HAMAMATSU

PHOTONICS DEUTSCHLAND GMBH

HiPic/HPD-TA Remote Control Programmers Handbook

HiPic/HPD-TA Version 6.3.0

WT, HPD, Document Version 1.0, 07.03.2003

Table of Content

Table of Content 2
General 7
About this manual 7
Purpose and general realization 7
The interface objects 7
Type of I/F objects and naming conventions 8
Availability of interface objects 9
Using the HiPic/HPD-TA User Interface 10
Showing and hiding dialogs 12
Standalone program and COM components 12
Object model and main objects 13
Using the I/F objects 14
General 14
Properties, Methods and Events 14
HCommand object 15
HCommand4Array object and other arrays 16
HMenu 17
HCheck 17
HFrame 17
HDisp 17
HEditString 17
HEditNumber 18
HEntry 18
HRadios 19
HTab 19
HSingleTab 20
HProgress 20
HWindow 20
HLut 22
HImageArea 23
HDevPar 26
Using the main objects in HiPic and HPD-TA 27
Startup (HInitHi and HInitTa) 27
The application object (HAppHiPic and HAppHPDTA) 28
The HImages and the HImage object 29
The HImages object 29

The HImage object 29
The HExternalDevices and the HExternalDevice object 30
The HExternalDevices object 30
The HExternalDevice object 32
HLUTControl 33
HSystemScaling 33
HProfileScalingData, HImageScalingData, HSytemScalingData and HPRFParametersData 33
HProfileScalingData 33
HSequence and HJitter 35
The Camera objects (HC4880, HC488080, HACam, HC474295, HC474298, HC7300, HC800010, HC800020, HFlatPanel) 35
The HAcq object 36
Utility objects (HLicence, HAsyncCommand, HMsgBox, HError) 36
Synchronous and asynchronous commands 37
Message boxes 38
Configuring a remote COM-client and COM-server environment 39
Target operating system platform issues 39
General Topics on Windows 95 and Windows 98 Platforms 39
General Topics on Windows NT4 and Windows 2000 Platforms 39
Networking issues 40
Configuring the COM-Server 40
a) Configuring a COM-Server on Windows 95 and Windows 98 Platforms 40
Configuring a COM-Server on Windows NT4 and Windows2000 Platforms 44
Configuring the COM Client 51
a) Configuring the COM-Client when using default COM access mechanism 51
b) Configuring the COM-Client when using explicit server names 53
License and Keys 53
Migrating from Version 6.1 to 6.2 53
Outline 53
Details 56
Migrating from Version 6.2 to 6.3 76
Outline 76
Details 77
Appendix A: All Public Properties/Functions/Subs/Events 85
HInitHi 85
HAppHiPic 87
HInitTa 93
HAppHPDTA 97
HImage 104
HImages 108
HLutControl 113
HSystemScaling 114

HSequence 120
HJitter 124
HGeneral 124
HACam 124
HC474295 127
HC4880 130
HC488080 134
HC474298 137
HC7300 140
HC800010 143
HC800020 145
HFlatPanel 147
HDCam 149
HExternalDevices 151
HDevPar 161
HExternalDevice 162
HDevPars 163
HAcq 164
HGrb 169
HLicence 175
HUtils 175
HAsyncCommand 175
HMsgBox 176
HError 176
HCheck 176
HCheck4Array 177
HCommand 177
HCommand4Array 178
HDisp 178
HDisp4Array 178
HEditNumber 178
HEditNumber4Array 179
HEditString 179
HEntry 180
HEntry4Array 181
HFrame 181
HFrame4Array 181
HMenu 182

HProgress 182
HProgress4Array 183
HRadios 183
HRadios4Array 184
HSingleTab 184
HTab 185
HWindow 186
HLut 188
HImageArea 189
HEditString4Array 190
HWindow20Array 191
IHCall2Appl 192
IHCall2Cam 192
IHCall2Exttrig 194
IHCall2Licence 194
IHCall2UserFunc 194
IHCall2LUTControl 194
IHCall2Sequence 195
IHCall2AcqControl 196
IhCall2ExtDevs 196
IhCall2Shutdown 196
IHCall2TransAbs 196
IHCall2MC 196
HProfileScalingData 196
HSystemScalingData 198
HImageScalingData 198
HPRFParametersData 199
Sample programs *201*
Sample program CltHiDll and CltTaDll 201
Basic structure of CltHiDll and CltTaDll 202
Startup 202
Initializing of the client program 204
Ending the component 205
Executing and availability of commands 205
Setting the exposure time 205
Using events 206
Saving and loading images specifying the filename and type 206
Executing image acquisitions and image analysis in one function 206
Using the HExternalDevices object (HPD-TA only) 207
Using the HExternalDevice object (HPD-TA only) 207
Using the HDevPar object (HPD-TA only) 207
Sample program CltHiExe and CltTaExe 208
Image display 208

Measurement cycles 208
Sample program CltHiExe2 and CltTaExe2 209
Copying data and display with the local component 209
TaMonit and HiMonit 211
TaStart_U and HiStart_U 211

General

The following document describes how the HiPic/HPD-TA can be controlled by another application. This documentation is written for programmers who intends to make a client program using the HiPic or HPD-TA remote control component. It is assumed that the programmer is familiar with the purpose and the usage of the standalone version of these programs. Its is also assumed that the programmer is familiar with a programming language capable of using a COM component. This document is no programming lecture for those not familiar with either of the a.m. topics. For those not familiar how to use the standalone versions of these programs it is recommended to read the HiPic or HPD-TA user manual first. It is also recommended to provide all the hardware which is needed to run the application which should be developed.

About this manual

As the concept for HiPic and HPD-TA is the same and most of objects are identical this manual describes both versions. The manual starts to describe the basic concept of programming the HiPic/HPD-TA ActiveX component. As a basic feature the interface objects are described next. The usage of the main objects (like the Application or the Images object) are described roughly mentioning the available features of all these main objects. The configuration of a remote server is described in detail in the following chapter. For those familiar with programming with HiPic/HPD-TA 6.1 a chapter describes the difference between version 6.1 and the current version 6.2. The Appendix describes all properties, methods and events of all public objects. The identical information is also contained in the component itself (You can view this information for example with the Visual Basic 6.0 object catalog or a similar tool). As the object catalog describes only the purpose of these properties, methods and events and not the parameters, these parameters are described additionally in the respective chapters. Finally sample programs are introduced and explained.

Purpose and general realization

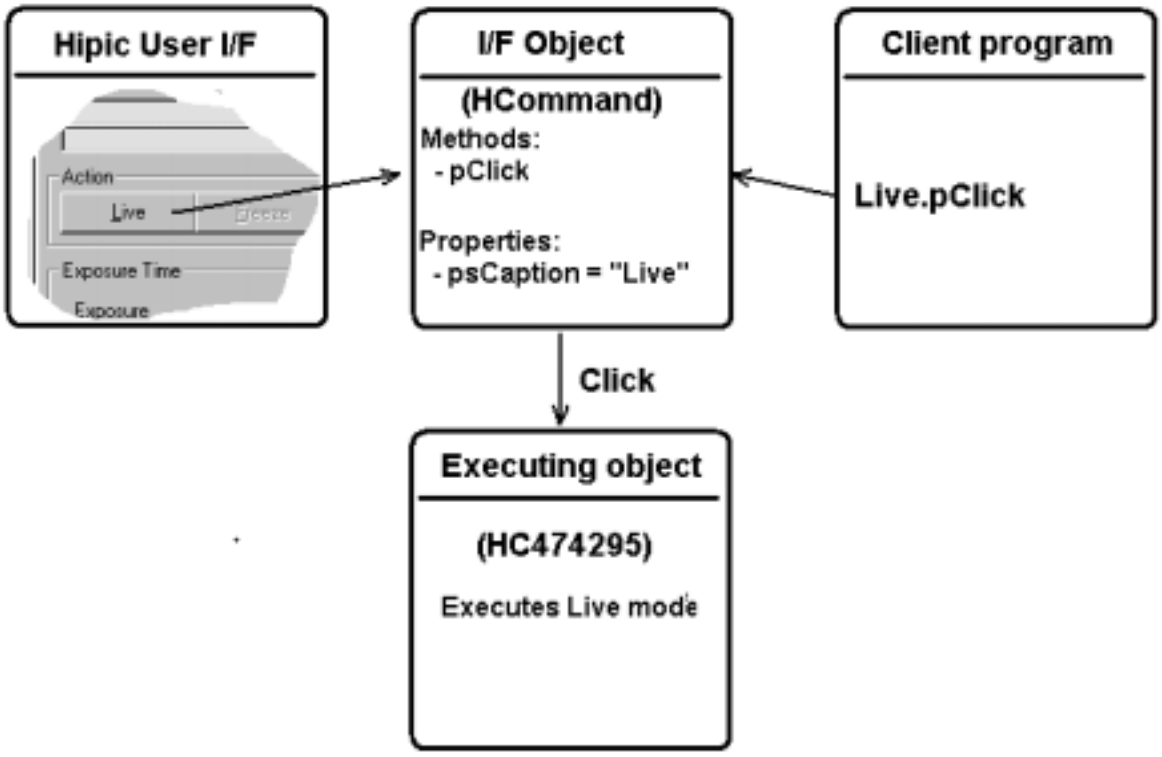
The purpose of the HiPic/HPD-TA remote control is to control the functions of the HiPic and the HPD-TA from another application (called the client program). To realize this, the mechanisms of COM (=component object model, also referred to as “Automation”) are used to control the functions of the HiPic/HPD-TA (called the automation-server). The general principle is that the client programmer can access every function which is accessible by the standard HiPic/HPD-TA user interface. The automation-server behaves then exactly as if the user had invoked the corresponding command on the user I/F (with some logical exceptions). We can say that "programming is as easy as using the standalone program".

The interface objects

In order to realize the target that "programming is as easy as using the standalone program" a set of interface objects have been created. Every interface object represents a control element on the HiPic/HPD-TA User I/F. For every type of control element a different type of interface object is created. There a object types for command buttons, check buttons, edit fields for numbers etc. These interface objects are logically located between the User I/F and the executing code. These interface objects can be accessed both by the HiPic/HPD-TA user I/F and the client program. The interface objects are independent of any user I/F. If, however, a

HiPic/HPD-TA user I/F is used every interface object has one (or several) so called reference objects. Any changes on the interface objects is reflected to these reference objects. If e.g. the user changes the value of a checkbox type interface object the reference checkbox changes its status simultaneously.

As an example see the following diagram. We use a very simple interface object: the HCommand object. In our example we use the HCommand object which is associated with the "Live" pushbutton on the C4742-95. This interface object can be accessed from both the HiPic/HPD-TA user interface and the client program. Every interface object has properties, methods and events.
One of the methods of the HCommand I/F object is the "pClick" method. Whenever the "pClick" method is invoked the object raises an event "Click". The executing object will detect this event and consequently execute the Live mode. The HiPic/HPD-TA user interface calls this method when the user clicks to the Live pushbutton on the C4742-95 camera dialog. This method can be executed from a client program as well. The result is the same: The Live mode starts. Of course this is true for all other pushbuttons which can be pressed on the user I/F. To explain the meaning of a property we take the psCaption property of the HCommand object. When this property is set the labeling on the corresponding pushbutton will be changed.



Type of I/F objects and naming conventions

There are many types of interface objects. Here is a list of all available types. The names of the interface objects follow a special naming convention. It is easy to find out the corresponding control element on the user interface when knowing the naming convention.

The following naming conventions are used to name I/F objects:

pHXxxYyyZzz (example: pHComMainLive)
P means public, H means HiPic/HPD-TA, Xxx stands for the type of class, Yyy is the name of the dialog, and Zzz is the name of the control (Where Zzz can consist of several parts in some cases. Example: phCh4CamClearFrameBuffer, Zzz=ClearFrameBuffer).
In our example Com=HCommand, Main=Main Dialog, Live=Live mode command

The following I/F classes are existing:

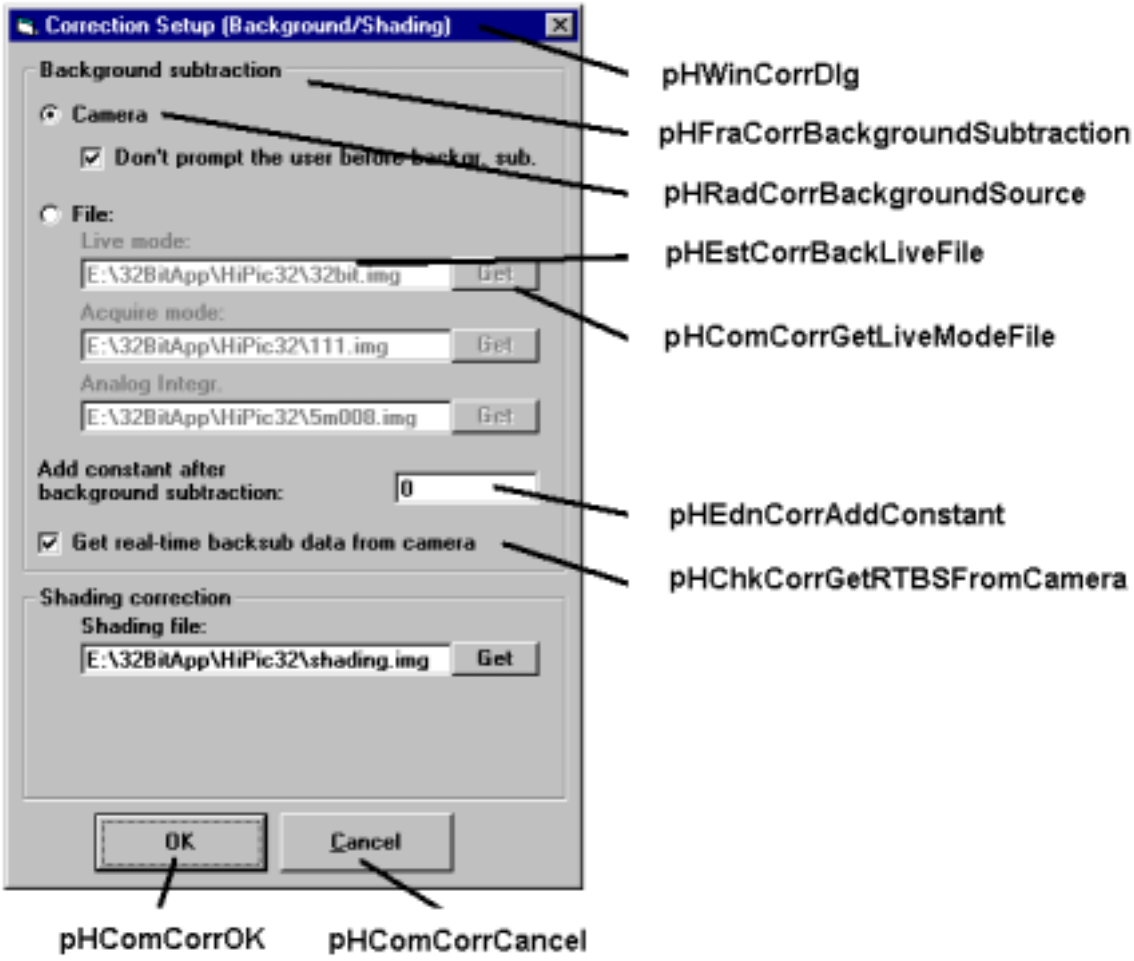
Short	Name	Description
Che	HCheck	Check Box
Ch4	HCheck4Array	Array of 4 check boxes (for the 4 acquisition modes)
Com	HCommand	Command button
Co4	HCommand4Array	Array of 4 command buttons (for the 4 acquisition modes)
Dis	HDisp	Display element
Di4	HDisp4Array	Array of 4 display elements (for the 4 acquisition modes)
Edn	HEditNumber	Edit box for number input
Ed4	HEditNumber4Array	Array of 4 edit boxes for number input (for the 4 acquisition modes)
Est	HEditString	Edit box for string input
Es4	HEditString4Array	Array of 4 edit boxes for string input (for the 4 acquisition modes)
Ent	HEntry	Entry input (slidebar with edit box or combo box)
En4	HEntry4Array	Array of 4 entry input (slidebar with edit box or combo box)
Fra	HFrame	Frame control
Fr4	HFrame4Array	Array of 4 frame controls (for the 4 acquisition modes)
Men	HMenu	Menu Entry
Pro	HProgress	Progress bar
Pr4	HProgress4Array	Array of 4 Progress bars (for the 4 acquisition modes)
Rad	HRadios	Group of radio buttons
Ra4	HRadios4Array	Array of 4 group of radio buttons s (for the 4 acquisition modes)
Sta	HSingleTab	Single Tab on a Tab array
Tab	HTab	Tab array
Win	HWindow	Dialog Window or a dependent window
W20	HWindow20Array	Array of 20 Window controls (for image windows)
Lut	HLut	LUT control within the LUT control dialog
Ima	HImageArea	Image display area on a image window
Dev	HDevPar	Device parameter (HPD-TA external devices only)

Availability of interface objects

To have access to a specified control element on any user interface (this is not only true for the HiPic/HPD-TA) several conditions have to be fulfilled:

- The dialog must be shown on the screen
- The control element must be visible and enabled
- The window where the control element is embedded must be visible and enabled
- Sometimes this window is also embedded in another window etc. which means there is a full hierarchy of windows which have to be enabled and visible to allow the control element to be accessible

This situation is reflected by the interface objects with the properties pfVisible pfEnable and pcParent. When a control is not visible on the screen the corresponding interface objects property pfVisible is FALSE, when a control is not enabled the corresponding interface objects property pfEnabled is FALSE. The property pcParent of the interface object contains a reference to the window where the control element is embedded. The last element in the hierarchy is the dialog window where all control elements are located.



We will look at the Correction setup dialog as an example. The HEditNumber object "pHEdnCorrAddControl" is embedded into the HFrame object "pHFraCorrbackgroundSource" The object "pHFraCorrbackgroundSource" is embedded in the "pHWinCorrDlg". Therefore the parents of these objects are:

```
pHEdnCorrAddControl.pcParent = pHFraCorrbackgroundSource
pHFraCorrbackgroundSource.pcParent = pHWinCorrDlg
pHWinCorrDlg.pcParent = Nothing
```

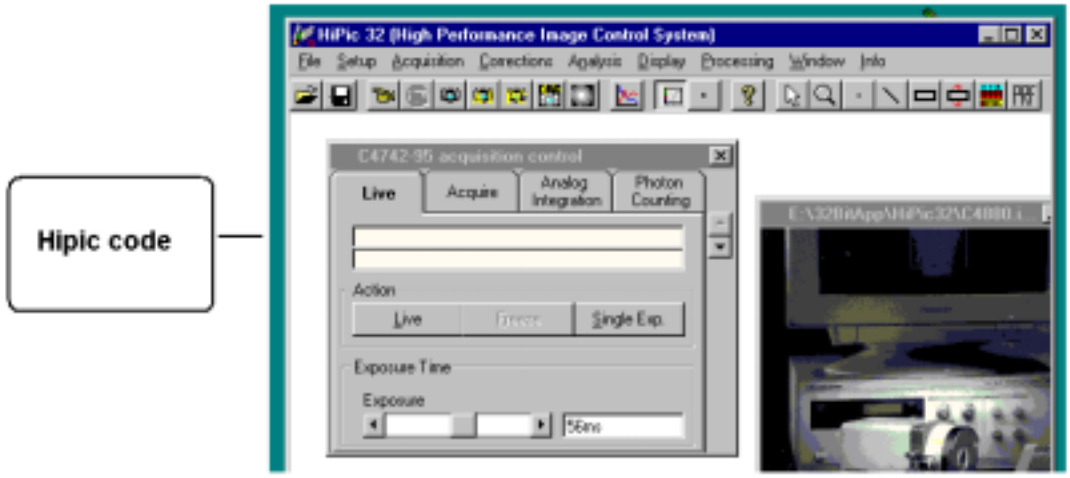
(This is the topmost element in the hierarchy)

To fascilitate the task to find out whether an interface object is accessible or not every interface object has an additional property pfControlAvail. If this is true the control element is accessible. The pfControlAvail property internally checks all pfVisible and pfEnabled properties of all higher order interface objects up to the dialogs window HWindow object.

Using the HiPic/HPD-TA User Interface

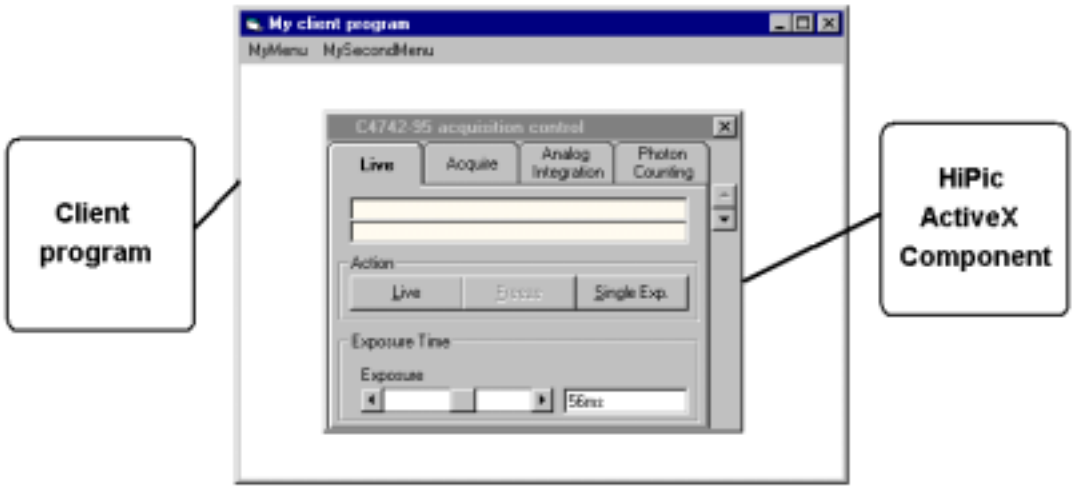
There are three ways of using the HiPic/HPD-TA User I/F.

The simplest way is to use the standalone version of the HiPic/HPD-TA. In this case the full HiPic/HPD-TA User I/F is shown. No other dialogs can be used.



HiPic/HPD-TA Standalone

The second way is using some (or all) dialogs from the HiPic/HPD-TA User I/F together with a client programs main window. The dialogs from the HiPic/HPD-TA User I/F can be placed into the client's main window like a child window or can be used as a standalone popup window. This is very convenient for the client programmer. No code has to be written for accessing the individual controls on these dialogs.
Showing these dialogs is done by setting the pfUserIF and the pfVisible properties of the main dialogs HWindow object to TRUE. By setting the plHWndParentWindow property of this HWindow object to the window handle of the window which should be the parent window.



Clientprogram is using dialogs from the HiPic/HPD-TA User I/F

The third way of using functionality within the HiPic/HPD-TA ActiveX component is to use the individual controls without showing the HiPic/HPD-TA User I/F. In this case individual access has to be made to the Interface Objects. This is mandatory if the client is running on a different computer than the server.

Client program using only the functionality of the HiPic without using the HiPic/HPD-TA User I/F.

Showing and hiding dialogs

To access any interface object on a dialog all involved interface objects must have set its pfVisible and pfEnabled properties to true. To allow to control such interface object without showing the corresponding dialog the HWindow object (the topmost interface object in the hierarchy) has an additional property pfUserIF. When this property is set to FALSE the dialog is not shown on the screen. The value of this property, however, has no influence on the accessibility of the interface objects on this dialog. Thus setting this property to TRUE will show the dialog and setting it to FALSE will not show the dialog.
This property allows to develop applications where no HiPic/HPD-TA user IF is used. During development the property can be set to TRUE to visually check what happens. When the development is finished the property can be set to FALSE the same things will happen but without showing the dialog on the screen (Of course showing the dialogs take some time so that the program may run a little bit faster without the user I/F). The dialog window and all the related controls are created as soon as both properties pfVisible and pfUserIF are set to TRUE. If one or both of these properties are set to FALSE the dialog will be unloaded.
Sometimes, however, one needs to access the window itself (maybe by using its window handle) before it is visible. In such case the properties pfVisible and pfUserIF must be set to TRUE (otherwise the window will not be created) but it should not be displayed. To archive this the property pfHideWindow can be used to hide the window. To perform this task set the pfHideWindow property to TRUE , then set the properties pfVisible and pfUserIF to TRUE, after that the window is created and can be accessed with its window handle. To show the window finally it is sufficient to set the pfHideWindow property back to FALSE.
Sometimes showing and hiding of a dialog is done by another command. Let us take the a.m. example of the correction setup. Showing is normally done by executing the menu command "Correction Setup" and hiding is done by the "OK" or "Cancel" pushbutton. It is recommended to do the same thing by calling the corresponding interface objects, not just setting the pfVisible property to TRUE and FALSE. The program may need to execute the code which is associated with the menu or pushbutton calls. E.g. the Setup will not correctly be performed if the dialog is just hidden by using the pfVisible property. Executing the pHComCorrOK.pClick method is necessary to make correct setup in this case.

Standalone program and COM components

For both HiPic and HPD-TA there will be two versions: An ActiveX-EXE and an ActiveX-DLL (ActiveX is another word for COM component).

The ActiveX-EXE (its filename is HiPic32u.exe and HPDTA32u.exe) can be started as a standalone program just by clicking on the icon. It can also be started from a client program and then has the function of a component. This component can be used both by a client on the same computer and a client on a different computer (DCOM). To use an ActiveX-EXE component provides more security because a fatal error in the component does not shutdown the client application. Direct memory access from the server to the client is not possible for an ActiveX-EXE component. Transferring image data has to be done by dumping and reading the data to a file.
The ActiveX-DLL (its filename will be HiPic32u.dll and HPDTA32u.dll) can only be started by a client program and runs in the same address space as the client program. This allows very fast access to the functions and data and allows direct access to image data by memory handles. It is only possible when the server and the client is running on the same computer. Depending on the host (=server application) there can be some restrictions concerning the display of non modal dialogs. (In some cases non modal dialogs cannot be shown. We do not have much information about this topic jet. The only thing which is clear that an application developed under VB5.0 or VB 6.0 is able to show non modal dialogs).

Object model and main objects

The HiPic and HPD-TA has a well defined object model. The topmost object in the object model of the HiPic or HPD-TA is the HAppHiPic or HAppHPDTA object. This object has several dependent objects (The interface objects which belongs to these objects are not listed) These are:

HAppHiPic:
- pHACam
- pHC4880
- pHC488080
- pHC474295
- pHC474298
- pHC7300
- pHC800010
- pHC800020
- pHFlatPanel
- pHAcq
- pHAsyncCommand
- pHImages
- pHLicence
- pHLUTControl
- pHMessageBox
- pHSequence
- pHSystemScaling

HAppHPDTA:
- pHACam
- pHC4880
- pHC488080
- pHC474295
- pHAcq
- pHAsyncCommand
- pHImages

pHLUTControl
pHMessageBox
pHSequence
pHSystemScaling
pHExternalDevices

Some of these objects again have some dependent objects these are:

HImages:
HImage (There are 20 dependent images)

HExternalDevices:
HExternalDevice (Currently there are 3 dependent ExternalDevices)

Some of these objects have more dependent objects but they are not public so these dependencies is not of interest here.

To give two examples:
To access the object HAcq one can write: HAppHipic.pHAcq.
To access one image object one can write: HAppHipic.pHImage.pcItem(Index).

Using the I/F objects

General
The I/F objects provides access to all features of the program. Making a client program will consist to a large extent on using properties, methods and events of the I/F objects. According to the different types of controls there are different types of I/F objects (We call the defining code of such an I/F type a "class"). There are many features which are common to all or several of these I/F classes. The following description will explain such common features only once describing a selected I/F class. The description is valid for all other classes with the same features. Thus it is recommended that you read the whole chapter as one unit because later parts of the chapter assume that you know the content of earlier parts. It starts form the simples interface objects and ends up with the most complex ones.

Properties, Methods and Events
Properties are referring to a single value like an integer number or a string. They can have read only or read/write attribute. Many properties have read only attribute. Only those properties which allows access to features which are also accessible to a user of the standalone program have read/write attribute. Properties normally can be read at any time, they can, however, be written only if the corresponding object is accessible. This is only true if the pfEnabled and pfVisible property of the object is TRUE and if this is true for all related parent objects (See the chapter "Availability of I/F objects" for details). As a property cannot have a return value there is no way to know whether the property could be set correctly. To inform the client program of any problem during access of a property an error is raised in such cases. It is the clients programs responsibility to correctly handle the errors raised by the component or to avoid such errors. An unhandled error will lead to a fatal problem in the client program. Normally the property has a valid range of values. Writing a value to a property which is out of range will also result in an error. A simple example of a property is the pfVisible property of most of the I/F classes which indicates whether the associated control is visible or not.

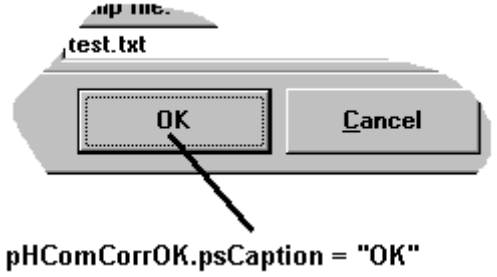
The I/F classes also contain methods. If the object is not enabled calling the method will also result in an error and the method is not executed. A simple example of a method is the pClick method of the HCommand object which can be executed when the corresponding command should be executed.
The I/F objects also raises events to inform the client program on changements of properties or similar information. A simple example of an event is the ChangeVisible event which is raised when the pfVisible property of the object changes. Some of the events are used by the HiPic/HPDTA to execute code. In this sense the HiPic/HPDTA is a client as well and it is not defined who will get the event first (An event is sent to all clients one after the other in an undefined order). As an example we will take the Click event of the HCommand object. This event is raised when the pClick method is executed by either the user or the client program. The HiPic/HPDTA will use the Click event to execute the associated command. When the client program gets the Click event the HiPic/HPDTA has already executed the associated command or it will do it after the client program get the event (This depends on the fact who gets the event first). As a consequence the program cannot rely on the fact that code which is executed with a certain event is already executed when the client program gets the event or not. In some cases there are several events in a series to get a well defined order.

HCommand object
The HCommand object is the simples I/F object. It has the following public properties / methods / events:

Event	ChangeEnabled()		Event which is raised when the pfEnabled property changes
Event	ChangeVisible()		Event which is raised when the pfVisible property changes
Event	Click()		Event which is raised when the pClick method is executed
Read only property	pcParent()	Object	Returns a reference to the parent object of this object. If this reference is Nothing this object is the topmost object within the object hierarchy (normally the HWindow object of the associated dialog)
Read only property	pfControlAvail()	Integer	Returns a value which specifies whether the object can be controlled by the client programmer (This is dependent on the pfVisible and pfEnabled properties and those of its parent)
Read only property	pfDlgHasUserIF()	Integer	Returns a value which specifies whether the associated dialog has a user I/F
Read only property	pfEnabled()	Integer	Returns a value which specifies whether the object is enabled
Read only property	pfVisible()	Integer	Returns a value which specifies whether the object is visible
Read only property	psName()	String	Returns the name of the object
Read/write property	psCaption()	String	Sets or returns the text which is used to label the associated control
Sub	pClick()		Method which raises the Click-Event and executes the associated command
Sub	pRegEvent(ByVal sEvent As String, ByVal fBasicEvent As Integer)		Should not be used by clients! (Registers an event to the error handler)

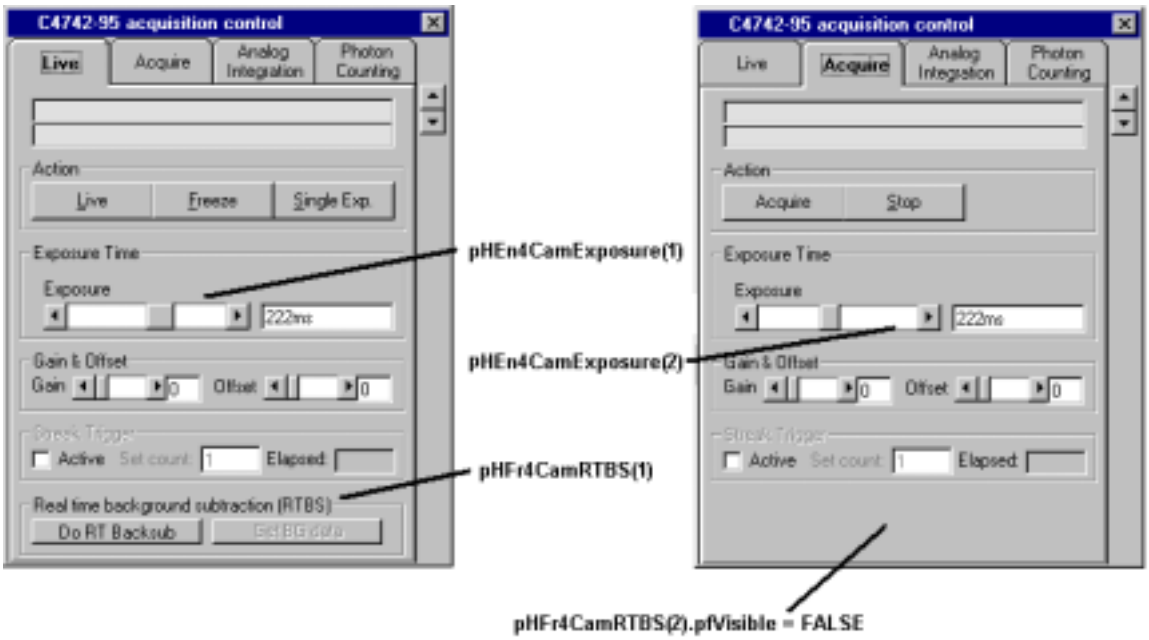
The HCommand has 3 events: ChangeEnabled, ChangeVisible and Click. The ChangeEnabled event is raised when the pfEnabled property changes. The ChangeVisible event is raised when the pfVisible property changes. The Click event is raised when the pClick method was called by either the HiPic/HPD-TA or the client program was called.
The HCommand has 6 read only properties: pcParent, pfControlAvail, pfDlgHasUserIF, pfEnabled, pfVisible and psName. The pfEnabled and pfVisible properties define whether the associated control is enabled or visible. The pfControlAvail property returns a value which specifies whether the object can be controlled by the client programmer (This is dependent on the pfVisible and pfEnabled properties and those of its parent). Use this property to make sure that you can write to read/write properties or execute methods of the object. The pfDlgHasUserIF property returns the UserIF property of the dialog on which the object is located. This may be of interest if you want to know whether there is already a user interface for the object you want to access (Sometimes it may be misleading or confusing if the client program changes values of control which are visible to the user). The pcParent property

returns a reference to the object where this object is embedded. As it is not known at compile time which object type will be the parent object the definition of pcParent is "As Object". The topmost object in the object hierarchy is the dialog window. In such case the pcParent propery is "Nothing".
Important: Do not confuse the pcParent property with the plHWndParent property of the HWindow object. Even if the dialog window is a set to be a child window of another window with the plHWndParent property the pcParent property of the dialog is still "Nothing". The reason of this is that a dialog can be placed as a child window to every other window where the window handle is know even if there is no HWindow object associated with this window. This is generally the case for windows the client program has created (A client program cannot create a HWindow object).
The psName property is the name under which the object is created in the HiPic/HPDTA. It is generally identical to the public name under which it is known to the client program with the exception that the first letter of the name is "m", whereas the first letter of the public object is "p" (The HMenu object to save an image has the psName property "mHMenMainSaveAs", but it is known as HAppHPDTA.pHMenMainSaveAs in the HPD-TA).
The HCommand object has one read/write property, the psCaption property. This property is used to label the associated pushbutton (see the screenshot below). The HiPic/HPD-TA sets this property to a reasonable value, however, the client program can change the property. As this property can be changed by the client you should not use it to identify the object. By using the write access of this property the client program can modify the naming of all labels. As one example you may wish to change the labeling "Acquire" to become "Get Image" on the camera dialog. Of course you have to take care that your label is fitting on the provided space on the command button.
The HCommand object has 4 methods: pClick, pRegEvent.
Only one should be used by the client programmer: the pClick method. It can be used to execute the associated command. As an example we take the pHImages.pHComOptOK object. By using the pHImages.pHComOptOK.pClick method we can execute the OK command within the Images options dialog.
The method pRegEvent is defined as public due to internal reasons and should not be used by clients.



