter mode.
Argument 2	Shutter mode setting status. [1] Locked shutter speed [0] Not Locked(default)
Argument 3	None
Argument 4	None
Remarks	

EXTRA_GET_SHUTTER_MODE

Functions	Gets an execution status of shutter mode.	
Argument 2	32bit type parameters that receive shutter mode status.	
	[1]	Locked shutter speed
	[0]	Not Locked(default)
Argument 3	None	
Argument 4	None	
Remarks		

EXTRA_SET_II_POWER

Functions	Sets power status of Image Intensifier.
Argument 2	"Image Intensifier" 's setting power status. II_POWER_OFF Power off II_POWER_ON Power on II_POWER_ON_LOAD Power on (Loading configuration on Camera)
Argument 3	None
Argument 4	None
Remarks	

EXTRA_GET_II_POWER

Functions	Gets an execution status of "Image Intensifier" 's power status.	
Argument 2	32bit type parameters that receive "Image Intensifier" 's power status.	
	II_POWER_OFF	Power off
	II_POWER_ON	Power on
Argument 3	None	
Argument 4	None	
Remarks		

EXTRA_SET_II_GAIN

Functions	Sets gain value of Image Intensifier.
Argument 2	Gain value of Image Intensifier. 1 - 50 (0.1V – 5.0V)
Argument 3	None
Argument 4	None
Remarks	

EXTRA_GET_II_GAIN

Functions	Gets gain value of Image Intensifier.
Argument 2	32bit type parameters that receive gain value of Image Intensifier. 1 - 50 (0.1V – 5.0V)
Argument 3	None
Argument 4	None
Remarks	

EXTRA_SET_II_GATE_SELECT

Functions Sets an execution status of "Image Intensifier" 's gate mode.

Argument 2 Gate mode of Image Intensifier.
 II_GATE_SELECT_OFF
 II_GATE_SELECT_CONTINUOUS
 II_GATE_SELECT_EXTERNAL
 II_GATE_SELECT_GATING
 II_GATE_SELECT_DELAY

Argument 3 None

Argument 4 None

Remarks

EXTRA_GET_II_GATE_SELECT

Functions	Gets gate mode of Image Intensifier.
Argument 2	32bit type parameters that receive gate mode of Image Intensifier. II_GATE_SELECT_OFF II_GATE_SELECT_CONTINUOUS II_GATE_SELECT_EXTERNAL II_GATE_SELECT_GATING II_GATE_SELECT_DELAY
Argument 3	None
Argument 4	None
Remarks	

EXTRA_SET_II_GATE_CYCLE

Functions	Sets CYCLE value of Image Intensifier.
Argument 2	CYCLE value of Image Intensifier.
Argument 3	None
Argument 4	None
Remarks	

EXTRA_GET_II_GATE_CYCLE

Functions	Gets CYCLE value of Image Intensifier.
Argument 2	32bit type parameters that receive CYCLE value of Image Intensifier.
Argument 3	None
Argument 4	None
Remarks	

EXTRA_SET_II_GATE_WIDTH

Functions Sets WIDTH value of Image Intensifier.

Argument 2 WIDTH value of Image Intensifier.

Argument 3 None

Argument 4 None

Remarks

EXTRA_GET_IL_GATE_WIDTH

Functions	Gets WIDTH value of Image Intensifier.
Argument 2	32bit type parameters that receive WIDTH value of Image Intensifier.
Argument 3	None
Argument 4	None
Remarks	

EXTRA_SET_II_GATE_TIMES

Functions Sets TIMES value of Image Intensifier.

Argument 2 TIMES value of Image Intensifier.

Argument 3 None

Argument 4 None

Remarks

EXTRA_GET_IL_GATE_TIMES

Functions	Gets TIMES value of Image Intensifier.
Argument 2	32bit type parameters that receive TIMES value of Image Intensifier.
Argument 3	None
Argument 4	None
Remarks	

EXTRA_SET_II_GATE_DELAY

Functions	Sets DELAY value of Image Intensifier.
Argument 2	DELAY value of Image Intensifier (nsec).
Argument 3	None
Argument 4	None
Remarks	

EXTRA_GET_II_GATE_DELAY

Functions	Gets DELAY value of Image Intensifier.
Argument 2	32bit type parameters that receive DELAY value of Image Intensifier.
Argument 3	None
Argument 4	None
Remarks	

EXTRA_GET_IL_GAIN_LIMIT

Functions	Gets Gain's limit value of Image Intensifier.
Argument 2	32bit type parameters that receive Gain value of Image Intensifier.
Argument 3	None
Argument 4	None
Remarks	

EXTRA_GET_EVENT_FRAME

Functions	Gets event frame number.
Argument 2	Event number (0 – 9)
Argument 3	Parameter that receive event frame number 0 ... No event
Argument 4	Parameter that receive random trigger times
Remarks	

EXTRA_SET_VARIABLE_LOAD

Functions	Set the state of a camera that is made into the setting value of a specification channel by variable setting.
Argument 2	The channel to be used (0-19)
Argument 3	None
Argument 4	None
Remarks	

EXTRA_SET_VARIABLE_ERASE

Functions	The setting value of a specification channel is deleted by variable setting.
Argument 2	The channel to delete (0-19)
Argument 3	None
Argument 4	None
Remarks	

EXTRA_GET_IL_GATE_WIDTH_LIMIT

Functions	Gets minimum WIDTH value of Image Intensifier.
Argument 2	Minimum WIDTH value of Image Intensifier (nsec).
Argument 3	None
Argument 4	None
Remarks	

EXTRA_SET_EDGE_ENHANCEMENT

Functions	A state setup of EDGE-ENHANCEMENT.		
Argument 2	mode	<div>EDGE_ENHANCEMENT_OFF OFF</div> <div>EDGE_ENHANCEMENT_MODE1 MODE1</div> <div>EDGE_ENHANCEMENT_MODE2 MODE2</div> <div>EDGE_ENHANCEMENT_MODE3 MODE3</div>	
Argument 3	None		
Argument 4	None		
Remarks	It is not compatible with EXTRA_SET_MONITOROUT_EDGEENHANCE.		