HCommand4Array object and other arrays
A group of 4 HCommand objects are grouped into one object: a HCommand4Array object. All HCommand objects can be accessed by one single object reference. Access to an individual HCommand can be done by the pcItem(Index) property. An event raised by an individual HCommand object is repeated by the HCommand4Array with an index indicating the index of the HCommand which has raised the event. Grouping of such objects is done because every camera dialog uses 4 Tabs for the 4 acquisition modes (Live, Acquire, AnalogIntegration and PhotonCounting). The same object is placed once on every Tab, thus enabling 4 different parameter settings one for every acquisition mode. Also sometimes a control is not visible on some Tabs (see the screenshot below). Sometimes other objects are

grouped like in the object HWindow20Array (an array of 20 HWindow objects used for the images), HExternalDevices (three external device types), HDevPars (containing all device parameters), HImages (containing all Images).
The following description does no longer refer to such groups it only refers to the single object.



HMenu
The HMenu object is almost identical to the HCommand object with one single exception: it has an additional property pfChecked and an event ChangeChecked. The pfChecked property defines whether the menu item is checked. Whenever the property pfChecked changes the event ChangeChecked will be raised.

HCheck
Apart from properties/methods and events which already had been described the HCheck object has one additional property and one more event: the pfValue property and the ChangeValue event. The pfValue is a read/write property and can be TRUE or FALSE depending on the status of the associated checkbox. Whenever the pfValue property changes the ChangeValue event will be raised.

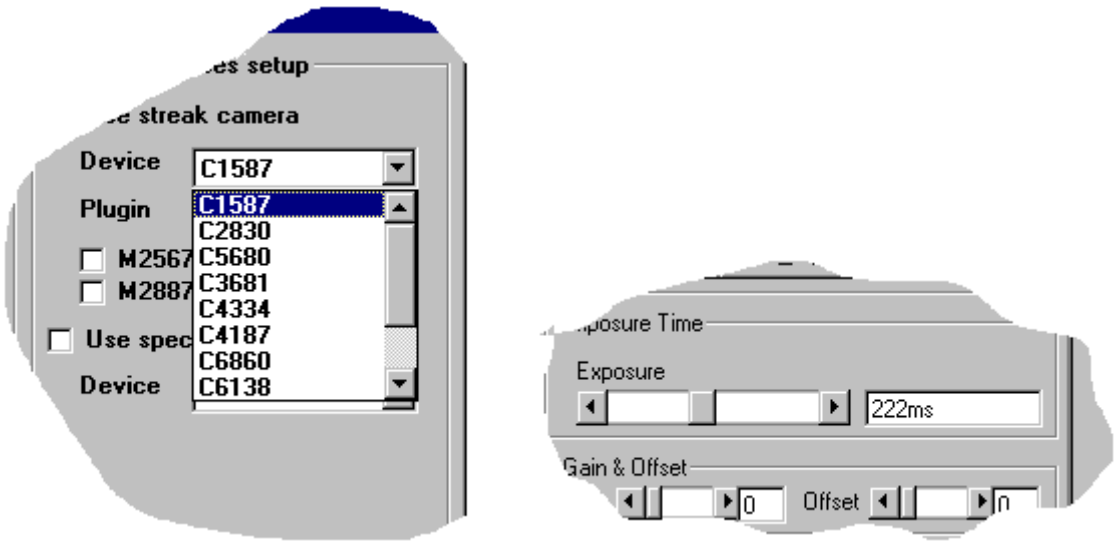
HFrame
The HFrame object is used to describe a frame used to group several controls. All properties/methods and events are already described.

HDisp
The HDisp object has one additional property and one more event: the psMessage property and the ChangeMessage event. The psMessage property is the string which will be displayed in the display box. When this string changes the ChangeMessage event will be raised.

HEditString
The HEditString object is used to enter a string value like file names or comments etc. It has the property psText describing the text which has been entered by the user. When this text changes the ChangeText event is raised.

HEditNumber
The HEditNumber is used to enter numbers. The number can have 4 different number types: integer (16bit), long (32bit), single (32bit) and double (64 bit). The number type is defined by the property piNumberType. There is a read/write property pvValue which returns the entered number. The data type of pvValue is Variant and can take any of the a.m. types. Suitable conversions are done with the characters the user is inputting. The property pvValue is limited by two values: pvMinValue and pvMaxValue. When the inputted number exceeds these limits an error is raised. When the limiting values are changed the event ChangeLimit is raised.

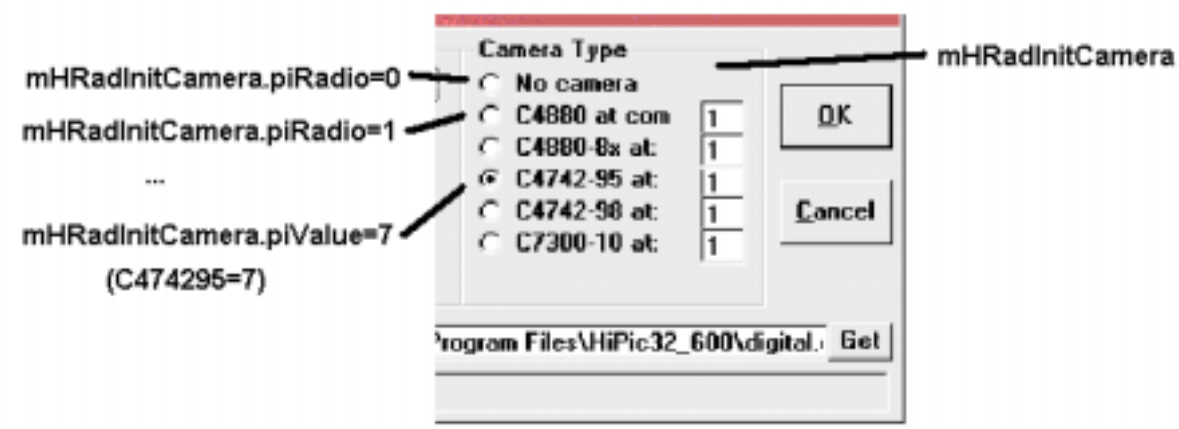
HEntry
The HEntry object represents a list of string entries. It can have two different "faces". It can consist of a combo-box or of slidebar in combination with a edit or display box (see screenshots below).



In the first case the Device can be selected out of a list represented by a combo-box, in the second case the Exposure time can be selected out of a list where the individual item can be selected with a slide bar or by entering the string into an edit box.
An important property is NoEntries which defines the number of entries in the list. The ChangeNoEntries event will be raised when the No of Entries changes. The currently selected string can be read or written by using the psKeyValue. Any Key value within the list can be read by using the psSetKeyValue(Index) property. The property piEntry can be used to set or get the index of the currently selected string item. Possible values for piEntry are in the range of [0 to NoEntries-1]. An additional string value which is associated to all entries is the psTag property (psSetTag(index) returns the psTag value of any index). This psTag property normally is used for internal purposes and should not be used by the client. The strings used in the list can have a special meaning. This meaning is defined in the property EntryType. There are 3 entry types: EntryTypeTime, EntryTypeString, EntryTypeNumber. If `EntryType=EntryTypeString` no special meaning is assigned to the string value. If `EntryType=EntryTypeNumber` the strings are representing numbers. If `EntryType=EntryTypeTime` the strings are representing time information. Normally they describe exposure times. If the property pfSearchNext is set to TRUE in combination with `EntryType=EntryTypeNumber` or `EntryType=EntryTypeTime` then the nearest value will be set. In the case of EntryType=EntryTypeTime the search algorithm additionally understands time information. In such cases 8050ms is smaller then 9s (the units m, s, ms, us, ns, ps can be used and are correctly detected).

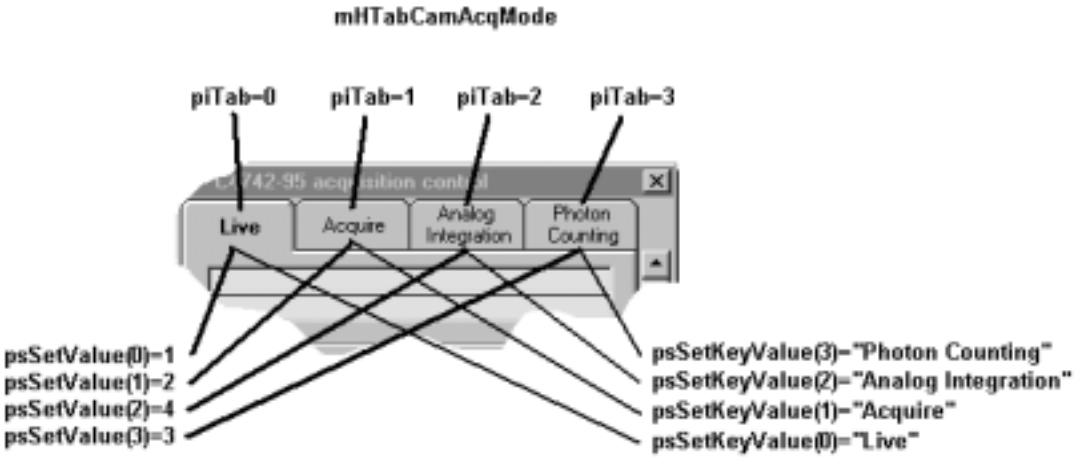
HRadios

The HRadios object represents a group of radio buttons. The HRadios object is similar to the HEntry object except that Every radio button can be disabled or set visible individually. There are the properties pfRadioEnabled(Index) and pfRadioVisible(Index) to define whether an individual radiobutton is enabled or visible. The events ChangeRadioEnabled(iRadio) and ChangeRadioVisible(iRadio) are raised when the corresponding properties are changed. Every radiobutton can have an associated string value and a numeric value. These are the properties psKeyValue and the piValue. Any string and numerical values can be get by the psSetKeyValue(iRadio) and the piSetValue(iRadio). The property piRadio sets or gets the index of the selected radio button. Possible values are in the range of [0 to NoEntries-1]. The psKeyValue is used to label the individual radio buttons. Sometimes one needs to know the piRadio value from a given Value (psSetValue(Index)), therefore a function piRadioFromValue(iValue) is provided for convenience.



HTab

The HTab object is very similar to the HRadios object the usage of course is a little bit different. It is used when controls are arranged on a tabbed control.
In opposite to the HRadios object the properties containing information about the visibility or enabled status of a single tab are pfTabVisible(iTab) and pfTabEnabled(iTab). The respective events are ChangeTabVisible and ChangeTabEnabled. is a string value and a numerical value associated to each Tab (psKeyValue and piValue, additionally there are psSetKeyValue(iTab) and psSetValue(iTab)). The property piTab sets or gets the index of the selected Tab. Possible values are in the range of [0 to NoEntries-1]. Sometimes one needs to know the piTab value from a given Value (psSetValue(Index)), therefore a function piTabFromValue(iValue) is provided for convenience.



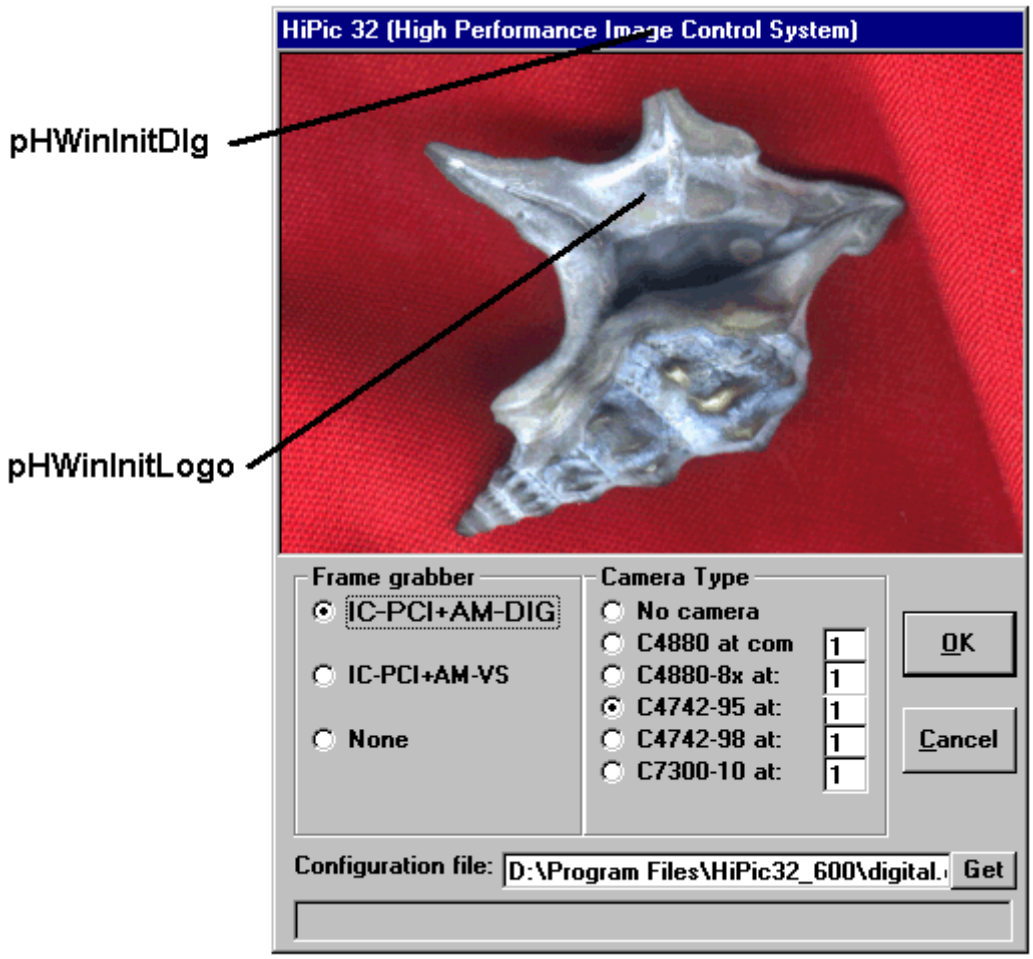
HSingleTab
Associated with every individual Tab on a Tab control is a HSingleTab object. This is made mainly to provide a chain of parent relationships up to the dialog window. The user normally doesn't need to access such HSingleTab object. All properties/methods and events used in this object are already described.

HProgress
The HProgress object is used to display the progress of an operation (lets say an exposure). It has two main property: the psMessage and the ftPercent. psMessage is the text which is displayed on the progress bar, ftPercent is the progress of the operation in percent. When the percent display changes the ChangePercent event will raised.

HWindow
There are two applications for a HWindow object:
a) The first (and more important) is to represent a dialog window.
b) The second is to represent a picture are located on a dialog.
Most of functions only refer to the case where the HWindow refers to a dialog. There is no other way to distinguish the two cases apart from the name. The name for a dialog always ends with the characters "dlg". As an example lets take the introduction screen (see the screenshot below). There are two HWindow objects: the pHWinInitDlg and the pHWinInitLogo. The pHWinInitDlg is the dialog window, the pHWinInitLogo is a picture box where the logo (intro-bitmap) is displayed. The following properties/methods or events does not refer to HWindow objects associated with a picturebox, do not use them in combination with such a type of window: BeforeWindowVisible, NowWindowVisible,psTag, piSetWindowPosition, ActivateWindow, KeyDownWindow, CloseWindow, pSetFocus, pClose, piClientAreaXOffset, piClientAreaYOffset, pfHideForm, pfUserIF, plHWndParentWindow.
The visibility of a picturebox type HWindow is same as for all other control it is defined by the flag pfVisible. The default for the property pfVisible is FALSE, so it must be set to visible explicitly.
The visibility of a dialog type HWindow is dependent on the properties pfVisible, pfUserIF and pfHideForm. This was already described in detail in the chapter "Showing and hiding dialogs". The following description will describe all properties/methods or events which are not jet described together with another object.

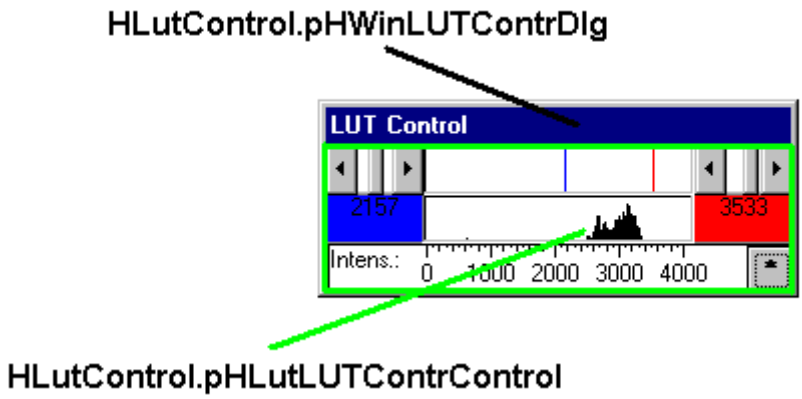
The pfWindowSizable property returns a value which defines whether the associated window is sizeable. The client program should not attempt to change the windows size if this property is FALSE.
Due to the borders and the window-caption the starting coordinates of the window and its client area are different. This difference is described by the properties piClientAreaXOffset and piClientAreaYOffset.
The plHWnd property returns the associated windows window handle. If there is no associated window this handle is zero. If you want to access the window by its window handle you first have to set the properties pfVisible and pfUserIF to TRUE, then the window will be created, the you can use the plHWnd property to get the Window handle. If the window is a picturebox type window you have to set the properties pfVisible and pfUserIF of the dialog window to TRUE.
Every dialog belongs to an object (the dialog shown below belongs to the HInitTa object for example). This object is the owner of this dialog and any access to the dialog must be passed to this object (To access the configuration file name you have to access the pHInitTa.pHEstInitConfiFile interface object). The property psOwnerName contains information about the owning object (In our case it would contain "HInitTa").
The HiPic/HPD-TA has a global option "Remember window positions". To realize this feature, every dialog has a property pfRestoreWindowPos which tells the dialog to save and restore its window position when it is opened or closed. Normally every main object (like the HC474295 object) also has such a property. If the property of a main object is set, this object sets the pfRestoreWindowPos property of all its dialogs (In the case of the HC474295 object this is the camera dialog and camera setup dialog) and of all other dependent objects (If you set the property pfRestoreWindowPos of the HAppHiPic object the properties of all dependent objects like the camera object etc. is set as well).
The plColor property sets or returns the color of the associated window.
Every dialog can be placed as a child window within another window. To do this the plHWndParentWindow property has to be set to the window handle of the of the parent window.
Notes:
1) Do not read or write the plHWndParentWindow for picturebox type windows. This feature is only intended for dialog windows.
2) The pcParent property does not return a correct HWindow reference if you use the plHWndParentWindow to place the dialog into another window. A dialog windows pcParent property always returns "Nothing".
The psPicture property sets or returns the file name of a bitmap image which is displayed on this window. The following formats are supported: BMP, ICO, CUR, RLE, WMF, EMF, GIF, JPG.
The psTag property sets or returns the Tag property of this object. The Tag property is used to communicate with the associated window (internal use only). Do not write to this property.
The piGetWindowPosition function can be used to get the current window position and window state. The coordinates returned by this function are always measured in pixels and its origin is the origin of the parent window (if no parent window is defined the screen is the origin). The piSetWindowPosition function can be used to set the current window position and window state. Do not change the window size for windows where the property pfWindowSizable is FALSE. Do not use this function for picturebox type windows.
The pActivateMainWindow method can be used to activate the topmost window in the parent relationship. If a dialog window is set as a child window of another window with the plHwndParent property this parent (or its parent) window is activated. The pClose method closes the associated window.

The pSetFocus method sets the focus to the associated window. It works only if the associated window is visible on screen and is a dialog window.
Several events are raised from a HWindow object:
The ActivateWindow event is raised when the associated window is activated. The BeforeWindowVisible event which is raised before the associated Window is shown on the screen. This event can be used to access the window before it is visible, but after it has been created. the NowWindowVisible event is raised after the associated Window is shown on the screen. This event can be used to get an information about the moment when the window is really displayed. The CloseWindow event is raised when the associated window will be closed. It can be used to save data associated with this window. This KeyDownWindow is raised when a key event as sent to the window. The MouseDown, MouseMove and MouseUp events are raised when the respective mouse events are sent to the window. The ResizeWindow event is raised whenever the window is resized.



HLut
The HLut object is used to enter LUT values. It is placed on the LUT control dialog (see screenshot below) which belongs to the HLutControl object. The HLut object has the following properties/methods and events: The piLUTSize property returns the size of the LUT. Possible values are: LUTSize8, LutSize10, LutSize12, LUTSize16, LUTSize812, LUTSize14 and LUTSize16x which stands for 8, 10, 12, 14, 16, 16x (which stands for extended 16 bit) and 8/12 bit (8/12 is a LUT for layered superimposed images). The corresponding size of the LUT is 256, 1024, 4096, 16384, 32768, 32768 (with bit shift) and 4096. The plMinCursorDiff property returns the minimum difference between the upper and lower value (cursor). The HLut object always forces these values to have the specified minimum difference. The plLowerValue and plUpperValue properties define the lower and upper LUT values. The

pSetAuto method executes the Auto LUT function. There are several events which informs the client program about changes on the HLut object. The AutoLUT event is raised when the AutoLUT function is executed. Setting the Cancel parameter to TRUE can suppress the Auto LUT function. The LimitsChanged is raised when the piLUTSize property changes. The LowerCursorChanged event is raised when the property plLowerValue changes. The UpperCursorChanged event is raised when the property pUpperValue changes. The CursorsChanged event is raised when either of the properties plLowerValue or plUpperValue changes. The Resize event is raised when the LUT control is resized. In the case of LUTSize16x a bit shift is performed prior to LUT operation. The property plMult defines the multiplication factor related to this shift. Example: If the shift is 3 the multiplication factor is 8.



HImageArea

The HImageArea object describes the area where an image is displayed. First of all let us look on the numerical values describing the image:
The image has a size of iDX(width) x iDY (height) pixels and an offset of iX (horizontal offset) and iY (vertical offset). For the ease of handling a structure of type "are" contains these numbers. This area is called the source area of the image (areSource).
The first screenshot shows the full image. In our case the size is:
areSource.iX=0
areSource.iY=0
areSource.iDX=512
areSource.iDY=512
A part of the image is displayed on the screen (see the second screen shot). The part of the image which is displayed in the second image is marked in the first image as an ROI. The part of the image which should be displayed is described in the structure areImgToDisplay. In our case this area is:
areImgToDisplay.iX=69
areImgToDisplay.iY=160
areImgToDisplay.iDX=388
areImgToDisplay.iDY=463
The part which should be displayed is displayed on a picturebox with the size X x Y (see the area marked in red). The size of this area on screen is described in a structure pntPicImage. In our case pntPicImage is:
pntPicImage.X=388
pntPicImage.Y=463
The whole image including horizontal and vertical slide bars is displayed in an area pntWindow. In our case pntWindow is larger than pntPicImage due to the slidebars which require some space. pntPicImage.X is identical to areImgToDisplay.iDX and pntPicImage.Y

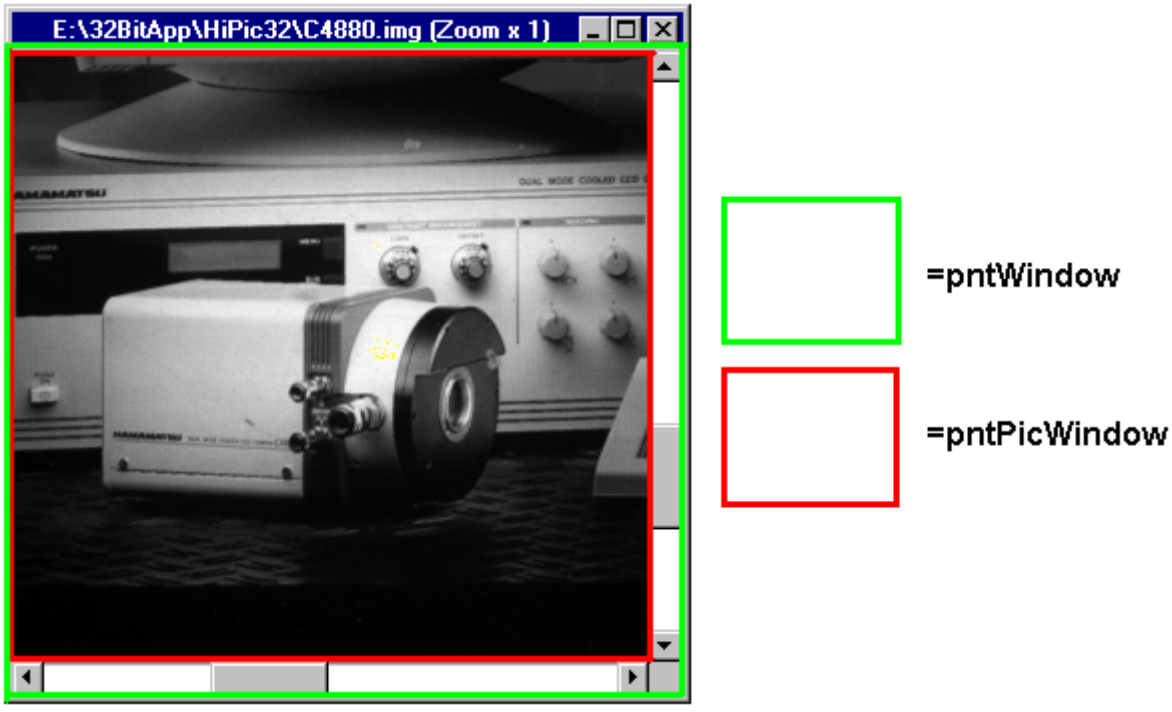
is identical to areImgToDisplay.iDY because the zoom factor is 1. If the zoom factor would be 2 pntPicImage.X would be 776 and pntPicImage.Y would be 962.

Now lets describe the properties you can get or set within the HImageArea object.
The method pGetpntWindow returns the size of pntWindow. The method pGetpntPicWindow returns the size of pntPicWindow. The method pGetareImgToDisplay returns the size of areImgToDisplay (The size of the image areSource is a property of the HImage object). The pGetftpntCenter and pLetftpntCenter functions allow to get and define the zooming center point (The zooming center is the point where the user clicks with the mouse when zooming). The property pftZoom sets or gets the current zooming factor.

Additionally there are several functions related to an ROI. An ROI has a start and an end point and a type. There are three different types of ROI: Point, Line and Rectangle. The structure areROI describes the location of the ROI in image coordinates. There is a corresponding structure wndROI which describes the ROI in screen coordinates (relative to the origin of PicImage). The function pGetareROI and pLetareROI allow to get and set the ROI in image coordinates. The function pGetwndROI returns the screen coordinates which are related to the ROI. The piROIType property sets or returns the ROI Type. Possible values are defined in the enumeration ROIType. The property piPRFDirection sets or returns the quick profile direction. Possible values are HORINTEGRPROFILE and VERINTEGRPROFILE. The plHWndImageArea returns the window handle of the window where the image is displayed (marked in red in the screenshot below). The method pEnlargeROI enlarges the ROI in the specified direction. Possible values are defined in the enumeration EnlargeDirection

The ChangeCenter event is raised when the center position changes. The ChangeZoom event is raised when the Zoning factor changes (The meaning of ChangeEnabled and ChangeVisible is identical to those of previous objects).

E:\32BitApp\HiPic32\C4880.img (Zoom x 1)
Panasonic
C4880

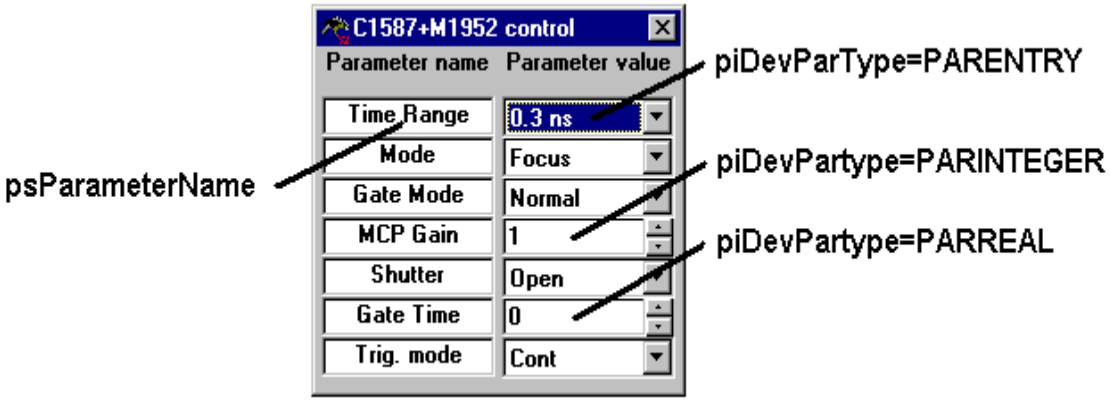

E:\32BitApp\HiPic32\C4880.img (Zoom x 1)
=pntWindow
=pntPicWindow

HDevPar

The HDevPar object describes a parameter of an external device. This objects are only used in combination with the HPD-TA. The HDevPar object is similar to a HEntryNumber type but it has some more properties. The HDevPar object communicates with the HPDTA32.DLL which provides access to all external devices. Exchange of parameter values is always done by one floating point number (double type). Every parameter is identified by a constant, the iParameter value. The parameters are arranged on the status/control box from top to down with an index iControl starting from 0. There are different types of Parameters: Entry, Integer and Real. An entry type parameter consists of a list of strings (like the HEntry object), an integer type can have only integer values and a real type parameter can have any real type value. The range of values is limited by a lower and upper limit (like with the HEditNumber entry) and eventually there is a step width defining the interval between two possible values. For any numeric type of parameters (no entry types) the value can be visualized using an exponent. The value of the parameter is then "value displayed in the edit box" * 10^{exponent}. This exponent can be set or inquired with the piExponent property. In the case of an entry type parameter the parameter value is used to index the string within the list. The entries are numbered starting from 1. There are different modes of a parameter depending whether the parameter value can be inquired or controlled or both.

The following is a description of the individual properties/methods and events:

The piDevParType property returns the ParameterType of the parameter. Possible values are described in the enumeration DevParType. The piMode property returns a value which defines the ParameterMode of the parameter. Possible values are described in the enumeration DevParMode. The piMode property is dependent on the two properties pfControlAvailable and pfStatusAvailable which describe whether the parameter can be inquired or controlled. Do not mix up the pfControlAvailable property which refers to the hardware with the pfControlAvail property which exists for every interface object and tells the client programmer that the interface object is currently available because it is visible and enabled etc. The piParameter returns a value identifying the parameter. Possible values of known parameters are described in the enumeration StreakParameter, SpectrographParameter and DelayParameter. The psParameterName property returns the parameter name. The pvValue propery returns the value of the parameter. For entry type parameter pvValue propery is the index within the list (starting from 1). Possible values pvValue are in the range between pvMinValue and pvMaxValue. The property pdbStep returns a value which defines the step width of two neighboring values. If it is zero all values between pvMinValue and pvMaxValue are possible. The property psKeyValue specifies the parameter value in string format. For entry type parameters the property psSetKeyValue(Index) can be used to get an individual entry out of the list.



Using the main objects in HiPic and HPD-TA

This part of the documentation explains how to use the main objects in the HiPic/HPD-TA. Every main object has one or more dialog for the communication with the user. Every dialog contains interface objects for the communication with the user. As these interface objects already have been described in detail in the previous chapter they will not be described here. This chapter will only describe properties/methods and events not related to the interface objects. Basically these features will give the client programmer extended possibilities compared to the user of the standalone program.

Startup (HInitHi and HInitTa)
The only object which can be created by a client program is the start object (HInitHi for the HiPic and HInitTa for the HPD-TA). The start is done in two steps: Showing the Intro screen and starting the program.
In detail the following things have to be done:
1) Creating an instance of HInitHi or HInitTa
The HInitHi or HInitTa has a multi use property, this means that several instances of the object can be created within the same instance of the component. By using this feature one can get access to an instance of the HiPic which is already running. The property piStartStatus gives information about the status of the program. There are three possibilities:

piStartStatus=StartStatusNone	The program is not running
piStartStatus=StartStatusInit	The program shows the Init dialog
piStartStatus=StartStatusRunning	The program has been started

If piStartStatus=StartStatusNone the Init dialog can be showed by calling the piInit method.
If piStartStatus=StartStatusInit the property pStartHInitHi or pStartHInitTa returns a reference to the HInitHi or HInitTa object which currently shows the Init dialog. Executing the pHComOK.pClick method will start the program.
If piStartStatus=StartStatusRunning the property pStartHAppHiPic or pStartHAppHPDTA returns a reference to the already existing Application object.

2) Showing the Init dialog
The piInit method is used to show the Init dialog and has the following parameters:

```
Public Function piInit(ByVal sIniFile As String, ByVal fInitUserIF As
Integer, ByVal fApplUserIF As Integer, ByVal fNoDialogs As Integer) As
Integer
```

The sIniFile parameter specifies the *.INI file which should be used for this session. The fInitUserIF defines whether the Init dialog should be started with or without UserIF. The fApplUserIF defines whether the whole application should be started with or without UserIF (This is more convenient then setting all pfUserIF properties of all dialogs manually). The fNoDialogs defines whether the program should suppress message boxes.

3.) Starting the program
Executing the pHComOK.pClick method of the HInitHi or HInitTa object will start the program. The client program has to wait until pfInitStatus>=0. If pfInitStatus=InitStatusInitialized the pHAppHiPic or pHAppHPDTA property will contain a reference to the application. If pfInitStatus>0 the creation of the Application has been failed due to some reason and the program has to end.

Apart from these properties the HInitHi or HInitTa object contain other information about the application. The piApplicationType property returns the application type. Possible values are defined in the enumeration ApplicationType. The psAppDate property returns the applications date. The psAppTitle and the psAppTitleLong properties return a short and a long version of the title. The psIniFile returns the *.ini file name. The psSoftwareVersion property returns the software version. The pGetLicenceKeys method returns information about all license keys which could be found. There are several events which are raised during startup informing the client program about details of the startup. These are: ApplicationCreated, ErrorDueToLicence, ErrorDuringInit, InitCancel, NoUserIFAllowed, MsgBoxResult and Message. The MsgBox event is similar to those raised by the HMsgBox object. It is however not possible to get a reference to the HMsgBox object during a certain period startup.

The application object (HAppHiPic and HAppHPDTA)

Shutdown
Before we talk about how to use the application object we should explain how to end the program properly.
The piEndProg method is used to shutdown the application. Before you use this method you should free all object references (except the reference to the main application object of course). After this you should free the reference to the application object.
As there is a possibility that the user or another client shuts down the application you should free all object references all well. Otherwise the component cannot shutdown correctly and will remain in the computers memory.
To be sure that you will be informed about the shutdown create a WithEvents object reference to the main dialogs HWindow object and release all object references in the Close event of this object like in the following code:

```
Private WithEvents mHWinMainDlg As HWindow
Private Sub mHWinMainDlg_CloseWindow(Cancel As Integer, ByVal Mode As Integer)
    'Free all object references here!
End Sub
```

Usage of the application object
Besides the interface objects the application objects returns object references to other main objects used in the HiPic or HPD-TA (See the chapter object model and main objects for details). Again beside these references the application object contains the following properties/methods and events:
The piCameraTyp, piFrameGrabber, psConfigFile and piAcquisitionModule properties return information about the camera and the frame grabber. The piApplicationType, psAppDate, psApplicationString, psAppTitle, psAppTitleLong, psSoftwareVersion properties returns

information about the application. The piOperatingSystem property return the operating system. The psApplicationDirectory property returns the application directory. The psWindowsDirectory property returns Windows directory where the component is running. The psGetStatus function returns the status of the application object. The psErrorString returns a comprehensive explanation of an error code returned by a function call.

The HImages and the HImage object

The HImages, HImage and the HImage area object are working together closely and they can be regarded as one unit. The HImages object is the topmost object of these objects in the object model.

The HImages object
The HImages objects is owning the ROI I/F and the options dialog, every HImage object owns one image display dialog window. Access to the individual HImage objects is done by the HImages.pcItem(Index) property. The pcItemCurrent property returns the currently selected HImage object. The piCurrentImageValid property tells the client whether there is any valid active image. Access to all Image-Dialog and Image-Picture (the area where the image is displayed) windows is possible through the pHWinImgDlg20Array and pHWinImgPicture20Array objects. Management of image numbers is also done within the HImage object. Special image numbers have special meaning. There is piIndexAcquireImg, piIndexCurrentImg, piIndexLiveImg, piIndexLoadImg, piIndexSequenceImg, piIndexAcqImg (see the object catalog for details of these index numbers). The HImages objects contains functions for loading and saving images with or without file dialog and with or without asynchronous execution. The available functions and properties related to file/save are:
pfSaveImage, pfLoadImage (ordinary save/load)
pAsyncSaveImage pAsyncLoadImage (asynchronous save/load)
pfLoadImageNoShow (load without show)
piGetLoadImageFile, piGetSaveImageFile (Opens file dialog to get file names)
piDefaultImageType, pfDisplayToTIFFMode, piSeqMode, psDefaultScalingDirectory, (special properties)
The pPrint methods prints an image. The pChangeActiveImage changes the active image. There are several events which are raised to inform the client about specific situations. These events are: CloseSequenceImage, CurrentImageChanged, FileNameChanged, ImageDataChanged, ImagePainted, Message and RemoveImage.

The HImage object
All properties of the HImage object are referring only to the specified image. There are some methods which exist for both HImages and HImage object. If you use the methods of the HImage object you can make sure that it is applied on this individual image. The methods which are similar for the HImage and HImage object are: pAsyncLoadImage, pAsyncSaveImage, pPrint, pfSaveImage, pfLoadImage. Access to the HImageArea object and the involved windows is possible by the pHImaImgArea, pHWinImgDlg and pHWinImgPicture properties.
There are several properties and methods describing the image, these are: piBytesPerPixel, pGetareSource, piReside, piDataType, psFileName, pfImageValid. Most important properties are piBytesPerPixel, pGetareSource which describes the size of the image.
Concerning storage and data access the following properties and methods are available: pfDataSaved, plDataBytes, plDataHandle, plDisplayBytes, plDisplayHandle, pfDumpDataToFile, pfGetPointIntensity, pfSetPointIntensity.

To get image data the functions pfGetDisplayData, pfGetImageData1Byte, pfGetImageData2Byte and pfGetImageData4Byte can be used.
To set image data the functions pfSetImageData1Byte, pfSetImageData2Byte and pfSetImageData4Byte can be used. After changing image data, the image size or the status the function pfUpdateImageAfterDataChange should be used.
Concerning display the following functions are available: pfDisplayOnVGA, pDisplayImage, pShowImage, pfCreateEmptyImage. The functions pDisplayImage, pShowImage are not needed normally. pDisplayImage can be used to display an image when its content has been changed. The property pfDisplayDataValid has to be set to FALSE first otherwise the display function does not recalculate the display data.
The function pfMemGetPRFData can be used to extract a profile out of the image. The piGetMaxBit returns the topmost bit set in the image.
For inquiry and setting of status information the follow functions can be used: psStatus, psStatusGetSection, psStatusGetString, pStatusWriteSection, pStatusWriteString.
To get or set scaling data the client can use the functions of the object pisdImage (see the describition about the HImageScalingData object):
The properties piMaxScalingX, pGetScalingXInfo, pGetScalingYInfo, pSetScaling, psGetPointScaling, psGetPointScalingXY, psGetWidthScaling ar kept for compatibility with older version and can alternatively be used to get or set scaling properties.

The HExternalDevices and the HExternalDevice object

The HExternalDevices object

The HExternalDevices object contains six dialogs:

- The Auto Delay dialog (pHWinAutoDelDlg)
- The device control options dialog (pHWinOptionsDlg)
- The prcessing dialog (pHWinProcDlg)
- The device control setup (pHWinSetupDlg)
- The trigger timing diagram (pHWinTimingDlg)
- The trigger setup dialog (HWinTrigSetDlg)

Apart from various I/F objects there are several properties indication the external devices setup, external devices options and trigger setup condition:

Read only property	pfDualTimeBaseExist()	Integer	Returns a flag which indicates whether a dual time base extender is used as a streak camera option
Read only property	pfStreakUseDTBE()	Integer	Tells the camera object that a Dual time base extender is currently used for trigger handshake
Read only property	piA6538Connected()	Integer	Returns a flag which defines whether an A6538 is connected to the trigger status adapter
Read only property	piCounterBoardInstalled()	Integer	Returns a flag which defines whether a Dt2819 board is installed
Read only property	piCounterBoardIOBase()	Integer	Returns the DT2819 board IO base adress
Read only property	piGPIBInstalled()	Integer	Returns a flag which defines whether a GP-IB board is installed in the system
Read only property	piGPIBIOBase()	Integer	Returns the GP-IB board IO base address
Read only property	piPostTriggerTime()	Integer	Returns the Post Trigger Time
Read only property	piStreakTriggerMethod()	Integer	Returns the currently selected streak trigger method
Read only property	piTDStatusCableC	Integer	Returns a flag which defines whether a status cable is

	onnected()		connected to the streak camera
Read only property	plTriggerDelay()	Long	Returns the trigger delay
Read/write property	pfAutoMCP()	Integer	Returns a flag which defines whether Auto MCP is active
Read/write property	pfAutoStreakShutter()	Integer	Returns a flag which defines whether Auto Streak Shutter is active
Read/write property	pfDoStatusRegularly()	Integer	Sets or returns a flag which defines whether the status should inquired regularly (Typically once per second)

The properties pfAutoMCP and pfAutoStreakShutter returns information about the AutoMCP and AutoDelay device control option. If the property pfDoStatusRegularly is TRUE the HPD-TA inquires the status from the external devices regularly. If it is FALSE the client programmer can update the status by using the pDoGetStatus method.

There are several properties which are declared as public due to internal reasons but which should not be used by the client programmer. They start with an r (even though they are public) and are: rfIsDualTimeBaseExtender, rTA_IdentOptionName, rTA_IdentOptionNo, rTA_IdentParEntryName, rTA_IdentParEntryNo, rTA_IdentPluginName, rTA_IdentPluginNo, rTA_Setup, rTA_SetupDevice, rTA_SetupOption, rTA_SetupPlugin, rGetDeviceList, rGetDualTimeBaseInfo, rTA_SetupCloseSession.
The pcItem(Index) property returns a reference to an individual HExternalDevice object. Index can be in the range piMinIndex and piMaxIndex. Possible values for Index are defined in the enumeration DeviceType.
Several events are raised to inform the client about changements with the external devices. These are DevicesBuilt, TASetupExecuted and Message.
The pEmergencyOff method can be used to switch off the streak cameras MCP, close the shutter and set the spectrographs slitwidth to zero (if available).
The pShowDialogs shows all external devices dialogs.
There is a set of functions where the client programmer can directly access GP-IB devices. In the case a device driver is provided by the HPDTA it is better and more easy to use the HExternalDevice object for communication to the device. In the case of a special or user provided device the client programmer can access these devices directly. The programmer should first check whether the HPD-TA has already got a handle to the GP-IB board. If this is not the case he can open a GP-IB session by himself. These are the corresponding functions:

Function	plIEEE488_AttachDevice(ByVal iBoardNo As Long, ByVal iIEEE488Address As Long, ByVal lpszOutEOS As String, ByVal lpszInEOS As String, ByVal fDeviceIsSlow As Long, ByVal uiTimeout As Long)	Long	Low Level IEEE 488 function: Attaches a device
Function	plIEEE488_Close(ByVal iBoardNo As Long)	Long	Low Level IEEE 488 function: Closes the GP-IB session
Function	plIEEE488_DetachDevice(ByVal lDeviceHandle As Long)	Long	Low Level IEEE 488 function: Detaches a device
Function	plIEEE488_IsSRQPending(ByVal iBoardNo As Long)	Long	Low Level IEEE 488 function: Inquires pending SRQs
Function	plIEEE488_Open(ByVal iIOAddress As Long, ByVal iMyIEEE488Address As Long, ByVal fController As Long)	Long	Low Level IEEE 488 function: Opens a GP-IB session
Function	plIEEE488_ParallelPoll	Long	Low Level IEEE 488 function: Executes a parallel poll

	(ByVal lDeviceHandle As Long, ByVal lpszPollStatus As String)		
Function	plIEEE488_ReadBinary (ByVal lDeviceHandle As Long, bByte()	Byte, ByVal iCount As Long) As Long	Low Level IEEE 488 function: Reads GP-IB data binary
Function	plIEEE488_ReadString (ByVal lDeviceHandle As Long, lpszString As String, ByVal iCount As Long)	Long	Low Level IEEE 488 function: Reads GP-IB data as string
Function	plIEEE488_SendBinary (ByVal lDeviceHandle As Long, lpData As Byte, ByVal iCount As Long, ByVal fEOI As Long)	Long	Low Level IEEE 488 function: Sends GB-IB data binary
Function	plIEEE488_SendString(ByVal lDeviceHandle As Long, ByVal lpszString As String)	Long	Low Level IEEE 488 function: Sends GB-IB data as string
Function	plIEEE488_SerialPoll(ByVal lDeviceHandle As Long, ByVal lpszPollStatus As String)	Long	Low Level IEEE 488 function: Executes serial poll
Function	plIEEE488_SetGotoLocal(ByVal lDeviceHandle As Long)	Long	Low Level IEEE 488 function: Sends Goto Local command
Function	plIEEE488_SetRemoteEnable(ByVal iBoardNo As Long)	Long	Low Level IEEE 488 function: Sends Set remote enable command
Function	plIEEE488_TA_IdentGPIBBoard()	Long	Returns the board handle in the case the HPDTA has already opened a GB-IB session
Sub	pTA_HandleSRQs()		Gives the device DLLs the opportunity to handle SRQs (if any)

The HExternalDevice object

The HExternalDevice object has several properties indicating the current device condition. The properties psDeviceName and psPluginName return the devices and its plugin name. The psDevType property returns a string describing the device type. The properties piNoOfOptions, piOptionInstalled(index) and psOptionName(index). Index can be in the range of 0 to piNoOfOptions-1.
The properties pfControlAvaileable and pfStatusAvaileable tells the client whether control and status information is available for the device (Every parameter has an individual pfControlAvaileable and pfStatusAvaileable property as well).
The properties pfStatusOutExist and pfTriggerConnectorExist refer only to streak cameras and gives an information about the status out port and the cabling of the streak trigger connectors for a single shot configuration (All these informations have to be specified in the device control setup first of course).
The pHWinDevDlg returns access to the device status/control dialog window. The piGPIBCableConnected and piGPIBBase properties returns information whether the device is connected by GP-IB and to which base address. The piUseDevice property informs the client whether the device is used at all, the piDeviceIndex and piPluginIndex returns the entry numbers in the device and plugin list provided by the HPDTA32.DLL.
The HDevPars property returns a reference to an array containing all parameters of this device. The piNoOfControl property informs the user about the number of controls arranged on the devices status/control box.
Though all parameters can be inquired and controlled by the HDevPars object some additional functions are provided for convenience. These are

pfGetParameter, pfGetParameterByString, pfParameter2Control, pfParameterName2Control, pfSetParameter, pfSetParameterByString, pfSetParameterToMaximum, pfSetParameterToMinimum.
To understand the differences of these functions it is useful to know that iParameter is referring to an integer constant associated with the parameter (like TimeRange=3) , iControl is the index how the Parameters are arranged on the status control box (the topmost parameter has iControl=0 etc.) and the postfix ByString always refers to the parameter or value name as a string (like "Time Range" for the parameter name or "1 ns" for the parameter value). Whenever the device controls setup is done all object references of the streak parameters are rebuilt. Therefore the client has to release the object references to these objects. The moment when this has to be done is indicated by the ReleaseParameters event.

HLUTControl

The HLUTControl object has two dialogs: the LUT Control dialog (pHWinLUTContrDlg) and the LUT Parameters dialog (pHWinLUTParDlg). Apart from the interface objects there is a function psGetStatus which returns the current LUT status.

HSystemScaling

The HSystemScaling object has five dialogs: the scaling setup dialog (pHWinScalSetDlg), the scaling editor dialog (pHWinScalEditDlg), the create polynomial dialog (pHWinCreaPolyDlg), the time scaling dialog (pHWinTimScalDlg, HPD-TA only) and the spectrograph scaling dialog (pHWinSpecScalDlg, HPD-TA only). The pssdSystem object can be used to get or set system scaling properties (see the description of the HSystemScalingData object).

HProfileScalingData, HImageScalingData, HSytemScalingData and HPRFParametersData

There are three locations where scaling data is active: System Scaling, Image Scaling and Profile scaling.
The current version does not jet allow to access profiles by automation (which may be possible with future versions). System and image scaling data, however, can be accessed by automation. The system and image scaling objects consits of two other objects: one HProfileScalingData object for each coordinate (X and Y). To understand how to access profile scaling the HProfileScalingData is most important.

HProfileScalingData

Internally the profile scaling can be of the type linear or table. The HProfileScalingData object contains functions and properties which refers directly to this type. There are, however, functions which hide this fact to the user and are therefore more convenient to use. Additionally there are functions to set different types of scaling.

The basic functions to get scaling are:

Read only property	psUnit()	String	Returns the unit
Read only property	pftValue(iEntry As Integer, Optional sError As Variant)	Single	Returns the value of a given index
Function	pfGetArrayOfValues(ftArray()	Single, ByVal iEntries As Integer,	Returns an array of values

		Optional sError As Variant) As Integer	
Function	pfValueExist(iEntry As Integer)	Integer	Returns a flag whether or not the value at the specified location exist
Function	piMaxEntries()	Integer	Returns the maximum number of entries

Functions to get a pixel index as a function of the scaled value

Function	pftProfileLocationFT(fScaled As Integer, ft As Single)	Single	Returns the profile location (floating point)
Sub	pLocationToIndex(ftLocation As Single, ftIndex As Single, fDisplayScaled As Integer)		Returns the index corresponding to the given location

Functions so set scaling

Function	pfSetLinearData(ftScale As Single, sUnit As String, Optional sError As Variant)	Integer	Sets linear scaling
Function	pfSetTableData(ftTable()	Single, iEntries As Integer, sUnit As String, Optional sError As Variant, Optional varCheckFromChannel As Variant) As Integer	Sets table scaling
Function	pfCreateDiffPoynomial(iOrder As Integer, ftCoeff()	Single, iNrValidChannels As Integer, sUnit As String, Optional sError As Variant) As Integer	Create differential polynomial
Function	pfCreateIntegPoynomial(iOrder As Integer, ftCoeff()	Single, iNrValidChannels As Integer, sUnit As String, Optional sError As Variant) As Integer	Create integral polynomial
Function	pfReadScalingTable(sFileName As String, sOffset As String, fCheck As Integer, sUnit As String, Optional sError As Variant, Optional varCheckFromChannel As Variant)	Integer	Reads a scaling table from file
Function	pfSetScaling(psd As HProfileScalingData, Optional sError As Variant)		Sets the scaling data of this object identical to the data of the specified object
Function	pSaveScalingFile(sFile As String, Optional sError As Variant)	Integer	Saves the scaling data to file

Internal or old type functions (only needed in special cases)

Read only property	pftScale()	Single	Returns the scaling factor
Read only property	piType()	Integer	Returns the type (linear/table)
Read only property	psOrigin()	String	Returns the scaling origin
Read only property	psScalingFile()	String	Returns the scaling file name
Function	gfCheckSwapScaling( )	Integer	Swaps the scaling and profile data in case the scaling is descending
Function	psScalingInfo()	String	Returns general scaling info
Sub	pGetMinAndMax(iLo wer As Integer, iUpper As Integer, ftMin As Single, ftMax As Single)		gets minimum and maximum value

HSequence and HJitter

The HSequence objects has two dialogs: the sequence dialog (pHWinSeqDlg) and the options dialog (pHWinOptDlg).
The pfSequenceIsInRAM property informs the client whether the sequence is loaded to RAM. There are several events informing the client about specific situations: EndSequence, ErrorEndSequence, Message, SeqSingleAcqEnded, StartSequence. The StartSequence and EndSequence events can be used to set some parameters at the start and end position of a sequence (like opening a shutter or switching on microscope illumination). In case of an error the ErrorEndSequence event will come instead. The SeqSingleAcqEnded event can be used to execute some code after an individual acquisition has been ended. It is the client programmers responsability not to take too long time within such code otherwise the acquisition will be delayed. The StartStreakTrigger is an event which has been introduced intermediately to make a specific sequence in combination with streak triggering possible. It will be removed as soon as the standard program is able to handle this sequence. Similar to the images object there are some functions for saving, loading, starting and correcting sequences. These are: pAsyncLoadSequence, pAsyncSaveSequence, pCorrectSequence, pLoadSequence, pRemoveSequence, pSaveSequence, pAsyncSeqStartAcquisition.

The HJitter object is not public, it is operated from a user I/F on the HSequence dialog so every interface object can be found in the HSequence object.

The Camera objects (HC4880, HC488080, HACam, HC474295, HC474298, HC7300, HC800010, HC800020, HFlatPanel)

Every camera object has at least two dialogs: the camera dialog (pHWinCamDlg) and the setup dialog (pHWinSetupDlg). Some cameras also have a subarray dialog like the HC4880 object (pHWinSubDlg). All camera dialogs are designed in a very similar way. According to the 4 acquisition modes (Live, Acquire, AnalogIntegration and PhotonCounting) the controls of every acquisition mode are located on an individual Tab of a tab control (pHTabCamAcqMode). To access an interface object one has to select the correct tab first. As the same interface object can appear on all or several tabs the interface objects are grouped as a so called 4Array. As an example we take the exposure time entry object (pHEn4CamExposure). Access to the individual item is done by the pcItem(Index) property where index stands for the acquisition mode. To access the exposure time of live mode we simple can write `pHEn4CamExposure.pcItem(Live).psKeyValue="20ms"`. There are interface objects with the same name on the camera dialog of different cameras (like the Live pushbutton array pHCo4CamLive). Therefore we can use a general function to access the

camera pHobjCamera to access the same control of any camera (like `pHobjCamera.pHEn4CamExposure.pcItem(Live).psKeyValue="20ms"`). Apart from the interface objects there are some functions and properties common to all camera objects. These are:
PiDatTypeForLive, piDatTypeForAcquire, piDatTypeForAI and piDatTypeForPC which returns the data types in the different acquisition modes: The property psCameraName returns the camera name. The psGetStatus property returns the current camera status. The method pStopAcquisition can be used to stop the currently running acquisition.

The HAcq object

The HAcq object is used to acquire images from any camera. It interacts very closely with the camera objects. The camera object is always the master object whereas the HAcq object has a possibility to interact with the camera object in a very restricted way by so called callback functions. The sequence object can also be the master of the HAcq object (These informations are only given for a general understanding of the program, they are not necessary to operate the HiPic/HPDTA component).
The HAcq object is the owner of the following dialogs: pHWinBackDlg(backsub dialog, used to indicate a backsub correction under progress) , pHWinCorrDlg (correction dialog), pHWinCurvcorrDlg (curvature correction dialog), pHWinOptDlg (options dialog).
There are several read-only properties to inform the client programmer about the status of the object. These are: pfAcqPending (an acquisition is currently pending), pfGrbIs4MType (the grabber is a 4 MB type), pfProcPending (a processing operation is pending), pfProcStopped (the processing operation has been stopped), pftDefaultExposureTime (the default exposure time), piAmMod (the acquisition module).
Some properties have read/write attribute. These are pfDisableAutoAction, pfDisableAutoInquiry, pfRestoreWindowPos, pfUserIF. pfDisableAutoAction disables the AutoAction (like auto-shutter or Auto-MCP) before an acquisition (HPD-TA only). If pfDisableAutoInquiry is active the program does not prompt the user before auto actions. These two properties are used by the sequence mode, so be careful if you want to use them together with sequence mode.
There are several functions in combination with acquisition processes: pAsyncLive, pAsyncAcquire, pAsyncAnalogIntegration, pAsyncPhotonCounting and pStopAcquisition. The client programmer does not need these functions, he can use the interface objects on the camera dialogs to initiate the same acquisition processes.
Several events can be used by the client programmer to be informed about actions happening within the object. These are: CameraTemperature, ChangeAcqMode, EndAcquisition, EndBacksub, Message, SeqSingleAcqEnded, StartAcquisition, StartBacksub. StartAcquisition and EndAcquisition indicates the starting and ending of an acquisition. StartBachksub and EndBacksub indicates the starting and ending of a backsub acquisition. Don't use SeqSingleAcqEnded of this object, use the same event which is raised from the HSequence object instead.

Utility objects (HLicence, HAsyncCommand, HMsgBox, HError)

There are some small utility objects which can be useful from time to time.
The HLicence object provides information about the licenses which are contained in the connected dongle (Use of the HiPic/HPDTA remote component without any license is impossible). The function pfGetCustomKey returns an information whether a custom made key is located on the dongle (special customers only).

The HAsyncCommand manages asynchronous commands. These are commands which returns control to the client program immediately but the execution of this command starts only in this moment. By using some properties of the HAsyncCommand the client programmer can get information about whether and what asynchronous command is currently running.

The HMsgBox object manages the output of message boxes. Under certain circumstances the program should not output message boxes. In such cases the text to be outputted will be contained in an event raised by the HMsgBox object.
The HError object handles error trapping and should not be used by the client program.

Synchronous and asynchronous commands

As long as a method is executed the client program cannot continue. If the method takes very long time there will be a timeout and the client will show ugly and not easy to understand messages. Most commands, however, are executed within short time and no problem occurs. Some methods, however, take a long (like long integration) or even undetermined time (like the LIVE mode) which result in such timeout. Also outputting message boxes lead to asituation that the component does not return to the client program for an undefined time (if the user does not respond). For such cases the HAsyncCommand object provides the means of starting commands in an asynchronous way. Once the method has been invoked the command starts but it returns control to the client program immediately. The real executing code starts after this moment (it is executed in a timer event started within the AsyncCommand). The event BeforeCommand will be raised before the asynchronous command has been executed, the event AfterCommand will be raised after the command has been executed. The command itself is triggered by the DoCommand event. Note: Don't use this event within the client program because you cannot know whether you will get this event before or after the execution of the command. The pfAsyncCommandActive property informs the client programmer that an asynchronous command is active, the psActiveAsyncCommand property tells which one. While an asynchronous command is active any normal command can be executed, but you cannot start a second asynchronous command. Asynchronous commands are:

- HAcq.pAsyncLive, HAcq.pAsyncAcquire, HAcq.pAsyncAnalogIntegration, HAcq.pAsyncPhotonCounting (These commands are executed from the corresponding Menu or Pushbutton objects on the main dialog, like pHComMainLive).
- HSequence.pAsyncSaveSequence, HImages.pAsyncSaveImage.
- HImage.pAsyncLoadImage and HImage.pAsyncSaveImage (The HImages.pAsyncLoadImage and HImages.pAsyncSaveImage methods are using these methods).
- HImages.pAsyncLoadImage and HImages.pAsyncSaveImage.
- The save and load Menu or Pushbutton objects on the main dialog are using the HSequence.pAsyncSaveSequence, HImages.pAsyncSaveImage or HImages.pAsyncLoadImage and HImages.pAsyncSaveImage methods.
- HInitHi.pAsyncStartProg and HInitTa.pAsyncStartProg (The pHComInitOK uses this method).
- The "Start Acquisition" pushbutton uses the HSequence.pAsyncSeqStartAcquisition function.

The string sCommand contains the name of the calling command. The following is a list of all AsyncCommands and their names (normally objects name and function name):

Function	sCommand
pAsyncLive	HAcq_mLive
PAsyncAcquire	HAcq_mAcquire
PasyncAnalogIntegration	HAcq_mAnalogIntegration
PAsyncPhotonCounting	HAcq_mPhotonCounting
pAsyncLoadImage	"HImages" + Format$(Index) + "pAsyncLoadImage"
pAsyncSaveImage	"HImages" + Format$(Index) + "pAsyncSaveImage"
pAsyncSaveSequence	HSequence_pSaveSequence
pAsyncLoadSequence	HSequence_pLoadSequence
pAsyncStartProg	HInitTa_mStartProg or HInitHi_mStartProg
mAsyncLoadImageOrSeq	HApplication_mLoadImageOrSeq
PasyncSeqStartAcquisition	HSequence_pAsyncSeqStartAcquisition

The HAsyncCommand returns control to the client program, however, it normally needs all or almost all of the processor time. You should keep in mind that only one part can run at a time: either the client program or the component. During an AsyncCommand normally three different things happens within the component:

1) Execution of the normal code of this command
2) Raising events
3) DoEvents to handle other events

If the client program executes another event like clicking on a pushbutton (e.g. Auto LUT or freeze) this can be done only when the component executes DoEvents. As long as the client event remains in this event execution of the components code does not proceed. This mean that you should not stay for a long time in such events. Also you should not execute a program path which waits for some event (Even calling DoEvents is not useful in such case). So the general rule for acting while an dAsyncCommand is running is: Start the AsyncCommand, Exit the function, then react on events, but do not wait in any loop. If you want to wait for some condition you can use a timer which should do nothing than checking the condition and exit again.
In most cases (like starting the LIVE mode) it is sufficient to start the AsyncCommand (like mobjHCamera.pHCo4CamLive.pcItem(1).pClick) exit the function immediately and stop the function with another command (call mobjHCamera.pHCo4CamFreeze.pcItem(1).pClick from your own pushbutton).

Message boxes

There are some cases where it is not useful or not possible to output message boxes. These cases are:
1) No user is sitting in front of the computer. The is the case if a component runs on a remote computer which is controlled by another computer.
2) An ordinary component function is called. If the user does not react on the message box the component function does not return to the client program which will after some time output a timeout message saying that the component request could not be finished. (This effect is quite common if you use standard Microsoft programs like Excel. In our case, however we do not like such messages because normally the client program want to hide the fact that another program is running at all).

According to the pfNoDialogs property of the MsgBox object component outputs Message boxes ot not. If no message box is outputted the HMsgBox raises an event instead (MsgBox).

If the client program wants to output a message box inside this MsgBox-Event it should distinguish whether the event has been raised due to an action of the component (either issued by the components user I/F or by an asyncronous command, in this case it is allowed to wait inifinitely) or by an action of the client program (For this purpose the samples uses the pfClientEvent property internally).

Configuring a remote COM-client and COM-server environment

This chapter is a summary report about the attempt to setup HPD-TA V.6.0 as a remote server (written in Visual Basic 6.0) application using different remote client applications written in Visual C++ 6.0 and Visual Basic 6.0.
The operating system platforms used for the server application were Win98, WindowsNT4 and Windows2000.
We also tried Window95 as server platform. In this case we found that COM communication basically worked, however we found that in case the Visual Basic server component was created using the option „WithEvents“ we got the error message „COM automation error“. Up to now we couldn’t find a solution for this problem.

Target operating system platform issues

General Topics on Windows 95 and Windows 98 Platforms

- Windows 9x has no intrinsic support for distributed COM applications. In order to enable distributed COM on these platforms the DCOM support has to be installed separately using the upgrade utility files DCOM95.EXE or DCOM98.EXE as available from Microsoft.
- When using Win9x as the client platform no additional considerations have to be taken.
- When using Win9x as server platform it has to be considered that DCOM for Windows 9x is not capable of automatically launching the COM server EXE file. Launching of the COM server application has to be done either manually or by including the EXE file into the autostart folder.
- In addition DCOM95 V.1.0 is not capable of automatically launching the remote procedure call service (rpcss.exe). In this case the rpcss.exe has to be included into the autostart folder of Win95 or Win95 must be upgraded to DCOM95 V.1.1
- DCOM has to be enabled using the DCOMCNFG.EXE utility.

General Topics on Windows NT4 and Windows 2000 Platforms

- Windows NT4 and Windows 2000 have an intrinsic DCOM support. These operating systems do not need additional upgrades.
- DCOM has to be enabled using the DCOMCNFG.EXE utility.

Networking issues

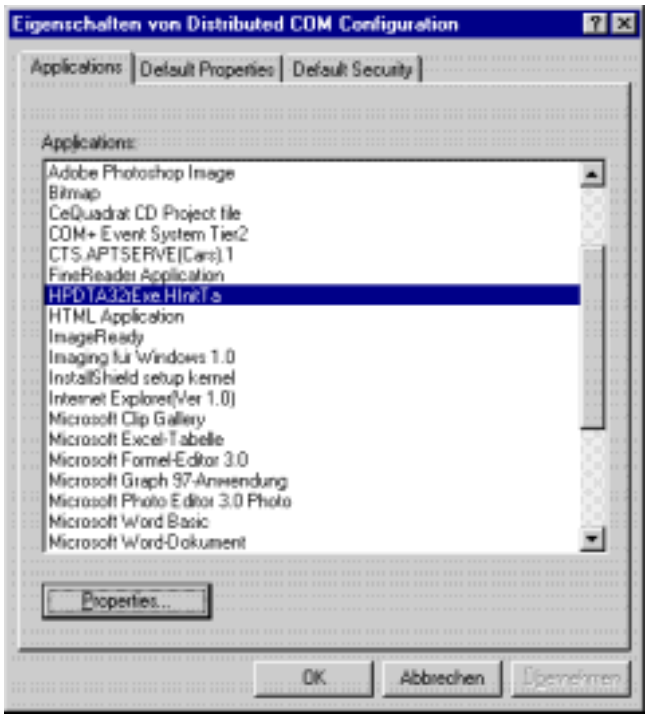
- The client computer and the server computer are both connected by a network.
- Both computers are part of the same network domain.
- The preferred network protocol is TCP/IP but also other protocols should be possible according to the COM specification.