EXTRA_GET_EDGE_ENHANCEMENT

Functions	Get the state of EDGE-ENHANCEMENT
Argument 2	DWORD-type pointer that stores the mode value of EDGE-ENHANCEMENT EDGE_ENHANCEMENT_OFF OFF EDGE_ENHANCEMENT_MODE1 MODE1 EDGE_ENHANCEMENT_MODE2 MODE2 EDGE_ENHANCEMENT_MODE3 MODE3
Argument 3	None
Argument 4	None
Remarks	It is not compatible with EXTRA_GET_MONITOROUT_EDGEENHANCE.

EXTRA_SET_AUTO_EXPOSURE

Functions	Sets status of Auto Exposure function.
Argument 2	Auto Exposure setting values [1] Auto Exposure function On [0] Auto Exposure function OFF
Argument 3	None
Argument 4	None
Remarks	When the Auto Exposure function is on, the shutter speed that can be gotten is indefinite.

EXTRA_GET_AUTO_EXPOSURE

Functions	Gets status of the Auto Exposure function.	
Argument 2	A 32bit-type address that stores status of the Auto Exposure function.	
	[1]	Auto Exposure function On
	[0]	Auto Exposure function Off
Argument 3	None	
Argument 4	None	
Remarks	When the Auto Exposure function is on, the shutter speed that can be gotten is indefinite.	

EXTRA_SET_AUTO_EXPOSURE_MAXSHUTTER

Functions	Sets the maximum exposure period for the Auto Exposure function.
Argument 2	Maximum exposure period Specifies the maximum exposure period (1/shutter speed) that can be set.
Argument 3	None
Argument 4	None
Remarks	Values that can be set can be gotten from the member function [GetShutterSpeedAEList].

EXTRA_GET_AUTO_EXPOSURE_MAXSHUTTER

Functions	Gets the maximum exposure period for the Auto Exposure function.
Argument 2	A DWORD-type pointer that stores status of the Auto Exposure function. Gets the maximum exposure period (1/shutter speed) that is set
Argument 3	None
Argument 4	None
Remarks	

EXTRA_SET_II_PROTECTION

Functions	Sets the level of hardware protection function for the image intensifier (I.I.).
Argument 2	Protection level value 1 to 5 Protection function level
Argument 3	None
Argument 4	None
Remarks	

EXTRA_GET_II_PROTECTION

Functions	Gets the level of hardware protection function for the image intensifier (I.I.).
Argument 2	A DWORD-type pointer that stores the protection level.
Argument 3	None
Argument 4	None
Remarks	

5. ImageData

This section discusses the overview of ImageData and specifications its functions..

Class in the following subsections:

- 5.1 [CImageData Overview](#)**
- 5.2 [CImageData Function List](#)**
- 5.3 [CImageData Function Specifications](#)**

5.1. ImageData Overview

5.1.1. About ImageData

ImageData Functions has such functions as writing still image data, storing, LUT (Look Up Table) Correction and Image Transform into Windows Format.

One still frame requires one class. After generating an instance of ImageData, you need to initialize to secure an image memory buffer using **InitImageData** function. When an **ExitImageData** function is called out, all buffers are opened.

The size of buffers is automatically readjusted by the size of image file being read.

Note: The total size of buffers secured in a class is normally larger than that of the original image file.

■ ImageData Functions

- Initialization Function
- Image operation functions
- File input/Output functions
- LUT functions

5.2. ImageData Function List

List of Member Functions of ImageData

Structure	
ImageData	ImageData Structure
Initialization functions	
InitImageData	Initialization processes
ExitImageData	Exiting processes
SetImageNumber	Set Operation picture number
GetImageNumber	Get Operation picture number
Image functions	
GetImageData	Gets image data
SetImageData	Sets image data
GetImageSize	Gets bitmap information
GetWindowsImage	Gets Windows display image
GetWindows16BitImage	Gets 16-bit Windows display image
GetBmpInfo	Gets BITMAP information
SetBmpInfo	Sets BITMAP information
File functions	
LoadFile	Reads file(Returned value:image format)
LoadRawFile	Reads RAW file
LoadRawwFile	Reads RAWW file
SaveFile	Stores File
SaveBmpFile	Stores BMP file
SaveTiffFile	Stores TIFF file
SaveTiffFileEx	Stores TIFF file
SaveJpegFile	Stores JPEG file
SavePnmFile	Stores PNM file
SaveRawFile	Stores RAW file
SaveRawwFile	Stores RAWW file
SavePngFile	Stores PNG file

LUT	
<u>InitLut</u>	Initializes LUT
<u>SetUseLut</u>	Use LUT ON/OFF
<u>GetUseLut</u>	Gets status of Use LUT ON/OFF
<u>SetLut</u>	Sets LUT
<u>SetLutR</u>	Sets LUT Red plane
<u>SetLutG</u>	Sets LUT Green plane
<u>SetLutB</u>	Sets LUT Blue plane
<u>GetLut</u>	Gets LUT
<u>GetLutR</u>	Gets LUT Red plane
<u>GetLutG</u>	Gets LUT Green plane
<u>GetLutB</u>	Gets LUT Blue plane
<u>SetLutGamma</u>	Sets LUT gamma correction
<u>SetLutContrast</u>	Sets LUT contrast correction
<u>SetLutBrightness</u>	Sets LUT brightness correction
<u>GetLutGamma</u>	Gets LUT gamma correction
<u>GetLutContrast</u>	Gets LUT contrast correction
<u>GetLutBrightness</u>	Gets LUT brightness correction

5.3. ImageData Function Specifications

This section describes specifications of member functions of the ImageData .

VisualBasic6.0

ImageData () As Long

VisualBasic.NET

ImageData () As Integer

Functions	Generates an instance of ImageData .
Arguments	None
Returned values	None
Remarks	One ImageData operates one still image.

VisualBasic6.0

InitImageData (ByVal buffer_size As Long) As Long

VisualBasic.NET

InitImageData (ByVal buffer_size As Integer) As Integer

Functions	Initializes a ImageData .
Arguments	<p>buffer_size</p> <p>Specifies image size to secure an image buffer for internal processing.</p> <p>Image size = Height x Width x Number of color channels</p> <p>The size when omitted is 0 x B4000 (w512 x h480 x p3)</p>
Returned values	A [1] is returned for successful initialization and a [0] for failure of initialization.
Remarks	One ImageData operates one still image.

VisualBasic6.0

ExitImageData () As Long

VisualBasic.NET

ExitImageData () As Integer

Functions	Exits ImageData process (releases work memory area).
Arguments	None
Returned values	A [1] is returned if successful, and a [0] if failed.
Remarks	This function is normally called from the default destructor.

VisualBasic6.0

SetImageNumber (ByVal number As Long) As Long

VisualBasic.NET

SetImageNumber (ByVal number As Integer) As Integer

Functions	Sets Operation picture number
Arguments	number The management image number in a library
Returned values	A [1] is returned when the process is success, and a [0] when in error.
Remarks	

VisualBasic6.0

GetImageNumber () As Long

VisualBasic.NET

GetImageNumber () As Integer

Functions	Gets Operation picture number
引数	none
Returned values	The management Image number in a library
Remarks	

VisualBasic6.0**GetData (ByRef image As Byte) As Long****VisualBasic.NET****GetData (ByRef image As Byte) As Integer**

Functions	Gets a buffer of image data.
Arguments	<div>image</div> <div>The BYTE type address which receives image data</div>
Returned values	<div>Returns a [0] when it is not initialized by a function InitImageData.</div> <div>It gets the uncorrected, original data even if LUT correction function is enabled.</div>
Remarks	The image data acquired by this function is a top-down image with its origin in the upper left corner. To get a Windows-standard bottom-up image, use GetWindowsImage function.

VisualBasic6.0**SetImageData (ByRef image As Byte) As Long****VisualBasic.NET****SetImageData (ByRef image As Byte) As Integer**

Functions	Sets a buffer of image data
Arguements	image The BYTE type address which passes image data to an image data buffer
Returned values	Returns a [0] when it is not initialized by a function InitImageData. It gets the uncorrected, original data even if LUT correction function is enabled.
Remarks	

VisualBasic6.0

GetImageSize (ByRef width As Long, ByRef height As Long)

VisualBasic.NET

GetImageSize (ByRef width As Integer, ByRef height As Integer)

Functions	Gets a size for image data.
Arguments	<div><div>width</div><div>Specifies 32bit type parameter that receives the image width.</div></div> <div><div>height</div><div>Specifies 32bit type parameter that receives the image height.</div></div>
Returned values	None
Remarks	When initialization has not been done with an Init function, or when there is no image data existing, both arguments - width and height - will have a "0".

VisualBasic6.0

**GetWindowsImage (ByRef image As Byte, ByRef buff As Byte,
ByRef bmpinfo As BITMAPINFO) As Long**

VisualBasic.NET

**GetWindowsImage (ByRef image As Byte, ByRef buff As Byte,
ByRef bmpinfo As BITMAPINFO) As Integer**

Functions Gets Windows-standard bottom-up image data obtained by correcting original image by the LUT.

Arguments image

The address of the BYTE type parameter which receives Windows standard image data.

buff

Specifies the address of a BYTE type parameter for top-down type image data that is to be transferred to Windows-type standard image data.

bmpinfo

Specifies the address of a BITMAPINFO structure that contains bitmap information of top-down type image data specified by "buff".

Returned values A [1] is returned when the process is success, and a [0] when in error. Returns a [0] if the system has not been initialized by an InitImageData function.
When LUT correction function is active, the data processed by the LUT correction function is obtained.

Remarks

The image data acquired by this function is a Windows-standard image with its origin in the lower left corner. To get a top-down type image, use a GetImageData function.

See [Platform SDK: Windows GDI] in the Microsoft Corporation MSDN Library for the details of BITMAPINFO structure.

VisualBasic6.0

**GetWindows16BitImage (ByRef image As Byte,
ByRef buff As Byte, ByRef bmpinfo As BITMAPINFO)
As Long**

VisualBasic.NET

**GetWindows16BitImage (ByRef image As Byte,
ByRef buff As Byte, ByRef bmpinfo As BITMAPINFO)
As Integer**

Functions	Gets Windows-standard bottom-up type image data processed in 16-bit data LUT correction.
Arguments	<p>image</p> <p>The address of the BYTE type parameter which receives Windows standard image data.</p> <p>buff</p> <p>Specifies the address of a BYTE type parameter for top-down type image data that is to be transferred to Windows-type standard image data.</p> <p>bmpinfo</p> <p>Specifies the address of a BITMAPINFO structure that contains bitmap information of top-down type image data specified by "buff".</p>
Returned values	<p>A [1] is returned when the process is success, and a [0] when in error. Returns a [0] if the system has not been initialized by an InitImageData function.</p> <p>When LUT correction function is active, the data processed by the LUT correction function is obtained.</p>
Remarks	<p>The image data acquired by this function is a Windows-standard image with its origin in the lower left corner. To get a top-down type image, use a GetImageBuff member function.</p> <p>See [Platform SDK: Windows GDI] in the Microsoft Corporation MSDN Library for the details of BITMAPINFO structure.</p>