Configuring the COM-Server

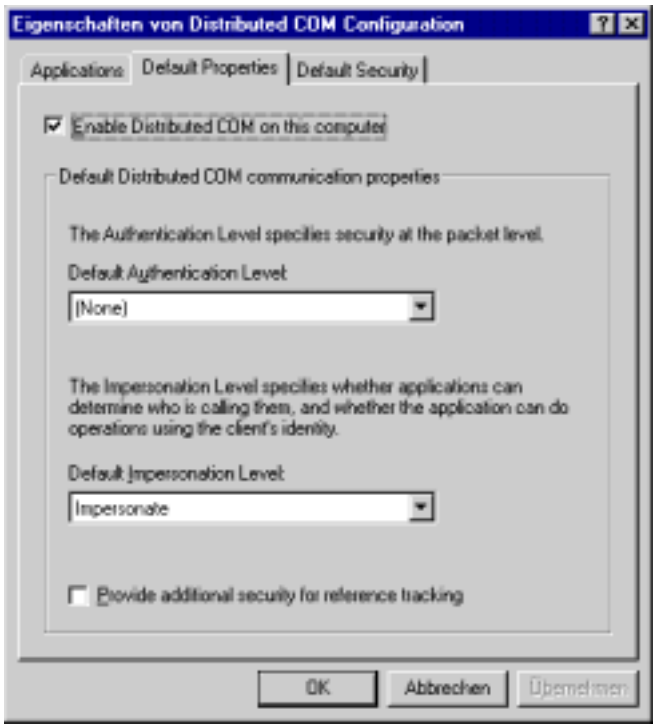
The COM server application has to be configured on the computer where it is located. This can either be done by manually modifying the registry or – more convenient – by using the DCOMCNFG.EXE utility that is part of DCOM. This utility is located in the \Windwos\System32 (on Win9x platforms) or in the \WinNT\System32 directory (on WinNT and Windows2000 platforms).
It is assumed that the server component has been successfully registered on the server-computer .

a) Configuring a COM-Server on Windows 95 and Windows 98 Platforms

When DCOMCNFG.EXE is launched it pops up with a dialog as displayed below. On the „Application“ TAB page you will find a list of all COM server applications that are registered on this machine.

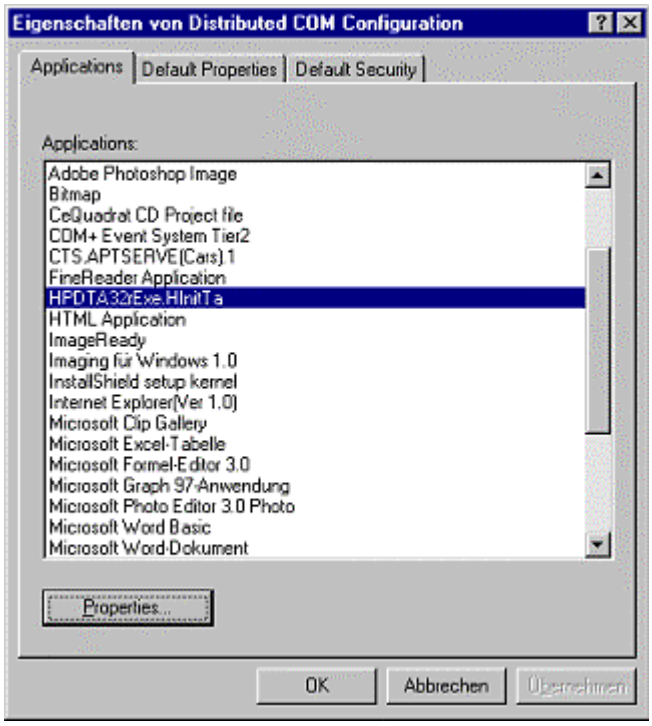


First select the TAB „Default Properties“. Ensure that the item „Enable Distributed COM on this computer“ is checked. As „Default Authentication Level“ select „(None)“ and as „Default Impersonation Level“ select „Impersonate“. Uncheck the item „Provide additional security for reference tracking“.

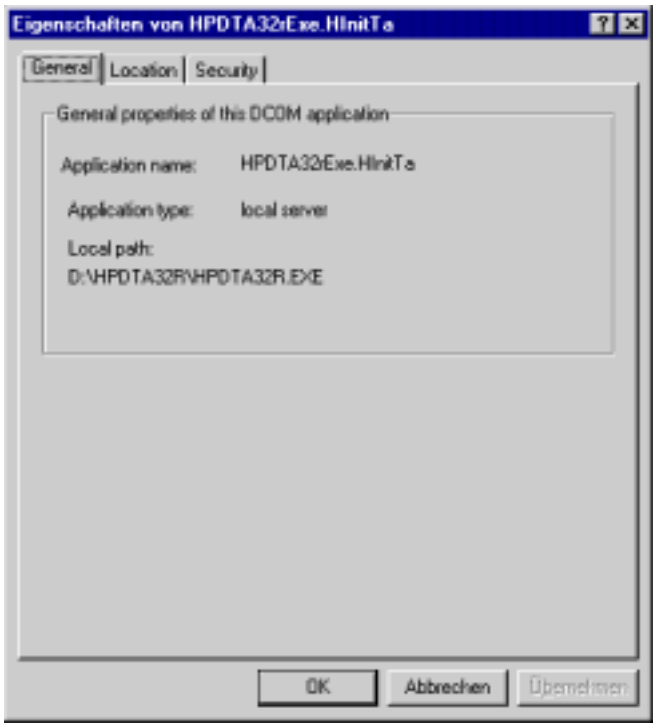


Then go to TAB page Default security and place a checkmark on item „Enable remote connection

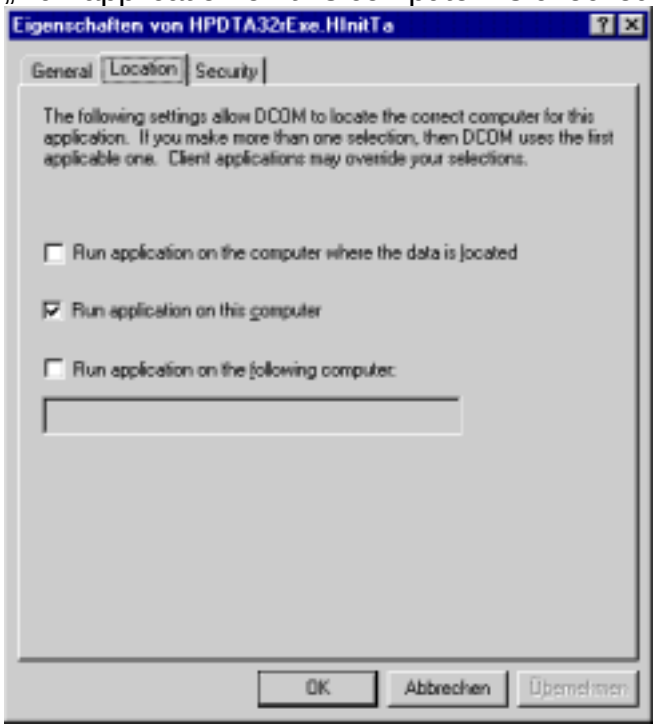
If you need to change specific properties of the COM server application, activate the „Application“ TAB page and from the „Applications“ list highlight the module you are interested in and click the „Properties“ button afterwards.



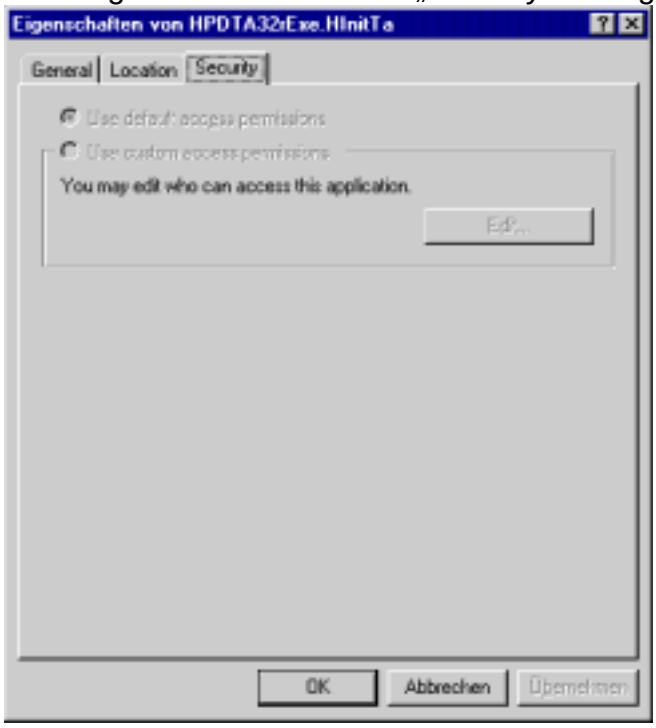
On the first page of the module properties dialog you can view some general information about the selected module



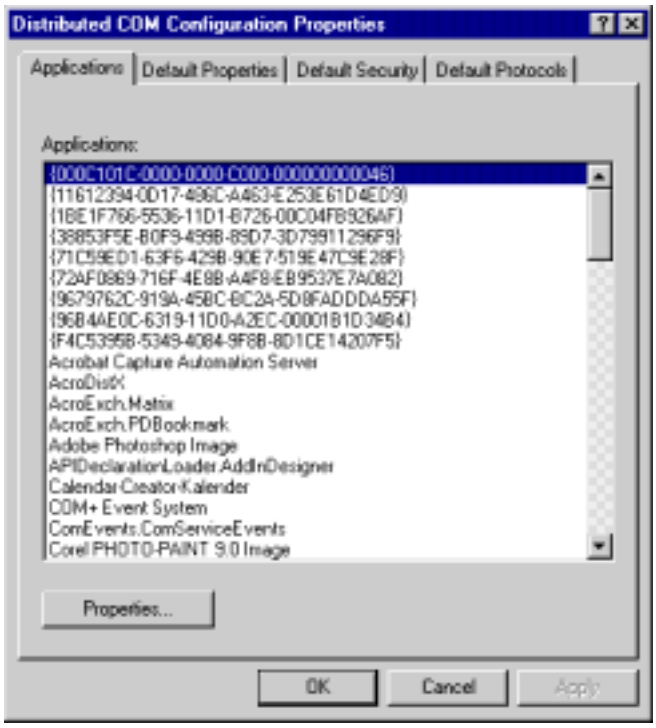
Go to the „Location“ page and verify that the item
„Run application on this computer“ is checked. Uncheck all other items.



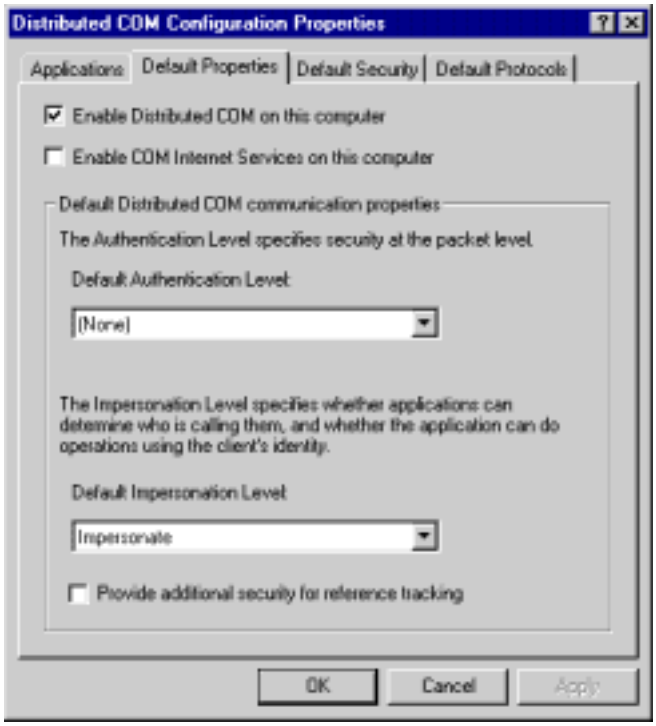
Because of the limited security capabilities on Win9x platforms there is nothing to configure on the module‘s „Security“ settings



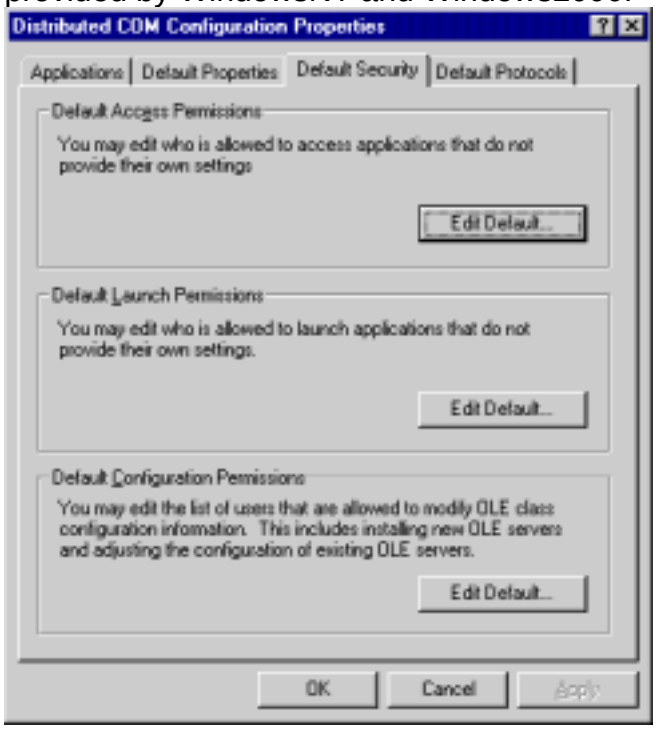
When DCOMCNFG.EXE is launched it pops up with a dialog window as displayed below. On the „Applications“ TAB-page you can find a list of all COM server modules that are installed on the computer.



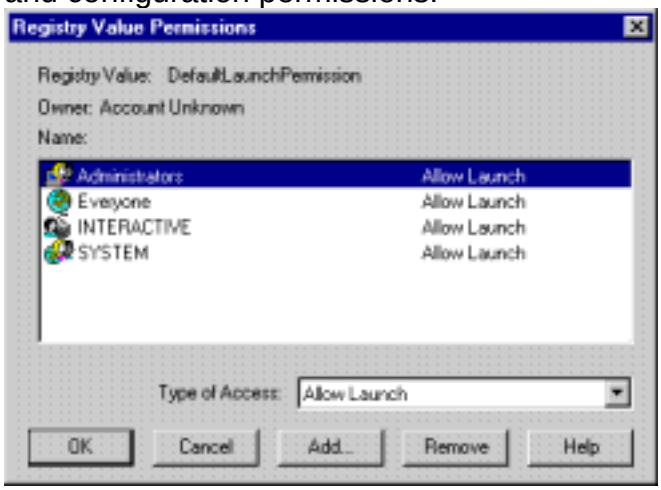
First select the TAB „Default Properties“. Ensure that the item „Enable Distributed COM on this computer“ is checked. As „Default Authentication Level“ select „(None)“ and as „Default Impersonation Level“ select „Impersonate“. Uncheck the items „Enable COM Internet Services on this computer“ and „Provide additional security for reference tracking“.



On the „Default Security“ TAB-page you can configure the security options provided by WindowsNT and Windows2000.

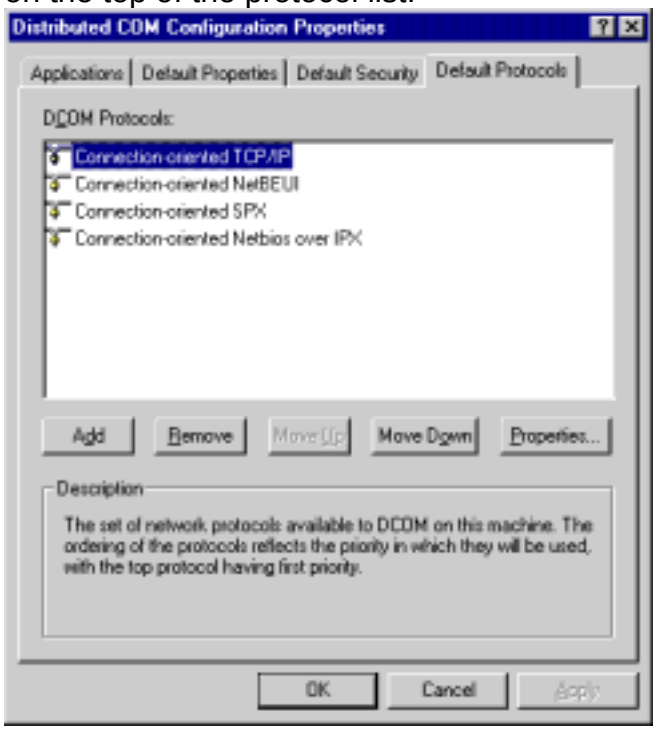


Press the „Edit Default ...“ button to change the settings for access, launching and configuration permissions.

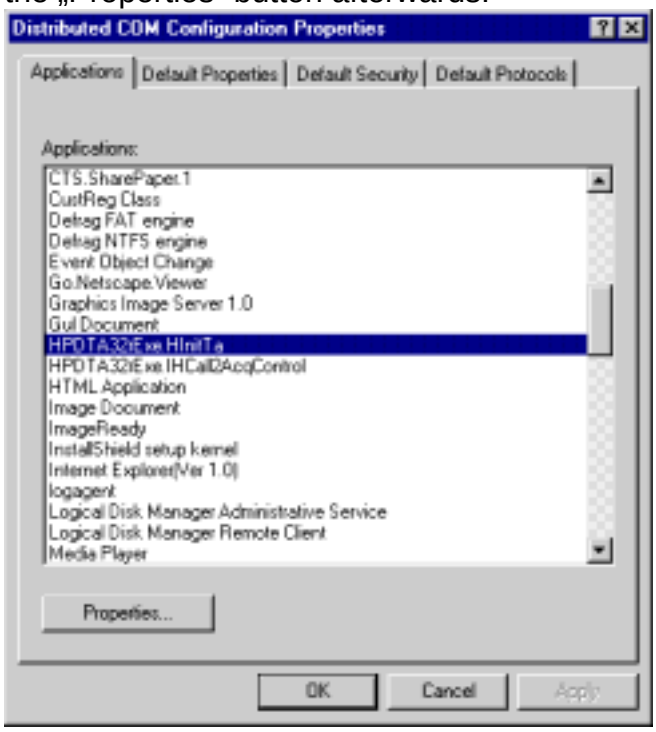


In case you do not include the user-group „Everyone“ you need either add every user that should have access to a server component to the Windows user lists of a matching using group or you can change the individual component settings as described below.

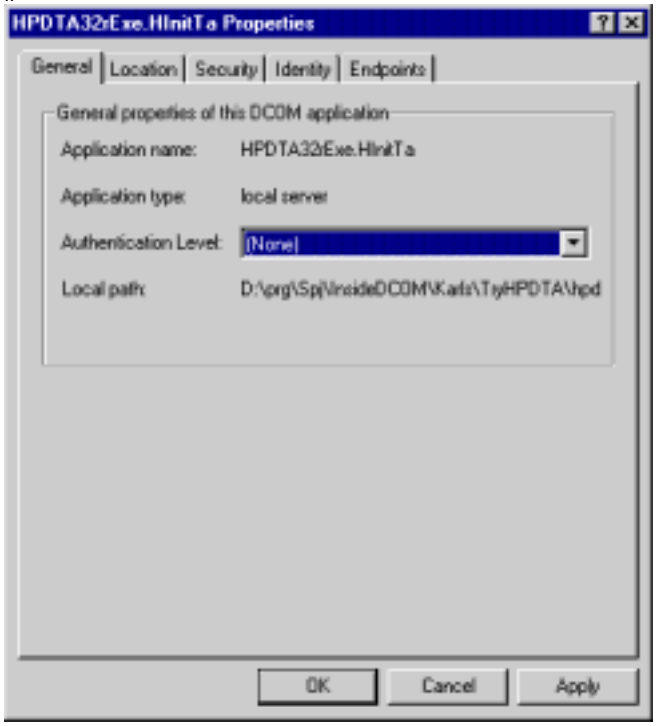
On the „Default Protocols“ TAB-page you can specify the network protocols that can be used to access the server component and the order in which they are used. In any case you should take care that TCP/IP is installed and placed on the top of the protocol list.



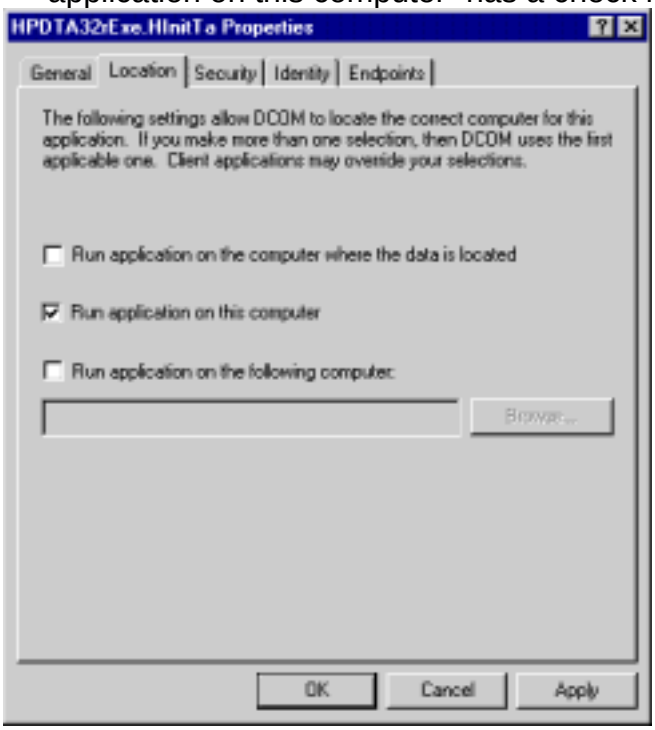
If you want to view or change the individual settings of a certain COM-server, go to the „Applications“ TAB-page and select the desired component. Press the „Properties“ button afterwards.



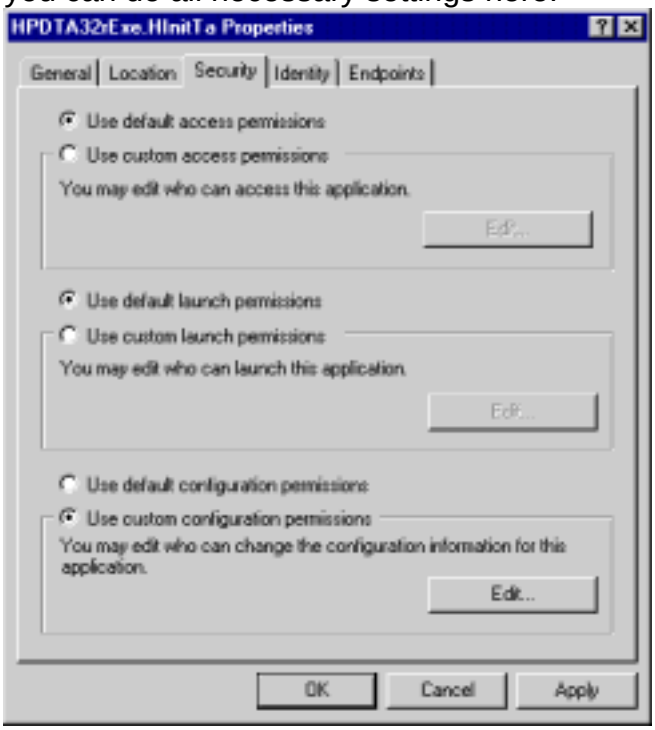
On the „General“ page you can view some general information related to the selected module. Additionally you can select an individual „Authentication Level“. I our case make sure that „(None)“ is selected.



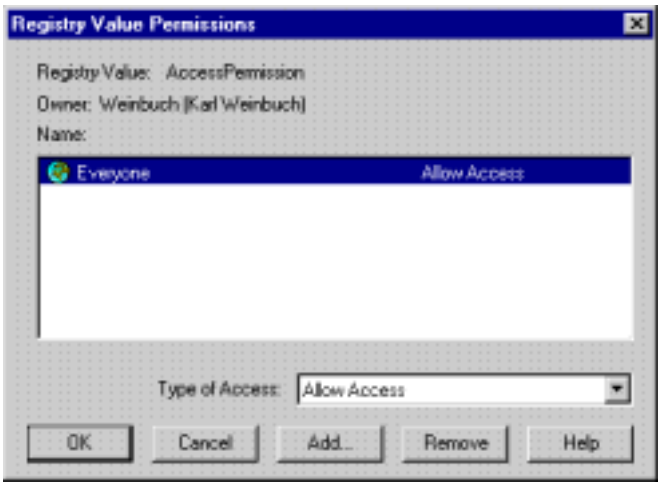
On the „Location“ page make sure that item „Run application on this computer“ has a check mark. Uncheck all other items.



On the security page you can configure the access, launch and configuration permissions individually for the selected component. This is very useful if you do not want to apply the default permissions. In case you do not grant permissions to the group „Everyone“ as the general default setting but you want to grant permissions to the group „Everyone“ for this specific component you can do all necessary settings here.

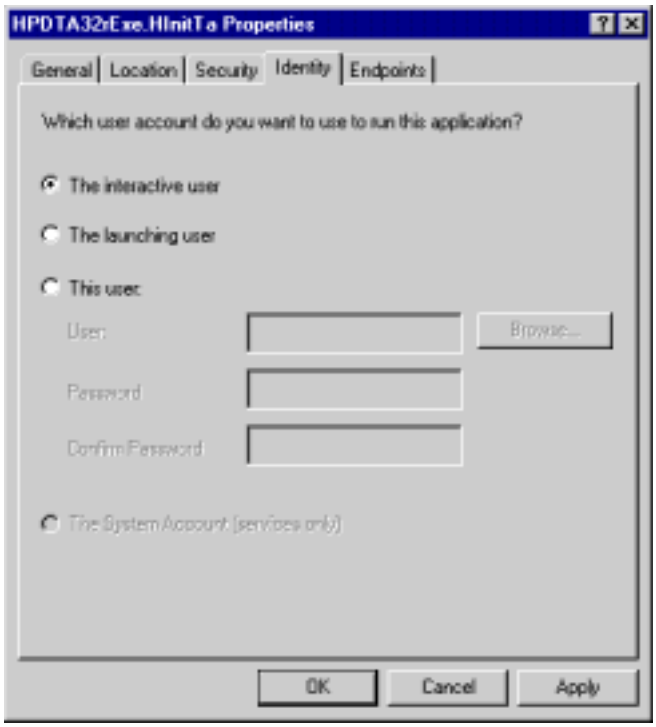


Simply select „Use custom access permissions“ and then press the „Edit“ button to open the Value Permissions dialog.

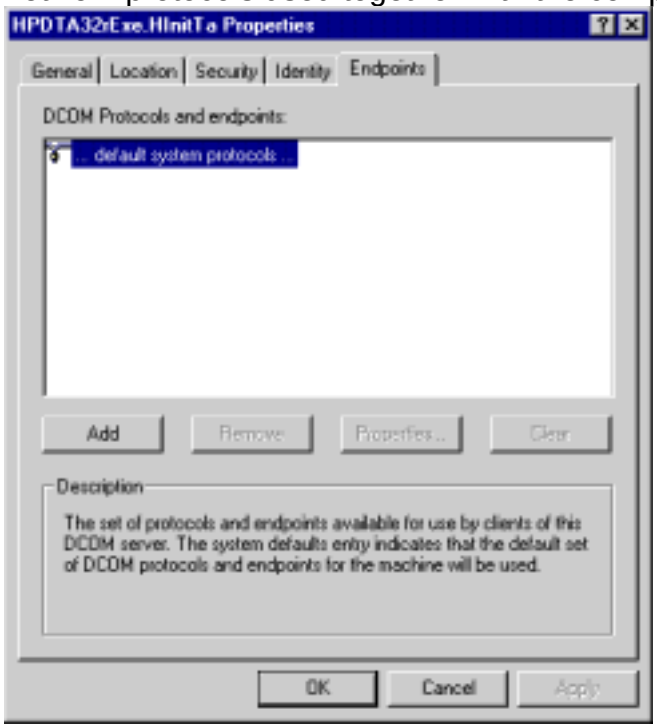


Add the desired user groups e.g. „Everyone“ and press the „Close“ button to return the properties dialog.

On the identity page you can specify the user account that is used when this server application runs on the computer. In case the server application pops up any type of window that should become visible on the monitor for any remote user logon then select „The interactive user“. If you do not want that the windows pop up on the screen select either „The launching user“ or „This user“

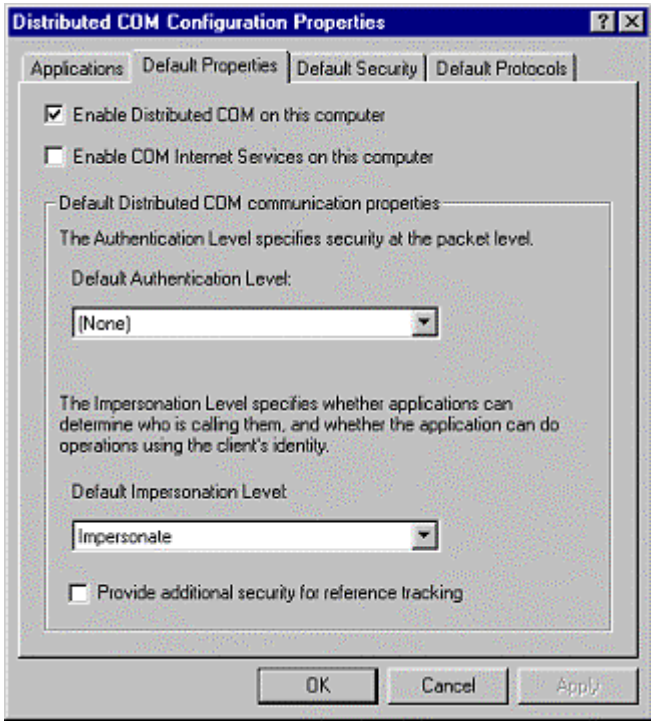


Finally on the „Endpoints“ page you can individually specify and configure the network protocols used together with the component.



Configuring the COM Client

The main topic for the client configuration is to enable DCOM. This can be done by using DCOMCNFG.EXE.
On the „Default Properties“ page make sure the item
„Enable Distributed COM on this computer“ bears a checkmark.

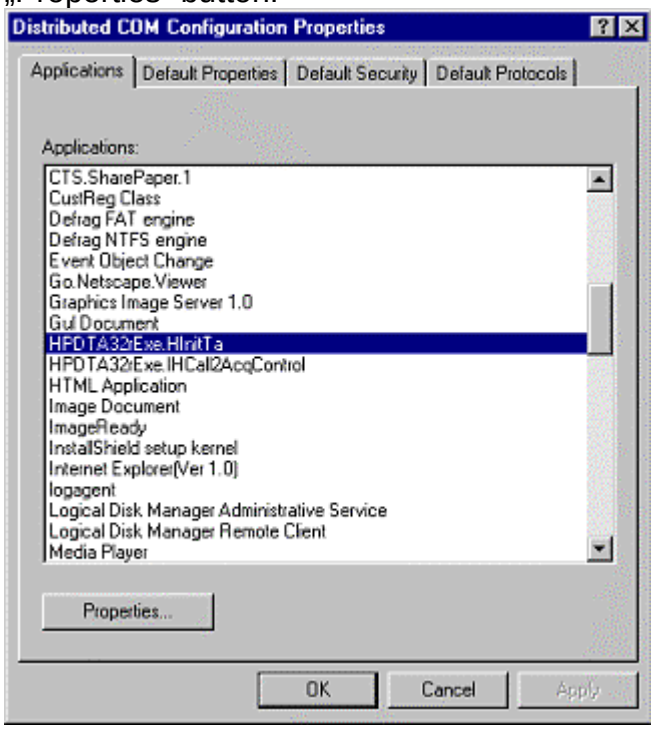


If a client needs further configuration mainly depends on the way how the client application tries to establish the connection to the server application.

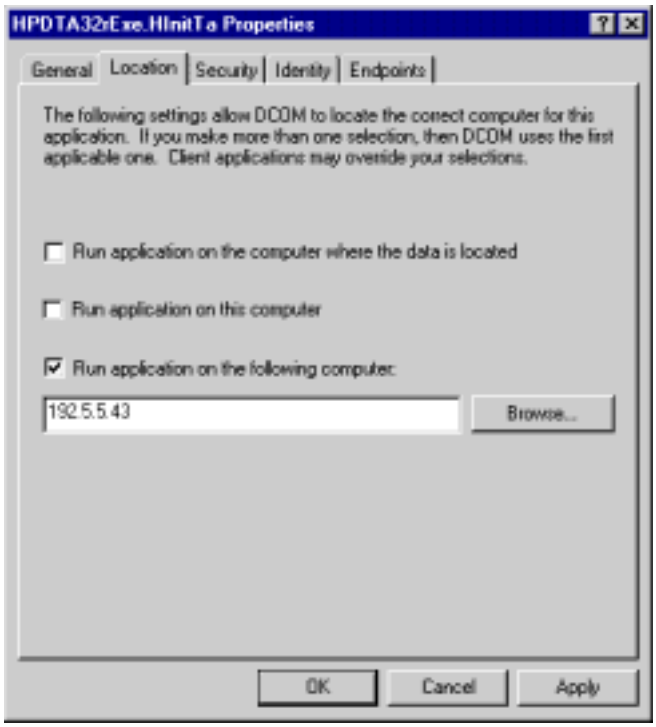
a) Configuring the COM-Client when using default COM access mechanism

If the client is using the default COM access mechanism the location of the server application has to be configured with DCOMCNFG.EXE. In this case the server application has to registered on the client computer additionally.

From the „Applications“ page select the server component, then press the „Properties“ button.



Select the „Location“ page and check the option „Run application on the following computer“. Uncheck the other options. On the edit field type in either the IP-address of the server computer e.g. 192.5.5.43 or the computer network name e.g. //kweinbuch. On WindowsNT and Windows2000 platforms you will find an additional „Browse“ button to browse for the server computer.



b) Configuring the COM-Client when using explicit server names
If the connection to the server is established by explicitly specifying the network address of the server computer no additional configuration of the client is necessary. Using this method has the additional advantage that the server application may not be registered on the client machine.