VisualBasic6.0

GetBmpInfo (ByRef bmpinfo As BITMAPINFO) As Long

VisualBasic.NET

GetBmpInfo (ByRef bmpinfo As BITMAPINFO) As Integer

Functions	Gets bitmap information of image data.
Arguments	<p>bmpinfo</p> <p>Adress of BITMAPINFO structure that contains the bitmap information of the image data.</p>
Returned values	<p>A [1] is returned when the process is success, and a [0] when in error.</p> <p>Returns a [0] when this function has not been initialized by an InitImageData function.</p>
Remarks	<p>When image data is written with a file input function, the bitmap information of the data is stored here.</p> <p>See [Platform SDK: Windows GDI] in the Microsoft Corporation MSDN Library for the details of BITMAPINFO structure.</p>

VisualBasic6.0

SetBmpInfo (ByRef bmi As BITMAPINFO) As Long

VisualBasic.NET

SetBmpInfo (ByRef bmi As BITMAPINFO) As Integer

Functions	Sets bitmap information of image data.
Arguments	bmi The adress of BITMAPINFO structure with which the bitmap information of the data is stored.
Returned values	A [1] is returned if successful, and a [0] if failed.
Remarks	When image data is written with a file input function, the bitmap information of the data is stored here. See [Platform SDK: Windows GDI] in the Microsoft Corporation MSDN Library for the details of BITMAPINFO structure.

VisualBasic6.0**LoadFile (ByVal filename As String) As Long****VisualBasic.NET****LoadFile (ByVal filename As String) As Integer**

Functions	Reads image data from a file.
Arguments	filename Specifies the file from which the data is read.
Returned values	Returns the format name of the file that has been read. When in error, a FORMAT_NONE is returned.
Remarks	See [On File Format] for information about files that can be read. About File Formats

VisualBasic6.0

**LoadRawFile (ByVal filename As String,
ByVal size_header As Long, ByVal width As Long,
ByVal height As Long, ByVal plane As Long,
ByVal interleave As Long) As Long**

VisualBasic.NET

**LoadRawFile (ByVal filename As String,
ByVal size_header As Integer, ByVal width As Integer,
ByVal height As Integer, ByVal plane As Integer,
ByVal interleave As Integer) As Integer**

Functions	Reads image data from a RAW file.
Arguments	<p>filename Specifies the file name to read.</p> <p>size_header Specifies the size of header information.</p> <p>width Specifies the width of image data.</p> <p>height Specifies the height of image data.</p> <p>plane Specifies the number of color planes.</p> <p>interleave Specifies whether the image data is in an interleave format. [1] Interleave [0] Non interleave The default is [1].</p>
Returned values	Returns a [1] when successful, and a [0] if failed.
Remarks	

VisualBasic6.0

**LoadRawwFile (ByVal filename As String,
ByVal size_header As Long, ByVal width As Long,
ByVal height As Long, ByVal plane As Long,
ByVal interleave As Long) As Long**

VisualBasic.NET

**LoadRawwFile (ByVal filename As String,
ByVal size_header As Integer, ByVal width As Integer,
ByVal height As Integer, ByVal plane As Integer,
ByVal interleave As Integer) As Integer**

Functions	Reads image data from a RAWW file.
Arguments	<p>filename Specifies the file name to read.</p> <p>size_header Specifies the size of header information.</p> <p>width Specifies the width of image data.</p> <p>height Specifies the height of image data.</p> <p>plane Specifies the number of color planes.</p> <p>interleave Specifies whether the image data is in an interleave format. [1] Interleave [0] Non interleave The default is [1].</p>
Returned value	Returns a [1] when successful, and a [0] if failed.
Remarks	

VisualBasic6.0

**SaveFile (ByVal filename As String, ByVal compress As Long,
ByVal comp_quality As Integer,
ByVal ascii As Integer) As Integer**

VisualBasic.NET

**SaveFile (ByVal filename As String, ByVal compress As Integer,
ByVal comp_quality As Integer,
ByVal ascii As Integer) As Integer**

Functions

Stores image data in a file.

Arguments

filename

Specifies the file name to store

compress

TIFF format:

Specifies whether PACKBIT compression should be applied.

Be 0 when omitting.

RAWW format:

Specifies grayscale of image data (default 16)

PNG format:

PNG_COMPRESS_NORMAL (default)

PNG_COMPRESS_SPEEDL

PNG_COMPRESS_SIZE

comp_quality

TIFF format:

Specifies the number of bits (8 or 16) per color plain.

JPEG format:

Specifies compression ratio with a value between 0 and 100.

The higher the value, the lower the compression rate with higher image quality.

Be 75 when omitting.

RAWW format:

Specifies 10- or 16-bit transfer

EFFECTIVE_BITS_LOWSIDE lower

EFFECTIVE_BITS_HIGHSIDE higher (default)

PNG format:

Specifies grayscale of image data (default 8)

ascii

PNM format:

Specifies the data recording format.

[1] Image data is stored in ASCII code.

[0] Image data is stored in binary code.

Be 0 when omitting.

RAWW format:

Specifies the front buffer.

Returned values A [1] is returned when the data is stored correctly, and a [0] when in error.

Remarks The file format for storage is determined by the file name extension.
For details of file formats that can be stored, see [\[File Format\]](#).

VisualBasic6.0

SaveBmpFile (ByVal filename As String) As Long

VisualBasic.NET

SaveBmpFile (ByVal filename As String) As Integer

Functions	Stores image data in a file in BMP format.
Arguments	filename Specifies the file name.
Returned values	A [1] is returned when the data is properly stored and a [0] when in error.
Remarks	

VisualBasic6.0

**SaveTiffFile (ByVal filename As String,
ByVal compress As Long) As Long**

VisualBasic.NET

**SaveTiffFile (ByVal filename As String,
ByVal compress As Integer) As Integer**

Functions	Stores image data in a file in TIFF format.
Arguments	<p>filename Specifies the file name.</p> <p>compress Specifies whether PACKBIT compression should be applied. Be 0 when omitting.</p>
Returned values	A [1] is returned when the data is properly stored and a [0] when in error.
Remarks	

VisualBasic6.0

**SaveTiffFileEx (ByVal filename As String,
ByVal compress As Long, ByVal depth As Long) As Long**

VisualBasic.NET

**SaveTiffFileEx (ByVal filename As String,
ByVal compress As Integer, ByVal depth As Integer) As Integer**

Functions	Stores image data in a file in TIFF format.
Arguments	<p>filename Specifies the file name.</p> <p>compress Specifies whether PACKBIT compression should be applied. Be 0 when omitting.</p> <p>depth Specifies the number of bits (8 or 16) per color plain.</p>
Returned values	A TRUE is returned when the data is properly stored and a FALSE when in error.
Remarks	

VisualBasic6.0

**SaveJpegFile (ByVal filename As String,
ByVal quality As Long) As Long**

VisualBasic.NET

**SaveJpegFile (ByVal filename As String,
ByVal quality As Integer) As Integer**

Functions	Stores image data in a file in JPEG format.
Arguments	<p>filename Specifies the file name.</p> <p>quality Specifies the required compression rate for JPEG storage by a value between 0 and 100. The higher the value, the lower the compression rate resulting in higher picture quality. Be 75 when omitting.</p>
Returned values	A [1] is returned when image data is properly stored, and a [0] when in error.
Remarks	Supports irreversible compression only.

VisualBasic6.0

**SavePnmFile (ByVal filename As String, ByVal ascii As Long)
As Long**

VisualBasic.NET

**SavePnmFile (ByVal filename As String, ByVal ascii As Integer)
As Integer**

Functions	Stores image data in a file in PNM format.
Arguments	<p>filename Specifies the file name.</p> <p>ascii Specifies the data storage format. [1] Stores image data portion in ASCII code. [0] Stores image data portion in binary code. Be 0 when omitting.</p>
Returned values	A [1] is returned when data is properly stored, and a [0] when in error.
Remarks	See [File Format] for details of PNM format.About File Formats .

VisualBasic6.0**SaveRawFile (ByVal filename As String) As Long****VisualBasic.NET****SaveRawFile (ByVal filename As String) As Integer**

Functions	Stores image data in a file in RAW format.
Arguments	filename Specifies the file name.
Returned values	A [1] is returned when data is properly stored and a [0] when in error.
Remarks	See [File Format] for details of RAW format About File Formats .

VisualBasic6.0

**SaveRawwFile (ByVal filename As String,
ByVal bit_depth As Long, ByVal bit_side As Long,
ByRef buff As Byte) As Long**

VisualBasic.NET

**SaveRawwFile (ByVal filename As String,
ByVal bit_depth As Integer, ByVal bit_side As Integer,
ByRef buff As Byte) As Integer**

Functions	Stores image data in a file in RAWW format.
Arguments	<p>filename Specifies the file name.</p> <p>bit_depth Grayscale of the image data (default 16)</p> <p>bit_side Specifies 10- or 16-bit transfer RAWW_EFFECTIVE_BITS_LOWSIDE lower RAWW_EFFECTIVE_BITS_HIGHSIDE higher (default)</p> <p>buff Specifies the top buffer.</p>
Returned values	A [1] is returned when the data is stored properly, and a [0] when in error.
Remarks	See [File Format] for the details of the RAWW format About File Formats ..

VisualBasic6.0

**SavePngFile (ByVal filename As String,
ByVal comp_mode As Long, ByVal depth As Long)
As Long**

VisualBasic.NET

**SavePngFile (ByVal filename As String,
ByVal comp_mode As Integer, ByVal depth As Integer)
As Integer**

Functions	Stores image data in a file in PNG format.
Arguments	<p>filename Specifies the file name.</p> <p>comp_mode Selects a compression mode: PNG_COMPRESS_NORMAL (default) PNG_COMPRESS_SPEEDL priority on speed PNG_COMPRESS_SIZE priority on size</p> <p>depth Specifies the grayscale of image data.</p>
Returned values	A [1] is returned when the data is stored properly, and a [0] when in error.
Remarks	See [File Format] for the details of the PNG format About File Formats .

VisualBasic6.0
InitLut ()

VisualBasic.NET
InitLut ()

Functions	Initializes the LUT (Look Up Table) Returns to the default value (IN = OUT)
Arguments	None
Returned values	None
Remarks	See [LUT (Look Up Table)] for the details of LUT.

VisualBasic6.0**SetUseLut (ByVal On As Long)****VisualBasic.NET****SetUseLut (ByVal On As Integer)**

Functions	Sets ON/OFF of the correction process in LUT.						
Arguemnts	<table><tr><td>on</td><td>Specifies ON or OFF of the LUT correction process</td></tr><tr><td>[1]</td><td>Process ON</td></tr><tr><td>[0]</td><td>Process OFF</td></tr></table>	on	Specifies ON or OFF of the LUT correction process	[1]	Process ON	[0]	Process OFF
on	Specifies ON or OFF of the LUT correction process						
[1]	Process ON						
[0]	Process OFF						
Returned values	None						
Remarks	See [LUT (Look Up Tabel)] for details of LUT.LUT.						

VisualBasic6.0

GetUseLut () As Long

VisualBasic.NET

GetUseLut () As Integer

Functions	Gets ON/OFF status of the LUT correction process.
Arguments	None
Returned values	Returns ON or OFF status of the LUT process. [1] Process ON [0] Process OFF
Remarks	See [LUT (Look Up Table)] for details of LUT.LUT.