License and Keys

The remote control component for the HiPic and the HPD-TA can be used with the standard keys (dongles) for these programs. No special key is necessary. A special key which do not allow to run the standard products, but which allow to use the component without any user I/F (for specific applications) can be provided.

Migrating from Version 6.1 to 6.2

Outline

The ActiveX components from version 6.2 are not compatible to those of version 6.1. The IF is changed due to further requirements. The filenames are different for both versions and both versions can be used at the same computer. A client program designed for version 6.1 will still run with the component of version 6.1 (If this version has not been deleted or overwritten of course). If you want to use the component of version 6.2 with a client designed for version 6.1 you have to recompile your application using a reference to the component of version 6.2. In addition if your client program uses functions methods or properties which are removed in version 6.2 you have to substitute them by other functions (Normally they are removed because they are replaced by functions handling the topic under question in a better way).

Generally the following changements have been made:

- New Functions added to the HInitHi and HInitTa main objects: psApplicationDirectory (Returns the application directory), psWindowsDirectory (Returns the Windows directory of the computer where the component runs), pfReadFile (Reads the content of a file stored at the remote computer), pfWriteFile (Writes a file at the remote computer). These functions can be used to transfer data to or from a remote computer prior to start the main application HiPic or HPD-TA. Be careful with the pfWriteFile function. In the worst case you can destroy essential files on the remote computer.
- The same functions (psApplicationDirectory, psWindowsDirectory, pfReadFile, pfWriteFile) are also available in the HAppHiPic and HAppHPDTA main objects.
- getting the start status is now made more easy. The properties pfApplicationCreated and pfApplicationCreationFailed has been removed. Everything can now be inquired with the pfInitStatus property. There are the following possibilities: InitStatusNotInitialized = -2, InitStatusInitializing = -1, InitStatusInitialized = 0, InitStatusErrorDuringInit = 1, InitStatusNoLicence = 2, InitStatusCancel = 3. Generally if pfInitStatus<0 means not initialized, pfInitStatus=0 means Initialization succesful, pfInitStatus>0 Initialization not succesful. Pls. distinguish pfInitStatus from piStartStatus. piStartStatus tells you if the application has already been started from another location (Basically ActiveX-EXE only). If piStartStatus=StartStatusNone the application has not been started. If piStartStatus=StartStatusInit the application just shows thew init screen. If piStartStatus=StartStatusRunning the application is running. In the latter two cases you can get objects references to the HInitHi/Ta or HAppHiPic/HPDTA objects by the mHInitHiStart.pStartHInitHi (or mHInitTaStart.pStartHInitTa) and mHInitHiStart.pStartHAppHiPic (or mHInitTaStart.pStartHAppHPDTA) properties (See the example CltHiExe or CltTaExe for details).

- The way some parameters are called have been changed to be more consequently. Input parameters are now always called by value. Sometimes in such cases the VB compiler gives an error message when compiling with the reference to version 6.2 saying that the parameter type is not indentival. To find out the correct prototyp it is most easy to outcomment the function and let the VB compiler write the prototype by just selecting the correct object procedure again. Then just copy back the code to the empty prototype. As an example take the Message event of the HInitHi main object. The old definition was:

```
Private Sub mHInitHi_Message(sMessage As String)
    Your code is here...
End Sub
```

  The new definition is

```
Private Sub mHInitHi_Message(ByVal sMessage As String)
    Your code is here...
End Sub
```

  The Definition of sMessage has consequently been changed to be ByVal.
- The HInitTa main object has been reorganized reflecting the fact that the properties to select the camera parameters are now in a separate window. Therefore as an example the interface object pHRadInitCamera has been renamed to pHRadCCDSetCamera. The main window contains now some new interface objects to display the selected devices like pHDisInitCCDCamera (showing the selected camera).
- All options in the area General, Acquisition, Images and Sequence (not true for other dialogs like correction) can now be accessed more easly. Instead of showing the corresponding dialog and setting the interface object to the correct value there are now properties in the corresponding main objects. Example: The option “Acquire always to the same window” in the Images options could (and still can) be selected by showing the dialog HImages (`mHAppHiPic.pHImages.pHWinOptDlg.pfVisible=TRUE`), than select the option (`mHAppHiPic.pHImages.pHChkOptAcqToSameWnd.pfValue=TRUE`), then execute the OK button of the dialog (`mHAppHiPic.pHImages.pHComOptOK.pClick`).
  Now to get the same result you can just write:
  `mHAppHiPic.pHImages.pfOptionsAcqToSameWnd=TRUE`.
- New Properties in the HAppHiPic object: pHFlatPanel, pHMenMainFreeze
- New Properties in the HAppHPDTA object: pHC488080, pHMenMainFreeze, pHMenMainShowDelay2StatusControl, pHMenMainTriggerSetup. The property pHMenMainStreakSetup has been removed.
- There is now a way to get information whether the CCD camera could be connected successfully. Th propety pfCommPortOpen of the HAllHipIc and HAppHPDTA tell it.
- Getting and transfering image data to and from an image is now made more easy. There are the functions pfGetDisplayData, pfGetImageData1Byte, pfGetImageData2Byte, pfGetImageData4Byte and pfSetImageData1Byte, pfSetImageData2Byte, pfSetImageData4Byte of the HImage main object. These functions can be used by both the ActiveX-DLL and ActiveX-EXE versions and do not use memory handles. To get the data arrays have to be used which can (and will) be resized by the component. The definition should be like

```
Private miData() As Integer
```

  for getting 2 Byte data with the function pfGetImageData2Byte.
  The functions pfGetImgData and pfSetImgData have been removed.
- The functions for loading and saving images have been improved. The now have a return value to show whether the function was successful or not (TRUE or FALSE). Therefore the functions have been renamed from pLoadImage to pfLoadImage and from pSaveImage to pfSaveImage. Loading and saving sequences are modified from pLoadSequence to pfLoadSequence and pSaveSequence to pfSaveSequence.
- New Options of the HImages main object dealing with Auto Live LUT and FWHM display.

- If new data has been written to an image the new pfUpdateImageAfterDataChange function of the HImages main object should be called. This is especially true if the data or LUT type of the image has been changed by the pfCreateEmptyImage function or by writing important parameters like LUT parameters to the status string.
- All camera objects now have the following properties which are used in combination with the new trigger setup: pfStreakOperate, pfStreakOperateDTBE, pfStreakUseDTBE, piStreakTriggerMethod, pGetStreakRelatedCCDCaps. These properties are public due to internal reasons, do not use them.
- The HC800020 object has two new properties pHCh4CamHighVoltage and pHEn4CamHVoltage dealing with high voltage control.
- The HExternalDevices main object has many new properties dealing with the new trigger setup, the auto delay function and the second delay generator (DELAY2). The DT2819 has been renamed to Counterboard because now two counter boards can be used. All releated controls have be renamed.
- The properties pfDlgHasUserIF and subroutines pfStartUserIFEvent and pfEndUserIFEvent have been removed because it has been found that they do fulfill their purpose correctly (the had been defined public due to internal reason, adn should not be used by the client programmer).
- The HExternalDevice main object has two new feature for convenience: piGPIBCableConnectionSuccess (it tell the client programmer whether the connection to the device was successful) and pfSetParameterToMaximum (Sets the specified parameter to the maximum value).
- The HAcq main object has a new function pUpdateCameraParms. This functions inquires the camera and sets the parameters image size and depth (bytes per pixel) to the correct values. The new function pGetAcqDim can than get the correct values.
- The HMsgBox object has been modified. If pfNoDialogs has been set to TRUE (or if the fNoDialogs parameter of the piInit Function druign startup has been set to TRUE) it outputs no message box on the screen but raises the MsgBox Event. This event has the parameters

  ```
  Event MsgBox(ByVal iID As Integer, ByVal sPrompt As String, ByVal sTitle As String, ByVal Style As Integer, ByVal Buttons As Integer, ByRef default As Integer)
  ```

  The iID parameters gives an indication which dialog text is used. I refers to the entry in the HiPicTa.Res file. The client program can the react according to this parameter. Example: iID = 2092: refers to „The currently selected image is not or not fully saved. Do you want to save it now? Image“. If e.g. the client program wants to make a silent shutdown he can set default = vbNo. sPrompt is the text normally outputted, sTitle the title which normally is located on the caption of the messagebox. Style can be one of the following: `vbOKOnly=0,vbCritical=16,vbQuestion=32,vbExclamation=48,vbInformation=64`, Buttons can be:

  `vbOKOnly=0,vbOKCancel=1,vbAbortRetryIgnore=2,vbYesNoCancel=3,vbYesNo=4,vbRetryCancel=5`.

  The client program can reply to the message box by setting the default value. Possible values are: `vbOK=1, vbCancel=2, vbAbort=3, vbRetry=4, vbIgnore=5, vbYes=6, vbNo=7`.

  There is one issue to take into account: If the message boxc is a consequence of an action issued by the client program this event should not output a messagebox to the user and wait for the reply because then the component request is pending. Therefore the client program should distinguish whether the event has been raised due to an action of the component (either issued by the components user I/F or by an asyncronous command, in this case it is allowed to wait inifinitely) or by an action of the client program (For this purpose the samples uses the pfClientEvent property internally).

Details

In Detail the following changements have been made from version 6.1 to version 6.2

HinitHi:
New:

Read only property	pfInitStatus()	Integer	Returns a value indicating the current status of initialization. Possible values are defined in the enumeration InitStatus
Read only property	psApplicationDirectory()	String	Returns the application directory
Read only property	psWindowsDirectory()	String	Returns the Windows directory of the computer where the component runs
Function	pfReadFile(ByVal sFileName As String, bArray()	Byte, ByRef lLength As Long, Optional sError As Variant)	Reads the content of a file stored at the remote computer
Function	pfWriteFile(ByVal sFileName As String, bArray()	Byte, ByRef lLength As Long, ByVal fDontOverwrite As Integer, Optional sError As Variant)	Writes a file at the remote computer

Modified:

Event	Message(ByVal sMessage As String)		Event which is raised to indicate processign steps during initialization
Function	piInit(ByVal sIniFile As String, ByVal fInitUserIF As Integer, ByVal fApplUserIF As Integer, ByVal fNoDialogs As Integer)	Integer	Initializes the HInit object. This places the INIT dialog on screen if started with InitUserIF
Sub	pGetLicenceKeys(ByRef fApplicationKeyFound%, ByRef fLicenceAcquire%, ByRef fLicenceFitting%, ByRef fLicenceRCOnly%, ByRef fLicenceSave%, ByRef fLicenceSequence%, ByRef fLicenceTransAbs%)		Returns information about all licence keys which could be found

Removed:
MsgBoxResult
pfApplicationCreated
pfApplicationCreation Failed

HAppHipic:
New:

Read only property	pHFlatPanel()	HFlatPanel	Returns an object reference to the FlatPanel Camera object (HFlatPanel). If this camera is not used NOTHING is returned
Read only property	pHMenMainFreeze()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Freeze' menu
Read only property	psWindowsDirectory()	String	Returns the Windows directory of the computer where the component runs
Read/write property	pfOptionLUTTool()	Integer	Sets or returns a value which defines whether the LUT tool should be shown on the main window
Read/write property	pfOptionRestoreWindowPos()	Integer	Sets or returns a value which defines whether window positions should be restored or not when the window is reactivated
Read/write property	pfOptionUserFunctions()	Integer	Sets or returns a value which defines whether the User function is called or not
Read/write property	plOptionBackgroundCo	Long	Sets or returns a value which defines the background color

	lor()		
Read only property	pfCommPortOpen()	Integer	Returns a value which defines whether the communication port could be opened succesfully for camera control
Function	pfReadFile(ByVal sFileName As String, bArray()	Byte, ByRef lLength As Long, Optional sError As Variant)	Reads the content of a file stored at the remote computer
Function	pfWriteFile(ByVal sFileName As String, bArray()	Byte, ByRef lLength As Long, ByVal fDontOverwrite As Integer, Optional sError As Variant)	Writes a file at the remote computer

Modified:

Function	psErrorString(ByVal Index As Integer)	String	Returns a description of the Error related to error No. Index

HinitTa:

New:

Event	MsgBox(ByVal iID As Integer, ByVal sPrompt As String, ByVal sTitle As String, ByVal Style As Integer, ByVal Buttons As Integer, ByRef default As Integer)		Event which is raised in remote control mode when the pfNoDialogs is set to true instead of showing a \r\r\nmessagebox. This event can be use for the client program to react on such messages
Read only property	pfInitStatus()	Integer	Returns a value indicating the current status of initialization. Possible values are defined in the enumeration InitStatus
Read only property	pHComCCDSetCancel( )	HCommand	
Read only property	pHComCCDSetGetConfigFile()	HCommand	Returns an object reference to the HCommand object associated with the init dialogs 'Get Config File' pushbutton
Read only property	pHComCCDSetOK()	HCommand	
Read only property	pHComInitSetupCCD()	HCommand	Returns an object reference to the HCommand object associated with the init dialogs 'Setup CCD' pushbutton
Read only property	pHComInitSetupStreakDevices()	HCommand	Returns an object reference to the HCommand object associated with the init dialogs 'Setup Streak Devices' pushbutton
Read only property	pHDisInitCCDCamera( )	HDisp	Returns an object reference to the HDisp object associated with the init dialogs 'CCD Camera' display area
Read only property	pHDisInitDevice1()	HDisp	Returns an object reference to the HDisp object associated with the init dialogs 'Device 1' display area
Read only property	pHDisInitDevice2()	HDisp	Returns an object reference to the HDisp object associated with the init dialogs 'Device 2' display area
Read only property	pHDisInitDevice3()	HDisp	Returns an object reference to the HDisp object associated with the init dialogs 'Device 3' display area
Read only property	pHDisInitDevice4()	HDisp	Returns an object reference to the HDisp object associated with the init dialogs 'Device 4' display area
Read only property	pHDisInitGPIBAddr1()	HDisp	Returns an object reference to the HDisp object associated with the init dialogs GPIB Address 1' display area
Read only property	pHDisInitGPIBAddr2()	HDisp	Returns an object reference to the HDisp object associated with the init dialogs GPIB Address 2' display area
Read only property	pHDisInitGPIBAddr3()	HDisp	Returns an object reference to the HDisp object associated with the init dialogs GPIB Address 3' display area
Read only property	pHDisInitGPIBAddr4()	HDisp	Returns an object reference to the HDisp object associated with the init dialogs GPIB Address 4' display area
Read only property	pHEdnCCDSetC474295ComPort()	HEditNumber	Returns an object reference to the HEditNumber object associated with init dialogs 'C474295 Com Port' editbox
Read only property	pHEdnCCDSetC474298ComPort()	HEditNumber	Returns an object reference to the HEditNumber object associated with init dialogs 'C474298 Com Port' editbox
Read only property	pHEdnCCDSetC488080ComPort()	HEditNumber	Returns an object reference to the HEditNumber object associated with init dialogs 'C4880-8X Com Port' editbox
Read only property	pHEdnCCDSetC4880ComPort()	HEditNumber	Returns an object reference to the HEditNumber object associated with init dialogs 'C4880 Com Port' editbox
Read only property	pHEdnCCDSetGrabberSysNo()	HEditNumber	Returns an object reference to the HEditNumber object associated with init dialogs 'Board number' editbox (Also referred to system number)
Read only property	pHEstCCDSetConfiFile()	HEditString	Returns an object reference to the HEditString object associated with the init dialogs 'Config File' editbox

Read only property	pHFraCCDSetCamera()	HFrame	Returns an object reference to the HFrame object associated with the init dialogs CCD Setup' frame
Read only property	pHRadCCDSetCamera( )	HRadios	Returns an object reference to the HRadios object associated with the init dialogs 'Camera' radiobutton group
Read only property	pHRadCCDSetFrameG rabber()	HRadios	Returns an object reference to the HRadios object associated with the init dialogs 'Frame Grabber' group
Read only property	psApplicationDirectory ()	String	Returns the application directory
Read only property	psWindowsDirectory()	String	Returns the Windows directory of the computer where the component runs
Function	pfReadFile(ByVal sFileName As String, bArray()	Byte, ByRef lLength As Long, Optional sError As Variant)	Reads the content of a file stored at the remote computer
Function	pfWriteFile(ByVal sFileName As String, bArray()	Byte, ByRef lLength As Long, ByVal fDontOverw rite As Integer, Optional sError As Variant)	Writes a file at the remote computer

Modified:

Event	Message(ByVal sMessage As String)		Event which is raised to indicate processign steps during initialization
Function	piInit(ByVal sIniFile As String, ByVal fInitUserIF As Integer, ByVal fApplUserIF As Integer, ByVal fNoDialogs As Integer)	Integer	Initializes the HInit object. This places the INIT dialog on screen if started with InitUserIF
Sub	pGetLicenceKeys(ByR ef fApplicationKeyFound %, ByRef fLicenceAcquire%, ByRef fLicenceFitting%, ByRef fLicenceRCOnly%, ByRef fLicenceSave%, ByRef fLicenceSequence%, ByRef fLicenceTransAbs%)		Returns information about all licence keys which could be found

Removed:
MsgBoxResult
pfApplicationCreated
pfApplicationCreation Failed
phComInitGetConfigFile

Read only property	pHEdnInitC474295Co mPort()	HEditNumb er	Returns an object reference to the HEditNumber object associated with init dialogs 'C474295 Com Port' editbox
Read only property	pHEdnInitC474298Co mPort()	HEditNumb er	Returns an object reference to the HEditNumber object associated with init dialogs 'C474298 Com Port' editbox
Read only property	pHEdnInitC4880ComP ort()	HEditNumb er	Returns an object reference to the HEditNumber object associated with init dialogs 'C4880 Com Port' editbox
Read only property	pHEdnInitGrabberSysN o()	HEditNumb er	Returns an object reference to the HEditNumber object associated with init dialogs 'Board number' editbox (Also referred to system number)
Read only property	pHEstInitConfiFile()	HEditString	Returns an object reference to the HEditString object associated with the init dialogs 'Config File' editbox
Read only property	pHFraInitCamera()	HFrame	Returns an object reference to the HFrame object associated with the init dialogs CCD Setup' frame
Read only property	pHRadInitCamera()	HRadios	Returns an object reference to the HRadios object associated with the init dialogs 'Camera' radiobutton group
Read only property	pHRadInitFrameGrabbe r()	HRadios	Returns an object reference to the HRadios object associated with the init dialogs 'Frame Grabber' group

HAppHPDTA
New:

Read only property	pfCommPortOpen()	Integer	Returns a value which defines whether the communication port could be opened succesfully for camera control

Read only property	pHC488080()	HC488080	Returns an object reference to the C4880-80 Camera object (HC488080). If this camera is not used NOTHING is returned
Read only property	pHComMainFreeze()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Freeze' pushbutton
Read only property	pHMenMainShowDelay2StatusControl()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Show Delay2 Status/Control' menu entry
Read only property	pHMenMainTriggerSetup()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Trigger Setup' menu entry
Read only property	psWindowsDirectory()	String	Returns the Windows directory of the computer where the component runs
Read/write property	pfOptionLUTTool()	Integer	Sets or returns a value which defines whether the LUT tool should be shown on the main window
Read/write property	pfOptionRestoreWindowPos()	Integer	Sets or returns a value which defines whether window positions should be restored or not when the window is reactivated
Read/write property	pfOptionUserFunctions()	Integer	Sets or returns a value which defines whether the User function is called or not
Read/write property	plOptionBackgroundColor()	Long	Sets or returns a value which defines the background color
Function	pfReadFile(ByVal sFileName As String, bArray()	Byte, ByRef lLength As Long, Optional sError As Variant)	Reads the content of a file stored at the remote computer
Function	pfWriteFile(ByVal sFileName As String, bArray()	Byte, ByRef lLength As Long, ByVal fDontOverwrite As Integer, Optional sError As Variant)	Writes a file at the remote computer

Removed:
pHMenMainStreakSetup
Modified:

Function	psErrorString(ByVal Index As Integer)	String	Returns a description of the Error related to error No. Index

Himage:
New:

Function	pfSetImageData1Byte(ByVal iX As Integer, ByVal iDX As Integer, ByVal iY As Integer, ByVal iDY As Integer, ByRef bImgData()	Byte, Optional sError As Variant) As Integer	Sets 1byte/pixel data (8 bit) within a specified area of the image. If the image contains another type of data an error is returned
Function	pfSetImageData2Byte(ByVal iX As Integer, ByVal iDX As Integer, ByVal iY As Integer, ByVal iDY As Integer, ByRef iImgData()	Integer, Optional sError As Variant) As Integer	Sets 2byte/pixel data (8 bit) within a specified area of the image. If the image contains another type of data an error is returned
Function	pfSetImageData4Byte(ByVal iX As Integer, ByVal iDX As Integer, ByVal iY As Integer, ByVal iDY As Integer, ByRef lImgData()	Long, Optional sError As Variant) As Integer	Sets 4byte/pixel data (8 bit) within a specified area of the image. If the image contains another type of data an error is returned
Function	pfLoadImage(Optional sFile As Variant, Optional fShow As Variant)	Integer	Loads an image
Function	pfSaveImage(Optional sFile As Variant, Optional sAreDefault As Variant, Optional fSaveROI As Variant)	Integer	Saves the image

Modified:

Function	pfGetDisplayData(ByVal iX As Integer, ByVal iDX As Integer, ByVal iY As Integer, ByVal iDY As Integer, ByRef	Byte, Optional sError As Variant) As Integer	Gets the display data (8 bit) within a specified area of the image

	bImgDisplay()		
Function	pfGetImageData1Byte(ByVal iX As Integer, ByVal iDX As Integer, ByVal iY As Integer, ByVal iDY As Integer, ByRef bImgData()	Byte, Optional sError As Variant) As Integer	Gets 1byte/pixel data (8 bit) within a specified area of the image. If the image contains another type of data an error is returned
Function	pfGetImageData2Byte(ByVal iX As Integer, ByVal iDX As Integer, ByVal iY As Integer, ByVal iDY As Integer, ByRef iImgData()	Integer, Optional sError As Variant) As Integer	Gets 2byte/pixel data (16 bit) within a specified area of the image. If the image contains another type of data an error is returned
Function	pfGetImageData4Byte(ByVal iX As Integer, ByVal iDX As Integer, ByVal iY As Integer, ByVal iDY As Integer, ByRef lImgData()	Long, Optional sError As Variant) As Integer	Gets 4byte/pixel data (32 bit) within a specified area of the image. If the image contains another type of data an error is returned
Function	pfMemGetPRFData(By Ref lProfile()	Long, ByRef iNrData As Integer, ByVal iX As Integer, ByVal iY As Integer, ByVal iDX As Integer, ByVal iDY As Integer, ByVal iProfileType As Integer, ByRef iDatType As Integer, ByVal iMaxLen As Integer, ByRef sError As String) As Integer	Returns an array of profile values within a specified ROI

Removed:
pfGetImgData
pfSetImgData
pLoadImage
pSaveImage

HImages:
New:

Read only property	pHChkOptAutoLiveLUT()	HCheck	Returns an object reference to the HCheck object associated with the options dialogs 'Auto Live LUT' checkbox
Read only property	pHChkOptDisplayFWHM()	HCheck	Returns an object reference to the HCheck object associated with the options dialogs 'Display FWHM' checkbox
Read only property	pHComFWHMColor()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'FWHM color' checkbox
Read only property	pHEdnOptFWHMNoOfDigits()	HEditNumber	Returns an object reference to the HEditNumber object associated with options dialogs 'FWHM No. of Digits' editbox
Read only property	pHEdnOptFWHMSize( )	HEditNumber	options dialogs 'FWHM Size' editbox
Read/write property	pfOptionsAcqToSameWnd()	Integer	Sets or returns a value which defines whether the acquisition should always be done in the same window
Read/write property	pfOptionsAutoLiveLUT()	Integer	Sets or returns a value which defines whether AUTO Live LUT should be excuted during Live mode
Read/write property	pfOptionsAutoLUT()	Integer	Sets or returns a value which defines whether Auot LUT should be excuted aftre image acquisition
Read/write property	pfOptionsFixedITEXHeader()	Integer	Sets or returns a value which defines whether the image should be saved with fixed header
Read/write property	pfOptionsROIUseMinAsZero()	Integer	Sets or returns a value which defines whether the minimum should be used as zero point for FWHM calculation
Read/write property	pfOptionsWarnUnsaved	Integer	Sets or returns a value which defines whether the user should be

	()		warned when an unsaved image is closed
Read/write property	pftOptionsDefZoomFactor()	Single	Sets or returns the default zooming factor
Read/write property	piOptionsROIPrf()	Integer	
Function	pfUpdateImageAfterDataChange(ByVal iImg As Integer, Optional sError As Variant)		A function that updates all relevant modules after the client programmer has changed the image data, its size, data depth or status
Sub	pGetFWHMProperties(ByRef fDisplayFWHM As Integer, ByRef lColor As Long, ByRef iFontSize As Integer, ByRef iNoOfDigits As Integer)		Returns all relevant informations for writing the FWHM assoziated with the quick profile

Modified:

Event	CloseSequenceImage(ByVal Index As Integer, ByVal Mode As Integer, ByRef Cancel As Integer, ByRef SaveQueryDone As Integer, ByRef ImageRemoved As Integer)		Event which is sent to tell other objects that a sequence image will now be closed. This event is used by the HSequence object to free all related memory
Event	CurrentImageChanged(ByVal iImg As Integer)		Event which is raised when the current image changes
Event	FileNameChanged(ByVal iImg As Integer)		Event which is raised when the File name has changed
Event	ImageDataChanged(ByVal iImg As Integer)		Event which is raised when the image data has changed
Event	ImageRemoved(ByVal Index As Integer)		Event which is raised when the image has been removed
Event	Message(ByVal sMessage As String)		Event which is raised to inform the user of ongoing operations
Event	RemoveImage(ByVal Index As Integer, ByRef Cancel As Integer)		Event which is raised when an image will be removed
Function	pfLoadImage(Optional sFile As Variant, Optional fShow As Variant, Optional iImgIndex As Integer)	Integer	Loads an image
Function	pfLoadImageNoShow(ByVal sFile As String, ByVal iType As Integer, ByRef iImg As Integer)	Integer	Loads an image without showing it

HSystemScaling:
Modified:

Event	Message(ByVal sMessage As String)		Event which is raised to inform the user of ongoing operations
Function	pfGetFromTextFile(ByVal sTextFile As String, Optional sError As Variant)	Integer	Executes the ""Get Form Text File"" function specifiying a file name for the text file

HSequence:
New:

Function	pfLoadSequence(Optional sFileOpt As Variant, Optional iSeqModeOpt As Variant)	Integer	Loads a sequence
Function	pfSaveSequence(Optional sFile As Variant, Optional sAreDefault As Variant, Optional fSaveROI As Variant)	Integer	Saves a sequence

Modified:

Event	SeqSingleAcqEnded(ByVal iIndex As Integer)		Event which is raised when a single acquisition within a sequence has been ended
Event	Message(ByVal sMessage As String)		Event which is raised to inform the user of ongoing operations

Removed:
StartStreakTrigger
pLoadSequence
pSaveSequence

HJitter:

Event	Message(ByVal sMessage As String)		Event which is raised to inform the user of ongoing operations

HACam:
New:

Read/write property	pfStreakOperate()	Integer	A value that tells the camera object whether vertical streak operation is switched on. Don't modify this entry otherwise streak trigger handshake may not longer work
Read/write property	pfStreakOperateDTBE()	Integer	A value that tells the camera object whether horizontal streak operation is switched on. Don't modify this entry otherwise streak trigger handshake may not longer work
Read/write property	pfStreakUseDTBE()	Integer	Tells the camera object that a Dual time base extender is currently used for trigger handshake
Read/write property	piStreakTriggerMethod()	Integer	Tells the camera (or acquisition/sequence) object the streak trigger method. Don't modify this entry otherwise Steak trigger handshake may no longer work
Function	pGetStreakRelatedCCDCaps(ByRef fCCDCanExtTrig As Integer, ByRef fCCDCanStartStop As Integer, ByRef sExtTrigCCDMethod As String, ByRef sExtTrigCPUMethod As String, ByRef sStartStopCCDMethod As String, ByRef sStartStopCPUMethod As String)		Returns streak trigger related capabilities of the CCD camera

HC474295:
New:

Read/write property	pfStreakOperate()	Integer	A value that tells the camera object whether vertical streak operation is switched on. Don't modify this entry otherwise streak trigger handshake may not longer work
Read/write property	pfStreakOperateDTBE()	Integer	A value that tells the camera object whether horizontal streak operation is switched on. Don't modify this entry otherwise streak trigger handshake may not longer work
Read/write property	pfStreakUseDTBE()	Integer	Tells the camera object that a Dual time base extender is currently used for trigger handshake
Read/write property	piStreakTriggerMethod()	Integer	Tells the camera (or acquisition/sequence) object the streak trigger method. Don't modify this entry otherwise Steak trigger handshake may no longer work
Function	pGetStreakRelatedCCDCaps(ByRef fCCDCanExtTrig As Integer, ByRef fCCDCanStartStop As Integer, ByRef sExtTrigCCDMethod As String, ByRef sExtTrigCPUMethod As String, ByRef sStartStopCCDMethod As String, ByRef sStartStopCPUMethod As String)		Returns streak trigger related capabilities of the CCD camera

HC4880:

New:

Read/write property	pfStreakOperate()	Integer	A value that tells the camera object whether vertical streak operation is switched on. Don't modify this entry otherwise streak trigger handshake may not longer work
Read/write property	pfStreakOperateDTBE( )	Integer	A value that tells the camera object whether horizontal streak operation is switched on. Don't modify this entry otherwise streak trigger handshake may not longer work
Read/write property	pfStreakUseDTBE()	Integer	Tells the camera object that a Dual time base extender is currently used for trigger handshake
Read/write property	piStreakTriggerMethod ()	Integer	Tells the camera (or acquisition/sequence) object the streak trigger method. Don't modify this entry otherwise Steak trigger handshake may no longer work
Function	pGetStreakRelatedCCD Caps(ByRef fCCDCanExtTrig As Integer, ByRef fCCDCanStartStop As Integer, ByRef sExtTrigCCDMethod As String, ByRef sExtTrigCPUMethod As String, ByRef sStartStopCCDMethod As String, ByRef sStartStopCPUMethod As String)		Returns streak trigger related capabilities of the CCD camera

HC488080:
New:

Read/write property	pfStreakOperate()	Integer	A value that tells the camera object whether vertical streak operation is switched on. Don't modify this entry otherwise streak trigger handshake may not longer work
Read/write property	pfStreakOperateDTBE( )	Integer	A value that tells the camera object whether horizontal streak operation is switched on. Don't modify this entry otherwise streak trigger handshake may not longer work
Read/write property	pfStreakUseDTBE()	Integer	Tells the camera object that a Dual time base extender is currently used for trigger handshake
Read/write property	piStreakTriggerMethod ()	Integer	Tells the camera (or acquisition/sequence) object the streak trigger method. Don't modify this entry otherwise Steak trigger handshake may no longer work
Function	pGetStreakRelatedCCD Caps(ByRef fCCDCanExtTrig As Integer, ByRef fCCDCanStartStop As Integer, ByRef sExtTrigCCDMethod As String, ByRef sExtTrigCPUMethod As String, ByRef sStartStopCCDMethod As String, ByRef sStartStopCPUMethod As String)		Returns streak trigger related capabilities of the CCD camera
Read only property	pHCh4CamStreakTrigg er()	HCheck4Arr ay	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Streak Trigger Active' checkbox
Read only property	pHDi4CamStreakTrigg er()	HDisp4Arra y	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Streak Trigger' display area
Read only property	pHEd4CamNrTrigger()	HEditNumb er4Array	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Number of Streak Trigger' editbox
Read only property	pHFr4CamStreakTrigge r()	HFrame4Arr ay	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Streak Trigger'

HC474298:
New:

Read/write property	pfStreakOperate()	Integer	A value that tells the camera object whether vertical streak operation is switched on. Don't modify this entry otherwise streak trigger handshake may not longer work
Read/write property	pfStreakOperateDTBE( )	Integer	A value that tells the camera object whether horizontal streak operation is switched on. Don't modify this entry otherwise streak trigger handshake may not longer work
Read/write property	pfStreakUseDTBE()	Integer	Tells the camera object that a Dual time base extender is currently used for trigger handshake

Read/write property	piStreakTriggerMethod()	Integer	Tells the camera (or acquisition/sequence) object the streak trigger method. Don't modify this entry otherwise Steak trigger handshake may no longer work
Function	pGetStreakRelatedCCDCaps(ByRef fCCDCanExtTrig As Integer, ByRef fCCDCanStartStop As Integer, ByRef sExtTrigCCDMethod As String, ByRef sExtTrigCPUMethod As String, ByRef sStartStopCCDMethod As String, ByRef sStartStopCPUMethod As String)		Returns streak trigger related capabilities of the CCD camera

HC800020:
New:

Read only property	pHCh4CamHighVoltage()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'High Voltage' checkbox
Read only property	pHEn4CamHVoltage()	HEntry4Array	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'High Voltage' entrybox

New Object HFlatPanel
All events, properties and methods are new.