VisualBasic6.0**SetLut (ByRef lut As Long, ByVal plane As Long)****VisualBasic.NET****SetLut (ByRef lut As Integer, ByVal plane As Long)**

Functions	Sets up the LUT.
Arguments	<div><div>lut</div><div>Specifies the address of an 32bit type memory array that contains table data. The number of elements is 256.</div></div> <div><div>plane</div><div>Planes to be set: LUT_PLANE_ALL All planes LUT_PLANE_R R (Red) plane LUT_PLANE_G G (Green) plane LUT_PLANE_B B (Blue) plane</div></div>
Returned values	None
Remarks	<p>When the image data is a grayscale (monochrome) or of the 256 palatte color, only the R table is used.</p> <p>See [On the LUT (Look Up Table)] for the details of the LUT.</p>

VisualBasic6.0**SetLutR (ByRef lut As Long)****VisualBasic.NET****SetLutR (ByRef lut As Integer)**

Functions	Sets the Red Table of the LUT.
Arguments	lut Specifies the address of 32bit type memory array containing the table data. The number of elements is 256.
Returned values	None
Remarks	Only this Red Table is used when the image data is in grayscale or in 256-color palette color. See [On the LUT (Look Up Table)] for details of LUT.LUT.

VisualBasic6.0

SetLutG (ByRef lut As Long)

VisualBasic.NET

SetLutG (ByRef lut As Integer)

Functions	Sets the Green Table of the LUT.
Arguments	<div>lut Specifies the address of 32bit type memory array containing the table data. The number of elements is 256.</div>
Returned values	None
Remarks	<p>This table is void when the image data is in grayscale or in 256-color palette color.</p> <p>See [On the LUT (Look Up Table)] for details of LUT.LUT.</p>

VisualBasic6.0**SetLutB (ByRef lut As Long)****VisualBasic.NET****SetLutB (ByRef lut As Integer)**

Functions	Sets the Blue Table of the LUT.
Arguments	lut Specifies the address of 32bit type memory array containing the table data. The number of elements is 256.
Returned values	None
Remarks	This Blue Table is void when the image data is in grayscale or in 256-color palette color. See [On the LUT (Look Up Table)] for details of LUT. LUT.

VisualBasic6.0

GetLut (ByRef lut As Long, ByVal plane As Long)

VisualBasic.NET

GetLut (ByRef lut As Integer, ByVal plane As Long)

Functions	Gets settings of the LUT.								
Arguments	lut	Specifies the address of an 32bit type memory array that receives the table data. The number of elements is 256 (1024 for LUT_PLANE_ALL).							
	plane	Planes to be set: <table><tr><td>LUT_PLANE_ALL</td><td>All planes</td></tr><tr><td>LUT_PLANE_R</td><td>R (Red) plane</td></tr><tr><td>LUT_PLANE_G</td><td>G (Green) plane</td></tr><tr><td>LUT_PLANE_B</td><td>B (Blue) plane</td></tr></table>	LUT_PLANE_ALL	All planes	LUT_PLANE_R	R (Red) plane	LUT_PLANE_G	G (Green) plane	LUT_PLANE_B
LUT_PLANE_ALL	All planes								
LUT_PLANE_R	R (Red) plane								
LUT_PLANE_G	G (Green) plane								
LUT_PLANE_B	B (Blue) plane								
Returned values	None								
Remarks	When the image data is a grayscale (monochrome) or of the 256-color palatte color, only the R table is used.								
	See [On the LUT (Look Up Table)] for the details of the LUT.LUT.								

VisualBasic6.0**GetLutR (ByRef lut As Long)****VisualBasic.NET****GetLutR (ByRef lut As Integer)**

Functions	Gets the setting of Red Table of the LUT.
Arguments	<div>lut Specifies the address of an 32bit type memory array that receives the table data. The number of elements is 256.</div>
Returned values	None
Remarks	<p>Only this Red Table is used when the image data is in grayscale or in 256-color palette color.</p> <p>See [On the LUT (Look Up Table)] for details of LUT.LUT.</p>

VisualBasic6.0**GetLutG (ByRef lut As Long)****VisualBasic.NET****GetLutG (ByRef lut As Integer)**

Functions	Gets the setting of Green Table of the LUT.
Arguments	<div>lut Specifies the address of an 32bit type memory array that receives the table data. The number of elements is 256.</div>
Returned values	None
Remarks	<p>This Green Table is void when the image data is in grayscale or in 256-color palette color.</p> <p>See [On the LUT (Look Up Table)] for details of LUT.LUT.</p>

VisualBasic6.0**GetLutB (ByRef lut As Long)****VisualBasic.NET****GetLutB (ByRef lut As Integer)**

Functions	Gets the setting of Blue Table of the LUT.		
Arguments	<table><tr><td>lut</td><td>Specifies the address of an 32bit type memory array that receives the table data. The number of elements is 256.</td></tr></table>	lut	Specifies the address of an 32bit type memory array that receives the table data. The number of elements is 256.
lut	Specifies the address of an 32bit type memory array that receives the table data. The number of elements is 256.		
Returned values	None		
Remarks	<p>This Blue Table is void when the image data is in grayscale or in 256-color palette color.</p> <p>See [On the LUT (Look Up Table)] for details of LUT.LUT.</p>		

VisualBasic6.0

SetLutGamma (ByVal gamma As Double, ByVal plane As Long)

VisualBasic.NET

**SetLutGamma (ByVal gamma As Double,
ByVal plane As Integer)**

Functions Sets the LUT with gamma correction values

Arguments **gamma**

 Sets the gamma correction values.

 The range of gamma is 0.1 to 2.0.

plane

 Specifies the color plane (table) that performs gamma correction.

 LUT_PLANE_R Red (R) Plane

 LUT_PLANE_G Green (G) Plane

 LUT_PLANE_B Blue (B) Plane

 LUT_PLANE_ALL All Planes

Returned values None

Remarks The plane values are disregarded when the image data is in grayscale or in 256-color palette color.