HExternalDevices:
New:

Read only property	pfStreakUseDTBE()	Integer	Tells the camera object that a Dual time base extender is currently used for trigger handshake
Read only property	pHCh4SetupOptionDelay2()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the device control setup dialogs 'Use Delay2 Option' checkbox
Read only property	pHChkAutoDelParameter00()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 00' checkbox
Read only property	pHChkAutoDelParameter01()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 01' checkbox
Read only property	pHChkAutoDelParameter02()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 02' checkbox
Read only property	pHChkAutoDelParameter03()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 03' checkbox
Read only property	pHChkAutoDelParameter04()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 04' checkbox
Read only property	pHChkAutoDelParameter05()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 05' checkbox
Read only property	pHChkAutoDelParameter06()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 07' checkbox
Read only property	pHChkAutoDelParameter07()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 07' checkbox
Read only property	pHChkAutoDelParameter08()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 08' checkbox
Read only property	pHChkAutoDelParameter09()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 09' checkbox
Read only property	pHChkAutoDelParameter10()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 10' checkbox
Read only property	pHChkAutoDelParameter11()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 11' checkbox
Read only property	pHChkAutoDelParameter12()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 12' checkbox
Read only property	pHChkAutoDelParameter13()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 13' checkbox
Read only property	pHChkAutoDelParameter14()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 14' checkbox
Read only property	pHChkAutoDelParameter15()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 15' checkbox
Read only property	pHChkAutoDelParameter16()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 16' checkbox
Read only property	pHChkAutoDelParameter17()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 17' checkbox
Read only property	pHChkAutoDelParameter18()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 18' checkbox
Read only property	pHChkAutoDelParamet	HCheck	Returns an object reference to the HCheck object associated with the

	er19()		device control setup dialogs 'Auto Delay parameter 19' checkbox
Read only property	pHChkAutoDelParameter20()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 20' checkbox
Read only property	pHChkAutoDelParameter21()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 21' checkbox
Read only property	pHChkAutoDelParameter22()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 22' checkbox
Read only property	pHChkAutoDelParameter23()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 23' checkbox
Read only property	pHChkAutoDelParameter24()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 24' checkbox
Read only property	pHChkAutoDelParameter25()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 25' checkbox
Read only property	pHChkAutoDelParameter26()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 26' checkbox
Read only property	pHChkAutoDelParameter27()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 27' checkbox
Read only property	pHChkAutoDelParameter28()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 28' checkbox
Read only property	pHChkAutoDelParameter29()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 29' checkbox
Read only property	pHChkOptionsAutoDelay2Delay()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay2 Delay' checkbox
Read only property	pHChkSetupCounterBoardInstalled()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Dt2819 Installed' checkbox
Read only property	pHChkSetupGPIBConnectedToDELAY2()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'GP-IB connected To Delay2' checkbox
Read only property	pHChkSetupUseDeviceDELAY2()	Hcheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Use Delaybox2' checkbox
Read only property	pHChkTrigSetConnectMonitorOut()	Hcheck	Returns an object reference to the HCheck object associated with the trigger setup dialogs 'Connect Monitor Out' checkbox
Read only property	pHChkTrigSetConnectResetIn()	Hcheck	Returns an object reference to the HCheck object associated with the trigger setup dialogs 'Conect Reset In' checkbox
Read only property	pHChkTrigSetUseDTBE()	HCheck	Returns an object reference to the HCheck object associated with the Trigger setup dialogs 'Use dual time base extender' checkbox
Read only property	pHComAutoDelCancel()	HCommand	Returns an object reference to the HCommand object associated with the auto delay setup dialogs 'Cancel' command button
Read only property	pHComAutoDelOK()	HCommand	Returns an object reference to the HCommand object associated with the auto delay setup dialogs 'OK' command button
Read only property	pHComAutoDelResetStoredValues()	HCommand	Returns an object reference to the HCommand object associated with the auto delay setup dialogs 'Reset Stored Values' command button
Read only property	pHComTrigSetCancel()	HCommand	Returns an object reference to the HCommand object associated with the trigger setup dialogs 'Cancel' command button
Read only property	pHComTrigSetOK()	HCommand	Returns an object reference to the HCommand object associated with the trigger setup dialogs 'OK' command button
Read only property	pHComTrigSetShowTiming()	HCommand	Returns an object reference to the HCommand object associated with the trigger setup dialogs 'Show Timing' command button
Read only property	pHDisAutoDelIntroduction()	HDisp	Returns an object reference to the HDisp object associated with the Auto delay dialogs 'Introduction' display area
Read only property	pHDisTrigSetCCDMode()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'CCD mode' display area
Read only property	pHDisTrigSetComputerExposureControl()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Computer exposure control' display area
Read only property	pHDisTrigSetComputerGPIB()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Computer GPIB' display area
Read only property	pHDisTrigSetComputerMonitorIn()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Computer Monitor In' display area
Read only property	pHDisTrigSetComputerResetOut()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Reset Out' display area
Read only property	pHDisTrigSetConfiguration()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Configuation' display area
Read only property	pHDisTrigSetDetectMonitorByGPIB()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Detect Monitor By GPIB' display area
Read only property	pHDisTrigSetDTBE()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Dual time base extender' display area
Read only property	pHDisTrigSetHMonitorOut()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'H-Monitor Out' display area
Read only property	pHDisTrigSetHResetIn()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'H-Reset In' display area
Read only property	pHDisTrigSetHTrigIn()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'H-Trigger In' display area
Read only property	pHDisTrigSetMainUnitType()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Main Unit Type' display area
Read only property	pHDisTrigSetPluginMode()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Plugin Mode' display area

Read only property	pHDisTrigSetPluginType()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Plugin Type' display area
Read only property	pHDisTrigSetResetByGPIB()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Reset by GPIB' display area
Read only property	pHDisTrigSetStreakCameraGPIB()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Streak camera GPIB' display area
Read only property	pHDisTrigSetTrigIn()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Trigger In' display area
Read only property	pHDisTrigSetVMonitorOut()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'V-Monitor Out' display area
Read only property	pHDisTrigSetVResetIn()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'V-Reset In' display area
Read only property	pHDisTrigSetVTrigIn()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'V-Trigger In' display area
Read only property	pHEdnTrigSetPostTriggerTime()	HEditNumber	Returns an object reference to the HEditNumber object associated with the Trigger setup dialogs 'Post Trigger Time' editbox
Read only property	pHEdnTrigSetResetDelay()	HEditNumber	Returns an object reference to the HEditNumber object associated with the Trigger setup dialogs 'Reset Delay' editbox
Read only property	pHRadTrigSetTriggerMethod()	HRadios	Returns an object reference to the HRadios object associated with the device control dialogs 'Trigger method' radiobutton group
Read only property	pHWinAutoDelDlg()	HWindow	Returns an object reference to the HWindow object associated with the Auto delay setup dialogs main window
Read only property	pHWinTimingDiagram()	HWindow	Returns an object reference to the HWindow object associated with the Timing dialogs timing diagram window
Read only property	pHWinTimingDlg()	HWindow	Returns an object reference to the HWindow object associated with the Timing dialogs main window
Read only property	pHWinTrigSetDlg()	HWindow	Returns an object reference to the HWindow object associated with the Trigger setup dialogs main window
Read only property	pHProProcProgress()	HProgress	Returns an object reference to the HProgress object associated with the Processing dialogs 'Progress' bar
Read only property	piCounterBoardInstalled()	Integer	Returns a flag which defines whether a Dt2819 board is installed
Read only property	piCounterBoardIOBase()	Integer	Returns the DT2819 board IO base adress
Read only property	piStreakTriggerMethod()	Integer	Returns the currently selected streak trigger method
Function	rTA_IdentProperty(ByVal iDeviceType As Integer, ByVal iPropertyID As Integer, varReturn As Variant)	Integer	Returns a property value of the specified device
Sub	pTA_HandleSRQs()		Gives the device DLLs the opportunity to handle SRQs (if any)
Function	rTA_TrigSet()	Integer	
Sub	pSetCCDCaps(fCCDCanExtTrig As Integer, fCCDCanStartStop As Integer, sExtTrigCCDMethod As String, sExtTrigCPUMethod As String, sStartStopCCDMethod As String, sStartStopCPUMethod As String)		Sets CCD capabilities related to streak trigger handshake

Modified:

Event	Message(ByVal sMessage As String)		Event which is raised to inform the user of ongoing operations

Removed:
pfTriggerControlAvail
pHChkSetupDt2819Installed
pHEdnSetupDT2819IOBase
piDT2819Installed
piDT2819IOBase
piTDTriggerCableConnected

HDevPar:

Modified:

Event	ChangeValue(ByVal dbOldValue As Double,		Event which is raised when the pfValue property changes

	ByVal fFromDevice As Integer)		

Removed:
pfDlgHasUserIF
pfEndUserIFEvent
pfStartUserIFEvent

HExternalDevice:
New:

Read only property	piGPIBCableConnectionSuccess()	Integer	Returns a flag which defines whether the device is connected to the GP-IB board successfully
Function	pfSetParameterToMaximum(ByVal iParameter As Integer)	Integer	Sets the specified parameter to the maximum value

HDevPars:
Modified:

Event	ChangeEnabled(ByVal Index As Integer)		Event which is raised when the pfEnabled property changes
Event	ChangeLimits(ByVal Index As Integer)		Event which is raised when either pvMinValue or pvMaxValue changes
Event	ChangeValue(ByVal Index As Integer, ByVal dbOldValue As Double, ByVal fFromDevice As Integer)		Event which is raised when the pfValue property changes
Event	ChangeVisible(ByVal Index As Integer)		Event which is raised when the pfVisible property changes

HAcq:
New:

Read/write property	pfOptionAutoBacksub()	Integer	Sets or returns a value which defines whether Auto backsub function should be executed
Read/write property	pfOptionAutoCurvature()	Integer	Sets or returns a value which defines whether Auto curvature function should be executed
Read/write property	pfOptionAutoShading()	Integer	Sets or returns a value which defines whether Auto shading function should be executed
Read/write property	pfOptionClipZero()	Integer	Sets or returns a value which defines whether clip to Zero function should be executed during background subtraction
Read/write property	pfOptionSlowDisplay()	Integer	Sets or returns a value which defines whether slow display function should be executed during LIVE mode
Read/write property	pfOptionWriteDPCFile()	Integer	Sets or returns a value which defines whether DPC files should be written during photon counting
Read/write property	piOptionAdditionalTimeout()	Integer	Sets or returns the value for additional timeout
Sub	pGetAcqDim(iX As Integer, iY As Integer, iDX As Integer, iDY As Integer, iBytesPerPixel As Integer)		Gets the acquisition dimensions for the current camera settings
Sub	pUpdateCameraParms()		

Modified:

Event	CameraTemperature(ByVal sTemp As String)		Event which is raised when the camera temperature changes
Event	ChangeAcqMode(ByVal iAcqMode As Integer)		Event which is raised when the Acquisiton mode changes
Event	Message(ByVal sMessage As String)		Event which is raised to inform the user of ongoing operations

Removed:
StartStreakTrigger

HGrb:
New:

Function	pfCanAcquireLargeFrames()	Integer	Returns a value which indicates that the current frame grabber configuration can acquire large frames (larger than the frame grabbers memory)

Modified:

Function	pfAcqEnterVB(ByVal fTargetField As Integer, ByVal ftTimeout As Single)	Integer	Wait until the camera enters the vertical blank
Function	pfAcqPassLine(ByVal lStartLine As Long, ByVal iVact As Integer, ByVal ftTimeout As Single)	Integer	Waits until the camera passes a specified line. The line number can alos have the meaning of a byte counter
Function	pfGrbCopyFBToBufferInParts(ByVal lpvBufferArea As Long, ByRef areBufferiX As Integer, ByRef areBufferiY As Integer, ByRef areBufferiDX As Integer, ByRef areBufferiDY As Integer, ByRef iBufferBytesPerPixel As Integer, ByVal iParts As Integer, ByVal fCheckValid As Integer)	Integer	Copies the content of the frame buffer into a computers memory buffer in parts
Function	pfGrbCopyFBToMem(ByRef HImage As HImage)	Integer	Copies the content of the frame buffer into an image memory
Function	pfGrbCopyFBToMemInParts(ByRef HImage As HImage, ByVal iParts As Integer, ByVal fCheckValid As Integer)	Integer	Copies the content of the frame buffer into an image memory in parts
Function	pfGrbSetInputLUTforPC(ByVal iThreshold As Integer, ByVal iDat As Integer)	Integer	Sets the Input LUT for photon counting
Function	piInit(ByVal sConfigFile As String, ByVal iGRB As Integer, ByVal iGrabberSysNo As Integer, ByVal iAmMod As Integer, ByVal iComPort As Integer, ByVal sError As String, ByRef HError As HError)	Integer	Initializes the frame grabber
Function	plAcqAmDigStartLine(ByRef areACQiX As Integer, ByRef areACQiY As Integer, ByRef areACQiDX As Integer, ByRef areACQiDY As Integer, ByVal lpvBufferArea As Long, ByRef iBufferBytesPerPixel As Integer, ByRef ftTimeForReadout As Single, ByRef ftFrameTime As Single, ByVal ftTimeoutFromCamera As Single, ByRef lLastLine As Long)	Long	Calculates the starting line (or byte counter) used for the CatCo strategy
Function	plCurrenLine(ByVal iDY As Integer)	Long	Returns the current line (or byte counter)
Function	pRSeqDoABSequence(ByVal areGRBScaniX As Integer, ByVal areGRBScaniY As Integer, ByVal areGRBScaniDX As	Long, ByRef sISeqTime() As String, ByRef dbRSeqStartTime As	Executes an AB sequence

	Integer, ByVal areGRBScaniDY As Integer, ByVal iBytesPerPixel As Integer, ByVal fWrap As Integer, lRSeqPtr()	Double, ByRef HEdnSeqCurrentSample As HEditNumber, ByVal fEnableStopinSequence As Integer, ByVal iNrSamples As Integer, ByRef IHCall2AcqControl As IHCall2AcqControl, ByVal ftTimeout As Single) As Integer	
Sub	pAcqAverageMultipleFrames(ByVal fICPAverageMultipleFrames As Integer, ByVal iICPAverageMultipleFramesCount As Integer, ByVal lpvBufferArea As Long, ByVal areBufferiX As Integer, ByVal areBufferiY As Integer, ByVal areBufferiDX As Integer, ByVal areBufferiDY As Integer, ByVal iBufferBytesPerPixel As Integer, ByVal lBuffer16Handle As Long)		Average a series of 8 bit images acquired from video camera with external trigger
Sub	pAcqGrbStartTrigSeqSnap(ByVal fICPAverageMultipleFrames As Integer, ByVal iICPAverageMultipleFramesCount As Integer, ByVal miAdditionalTimeout As Integer, ByRef IHCall2AcqControl As IHCall2AcqControl)		Start a triggered sequential snap
Sub	pCreateFrame(ByVal iBits As Integer, ByVal fSeqFrame As Integer, ByVal iNrSeqFrames As Integer)		Creates a frame on the frame grabber
Sub	pfGrbCopyFBToBuffer(ByVal lpvBufferArea As Long, ByRef areBufferiX As Integer, ByRef areBufferiY As Integer, ByRef areBufferiDX As Integer, ByRef areBufferiDY As Integer, ByRef iBufferBytesPerPixel As Integer)		Copies the content of the frame buffer into a computers memory buffer
Sub	pGetVideoCameraInfo(ByRef iCameraDX As Integer, ByRef iCameraDY As Integer, ByRef ftVideoFrameTime As Single)		Get info about the cameras frame
Sub	pGetWindow(ByRef		Gets the hardware window parameters for a given ROI

	hOff%, ByRef hAct%, ByRef vOff%, ByRef vAct%)		
Sub	pGrbGetImgAttrib(ByRef areSourceiX As Integer, ByRef areSourceiY As Integer, ByRef areSourceiDX As Integer, ByRef areSourceiDY As Integer, ByRef iBytesPerPixel As Integer)		Gets the attributes of the image in frame grabber
Sub	pGrbGetScanArea(ByVal iCameraDX As Integer, ByVal iCameraDY As Integer, ByRef pntDummyX As Integer, ByRef pntDummyY As Integer, ByRef pntCamWidthX As Integer, ByRef pntCamWidthY As Integer, ByRef areGRBScaniX As Integer, ByRef areGRBScaniY As Integer, ByRef areGRBScaniDX As Integer, ByRef areGRBScaniDY As Integer, ByRef areValidIMGiX As Integer, ByRef areValidIMGiY As Integer, ByRef areValidIMGiDX As Integer, ByRef areValidIMGiDY As Integer, ByRef pntOrigFBX As Integer, ByRef pntOrigFBY As Integer)		Get the scan area
Sub	pGrbGetWindowForSubArea(ByVal iX%, ByVal iY%, ByVal iDX%, ByVal iDY%, ByRef hOff%, ByRef hAct%, ByRef vOff%, ByRef vAct%, ByRef sError$)		Get the window generator hardware parameters for a given subarrea
Sub	pGrbSetDefaultSettings (ByVal iFrameDX As Integer, ByVal iFrameDY As Integer)		Set default settings
Sub	pGrbWaitEndOfImgReadout(ByVal ftTimeout As Single)		Wait unti end of image readout
Sub	pGrbWriteReducedLinLUT(ByVal iDat As Integer, ByVal iThreshold As Integer)		Write a linear LUT which reduces the image to 8 bit
Sub	pRSeqDoAmvsCatcoSequence(ByVal fWrap As Integer, ByRef iErrors As Integer, ByRef HEdnSeqCurrentSample As HEditNumber, ByRef lRSeqPtr()	Long, ByRef dbRSeqStartTime As Double, ByRef sISeqTime() As String, ByVal fEnableStopinSequence As Integer, ByVal iNrSamples As Integer,	Executes CatCo sequence for video module

		ByRef IHCall2Acq Control As IHCall2Acq Control)	
Sub	pSetWindow(ByVal hOff%, ByVal hAct%, ByVal vOff%, ByVal vAct%)		Sets the window generator harware parameters

HUtils:
Modified:

Event	GeneralMessage(ByVal sMessage As String)		

HAsyncCommand:
Modified:

Event	AfterCommand(ByVal sCommand As String)		Event which is raised after the execution of an asynchronous command
Event	BeforeCommand(ByVal sCommand As String)		Event which is raised before the execution of an asynchronous command
Event	DoCommand(ByVal sCommand As String, ByVal vParam0 As Variant, ByVal vParam1 As Variant, ByVal vParam2 As Variant, ByVal vParam3 As Variant, ByVal vParam4 As Variant)		Event which is raised to execute an asynchronous command

HMsgBox:
New:

Event	MsgBox(ByVal iID As Integer, ByVal sPrompt As String, ByVal sTitle As String, ByVal Style As Integer, ByVal Buttons As Integer, ByRef default As Integer)		Event which is raised in remote control mode when the pfNoDialogs property is set to true instead of showing a messagebox. This event can be use for the client program to react on such messages

Removed:
MsgBoxResult

HError:
Removed:
pfDlgHasUserIF
pfEndUserIFEvent
pfStartUserIFEvent

HCheck4Array:
Modified:

Event	ChangeEnabled(ByVal Index As Integer)		Event which is raised when the pfEnabled property changes
Event	ChangeValue(ByVal Index As Integer)		Event which is raised when the pfValue property changes
Event	ChangeVisible(ByVal Index As Integer)		Event which is raised when the pfVisible property changes

HDisp:
Removed:
pfDlgHasUserIF
pfEndUserIFEvent
pfStartUserIFEvent

HEditNumber:
Removed:
pfDlgHasUserIF

pfEndUserIFEvent
pfStartUserIFEvent

HEditNumber4Array:
Modified:

Event	ChangeEnabled(ByVal Index As Integer)		Event which is raised when the pfEnabled property changes
Event	ChangeLimits(ByVal Index As Integer)		
Event	ChangeValue(ByVal Index As Integer)		Event which is raised when the pfValue property changes
Event	ChangeVisible(ByVal Index As Integer)		Event which is raised when the pfVisible property changes

HEditString:
Removed:
pfDlgHasUserIF
pfEndUserIFEvent
pfStartUserIFEvent

HEntry:
New:

Read/write property	pftValue()	Single	Sets or returns a numerical value associated to the entry type. Time values in ms, numerical values or entry indexes are used depending on the EntryType

Removed:
pfDlgHasUserIF
pfEndUserIFEvent
pfStartUserIFEvent

HEntry4Array:
Modified:

Event	ChangeEnabled(ByVal Index As Integer)		Event which is raised when the pfEnabled property changes
Event	ChangeLimits(ByVal Index As Integer)		
Event	ChangeValue(ByVal Index As Integer)		Event which is raised when the pfValue property changes
Event	ChangeVisible(ByVal Index As Integer)		Event which is raised when the pfVisible property changes

HFrame4Array:
Modified:

Event	ChangeEnabled()		Event which is raised when the pfEnabled property changes
Event	ChangeVisible()		Event which is raised when the pfVisible property changes

HMenu:
Removed:
pfDlgHasUserIF
pfEndUserIFEvent
pfStartUserIFEvent

HProgress:
Removed:
pfDlgHasUserIF
pfEndUserIFEvent
pfStartUserIFEvent

HProgress4Array
Modified:

Event	ChangePercent()		Event which is raised when the pfPercent property changes
Event	ChangeVisible()		Event which is raised when the pfVisible property changes

HRadios:

Modified

Event	ChangeRadioEnabled(ByVal iRadio As Integer)		Event which is raised when the pfRadioEnabled property changes
Event	ChangeRadioVisible(ByVal iRadio As Integer)		Event which is raised when the pfRadioVisible property changes

Removed:
pfDlgHasUserIF
pfEndUserIFEvent
pfStartUserIFEvent

HRadios4Array
Modified:

Event	ChangeEnabled(ByVal Index As Integer)		Event which is raised when the pfEnabled property changes
Event	ChangeNoEntries(ByVal Index As Integer)		Event which is raised when the piNoEntries property changes
Event	ChangeRadio(ByVal Index As Integer)		
Event	ChangeRadioEnabled(ByVal iIndex As Integer, ByVal Index As Integer)		Event which is raised when the pfRadioEnabled property changes
Event	ChangeRadioVisible(ByVal iIndex As Integer, ByVal Index As Integer)		Event which is raised when the pfRadioVisible property changes
Event	ChangeVisible(ByVal Index As Integer)		Event which is raised when the pfVisible property changes

HTap:
Modified:

Event	ChangeTabEnabled(ByVal iTab As Integer)		Event which is raised when the pfTabEnabled property changes
Event	ChangeTabVisible(ByVal iTab As Integer)		Event which is raised when the pfTabVisible property changes

Removed:
pfDlgHasUserIF
pfEndUserIFEvent
pfStartUserIFEvent

HLut:
Modified:

Event	AutoLUT(ByRef Cancel As Integer)		Event which is raised when the user presses the AutoLUT key
Event	CursorsChanged(ByVal LowerCursor As Long, ByVal UpperCursor As Long)		Event which is raised when either of the two LUt cursors have been changed
Event	LimitsChanged(ByVal iSize As Integer)		Event which is raised when the the LUT limits changes
Event	LowerCursorChanged(ByVal LowerCursor As Long)		Event which is raised when the lower cursor changes
Event	Resize()		Event which is raised when the LUT control is resized
Event	UpperCursorChanged(ByVal UpperCursor As Long)		Event which is raised when the upper cursor changes

Removed:
pfDlgHasUserIF
pfEndUserIFEvent
pfStartUserIFEvent

HImageArea:
Removed:
pfDlgHasUserIF
pfEndUserIFEvent
pfStartUserIFEvent

HEditString4Array:
Modified:

Event	ChangeEnabled(ByVal Index As Integer)		Event which is raised when the pfEnabled property changes
Event	ChangeText(ByVal Index As Integer)		Event which is raised when the psText property changes
Event	ChangeVisible(ByVal Index As Integer)		Event which is raised when the pfVisible property changes

IHCall2Appl:
Modified:

Sub	GetImageScaling(ByVal iImg As Integer)		Assigns image scaling information to the current image according the current system scaling
Sub	GetImageStatus(ByVal iImg As Integer)		Assigns image status information to the current image according the current status

IHCall2Cam:
Modified:

Sub	RestoreAcqControls(ByVal iAcqMode As Integer)		Restore the live mode controls to its initial values (Which had been present before SetLiveControlsToPC)
Sub	SetAcqControlsToPC(ByVal iAcqMode As Integer)		Sets the live mode controls to the values which exists in photon counting mode
Sub	SetClearFBOnStart(ByVal fClearFBOnStart As Integer, ByVal iAcqMode As Integer)		

New object: IHCall2MC
All events properties and methods are new

ProfileScalingData:
Modified:

Function	pfCreateDiffPoynomial (ByVal iOrder As Integer, ByRef ftCoeff()	Single, ByVal iNrValidChannels As Integer, ByVal sUnit As String, Optional sError As Variant) As Integer	Create differential polynomial
Function	pfCreateIntegPoynomial(ByVal iOrder As Integer, ByRef ftCoeff()	Single, ByVal iNrValidChannels As Integer, ByVal sUnit As String, Optional sError As Variant) As Integer	Create integral polynomial
Function	pfGetArrayOfValues(ByRef ftArray()	Single, ByVal iEntries As Integer, Optional sError As Variant) As Integer	Returns an array of values
Function	pfReadScalingTable(ByVal sFileName As String, ByVal sOffset As String, ByVal fCheck As Integer, ByVal sUnit As String, Optional sError As Variant, Optional	Integer	Reads a scaling table from file

	varCheckFromChannel As Variant)		
Function	pfSetLinearData(ByVal ftScale As Single, ByVal sUnit As String, Optional sError As Variant)	Integer	Sets linear scaling
Function	pfSetScaling(ByRef psd As HProfileScalingData, Optional sError As Variant)		Sets the scaling data of this object identical to the data of the specified object
Function	pfSetTableData(ByRef ftTable()	Single, ByVal iEntries As Integer, ByVal sUnit As String, Optional sError As Variant, Optional varCheckFromChannel As Variant) As Integer	Sets table scaling
Function	pftProfileLocationFT(ByVal fScaled As Integer, ByVal ft As Single)	Single	Returns the profile location (floating point)
Function	pfValueExist(ByVal iEntry As Integer)	Integer	Returns a flag whether or not the value at the specified location exist
Function	pSaveScalingFile(ByVal sFile As String, Optional sError As Variant)	Integer	Saves the scaling data to file
Sub	pGetMinAndMax(ByVal iLower As Integer, ByVal iUpper As Integer, ByRef ftMin As Single, ByRef ftMax As Single)		gets minimum and maximum value
Sub	pLocationToIndex(ByVal ftLocation As Single, ByRef ftIndex As Single, ByVal fDisplayScaled As Integer)		Returns the index corresponding to the given location

HPRFParametersData:
Modified:

Read/write property	pftData(ByVal iData As Integer, Optional sError As Variant)	Single	Returns a single data value
Function	pfGetArrayOfData(ByRef ftData()	Single, ByVal iNrData As Integer, Optional sError As Variant) As Integer	Returns an array of data
Function	pfLoadPRF(ByRef sCompleteFile As String, ByVal fInquireFileName As Integer, ByRef sMessage As String, ByVal fReadData As Integer)	Integer	Loads a profile from file
Function	pfSearchValue(ByVal fScaled As Integer, ByVal iStartIndex As Integer, ByVal iEndIndex As Integer, ByVal iStep As Integer,	Integer	Searches a value within a profile

	ByVal ftSearchVal As Single, ByRef ftScaledValue As Single)		
Function	pftProfileLocationFT(ByVal fScaled As Integer, ByVal ft As Single)	Single	Returns the location within a profile (floating point)
Function	pftProfileLocationINT(ByVal fScaled As Integer, ByVal i As Integer)	Single	Returns the location within a profile (integer)
Sub	pCalcProfileDef(ByVal fDisplayScaled As Integer)		Calculate profile default values
Sub	pFitGetPDWDefault(ByVal fAbs As Integer)		Get the profile display window
Sub	pfSetHPRFParameters Data(ByRef ppd As HPRFParametersData)		Set all data associated with this profile to identical to the specified profile
Sub	pGetMinAndMax(ByVal iLower As Integer, ByVal iUpper As Integer, ByRef ftMin As Single, ByRef ftMax As Single)		Returns minimum and maximum values
Sub	pGetSingleMinMaxPos (ByVal iFirst As Integer, ByVal iLast As Integer, ByRef iMinPos As Integer, ByRef iMaxPos As Integer)		get the minimum and maximum positions
Sub	pProfileLocationToValue(ByVal ftLocation As Single, ByRef iIndex As Integer, ByRef ftIndex As Single, ByRef ftValue As Single, ByVal fDisplayScaled As Integer)		Returns the Value at a specified location

Migrating from Version 6.2 to 6.3

Outline

The ActiveX components from version 6.3 are not compatible to those of version 6.2. The IF is changed due to further requirements. The filenames are different for both versions and both versions can be used at the same computer. A client program designed for version 6.2 will still run with the component of version 6.2 (If this version has not been deleted or overwritten of course). If you want to use the component of version 6.3 with a client designed for version 6.2 you have to recompile your application using a reference to the component of version 6.3. In addition if your client program uses functions methods or properties which are removed in version 6.3 you have to substitute them by other functions (Normally they are removed because they are replaced by functions handling the topic under question in a better way).

Generally the following changements have been made:

- New interface objects in the HiInitHi and HiInitTa main object dealing with DCam cameras.
- New interface objects in the HAppHiPic and HAppHPDTA main objects: pHDCam, pHLicence and pHMenMainDefectPixelCorrectionSetup.

- Modified variable type for all variables describing sequence indexes and numbers from integer to long.
- New interface objects in the HImages main object dealing with Auto LUT and FWHM display.
- New event in HAcq and HSequence: CameraIsStarted. This event defines the time when the camera has started to collect light.
- New interface objects in the HC474295 main object dealing with temperature control.
- New interface objects in the HAcq mian object dealing with defect pixel correction.
- Some properties of the I/F objects have now optional parameters to facilitate the handling of these properties. The optional parameters are: Optional DoEvent As Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer. The DoEvent Parameter can have the three values DoEventOnChange = 0 (Default), DoEventAlways = 1, DoEventNever = 2. If DoEventOnChange is specified the event is raised when the value changes. If DoEventAlways is specified the event is always raised when the value is set. If DoEventNever is specified the event is not raised. If IgnoreDisable = TRUE the value is changed even if the I/F object is disabled (Be careful with this option, normally there is a reason why the I/F object is disabled). If IgnoreDisable = FALSE (Default) is specified the value is only changed when the I/F object is available. If NoError = TRUE is specified then the no error is raised even if an error occurs. If NoError = FALSE (Default) then an error is raised if the I/F object is not available or if the a wrong value is specified. You have to handle such error otherwise your program will abort.
- New property piAutoOption in the HLut I/F object: If piAutoOption=2 then the lower LUT cursor will be set to Zero in the case of AutoLUT. if piAutoOption=0 then the lower LUT cursor will be set according to the histogram. Do not use this feature, use the option pfOptionsZeroLower within the HImages object instead.

Details

The following is a list of modification of the IF in detail:

HInitHi
New:

Read only property	pHDisInitBus()	HDisp	
Read only property	pHDisInitCameraID()	HDisp	
Read only property	pHDisInitCameraVersion()	HDisp	
Read only property	pHDisInitDCamAPIVersion()	HDisp	
Read only property	pHDisInitDriverVersion()	HDisp	
Read only property	pHDisInitModel()	HDisp	
Read only property	pHDisInitModuleVersion()	HDisp	
Read only property	pHEdnInitOrcaHRComPort()	HEditNumber	
Read only property	pHEntInitModelNr()	HEntry	
Read only property	pHFraInitDCamAPI()	HFrame	
Read only property	pHRadInitCCDCameraAccess()	HRadios	

HAppHiPic

New:

Read only property	pHDCam()	HDCam	
Read only property	pHLicence()	HLicence	
Read only property	pHMenMainDefectPixelCorrectionSetup()	HMenu	

HInitTA
New:

Read only property	pHDisCCDSetBus()	HDisp	
Read only property	pHDisCCDSetCameraID()	HDisp	
Read only property	pHDisCCDSetCameraVersion()	HDisp	
Read only property	pHDisCCDSetDCamAPIVersion()	HDisp	
Read only property	pHDisCCDSetDriverVersion()	HDisp	
Read only property	pHDisCCDSetModel()	HDisp	
Read only property	pHDisCCDSetModuleVersion()	HDisp	
Read only property	pHDisCCDSetVendor()	HDisp	
Read only property	pHEdnCCDSetOrcaHRComPort()	HEditNumber	
Read only property	pHEntCCDSetModelNr()	HEntry	
Read only property	pHExternalDevices()	HExternalDevices	
Read only property	pHFraCCDSetDCamAPI()	HFrame	

HAppHPDTA
New:

Read only property	pHDCam()	HDCam	
Read only property	pHMenMainDefectPixelCorrectionSetup()	HMenu	

Modified:

Sub	pUserFunction(ByVal iIndex As Integer, ByVal lCycleIndex As Long, ByRef sOut As String, ByVal fGetMemoryProperties As Integer)		Executes User Function. This function is kept for compatibility reason. It is no longer recommended to use this function

ByVal iCycle as integer -> ByVal lCycleIndex As Long

HImages:
New:

Read only property	pHChkOptAutoLutInROI()	HCheck	
Read only property	pHChkOptZeroLower()	HCheck	
Read/write property	pfOptionsAutoLutInROI()	Integer	

Read/write property	pfOptionsDisplayFWHM()	Integer	
Read/write property	pfOptionsZeroLower()	Integer	
Read/write property	piOptionFWHMNoOfDigits()	Integer	
Read/write property	piOptionFWHMSize()	Integer	
Read/write property	plOptionFWHMColor()	Long	

HSequence
New:

Event	CameraIsStarted()		

Modified:

Event	SeqSingleAcqEnded(ByVal lIndex As Long)		

ByVal iIndex As integer -> ByVal lIndex As Long

HC474298
New:

Read only property	pHChkSetupCooler()	HCheck	
Read only property	pHDisSetupActualTemperature()	HDisp	
Read only property	pHDisSetupConfiguration()	HDisp	
Read only property	pHDisSetupVolGain()	HDisp	
Read only property	pHDisSetupVolOffset()	HDisp	
Read only property	pHEdnSetupTargetTemperature()	HEditNumber	
Read only property	pHFraSetupCameraInfo()	HFrame	
Read only property	pHFraSetupTemperature()	HFrame	
Read only property	pHRadSetupLightMode()	HRadios	
Read only property	pHRadSetupShutterAction()	HRadios	
Read only property	pHRadSetupTriggerSource()	HRadios	

New Object: HDCam

HExternalDevices
New:

Read only property	pHEdnTrigSetExposure()	HEditNumber	

HExternalDevice
New:

Read only property	piNoOfUnknownPar()	Integer	Returns the number of unknown parameters

HAcq
New:

Event	CameraIsStarted()		
Read only property	pfOptionAutoBacksub()	Integer	
Read only property	pHChkOpt32BitInAI()	HCheck	
Read only property	pHChkOptDefectCorrection()	HCheck	
Read only property	pHComDefPixCalculateDefectPixels()	HCommand	
Read only property	pHComDefPixClose()	HCommand	
Read only property	pHComDefPixGetImageDeadPixels()	HCommand	
Read only property	pHComDefPixGetImageHotPixels()	HCommand	
Read only property	pHComDefPixReCalculateDeadPixels()	HCommand	
Read only property	pHComDefPixReCalculateHotPixels()	HCommand	
Read only property	pHComDefPixSaveToINIFile()	HCommand	
Read only property	pHComOptGetDefectPixelFile()	HCommand	
Read only property	pHDisDefPixAverageDeadPixel()	HDisp	
Read only property	pHDisDefPixAverageHotPixel()	HDisp	
Read only property	pHDisDefPixNrDefectColumns()	HDisp	
Read only property	pHDisDefPixNrDefectLines()	HDisp	
Read only property	pHDisDefPixNrDefectPixels()	HDisp	
Read only property	pHDisDefPixStandDevDeadPixel()	HDisp	
Read only property	pHDisDefPixStandDevHotPixel()	HDisp	
Read only property	pHDisDefPixStatus()	HDisp	
Read only property	pHEdnDefPixLineColumnsPercentage()	HEditNumber	
Read only property	pHEdnDefPixThresholdDeadLines()	HEditNumber	
Read only property	pHEdnDefPixThresholdDeadPixels()	HEditNumber	
Read only property	pHEdnDefPixThresholdHotLines()	HEditNumber	
Read only property	pHEdnDefPixThresholdHotPixels()	HEditNumber	
Read only property	pHEstDefPixImageDeadPixel()	HEditString	
Read only property	pHEstDefPixImageHotPixel()	HEditString	
Read only property	pHEstOptDefectPixelFile()	HEditString	
Read only property	pHFraDefPixDeadPixels()	HFrame	

Read only property	pHFraDefPixHotPixels()	HFrame	
Read only property	pHFraDefPixResults()	HFrame	
Read only property	pHRadDefPixMethod()	HRadios	
Read only property	pHWinDefPixDlg()	HWindow	
Read/write property	pfOptionDefectCorrection()	Integer	
Read/write property	pfStartImagesIndividually()	Integer	
Read/write property	psOptionDefectPixelFile()	String	
Sub	pUpdateCameraParms()		

HGrb
New:

Function	Test()		
Sub	pSetHMsgBox(HMsgBox As HMsgBox)		

HCheck
Modified:

Read/write property	pfValue(Optional DoEvent As Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer)	Integer	Sets or returns the value of the object (true if checked and false if unchecked)

Optional parameters added

HEditNumber:
Modified:

Read/write property	pvValue(Optional DoEvent As Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer)	Variant	Returns the value of the object

Optional parameters added

HEditString

Read/write property	psText(Optional DoEvent As Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer)	String	Sets or returns the text of the object

Optional parameters added

HEntry

Read/write property	pftValue(Optional	Single	Sets or returns a numerical value associated to the entry

	DoEvent As Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer)		type. Time values in ms, numerical values or entry indexes are used depending on the EntryType
Read/write property	piEntry(Optional DoEvent As Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer)	Integer	Sets or returns the entry index (Entries are numbered from piEntry=0 to piNrEntries-1)
Read/write property	psKeyValue(Optional DoEvent As Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer)	String	Sets or returns the key value of the object. This is a string value of the entry. When this value is set the all entries are searched whether the contain the specified string.