 See [\[On the LUT \(Look Up Table\)\]](#) for details of LUT. LUT

VisualBasic6.0

SetLutContrast (ByVal contrast As Long, ByVal plane As Long)

VisualBasic.NET

**SetLutContrast (ByVal contrast As Integer,
ByVal plane As Integer)**

Functions	Sets the LUT with contrast correction values								
Arguments	gamma	Sets the contrast correction values. The range of contrast is between −256 and + 256.							
	plane	Specifies the color plane (table) that performs contrast correction. <table><tr><td>LUT_PLANE_R</td><td>Red (R) Plane</td></tr><tr><td>LUT_PLANE_G</td><td>Green (G) Plane</td></tr><tr><td>LUT_PLANE_B</td><td>Blue (B) Plane</td></tr><tr><td>LUT_PLANE_ALL</td><td>All Planes</td></tr></table>	LUT_PLANE_R	Red (R) Plane	LUT_PLANE_G	Green (G) Plane	LUT_PLANE_B	Blue (B) Plane	LUT_PLANE_ALL
LUT_PLANE_R	Red (R) Plane								
LUT_PLANE_G	Green (G) Plane								
LUT_PLANE_B	Blue (B) Plane								
LUT_PLANE_ALL	All Planes								
Returned values	None								
Remarks	The plane values are disregarded when the image data is in grayscale or in 256-color palette color.								
	See [On the LUT (Look Up Table)] for details of LUT. LUT.								

VisualBasic6.0

**SetLutBrightness (ByVal brightness As Long,
ByVal plane As Long)**

VisualBasic.NET

**SetLutBrightness (ByVal brightness As Integer,
ByVal plane As Integer)**

Functions Sets the LUT with brightness correction values

Arguments **gamma**

Sets the brightness correction values.
The range of gamma is between -256 and +256.

plane

Specifies the color plane (table) that performs
brightness correction.

LUT_PLANE_R	Red (R) Plane
LUT_PLANE_G	Green (G) Plane
LUT_PLANE_B	Blue (B) Plane
LUT_PLANE_ALL	All Planes

Returned values None

Remarks The plane values are disregarded when the image data is in grayscale or
in 256-color palette color.

See [\[On the LUT \(Look Up Table\)\]](#) for details of LUT. LUT.

VisualBasic6.0**GetLutGamma (ByVal plane As Long) As Double****VisualBasic.NET****GetLutGamma (ByVal plane As Integer) As Double**

Functions	Gets the LUT gamma correction values.	
Arguments	plane	Specifies the color planes (table) that get a gamma correction value.
	LUT_PLANE_R	Red (R) Plane
	LUT_PLANE_G	Green (G) Plane
	LUT_PLANE_B	Blue (B) Plane
	LUT_PLANE_ALL	All Planes
Returned values	Returns the current gamma correction values.	
	If LUT_PLANE_ALL is specified after each plane is given a correction value independently, the returned value will be indefinite.	
Remarks	The values in the plane are disregarded when the image data is in grayscale or in 256-color palette color.	
	See [On the LUT (Look Up Table)] for details of LUT. LUT.	

VisualBasic6.0**GetLutContrast (ByVal plane As Long) As Long****VisualBasic.NET****GetLutContrast (ByVal plane As Integer) As Integer**

Functions Gets the LUT contrast correction values.

Arguments **plane**

Specifies the color planes (table) that get a contrast correction value.

LUT_PLANE_R	Red (R) Plane
LUT_PLANE_G	Green (G) Plane
LUT_PLANE_B	Blue (B) Plane
LUT_PLANE_ALL	All Planes

Returned values Returns the current contrast correction values.
If LUT_PLANE_ALL is specified after each plane is given a correction value independently, the returned value will be indefinite.

Remarks The values in the plane are disregarded when the image data is in grayscale or in 256-color palette color.

See [\[On the LUT \(Look Up Table\)\]](#) for details of LUT. LUT.

VisualBasic6.0

GetLutBrightness (ByVal plane As Long) As Long

VisualBasic.NET

GetLutBrightness (ByVal plane As Integer) As Integer

Functions	Gets the LUT brightness correction values.	
Arguments	plane	Specifies the color planes (table) that get a brightness correction value.
	LUT_PLANE_R	Red (R) Plane
	LUT_PLANE_G	Green (G) Plane
	LUT_PLANE_B	Blue (B) Plane
	LUT_PLANE_ALL	All Planes
Returned values	Returns the current brightness correction values.	
	If LUT_PLANE_ALL is specified after each plane is given a correction value independently, the returned value will be indefinite.	
Remarks	The values in the plane are disregarded when the image data is in grayscale or in 256-color palette color.	
	See [On the LUT (Look Up Table)] for details of LUT.LUT.	

6. Compatible with Old Version SDK (Unused) /Compatible with Old Version SDK

Extra commands are issued for each of camera models using the CameraControl [GetExtraCommand] / [SetExtraCommand] functions.
Extra commands are shown below:

Compatible with Old Version SDK (Unused)	
EXTRA_SET_MONITOROUT_EDGEENHANCE	SetEdgeEnhancementMode
EXTRA_GET_MONITOROUT_EDGEENHANCE	HaveEdgeEnhancement
EXTRA_SET_MONITOROUT_ZOOM	SetZoomMode (HaveZoomMode)
EXTRA_GET_MONITOROUT_ZOOM	GetZoomMode (HaveZoomMode)
EXTRA_SET_MONITOROUT_DISP_STATUS	SetStatusDisplayMode
EXTRA_GET_MONITOROUT_DISP_STATUS	GetStasusDisplayMode
EXTRA_MONITOROUT_PLAY	Play
EXTRA_MONITOROUT_PAUSE	Pause
EXTRA_MONITOROUT_STOP	Stop
EXTRA_MONITOROUT_STEP_FORWARD	StepForward
EXTRA_MONITOROUT_STEP_BACK	StepBack
EXTRA_MONITOROUT_GOTO_ANYFRAME	GoAnyFrame
EXTRA_MONITOROUT_GOTO_START	GoStart
EXTRA_MONITOROUT_GOTO_END	GoEnd
EXTRA_MONITOROUT_GOTO_TRIGGER	GoTrigger
Compatible with Old Version SDK	
EXTRA_SET_MONITOROUT_MODE	SetMonitorOutMode (HaveMonitorOutMode)
EXTRA_GET_MONITOROUT_MODE	GetMonitorOutMode (HaveMonitorOutMode)
EXTRA_SET_ENABLE_MCDL	SetEnableMCDL (HaveMCDL)
EXTRA_GET_ENABLE_MCDL	GetEnableMCDL (HaveMCDL)
EXTRA_SET_ENABLE_IRIG	SetEnebleIRIG (HaveIRIG)
EXTRA_GET_ENABLE_IRIG	GetEnableIRIG (HaveIRIG)
EXTRA_TRANSFER_MCDL	TransferMCDL
EXTRA_TRANSFER_IRIG	TransferIRIG
EXTRA_SHADING_COMPENSATION	ShadingCompensation

7. File Formats

File formats that can be used in this library are shown below:

7.1. BMP File (*.bmp)

The bitmap format that is used as a standard format in Windows environment.
Compatible with full color, 8-bit palette color and grayscale.

7.2. TIFF File (*.tif)

Another bitmap file format (Tagged Image File Format)
Supports noncompression and PackBit compression files only. Default is noncompression.

7.3. JPEG File (*.jpg)

A file format for compressed still image.
This library supports Lossy (non-reversible) compression only.
The compression parameter can be set between 0 and 100. The higher the number, the lower the compression rate resulting in higher picture quality. Default is 75.

7.4. PNM File (*.ppm / *.pgm)

Another bitmap file format (Tagged Image File Format).
A file extension changes with contents of a image.
PNM is the general term of the following format.

Color :PPM (Portable Pixel Map)
Grayscale :PGB (Portable Gray Map)
2Value :PBM (Portable Bit Map) (not supported)

Image data portion can be saved in binary data or ASCII format.

7.5. RAW File (*.raw / *.raww)

A binary data file format without header information.
Data of color image is stored in the order of interleaved (RGBRGBRGB..etc.).
RAWW file consists of 16-bit data.

7.6. PNG File (*.png)

A file format for compressed still image data which uses reversible compression technique.
Three compression parameters are available to choose from: priority on image quality, normal, and priority on speed.

8. Look Up Table (LUT)

The ImageData have LUT functions that offer the capability to freely correct displayed images without affecting the original image data. Image data corrected on LUT is acquired by GetWindowsImage function of the ImageData . Valid/Invalid of LUT can be switched by the [SetUseLut] function.

The LUT prepares and sets an 32bit array for each of the color planes. Each array contains the value converted from the 256-step density scale of the original image. For grayscale of 256 steps, only the [SetLutR] function is used.

***Note: For 256-color palette color, the whole palette is converted.**

8.1. Example of application: Negative inversion of image

Ex) VisualBasic 6.0

```
//LUT data of Red, Green, Blue planes
Dim lut_r(255) As Long, lut_g(255) As Long, lut_b(255) As Long

//Store density-reversed values

For i = 0 To 255
    lut_r(i) = 255 - i
    lut_g(i) = 255 - i
    lut_b(i) = 255 - i
Next i

// Acquire an original image
nSts = LoadFile("Sample.bmp")

// Set LUT
iret2 = SetLutR(lut_r(0))
iret2 = SetLutG(lut_g(0))
iret2 = SetLutB(lut_b(0))

// Make LUT valid
iret2 = SetUseLut(1)

// Acquire LUTcorrected image
'BITMAPINFOHEADER
BIH.biSize = 40
BIH.biWidth = lx
BIH.biHeight = ly
BIH.biPlanes = 1
BIH.biBitCount = 24
BIH.biCompression = 0
BIH.biSizeimage = CLng(lx) * ly * 3
iret2 = GetImageData(m_buffer(0))
iret2 = GetWindowsImage(image_buffer(0), m_buffer(0), BIH)
```

Ex) VisualBasic.NET

```
//LUT data of Red, Green, Blue planes
Dim lut_r(255) As Integer, lut_g(255) As Integer, lut_b(255) As Integer
```

```
//Store density-reversed values

For i = 0 To 255
    lut_r(i) = 255 - i
    lut_g(i) = 255 - i
    lut_b(i) = 255 - i
Next i

// Acquire an original image
nSts = LoadFile("Sample.bmp")

// Set LUT
iret2 = SetLutR(lut_r(0))
iret2 = SetLutG(lut_g(0))
iret2 = SetLutB(lut_b(0))

// Make LUT valid
iret2 = SetUseLut(1)

// Acquire LUTcorrected image
'BITMAPINFOHEADER
BIH.biSize = 40
BIH.biWidth = lx
BIH.biHeight = ly
BIH.biPlanes = 1
BIH.biBitCount = 24
BIH.biCompression = 0
BIH.biSizeimage = CInt(lx) * ly * 3
iret2 = GetImageData(m_buffer(0))
iret2 = GetWindowsImage(image_buffer(0), m_buffer(0), BIH)
```

www.photron.com

In Americas and Antepodes

PHOTRON USA, INC.

9520 Padgett Street, Suite 110

San Diego, CA 92126-4446, USA

Phone: 858-684-3555

Fax: 858-684-3558

E-mail: image@photron.com

In Europe:

PHOTRON EUROPE LIMITED

Willowbank House

84 Station Road

Marlow, Bucks SL7, UK

Phone: +44(0) 1628 89 4353

Fax: +44(0) 1628 89 4354

E-mail: image@photron.com

In other areas:

PHOTRON LIMITED

Chiyodafujimi BLDG.,

Fujimi 1-1-8, Chiyoda-Ku

Tokyo 102-0071, Japan

Phone: +81 3 3238 2106

Fax: +81 3 3238 2109

E-mail: image@photron.com

FASTCAM Control SDK Library "Pcc4VB" Reference Manual

English Edition

April, 2006

Rev. 2.976