Optional parameters added

HRadios

Read only property	pfRadioEnabled(ByVal iRadio As Integer, Optional DoEvent As Variant)	Integer	Returns a value which specifies the enabled status of the indicated radiobutton
Read only property	pfRadioVisible(ByVal iRadio As Integer, Optional DoEvent As Variant)	Integer	Returns a value which specifies the enabled status if the indicated radiobutton
Read/write property	piRadio(Optional DoEvent As Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer)	Integer	Sets or returns the index of the currenly selected radiobutton. It is numbered from (piRadio=0 to piNoEntries-1)
Read/write property	piValue(Optional DoEvent As Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer)	Integer	Returns the value which is associated with the currently selected radiobutton
Read/write property	psKeyValue(Optional DoEvent As Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer)	String	Sets or returns the key value of the object. When this value is set the radiobutton with the specified key value is selected

Optional parameters added

HTab

Read only property	pfTabEnabled(ByVal iTab As Integer, Optional DoEvent As	Integer	Returns a value which specifies whether the specified HSingleTab object is enabled

	Variant)		
Read only property	pfTabVisible(ByVal iTab As Integer, Optional DoEvent As Variant)	Integer	Returns a value which specifies whether the specified HSingleTab object is visible
Read/write property	piTab(Optional DoEvent As Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer)	Integer	Sets or returns the index of the currently selected Tab
Read/write property	piValue(Optional DoEvent As Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer)	Integer	Returns the value which is associated with the currently selected Tab
Read/write property	psKeyValue(Optional DoEvent As Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer)	String	Sets or returns the key value of the object. When this value is set the Tab with the specified key value is selected

Optional parameters added

HLut
New:

Read/write property	piAutoOption()	Integer	

IHCall2Appl
Modified

Sub	GetImageStatus(ByVal iImg As Integer, ByVal fIncludeCameraStatus As Integer)		Assigns image status information to the current image according the current status

Added fIncludeCameraStatus

IHCall2Exttrig
Modified:

Sub	SetupExtTrig(ByVal iMode As Integer, ByVal fPolarityPositiv As Integer, ByVal ftInterval As Single)		Setup external triggering

Added iMode, ftInterval

IHCall2LUTControl
New:

Sub	SetZeroLower(ByVal fZeroLower)		

IHCall2Sequence
Modified:

Function	fPSeqGetMask(ByVal sFile As	Integer	Gets the mask string for the specified filename

	String, ByRef sMask As String, ByRef lNr As Long, ByRef iPrf As Integer)		
Function	SeqGetCurrentSample()	Long	Returns the current sample
Sub	SeqDisplaySample (ByVal lNr As Long)		Displays the specified sample
Sub	SeqSingleAcqEnded(lC As Long)		
Sub	ValueExcludedSamples(ByVal lExcludedSamples As Long)		Returns the number of excluded samples
Sub	ValueUndefinedSamples(ByVal lUndefinedSamples As Long)		Returns the number of undefined samples

All references to the number of samples or sample index modified to long instead of integer

Appendix A: All Public Properties/Functions/Subs/Events

This is a complete listing of all public properties, functions, subs and events a client programmer can use. The same information is also available in the object catalog of Visual Basic 6.0 or similar development tools.

HInitHi

Event	ApplicationCreated(ByRef HAppHiPic As HAppHiPic)		Event which is raised when the Application is successfully created. This event returns an object reference to the HAppHipic object
Event	ErrorDueToLicence()		Event which is raised when the application could not be created due to missing licence
Event	ErrorDuringInit()		Event which is raised when the application could not be created due to an error during initialization
Event	InitCancel()		Event which is raised when the application was not created because the creation has been cancelled by user
Event	Message(ByVal sMessage As String)		Event which is raised to indicate processign steps during initialization
Event	MsgBox(ByVal iID As Integer, ByVal sPrompt As String, ByVal sTitle As String, ByVal Style As Integer, ByVal Buttons As Integer, ByRef default As Integer)		Event which is raised in remote control mode when the pfNoDialogs is set to true instead of showing a messagebox. This event can be use for the client program to react on such messages
Event	NoUserIFAllowed()		Event which is raised to indicate that no user I/F is allowed
Read only property	pfApplUserIF()	Integer	Retruns a flag which defines whether the Appliaction will run without User I/F (Default)
Read only property	pfInitStatus()	Integer	Returns a value indicating the current status of initialization. Possible values are defined in the enumeration InitStatus
Read only property	pfInitUserIF()	Integer	Returns a flag which defines whether the Initialisation will run without User I/F (Default)
Read only property	pHAppHipic()	HAppHiPic	Returns an object reference to the HAppHiPic object if successfully created
Read only property	pHComInitCancel()	HCommand	Returns an object reference to the HCommand object associated with the init dialogs 'Cancel' pushbutton
Read only property	pHComInitGetConfigFile()	HCommand	Returns an object reference to the HCommand object associated with the init dialogs 'Get Config File' pushbutton
Read only property	pHComInitOK()	HCommand	Returns an object reference to the HCommand object associated with the init dialogs 'OK' pushbutton
Read only property	pHDisInitBus()	HDisp	
Read only property	pHDisInitCameraID()	HDisp	
Read only property	pHDisInitCameraVersion()	HDisp	
Read only property	pHDisInitDCamAPIVersion()	HDisp	
Read only property	pHDisInitDriverVersion()	HDisp	

Read only property	pHDisInitMessage()	HDisp	Returns an object reference to the HDisp object associated with the init dialogs 'Message' display area
Read only property	pHDisInitModel()	HDisp	
Read only property	pHDisInitModuleVersion()	HDisp	
Read only property	pHDisInitVendor()	HDisp	
Read only property	pHEdnInitC474295ComPort()	HEditNumber	Returns an object reference to the HEditNumber object associated with init dialogs 'C474295 Com Port' editbox
Read only property	pHEdnInitC474298ComPort()	HEditNumber	Returns an object reference to the HEditNumber object associated with init dialogs 'C474298 Com Port' editbox
Read only property	pHEdnInitC488080ComPort()	HEditNumber	Returns an object reference to the HEditNumber object associated with init dialogs 'C488080 Com Port' editbox
Read only property	pHEdnInitC4880ComPort()	HEditNumber	Returns an object reference to the HEditNumber object associated with init dialogs 'C4880 Com Port' editbox
Read only property	pHEdnInitC7300ComPort()	HEditNumber	Returns an object reference to the HEditNumber object associated with init dialogs 'C7300 Com Port' editbox
Read only property	pHEdnInitC800010ComPort()	HEditNumber	Returns an object reference to the HEditNumber object associated with init dialogs 'C8000-10 Com Port' editbox
Read only property	pHEdnInitC800020ComPort()	HEditNumber	Returns an object reference to the HEditNumber object associated with init dialogs 'C8000-20 Com Port' editbox
Read only property	pHEdnInitGrabberSysNo()	HEditNumber	Returns an object reference to the HEditNumber object associated with init dialogs 'Grabber System Number' editbox
Read only property	pHEdnInitOrcaHRComPort()	HEditNumber	
Read only property	pHEntInitModelNr()	HEntry	
Read only property	pHEstInitConfiFile()	HEditString	Returns an object reference to the HEditString object associated with the init dialogs 'Config File' editbox
Read only property	pHFraInitCamera()	HFrame	Returns an object reference to the HFrame object associated with the init dialogs 'Camera' frame
Read only property	pHFraInitDCamAPI()	HFrame	
Read only property	pHRadInitCamera()	HRadios	Returns an object reference to the HRadios object associated with the init dialogs 'Camera' radiobutton group
Read only property	pHRadInitCCDCameraAccess()	HRadios	
Read only property	pHRadInitFrameGrabber()	HRadios	Returns an object reference to the HRadios object associated with the init dialogs 'Frame Grabber' group
Read only property	pHWinInitDlg()	HWindow	Returns an object reference to the HWindow object associated with the init dialogs main window
Read only property	pHWinInitLogo()	HWindow	Returns an object reference to the HWindow object associated with the init dialogs logo picture box
Read only property	piApplicationType()	Integer	Returns the application type. Possible values are defined in the enumeration ApplicationType
Read only property	piStartStatus()	Integer	Returns the current start status of the component. StartStatusNone=newly started, StartStatusInit=init dialog is shown, StartStatusRunning=running
Read only property	psAppDate()	Integer	Returns the applications date
Read only property	psApplicationDirectory()	String	Returns the application directory
Read only property	psAppTitle()	String	Returns the title of the application
Read only property	psAppTitleLong()	String	Returns the long version of the title of the application
Read only property	psIniFile()	String	Sets or returns the name of the *.ini file used for storage of permanent parameters
Read only property	psSoftwareVersion()	String	Returns the software version
Read only property	pStartHAppHiPic()	HAppHiPic	Returns an object reference to the HAppHiPic object if piStartStatus = StartStatusRunning

Read only property	pStartHInitHi()	HInitHi	Returns an object reference to the HInitHi object if piStartStatus = StartStatusInit
Read only property	psWindowsDirectory()	String	Returns the Windows directory of the computer where the component runs
Function	pfReadFile(ByVal sFileName As String, bArray()	Byte, ByRef lLength As Long, Optional sError As Variant)	Reads the content of a file stored at the remote computer
Function	pfWriteFile(ByVal sFileName As String, bArray()	Byte, ByRef lLength As Long, ByVal fDontOverwrite As Integer, Optional sError As Variant)	Writes a file at the remote computer
Function	piInit(ByVal sIniFile As String, ByVal fInitUserIF As Integer, ByVal fApplUserIF As Integer, ByVal fNoDialogs As Integer)	Integer	Initializes the HInit object. This places the INIT dialog on screen if started with InitUserIF
Sub	pAsyncStartProg()		Starts the program asynchronously
Sub	pGetLicenceKeys(ByRef fApplicationKeyFound%, ByRef fLicenceAcquire%, ByRef fLicenceFitting%, ByRef fLicenceRCOnly%, ByRef fLicenceSave%, ByRef fLicenceSequence%, ByRef fLicenceTransAbs%)		Returns information about all licence keys which could be found
Sub	pMode(ByVal s As String, ByVal ft As Single)		

HAppHiPic

Event	Message(ByVal s As String)		Event which is raised to inform the user of ongoing operations
Read only property	pfCommPortOpen()	Integer	Returns a value which defines whether the communication port could be opened succesfully for camera control
Read only property	pHACam()	HACam	
Read only property	pHAcq()	HAcq	Returns an object reference to the Acquisition object (HAcq)
Read only property	pHAsyncCommand()	HAsyncCommand	Returns an object reference to the Asynchronous Command object (HAsyncCommand)

Read only property	pHC474295()	HC474295	Returns an object reference to the C4742-95 Camera object (HC474295) If this camera is not used NOTHING is returned
Read only property	pHC474298()	HC474298	Returns an object reference to the C4742-98 Camera object (HC474298). If this camera is not used NOTHING is returned
Read only property	pHC4880()	HC4880	Returns an object reference to the C4880 Camera object (HC4880). If this camera is not used NOTHING is returned
Read only property	pHC488080()	HC488080	Returns an object reference to the C4880-80 Camera object (HC488080). If this camera is not used NOTHING is returned
Read only property	pHC7300()	HC7300	Returns an object reference to the C7300 Camera object (HC7300). If this camera is not used NOTHING is returned
Read only property	pHC800010()	HC800010	Returns an object reference to the C800010 Camera object (HC800010). If this camera is not used NOTHING is returned
Read only property	pHC800020()	HC800020	Returns an object reference to the C800020 Camera object
Read only property	pHChkMainLUTTool()	HCheck	Returns an object reference to the HCheck object associated with the main dialogs 'LUT Tool' checkbox
Read only property	pHChkMainPRF()	HCheck	Returns an object reference to the HCheck object associated with the main dialogs 'Quick Profile' checkbox
Read only property	pHChkMainProfile()	HCheck	Returns an object reference to the HCheck object associated with the main dialogs 'Profile' checkbox
Read only property	pHChkMainROI()	HCheck	Returns an object reference to the HCheck object associated with the main dialogs 'ROI I/F' checkbox
Read only property	pHChkOptLUTTool()	HCheck	Returns an object reference to the HCheck object associated with the option dialogs 'LUT Tool' checkbox
Read only property	pHChkOptRestoreWindowPos()	HCheck	Returns an object reference to the HCheck object associated with the option dialogs 'Restore Window Position' checkbox
Read only property	pHChkOptUserFunctions()	HCheck	Returns an object reference to the HCheck object associated with the option dialogs 'User Functions' checkbox
Read only property	pHComAllOptCancel()	HCommand	Returns an object reference to the HCommand object associated with the all options dialogs 'Cancel' pushbutton
Read only property	pHComAllOptOK()	HCommand	Returns an object reference to the HCommand object associated with the all options dialogs 'OK' pushbutton
Read only property	pHComMainAcquire()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Acquire' pushbutton
Read only property	pHComMainAI()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Analog Integration' pushbutton
Read only property	pHComMainAutoLUT()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Auto LUT' pushbutton
Read only property	pHComMainBacksub()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Backsub' pushbutton
Read only property	pHComMainFreeze()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Freeze' pushbutton
Read only property	pHComMainInfo()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Info' pushbutton
Read only property	pHComMainLive()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Live' pushbutton
Read only property	pHComMainOpen()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs ""Open"" pushbutton
Read only property	pHComMainPC()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Photon Counting'

			pushbutton
Read only property	pHComMainSave()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Save' pushbutton
Read only property	pHComMainShading()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Shading correction' pushbutton
Read only property	pHComOptBackgroundColor()	HCommand	Returns an object reference to the HCommand object associated with the options dialogs 'Background Color' pushbutton
Read only property	pHComOptCancel()	HCommand	Returns an object reference to the HCommand object associated with the options dialogs 'Cancel' pushbutton
Read only property	pHComOptOK()	HCommand	Returns an object reference to the HCommand object associated with the options dialogs 'OK' pushbutton
Read only property	pHDCam()	HDCam	
Read only property	pHDisMainMessage()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Message' display area
Read only property	pHDisMainTemperature()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Temperature' display area
Read only property	pHFlatPanel()	HFlatPanel	Returns an object reference to the FlatPanel Camera object (HFlatPanel). If this camera is not used NOTHING is returned
Read only property	pHImages()	HImages	Returns an object reference to the Images object (HImages)
Read only property	pHLicence()	HLicence	
Read only property	pHLUTControl()	HLUTControl	Returns an object reference to the LUT Control object (HLUTControl)
Read only property	pHMenMain3DData()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs '3DData' menu entry
Read only property	pHMenMainAbout()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'About' menu entry
Read only property	pHMenMainAcquire()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Acquire' menu entry
Read only property	pHMenMainAcquisition()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Acquisition' menu
Read only property	pHMenMainAnalogIntegration()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Analog Integration' menu entry
Read only property	pHMenMainAnalysis()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Analysis' menu
Read only property	pHMenMainArithmetic()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Arithmetic' menu entry
Read only property	pHMenMainBackgroundSubtraction()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Background Subtraction' menu entry
Read only property	pHMenMainBlank1()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Blank1' menu entry
Read only property	pHMenMainBlank2()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Blank2' menu entry
Read only property	pHMenMainBlank3()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Blank3' menu entry
Read only property	pHMenMainBlank6()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Blank6' menu entry
Read only property	pHMenMainBlank7()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Blank7' menu entry
Read only property	pHMenMainCamera()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Camera Setup' menu entry
Read only property	pHMenMainClearI	HMenu	Returns an object reference to the HMenu object

	mage()		associated with the main dialogs 'Clear Image' menu entry
Read only property	pHMenMainCloseAll()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Close All' menu entry
Read only property	pHMenMainCorrections()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Corrections' menu
Read only property	pHMenMainCorrectionSetup()	HMenu	
Read only property	pHMenMainCreateScalingFiles()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Create Scaling Files' menu entry
Read only property	pHMenMainDefectPixelCorrectionSetup()	HMenu	
Read only property	pHMenMainDisplay()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Display' menu
Read only property	pHMenMainExit()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Exit' menu entry
Read only property	pHMenMainFile()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'File' menu
Read only property	pHMenMainFreeze()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Freeze' menu
Read only property	pHMenMainHistogram()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Histogram' menu entry
Read only property	pHMenMainImageStatus()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Image Status' menu entry
Read only property	pHMenMainInfo()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Info' menu
Read only property	pHMenMainLive()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Live' menu
Read only property	pHMenMainLUT()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'LUT' menu entry
Read only property	pHMenMainMapLUT()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Map Values By LUT' menu entry
Read only property	pHMenMainOpen()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Open' menu entry
Read only property	pHMenMainOptions()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Options' menu entry
Read only property	pHMenMainPhotonCounting()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Photon Counting' menu entry
Read only property	pHMenMainPrint()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Print' menu entry
Read only property	pHMenMainProcessing()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Processing' menu
Read only property	pHMenMainProfile()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Profile' menu entry
Read only property	pHMenMainRS232()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'RS232' menu entry
Read only property	pHMenMainSaveAs()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Save As' menu entry
Read only property	pHMenMainSaveROIAs()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Save ROI As' menu entry
Read only property	pHMenMainScaling()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Scaling' menu entry
Read only property	pHMenMainSequence()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Sequence' menu entry

Read only property	pHMenMainSetup()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Setup' menu
Read only property	pHMenMainShadingCorrection()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Shading Correction' menu entry
Read only property	pHMenMainSuperimpose()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Superimpose' menu entry
Read only property	pHMenMainUserFunction()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'User Function' menu entry
Read only property	pHMenMainWindow()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window' menu
Read only property	pHMenMainWindowFile00()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 0' menu entry
Read only property	pHMenMainWindowFile01()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 1' menu entry
Read only property	pHMenMainWindowFile02()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 2' menu entry
Read only property	pHMenMainWindowFile03()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 3' menu entry
Read only property	pHMenMainWindowFile04()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 4' menu entry
Read only property	pHMenMainWindowFile05()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 5' menu entry
Read only property	pHMenMainWindowFile06()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 6' menu entry
Read only property	pHMenMainWindowFile07()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 7' menu entry
Read only property	pHMenMainWindowFile08()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 8' menu entry
Read only property	pHMenMainWindowFile09()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 9' menu entry
Read only property	pHMenMainWindowFile10()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 10' menu entry
Read only property	pHMenMainWindowFile11()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 11' menu entry
Read only property	pHMenMainWindowFile12()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 12' menu entry
Read only property	pHMenMainWindowFile13()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 13' menu entry
Read only property	pHMenMainWindowFile14()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 14' menu entry
Read only property	pHMenMainWindowFile15()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 15' menu entry

Read only property	pHMenMainWindowFile16()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 16' menu entry
Read only property	pHMenMainWindowFile17()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 17' menu entry
Read only property	pHMenMainWindowFile18()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 18' menu entry
Read only property	pHMenMainWindowFile19()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 19' menu entry
Read only property	pHMessageBox()	HMsgBox	Returns an object reference to the Message Box object (HMessageBox)
Read only property	pHRadMainROITools()	HRadios	Returns an object reference to the HRadios object associated with the main dialogs 'ROI Tools' radiobutton group
Read only property	pHSequence()	HSequence	Returns an object reference to the Sequence object (HSequence)
Read only property	pHSystemScaling()	HSystemScaling	Returns an object reference to the System Scaling object (HSystemScaling)
Read only property	pHTabAllOpt()	HTab	Returns an object reference to the HTab object associated with the all options dialogs 'Option' tab group
Read only property	pHWinAllOptAcq()	HWindow	Returns an object reference to the HWindow object associated with the all options acquisition picture box
Read only property	pHWinAllOptDlg()	HWindow	Returns an object reference to the HWindow object associated with the all options main window
Read only property	pHWinAllOptGeneral()	HWindow	Returns an object reference to the HWindow object associated with the all options general picture box
Read only property	pHWinAllOptImages()	HWindow	Returns an object reference to the HWindow object associated with the all options images picture box
Read only property	pHWinAllOptSequence()	HWindow	Returns an object reference to the HWindow object associated with the all options sequence picture box
Read only property	pHWinMainClientArea()	HWindow	Returns an object reference to the HWindow object associated with the main windows client area picture box
Read only property	pHWinMainDlg()	HWindow	Returns an object reference to the HWindow object associated with the main dialogs main window
Read only property	pHWinOptDlg()	HWindow	Returns an object reference to the HWindow object associated with the options main window
Read only property	pHWinOptOptions()	HWindow	Returns an object reference to the HWindow object associated with the options dialogs options picturebox
Read only property	piAcquisitionModule()	Integer	Returns the computers acquisition module. Possible values are defined in the enumeration AcquisitionModule.
Read only property	piApplicationType()	Integer	Returns the application type. Possible values are defined in the enumeration ApplicationType
Read only property	piCameraTyp()	Integer	Returns the camera type. Possible values are defined in the enumeration CameraType.
Read only property	piFrameGrabber()	Integer	Returns the computers frame grabber. Possible values are defined in the enumeration FrameGrabber.
Read only property	piOperatingSystem()	Integer	Returns the computers frame grabber. Possible values are defined in the enumeration FrameGrabber.
Read only property	pobjHCamera()	Object	Returns an object reference to the camera object. Its type is dependent on the used camera
Read only property	psAppDate()	String	Returns the application date.
Read only property	psApplicationDirectory()	String	Returns the application directory
Read only property	psApplicationString()	String	Returns a string identifying the application
Read only property	psAppTitle()	String	Returns the title of the application

Read only property	psAppTitleLong()	String	Returns the long version of the title of the application
Read only property	psConfigFile()	String	Sets or returns the frame grabbers configuration file
Read only property	psIniFile()	String	Sets or returns the name of the *.ini file used for storage of permanent parameters
Read only property	psSoftwareVersion()	String	Returns the software version
Read only property	psWindowsDirectory()	String	Returns the Windows directory of the computer where the component runs
Read/write property	pfOptionLUTTool()	Integer	Sets or returns a value which defines whether the LUT tool should be shown on the main window
Read/write property	pfOptionRestoreWindowPos()	Integer	Sets or returns a value which defines whether window positions should be restored or not when the window is reactivated
Read/write property	pfOptionUserFunctions()	Integer	Sets or returns a value which defines whether the User function is called or not
Read/write property	pfRestoreWindowPos()	Integer	Sets or returns a value which defines whether the window position, its size and window state is restored when the window is displayed again after it has been closed
Read/write property	pfUserIF()	Integer	Sets or returns a value which defines whether the associated window should be shown on screen. A window is only displayed under the following condition: pfUserIF=TRUE, pfVisible=TRUE, pfHideForm=FALSE
Read/write property	plOptionBackgroundColor()	Long	Sets or returns a value which defines the background color
Function	pfReadFile(ByVal sFileName As String, bArray()	Byte, ByRef lLength As Long, Optional sError As Variant)	Reads the content of a file stored at the remote computer
Function	pfWriteFile(ByVal sFileName As String, bArray()	Byte, ByRef lLength As Long, ByVal fDontOverwrite As Integer, Optional sError As Variant)	Writes a file at the remote computer
Function	piEndProg()	Integer	Ends the program. This should be the last method in the application program.
Function	psErrorString(ByVal Index As Integer)	String	Returns a description of the Error related to error No. Index
Function	psGetStatus()	String	Returns the current application status string

HInitTa

Event	ApplicationCreated(ByRef HAppHPDTA As HAppHPDTA)		Event which is raised when the Application is successfully created. This event returns an object reference to the HAppHPDTA object
Event	ErrorDueToLicence()		Event which is raised when the application could not be created due to missing licence
Event	ErrorDuringInit()		Event which is raised when the application could not be created due to an error during initialization
Event	InitCancel()		Event which is raised when the application was not created because the creation has been cancelled by user

Event	Message(ByVal sMessage As String)		Event which is raised to indicate processign steps during initialization
Event	MsgBox(ByVal iID As Integer, ByVal sPrompt As String, ByVal sTitle As String, ByVal Style As Integer, ByVal Buttons As Integer, ByRef default As Integer)		Event which is raised in remote control mode when the pfNoDialogs is set to true instead of showing a \r\r\nmessagebox. This event can be use for the client program to react on such messages
Event	NoUserIFAllowed()		Event which is raised to indicate that no user I/F is allowed
Read only property	pfApplUserIF()	Integer	Retruns a flag which defines whether the Appliaction will run without User I/F (Default)
Read only property	pfInitStatus()	Integer	Returns a value indicating the current status of initialization. Possible values are defined in the enumeration InitStatus
Read only property	pfInitUserIF()	Integer	Retruns a flag which defines whether the Initialisation will run without User I/F (Default)
Read only property	pHAppHPDTA()	HAppHPDTA	Returns an object reference to the HAppHPDTA object if successfully created
Read only property	pHComCCDSetCancel()	HCommand	
Read only property	pHComCCDSetGetConfigFile()	HCommand	Returns an object reference to the HCommand object associated with the init dialogs 'Get Config File' pushbutton
Read only property	pHComCCDSetOK()	HCommand	
Read only property	pHComInitCancel()	HCommand	Returns an object reference to the HCommand object associated with the init dialogs 'Cancel' pushbutton
Read only property	pHComInitOK()	HCommand	Returns an object reference to the HCommand object associated with the init dialogs 'OK' pushbutton
Read only property	pHComInitSetupCCD()	HCommand	Returns an object reference to the HCommand object associated with the init dialogs 'Setup CCD' pushbutton
Read only property	pHComInitSetupStreakDevices()	HCommand	Returns an object reference to the HCommand object associated with the init dialogs 'Setup Streak Devices' pushbutton
Read only property	pHDisCCDSetBus()	HDisp	
Read only property	pHDisCCDSetCameraID()	HDisp	
Read only property	pHDisCCDSetCameraVersion()	HDisp	
Read only property	pHDisCCDSetDCamAPIVersion()	HDisp	
Read only property	pHDisCCDSetDriverVersion()	HDisp	
Read only property	pHDisCCDSetModel()	HDisp	
Read only property	pHDisCCDSetModuleVersion()	HDisp	
Read only property	pHDisCCDSetVendor()	HDisp	
Read only property	pHDisInitCCDCamera()	HDisp	Returns an object reference to the HDisp object associated with the init dialogs 'CCD Camera' display area
Read only property	pHDisInitDevice1()	HDisp	Returns an object reference to the HDisp object associated with the init dialogs 'Device 1' display area

Read only property	pHDisInitDevice2()	HDisp	Returns an object reference to the HDisp object associated with the init dialogs 'Device 2' display area
Read only property	pHDisInitDevice3()	HDisp	Returns an object reference to the HDisp object associated with the init dialogs 'Device 3' display area
Read only property	pHDisInitDevice4()	HDisp	Returns an object reference to the HDisp object associated with the init dialogs 'Device 4' display area
Read only property	pHDisInitGPIBAddr1()	HDisp	Returns an object reference to the HDisp object associated with the init dialogs GPIB Address 1' display area
Read only property	pHDisInitGPIBAddr2()	HDisp	Returns an object reference to the HDisp object associated with the init dialogs GPIB Address 2' display area
Read only property	pHDisInitGPIBAddr3()	HDisp	Returns an object reference to the HDisp object associated with the init dialogs GPIB Address 3' display area
Read only property	pHDisInitGPIBAddr4()	HDisp	Returns an object reference to the HDisp object associated with the init dialogs GPIB Address 4' display area
Read only property	pHDisInitMessage()	HDisp	Returns an object reference to the HDisp object associated with the init dialogs 'Message' display area
Read only property	pHEdnCCDSetC474295ComPort()	HEditNumber	Returns an object reference to the HEditNumber object associated with init dialogs 'C474295 Com Port' editbox
Read only property	pHEdnCCDSetC474298ComPort()	HEditNumber	Returns an object reference to the HEditNumber object associated with init dialogs 'C474298 Com Port' editbox
Read only property	pHEdnCCDSetC488080ComPort()	HEditNumber	Returns an object reference to the HEditNumber object associated with init dialogs 'C4880-8X Com Port' editbox
Read only property	pHEdnCCDSetC4880ComPort()	HEditNumber	Returns an object reference to the HEditNumber object associated with init dialogs 'C4880 Com Port' editbox
Read only property	pHEdnCCDSetGrabberSysNo()	HEditNumber	Returns an object reference to the HEditNumber object associated with init dialogs 'Board number' editbox (Also referred to system number)
Read only property	pHEdnCCDSetOrcaHRComPort()	HEditNumber	
Read only property	pHEntCCDSetModelNr()	HEntry	
Read only property	pHEstCCDSetConfiFile()	HEditString	Returns an object reference to the HEditString object associated with the init dialogs 'Config File' editbox
Read only property	pHExternalDevices()	HExternalDevices	
Read only property	pHFraCCDSetCamera()	HFrame	Returns an object reference to the HFrame object associated with the init dialogs CCD Setup' frame
Read only property	pHFraCCDSetDCamAPI()	HFrame	
Read only property	pHFraInitCamera()	HFrame	Returns an object reference to the HFrame object associated with the init dialogs 'Camera' frame
Read only property	pHFraInitStreakDevices()	HFrame	Returns an object reference to the HFrame object associated with the init dialogs 'Streak devices' frame
Read only property	pHRadCCDSetCamera()	HRadios	Returns an object reference to the HRadios object associated with the init dialogs 'Camera' radiobutton group
Read only property	pHRadCCDSetCCDCameraAccess()	HRadios	
Read only property	pHRadCCDSetFrameGrabber()	HRadios	Returns an object reference to the HRadios object associated with the init dialogs 'Frame Grabber' group
Read only property	pHWinInitDlg()	HWindow	Returns an object reference to the HWindow object associated with the init dialogs main window
Read only property	pHWinInitLogo()	HWindow	Returns an object reference to the HWindow object associated with the init dialogs logo picture box
Read only property	piApplicationType	Integer	Returns the application type. Possible values are defined

	()		in the enumeration ApplicationType
Read only property	piStartStatus()	Integer	Returns the current start status of the component. StartStatusNone=newly started, StartStatusInit=init dialog is shown, StartStatusRunning=running
Read only property	psAppDate()	Integer	Returns the applications date
Read only property	psApplicationDirectory()	String	Returns the application directory
Read only property	psAppTitle()	String	Returns the title of the application
Read only property	psAppTitleLong()	String	Returns the long version of the title of the application
Read only property	psIniFile()	String	Sets or returns the name of the *.ini file used for storage of permanent parameters
Read only property	psSoftwareVersion()	String	Returns the software version
Read only property	pStartHAppHPDTA()	HAppHPDTA	Returns an object reference to the HAppHPDTA object if piStartStatus = StartStatusRunning
Read only property	pStartHInitTa()	HInitTa	Returns an object reference to the HInitTA object if piStartStatus = StartStatusInit
Read only property	psWindowsDirectory()	String	Returns the Windows directory of the computer where the component runs
Function	pfReadFile(ByVal sFileName As String, bArray()	Byte, ByRef lLength As Long, Optional sError As Variant)	Reads the content of a file stored at the remote computer
Function	pfWriteFile(ByVal sFileName As String, bArray()	Byte, ByRef lLength As Long, ByVal fDontOverwrite As Integer, Optional sError As Variant)	Writes a file at the remote computer
Function	piInit(ByVal sIniFile As String, ByVal fInitUserIF As Integer, ByVal fApplUserIF As Integer, ByVal fNoDialogs As Integer)	Integer	Initializes the HInit object. This places the INIT dialog on screen if started with InitUserIF
Sub	pAsyncStartProg()		Starts the program asynchronously
Sub	pGetLicenceKeys(ByRef fApplicationKeyFound%, ByRef fLicenceAcquire%, ByRef fLicenceFitting%, ByRef fLicenceRCOnly%, ByRef fLicenceSave%, ByRef fLicenceSequence%, ByRef fLicenceTransAbs		Returns information about all licence keys which could be found

	%)		
Sub	pMode(ByVal s As String, ByVal ft As Single)		

HAppHPDTA

Event	Message(ByVal s As String)		Event which is raised to inform the user of ongoing operations
Read only property	pfCommPortOpen()	Integer	Returns a value which defines whether the communication port could be opened succesfully for camera control
Read only property	pHACam()	HACam	Returns an object reference to the Analog Camera object (HACam). If this camera is not used NOTHING is returned
Read only property	pHAcq()	HAcq	Returns an object reference to the Acquisition object (HAcq)
Read only property	pHAsyncCommand()	HAsyncCommand	Returns an object reference to the Asynchronous Command object (HAsyncCommand)
Read only property	pHC474295()	HC474295	Returns an object reference to the C4742-95 Camera object (HC474295) If this camera is not used NOTHING is returned
Read only property	pHC474298()	HC474298	Returns an object reference to the C474298 Camera object (HC474298). If this camera is not used NOTHING is returned
Read only property	pHC4880()	HC4880	Returns an object reference to the C4880 Camera object (HC4880). If this camera is not used NOTHING is returned
Read only property	pHC488080()	HC488080	Returns an object reference to the C4880-80 Camera object (HC488080). If this camera is not used NOTHING is returned
Read only property	pHChkMainLUTTool()	HCheck	Returns an object reference to the HCheck object associated with the main dialogs 'LUT Tool' checkbox
Read only property	pHChkMainPRF()	HCheck	Returns an object reference to the HCheck object associated with the main dialogs 'Quick Profile' checkbox
Read only property	pHChkMainProfile()	HCheck	Returns an object reference to the HCheck object associated with the main dialogs 'Profile' checkbox
Read only property	pHChkMainROI()	HCheck	Returns an object reference to the HCheck object associated with the main dialogs 'ROI I/F' checkbox
Read only property	pHChkOptLUTTool()	HCheck	Returns an object reference to the HCheck object associated with the option dialogs 'LUT Tool' checkbox
Read only property	pHChkOptRestoreWindowPos()	HCheck	Returns an object reference to the HCheck object associated with the option dialogs 'Restore Window Position' checkbox
Read only property	pHChkOptUserFunctions()	HCheck	Returns an object reference to the HCheck object associated with the option dialogs 'User Functions' checkbox
Read only property	pHComAllOptCancel()	HCommand	Returns an object reference to the HCommand object associated with the all options dialogs 'Cancel' pushbutton
Read only property	pHComAllOptOK()	HCommand	Returns an object reference to the HCommand object associated with the all options dialogs 'OK' pushbutton
Read only property	pHComMainAcquire()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Acquire' pushbutton
Read only property	pHComMainAI()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Analog Integration' pushbutton
Read only property	pHComMainAutoLUT()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Auto LUT' pushbutton
Read only property	pHComMainBacksub()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Backsub' pushbutton

Read only property	pHComMainFreeze()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Freeze' pushbutton
Read only property	pHComMainInfo()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Info' pushbutton
Read only property	pHComMainLive()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Live' pushbutton
Read only property	pHComMainOpen()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Open' pushbutton
Read only property	pHComMainPC()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Photon Counting' pushbutton
Read only property	pHComMainRecall1()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Recall1' pushbutton
Read only property	pHComMainRecall2()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Recall2' pushbutton
Read only property	pHComMainSave()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Save' pushbutton
Read only property	pHComMainSave1()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Save1' pushbutton
Read only property	pHComMainSave2()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Save2' pushbutton
Read only property	pHComMainShading()	HCommand	Returns an object reference to the HCommand object associated with the main dialogs 'Shading correction' pushbutton
Read only property	pHComOptBackgroundColor()	HCommand	Returns an object reference to the HCommand object associated with the options dialogs 'Background Color' pushbutton
Read only property	pHComOptCancel()	HCommand	Returns an object reference to the HCommand object associated with the options dialogs 'Cancel' pushbutton
Read only property	pHComOptOK()	HCommand	Returns an object reference to the HCommand object associated with the options dialogs 'OK' pushbutton
Read only property	pHDCam()	HDCam	
Read only property	pHDisMainGateMode()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Gate Mode' display area
Read only property	pHDisMainMCPGain()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'MCP Gain' display area
Read only property	pHDisMainMessage()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Message' display area
Read only property	pHDisMainMode()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Mode' display area
Read only property	pHDisMainPlugin()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Plugin' display area
Read only property	pHDisMainShutter()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Shutter' display area
Read only property	pHDisMainStreakCamera()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Streak Camera' display area
Read only property	pHDisMainTemperature()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Temperature' display area
Read only property	pHDisMainTimeRange()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Time Range' display area
Read only property	pHExternalDevices()	HExternalDevices	Returns an object reference to the External Devices object (HExternalDevices)

Read only property	pHImages()	HImages	Returns an object reference to the Images object (HImages)
Read only property	pHLicence()	HLicence	Returns an object reference to the Licence object (HLicence)
Read only property	pHLUTControl()	HLUTControl	Returns an object reference to the LUT Control object (HLUTControl)
Read only property	pHMenMain3DData()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs '3DData' menu entry
Read only property	pHMenMainAbout()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'About' menu entry
Read only property	pHMenMainAcquire()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Acquire' menu entry
Read only property	pHMenMainAcquisition()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Acquisition' menu
Read only property	pHMenMainAnalogIntegration()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Analog Integration' menu entry
Read only property	pHMenMainAnalysis()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Analysis' menu
Read only property	pHMenMainArithmetic()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Arithmetic' menu entry
Read only property	pHMenMainBackgroundSubtraction()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Background Subtraction' menu entry
Read only property	pHMenMainBlank1()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Blank1' menu entry
Read only property	pHMenMainBlank2()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Blank2' menu entry
Read only property	pHMenMainBlank3()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Blank3' menu entry
Read only property	pHMenMainBlank6()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Blank6' menu entry
Read only property	pHMenMainBlank7()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Blank7' menu entry
Read only property	pHMenMainCamera()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Camera Setup' menu entry
Read only property	pHMenMainClearImage()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Clear Image' menu entry
Read only property	pHMenMainCloseAll()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Claose All' menu entry
Read only property	pHMenMainCorrections()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Corrections' menu
Read only property	pHMenMainCorrectionSetup()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Correction Setup' menu entry
Read only property	pHMenMainCreateScalingFiles()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Create Scaling Files' menu entry
Read only property	pHMenMainCurvatureCorrection()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Curvature Correction' menu entry
Read only property	pHMenMainCurvatureCorrectionSetup()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Curvature Correction Setup' menu entry
Read only property	pHMenMainDefectPixelCorrectionSetup()	HMenu	
Read only property	pHMenMainDevic	HMenu	Returns an object reference to the HMenu object

	eControlOptions()		associated with the main dialogs 'Device Control Options' menu entry
Read only property	pHMenMainDisplay()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Display' menu
Read only property	pHMenMainEmergencyOff()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Emergency Off' menu entry
Read only property	pHMenMainExit()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Exit' menu entry
Read only property	pHMenMainFile()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'File' menu
Read only property	pHMenMainFitting()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Fitting' menu
Read only property	pHMenMainFraming2Sequence()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Framing To Sequence' menu
Read only property	pHMenMainFramingSetup()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Framing Setup' menu
Read only property	pHMenMainFreeze()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Freeze' menu
Read only property	pHMenMainHistogram()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Histogram' menu entry
Read only property	pHMenMainImageStatus()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Image Status' menu entry
Read only property	pHMenMainInfo()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Info' menu
Read only property	pHMenMainLive()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Live' menu
Read only property	pHMenMainLUT()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'LUT' menu entry
Read only property	pHMenMainMapLUT()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Map Values By LUT' menu entry
Read only property	pHMenMainOpen()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Open' menu entry
Read only property	pHMenMainOptions()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Options' menu entry
Read only property	pHMenMainPhotonCounting()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Photon Counting' menu entry
Read only property	pHMenMainPrint()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Print' menu entry
Read only property	pHMenMainProcessing()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Processing' menu
Read only property	pHMenMainProfile()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Profile' menu entry
Read only property	pHMenMainRS232()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'RS232' menu entry
Read only property	pHMenMainSaveAs()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Save As' menu entry
Read only property	pHMenMainSaveROIAs()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Save ROI As' menu
Read only property	pHMenMainScaling()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Scaling' menu entry
Read only property	pHMenMainSequence()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Sequence' menu entry
Read only property	pHMenMainSetup()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Setup' menu
Read only property	pHMenMainShadi	HMenu	Returns an object reference to the HMenu object

	ngCorrection()		associated with the main dialogs 'Shading Correction' menu entry
Read only property	pHMenMainShowDelay2StatusControl()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Show Delay2 Status/Control' menu entry
Read only property	pHMenMainShowDelayStatusControl()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Show Delay Status/Control' menu entry
Read only property	pHMenMainShowSpecStatusControl()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Show Spec Status/Control' menu entry
Read only property	pHMenMainShowStreakStatusControl()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Show Streak Status/Control' menu entry
Read only property	pHMenMainSuperimpose()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Superimpose' menu entry
Read only property	pHMenMainTransientAbsorption()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Transient Absorption' menu entry
Read only property	pHMenMainTriggerSetup()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Trigger Setup' menu entry
Read only property	pHMenMainUserFunction()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'User Function' menu entry
Read only property	pHMenMainWindow()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window' menu
Read only property	pHMenMainWindowFile00()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 0' menu entry
Read only property	pHMenMainWindowFile01()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 1' menu entry
Read only property	pHMenMainWindowFile02()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 2' menu entry
Read only property	pHMenMainWindowFile03()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 3' menu entry
Read only property	pHMenMainWindowFile04()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 4' menu entry
Read only property	pHMenMainWindowFile05()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 5' menu entry
Read only property	pHMenMainWindowFile06()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 6' menu entry
Read only property	pHMenMainWindowFile07()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 7' menu entry
Read only property	pHMenMainWindowFile08()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 8' menu entry
Read only property	pHMenMainWindowFile09()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 9' menu entry
Read only property	pHMenMainWindowFile10()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 10' menu entry

Read only property	pHMenMainWindowFile11()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 11' menu entry
Read only property	pHMenMainWindowFile12()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 12' menu entry
Read only property	pHMenMainWindowFile13()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 13' menu entry
Read only property	pHMenMainWindowFile14()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 14' menu entry
Read only property	pHMenMainWindowFile15()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 15' menu entry
Read only property	pHMenMainWindowFile16()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 16' menu entry
Read only property	pHMenMainWindowFile17()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 17' menu entry
Read only property	pHMenMainWindowFile18()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 18' menu entry
Read only property	pHMenMainWindowFile19()	HMenu	Returns an object reference to the HMenu object associated with the main dialogs 'Window File 19' menu entry
Read only property	pHMessageBox()	HMsgBox	Returns an object reference to the Message Box object (HMessageBox)
Read only property	pHRadMainROITools()	HRadios	Returns an object reference to the HRadios object associated with the main dialogs 'ROI Tools' radiobutton group
Read only property	pHSequence()	HSequence	Returns an object reference to the Sequence object (HSequence)
Read only property	pHSystemScaling()	HSystemScaling	Returns an object reference to the System Scaling object (HSystemScaling)
Read only property	pHTabAllOpt()	HTab	Returns an object reference to the HTab object associated with the all options dialogs 'Option' tab group
Read only property	pHWinAllOptAcq()	HWindow	Returns an object reference to the HWindow object associated with the all options acquisition picture box
Read only property	pHWinAllOptDlg()	HWindow	Returns an object reference to the HWindow object associated with the all options main window
Read only property	pHWinAllOptGeneral()	HWindow	Returns an object reference to the HWindow object associated with the all options general picture box
Read only property	pHWinAllOptImages()	HWindow	Returns an object reference to the HWindow object associated with the all options images picture box
Read only property	pHWinAllOptSequence()	HWindow	Returns an object reference to the HWindow object associated with the all options sequence picture box
Read only property	pHWinMainClientArea()	HWindow	Returns an object reference to the HWindow object associated with the main windows client area picture box
Read only property	pHWinMainDlg()	HWindow	Returns an object reference to the HWindow object associated with the main dialogs main window
Read only property	pHWinOptDlg()	HWindow	Returns an object reference to the HWindow object associated with the options main window
Read only property	pHWinOptOptions()	HWindow	Returns an object reference to the HWindow object associated with the options dialogs options picturebox
Read only property	piAcquisitionModule()	Integer	Returns the computers acquisition module. Possible values are defined Returns the computers acquisition module. Possible values are defined in the enumeration AcquisitionModule.
Read only property	piApplicationType	Integer	Returns the application type. Possible values are defined

	()		in the enumeration ApplicationType
Read only property	piCameraTyp()	Integer	Returns the camera type. Possible values are defined in the enumeration CameraType.
Read only property	piFrameGrabber()	Integer	Sets or returns the computers frame grabber. Possible values are defined in the enumeration FrameGrabber.
Read only property	piOperatingSystem()	Integer	Returns the operating system (Win95 = 1, WINNT = 2)\r\n
Read only property	pobjHCamera()	Object	Returns an object reference to the camera object. Its type is dependent on the used camera
Read only property	psAppDate()	String	Returns the application date.
Read only property	psApplicationDirectory()	String	Returns the application directory.
Read only property	psApplicationString()	String	Returns a string identifying the application
Read only property	psAppTitle()	String	Returns the title of the application.
Read only property	psAppTitleLong()	String	Returns the long version of the title of the application
Read only property	psConfigFile()	String	Sets or returns the frame grabbers configuration file.
Read only property	psIniFile()	String	Sets or returns the name of the *.ini file used for storage of permanent parameters.
Read only property	psSoftwareVersion()	String	Returns the software version
Read only property	psWindowsDirectory()	String	Returns the Windows directory of the computer where the component runs
Read/write property	pfOptionLUTTool()	Integer	Sets or returns a value which defines whether the LUT tool should be shown on the main window
Read/write property	pfOptionRestoreWindowPos()	Integer	Sets or returns a value which defines whether window positions should be restored or not when \r\r\nthe window is reactivated
Read/write property	pfOptionUserFunctions()	Integer	Sets or returns a value which defines whether the User function is called or not
Read/write property	pfRestoreWindowPos()	Integer	Sets or returns a value which defines whether the window position, its size and window state is restored when the window is displayed again after it has been closed
Read/write property	pfUserIF()	Integer	Sets or returns a value which defines whether the associated window should be shown on screen. A window is only displayed under the following condition: pfUserIF=TRUE, pfVisible=TRUE, pfHideForm=FALSE
Read/write property	plOptionBackgroundColor()	Long	Sets or returns a value which defines the background color
Function	pfReadFile(ByVal sFileName As String, bArray()	Byte, ByRef lLength As Long, Optional sError As Variant)	Reads the content of a file stored at the remote computer
Function	pfWriteFile(ByVal sFileName As String, bArray()	Byte, ByRef lLength As Long, ByVal fDontOverwrite As Integer, Optional sError As Variant)	Writes a file at the remote computer
Function	piEndProg()	Integer	Ends the program. This should be the last method in the application program

Function	psErrorString(ByVal Index As Integer)	String	Returns a string explaining the error with the specified error code
Function	psGetStatus()	String	Returns the current application status string
Sub	pUserFunction(ByVal iIndex As Integer, ByVal lCycleIndex As Long, ByRef sOut As String, ByVal fGetMemoryProperties As Integer)		Executes User Function. This function is kept for compatibility reason. It is no longer recommended to use this function

HImage

Read only property	pfDataSaved()	Integer	Returns a flag defining whether the image data has been saved
Read only property	pfDisplayOnVGA()	Integer	Returns a flag defining whether the Image is displayed in the screen
Read only property	pfImageValid()	Integer	Returns a flag which defines whether the image contains vaild data
Read only property	pHImaImgArea()	HImageArea	Returns an object reference to the images HImageArea object
Read only property	pHWinImgDlg()	HWindow	Returns an object reference to the images main dialog HWindow object
Read only property	pHWinImgPicture()	HWindow	Returns an object reference to the images picture area HWindow object. This is the location where the image is displayed
Read only property	piBytesPerPixel()	Integer	Returns a value defining the number of bytes per pixel used to store the image
Read only property	piMaxScalingX()	Integer	Returns a value which defines the maximum pixel index useable for scaling in x direction
Read only property	piReside()	Integer	Returns a value which defines where the image data resides. Possible values are described in the enumeration ImageResidence
Read only property	pisdImage()	HImageScalingData	Returns an object variable to the HImageScalingData object describing the scaling data associated with this image
Read only property	plDataBytes()	Long	Returns the number of bytes in the image data
Read only property	plDataHandle()	Long	Returns the handle to the image data
Read only property	plDisplayBytes()	Long	Returns the number of bytes in the image display data
Read only property	plDisplayHandle()	Long	Returns the handle to the image display data
Read/write property	pfDisplayDataValid()	Long	Returns a flag defining whether the display data is valid or not. If true pDisplayImage does not recalculate the display data
Read/write property	pfRestoreWindowPos()	Integer	Sets or returns a value which defines whether the window position, its size and window state is restored when the window is displayed again after it has been closed
Read/write property	pfUserIF()	Integer	Sets or returns a value which defines whether the associated window should be shown on screen. A window is only displayed under the following condition: pfUserIF=TRUE, pfVisible=TRUE, pfHideForm=FALSE
Read/write property	piDataType()	Integer	Returns a value defining the data type to store the image
Read/write property	psFileName()	String	Returns the images file name
Read/write property	psStatus()	String	Returns the image status string
Function	pfCreateEmptyImage(ByVal iX As Integer, ByVal iY As Integer, ByVal iDX As Integer,	Integer	Creates an empty (black) image

	ByVal iDY As Integer, ByVal iBytesPerPixel As Integer, ByVal sFileName As String, Optional fShow As Variant)		
Function	pfDumpDataToFile(ByVal fDumpDisplayData As Integer, ByVal sFile As String, ByVal fOverwrite As Integer, ByVal fFullImage As Integer, ByVal X As Integer, ByVal Y As Integer, ByVal dx As Integer, ByVal dy As Integer)	Integer	Dumps the image or display data to a file. This can be used to exchange data to and from an ActiveX-EXE file
Function	pfGetDisplayData(ByVal iX As Integer, ByVal iDX As Integer, ByVal iY As Integer, ByVal iDY As Integer, ByRef bImgDisplay()	Byte, Optional sError As Variant) As Integer	Gets the display data (8 bit) within a specified area of the image
Function	pfGetImageData1Byte(ByVal iX As Integer, ByVal iDX As Integer, ByVal iY As Integer, ByVal iDY As Integer, ByRef bImgData()	Byte, Optional sError As Variant) As Integer	Gets 1byte/pixel data (8 bit) within a specified area of the image. If the image contains another type of data an error is returned
Function	pfGetImageData2Byte(ByVal iX As Integer, ByVal iDX As Integer, ByVal iY As Integer, ByVal iDY As Integer, ByRef iImgData()	Integer, Optional sError As Variant) As Integer	Gets 2byte/pixel data (16 bit) within a specified area of the image. If the image contains another type of data an error is returned
Function	pfGetImageData4Byte(ByVal iX As Integer, ByVal iDX As Integer, ByVal iY As Integer, ByVal iDY As Integer, ByRef lImgData()	Long, Optional sError As Variant) As Integer	Gets 4byte/pixel data (32 bit) within a specified area of the image. If the image contains another type of data an error is returned
Function	pfGetPointIntensity(ByVal iX As Integer, ByVal iY As Integer, ByRef lIntensity As Long)	Integer	Returns a pixel intensity value

Function	pfLoadImage(Optional sFile As Variant, Optional fShow As Variant)	Integer	Loads an image
Function	pfMemGetPRFData(ByRef lProfile()	Long, ByRef iNrData As Integer, ByVal iX As Integer, ByVal iY As Integer, ByVal iDX As Integer, ByVal iDY As Integer, ByVal iProfileType As Integer, ByRef iDatType As Integer, ByVal iMaxLen As Integer, ByRef sError As String) As Integer	Returns an array of profile values within a specified ROI
Function	pfSaveImage(Optional sFile As Variant, Optional sAreDefault As Variant, Optional fSaveROI As Variant)	Integer	Saves the image
Function	pfSetImageData1Byte(ByVal iX As Integer, ByVal iDX As Integer, ByVal iY As Integer, ByVal iDY As Integer, ByRef bImgData()	Byte, Optional sError As Variant) As Integer	Sets 1byte/pixel data (8 bit) within a specified area of the image. If the image contains another type of data an error is returned
Function	pfSetImageData2Byte(ByVal iX As Integer, ByVal iDX As Integer, ByVal iY As Integer, ByVal iDY As Integer, ByRef iImgData()	Integer, Optional sError As Variant) As Integer	Sets 2byte/pixel data (8 bit) within a specified area of the image. If the image contains another type of data an error is returned
Function	pfSetImageData4B	Long,	Sets 4byte/pixel data (8 bit) within a specified area of the

	yte(ByVal iX As Integer, ByVal iDX As Integer, ByVal iY As Integer, ByVal iDY As Integer, ByRef lImgData()	Optional sError As Variant) As Integer	image. If the image contains another type of data an error is returned
Function	pfSetPointIntensity (ByVal iX As Integer, ByVal iY As Integer, ByVal lIntensity As Long)	Integer	Sets a pixel intensity value
Function	piGetMaxBit()	Integer	Returns the number of the topmost bit which is set
Function	psStatusGetSection (ByVal sSectionIdentifier As String)	String	Returns the specified section within the image status string
Function	psStatusGetString(ByVal sSectionIdentifier As String, ByVal sToken As String, ByVal sDefault As String)	String	Returns a string associated to a specified token and section within the image status string
Sub	pAsyncLoadImage (Optional sFile As Variant, Optional fShow As Variant)		Loads an image asynchronously
Sub	pAsyncSaveImage (Optional sFile As Variant, Optional sAreDefault As Variant, Optional fSaveROI As Variant)		Saves the image asynchronously
Sub	pDisplayImage()		Displays the current image in the already shown window
Sub	pGetareSource(By Ref iX As Integer, ByRef iY As Integer, ByRef iDX As Integer, ByRef iDY As Integer)		Returns the source area of the image
Sub	pGetScalingXInfo(ByRef iType As Integer, ByRef ftScale As Single, ByRef sUnit As String, ByRef sScalingFile As String)		Gets the images scaling info in X direction
Sub	pGetScalingYInfo(ByRef iType As Integer, ByRef ftScale As Single, ByRef sUnit As String, ByRef sScalingFile As String)		Gets the images scaling info in Y direction
Sub	pPrint(ByVal		Prints the image

	hwndOwner As Long)		
Sub	pRemoveImage()		Removes the image. This frees all associated memory. All memory handles become invalid afterwards
Sub	psGetPointScaling(ByVal iX As Integer, ByVal iY As Integer, ByRef sX As String, ByRef sY As String)		Returns the scaling oft the indicated pixel, the resulting value for every direction stored in one string
Sub	psGetPointScaling XY(ByVal iX As Integer, ByVal iY As Integer, ByRef ftXValue As Single, ByRef sXUnit As String, ByRef ftYValue As Single, ByRef sYUnit As String)		Returns the scaling oft the indicated pixel, the resulting value for every direction stored in a numerical value and a string
Sub	psGetWidthScalin g(ByVal iX As Integer, ByVal iY As Integer, ByVal iDX As Integer, ByVal iDY As Integer, ByRef sX As String, ByRef sY As String)		Returns the distance of two pixels in scaled units
Sub	pShowImage(ByV al fAdjust As Integer)		Shows the image. This places the associated window on screen
Sub	pStatusWriteSectio n(ByVal sSection As String)		Writes the specified section within the image status string
Sub	pStatusWriteString (ByVal sSectionIdentifier As String, ByVal sToken As String, ByVal sValue As String, Optional fAddQuotes)		Writes a string associated to a specified token and section within the image status string

HImages

Event	CloseSequenceIma ge(ByVal Index As Integer, ByVal Mode As Integer, ByRef Cancel As Integer, ByRef SaveQueryDone As Integer, ByRef ImageRemoved As Integer)		Event which is sent to tell other objects that a sequence image will now be closed. This event is used by the HSequence object to free all related memory
Event	CurrentImageChan ged(ByVal iImg As Integer)		Event which is raised when the current image changes
Event	FileNameChanged (ByVal iImg As Integer)		Event which is raised when the File name has changed

Event	ImageDataChanged(ByVal iImg As Integer)		Event which is raised when the image data has changed
Event	ImagePainted(ByVal iImg As Integer, ByVal ftFWHM As Single, ByVal sUnit As String)		Event which is raised when the image has been painted
Event	ImageRemoved(ByVal Index As Integer)		Event which is raised when the image has been removed
Event	Message(ByVal sMessage As String)		Event which is raised to inform the user of ongoing operations
Event	RemoveImage(ByVal Index As Integer, ByRef Cancel As Integer)		Event which is raised when an image will be removed
Read only property	pcItem(ByVal Index As Integer)	HImage	Returns a reference to the specified image
Read only property	pcItemCurrent()	HImage	Returns a reference to the currently active image. If no image is currently active this function does not return any value.
Read only property	pfFixedITEXHeader()	Integer	Sets or returns a value defining whether images are saved with a fixed header (10K) in the case ITEX file type is used
Read only property	pHChkOptAcqToSameWnd()	HCheck	Returns an object reference to the HCheck object associated with the options dialogs 'Acquire Always To Same Window' checkbox
Read only property	pHChkOptAutoLiveLUT()	HCheck	Returns an object reference to the HCheck object associated with the options dialogs 'Auto Live LUT' checkbox
Read only property	pHChkOptAutoLUT()	HCheck	Returns an object reference to the HCheck object associated with the options dialogs 'Auto LUT' checkbox
Read only property	pHChkOptAutoLutInROI()	HCheck	
Read only property	pHChkOptDisplayFWHM()	HCheck	Returns an object reference to the HCheck object associated with the options dialogs 'Display FWHM' checkbox
Read only property	pHChkOptFixedITEXHeader()	HCheck	Returns an object reference to the HCheck object associated with the options dialogs 'Fixed ITEX header' checkbox
Read only property	pHChkOptROIUseMinAsZero()	HCheck	Returns an object reference to the HCheck object associated with the options dialogs 'Use Minimum as zero for ROI' checkbox
Read only property	pHChkOptWarnUnsaved()	HCheck	Returns an object reference to the HCheck object associated with the options dialogs 'Warn Whne Unsaved Images Are Closed' checkbox
Read only property	pHChkOptZeroLower()	HCheck	
Read only property	pHChkRoiCalibrated()	HCheck	Returns an object reference to the HCheck object associated with the ROI Interface dialogs 'Calibrated' checkbox
Read only property	pHComFWHMColor()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'FWHM color' checkbox
Read only property	pHComOptCancel()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'Cancel' checkbox
Read only property	pHComOptOK()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'OK' checkbox
Read only property	pHDisRoiAreaSize	HDisp	Returns an object reference to the HDisp object

	()		associated with the ROI interface dialogs 'Area Size' display area
Read only property	pHDisRoiDiagonal()	HDisp	Returns an object reference to the HDisp object associated with the ROI interface dialogs 'Diagonal' display area
Read only property	pHDisRoiIntensity()	HDisp	Returns an object reference to the HDisp object associated with the ROI interface dialogs 'Intensity' display area
Read only property	pHDisRoiSlope()	HDisp	Returns an object reference to the HDisp object associated with the ROI interface dialogs 'Slope' display area
Read only property	pHDisRoiXUnit()	HDisp	Returns an object reference to the HDisp object associated with the ROI interface dialogs 'X Unit' display area
Read only property	pHDisRoiYUnit()	HDisp	Returns an object reference to the HDisp object associated with the ROI interface dialogs 'Y Unit' display area
Read only property	pHEdnOptFWHMNoOfDigits()	HEditNumber	Returns an object reference to the HEditNumber object associated with options dialogs 'FWHM No. of Digits' editbox
Read only property	pHEdnOptFWHMSize()	HEditNumber	options dialogs 'FWHM Size' editbox
Read only property	pHEdnRoiXEnd()	HEditNumber	Returns an object reference to the HEditNumber object associated with ROI interface dialogs 'X End' editbox
Read only property	pHEdnRoiXStart()	HEditNumber	Returns an object reference to the HEditNumber object associated with ROI interface dialogs 'X Start' editbox
Read only property	pHEdnRoiXWidth()	HEditNumber	Returns an object reference to the HEditNumber object associated with ROI interface dialogs 'X Width' editbox
Read only property	pHEdnRoiYEnd()	HEditNumber	Returns an object reference to the HEditNumber object associated with ROI interface dialogs 'Y End' editbox
Read only property	pHEdnRoiYStart()	HEditNumber	Returns an object reference to the HEditNumber object associated with ROI interface dialogs 'Y Start' editbox
Read only property	pHEdnRoiYWidth()	HEditNumber	Returns an object reference to the HEditNumber object associated with ROI interface dialogs 'Y Width' editbox
Read only property	pHFraOptROIPrf()	HFrame	Returns an object reference to the HFrame object associated with the options dialogs 'ROI profile default direction' frame
Read only property	pHRadOptDefZoomFactor()	HRadios	Returns an object reference to the HFrame object associated with the options dialogs 'Default Zoom factor' frame
Read only property	pHRadOptROIPrf()	HRadios	Returns an object reference to the HRadios object associated with the options dialogs 'ROI profile default direction' radiobutton group
Read only property	pHRadRoiPrfDirection()	HRadios	Returns an object reference to the HRadios object associated with the ROI interface dialogs 'ROI profile direction' radiobutton group
Read only property	pHWinImgDlg20Array()	HWindow20Array	Returns an object reference to the HWindow20Array object associated with the images main windows
Read only property	pHWinImgPicture20Array()	HWindow20Array	Returns an object reference to the HWindow20Array object associated with the images imagedisplay pictureboxes
Read only property	pHWinOptDlg()	HWindow	Returns an object reference to the HWindow object associated with the options dialogs main window
Read only property	pHWinOptOptions()	HWindow	Returns an object reference to the HWindow object associated with the options dialogs options picturebox
Read only property	pHWinRoiDlg()	HWindow	Returns an object reference to the HWindow object associated with the ROI interface dialogs main window
Read only property	piCurrentImageValid()	Integer	Returns a value which defines whether the current image is valid
Read only property	piIndexCurrentImg()	Integer	Returns the image index of the current image.

Read only property	piIndexLoadImg()	Integer	Returns the image index of the currently or last loaded image.
Read only property	piIndexSequenceImg()	Integer	Returns the image index of the sequence image. If there is no sequence image the value is -1
Read only property	piItemsMax()	Integer	Returns the maximum number of images.
Read only property	psDefaultScalingDirectory()	String	Returns the default scaling directory. This is only necessary for old type file where scaling tabels are not saved in the image and only a file reference is stored in the image
Read/write property	pfDisplayToTIFFMode()	Integer	Sets or returns a value defining whether images are saved in DisplayToTiff mode in the case TIFF is selected as the file type
Read/write property	pfOptionsAcqToSameWnd()	Integer	Sets or returns a value which defines whether the acquisition should always be done in the same window
Read/write property	pfOptionsAutoLiveLUT()	Integer	Sets or returns a value which defines whether AUTO Live LUT should be excuted during Live mode
Read/write property	pfOptionsAutoLUT()	Integer	Sets or returns a value which defines whether Auot LUT should be excuted aftre image acquisition
Read/write property	pfOptionsAutoLutInROI()	Integer	
Read/write property	pfOptionsDisplayFWHM()	Integer	
Read/write property	pfOptionsFixedITEXHeader()	Integer	Sets or returns a value which defines whether the image should be saved with fixed header
Read/write property	pfOptionsROIUseMinAsZero()	Integer	Sets or returns a value which defines whether the minimum should be used as zero point for FWHM calculation
Read/write property	pfOptionsWarnUnsaved()	Integer	Sets or returns a value which defines whether the user should be warned when an unsaved image is closed
Read/write property	pfOptionsZeroLower()	Integer	
Read/write property	pfRestoreWindowPos()	Integer	Sets or returns a value which defines whether the window position, its size and window state is restored when the window is displayed again after it has been closed
Read/write property	pftOptionsDefZoomFactor()	Single	Sets or returns the default zooming factor
Read/write property	pfUserIF()	Integer	Sets or returns a value which defines whether the associated window should be shown on screen. A window is only displayed under the following condition: pfUserIF=TRUE, pfVisible=TRUE, pfHideForm=FALSE
Read/write property	piDefaultImageType()	Integer	Sets or defines the current default image type
Read/write property	piIndexAcqImg()	Integer	Returns the image index of the currently or last acquired image
Read/write property	piIndexAcquireImg()	Integer	Returns the image index of the currently or last ACQUIRE image. This is every acquired single image except the live image
Read/write property	piIndexLiveImg()	Integer	Returns the image index of the currently or last LIVE image.
Read/write property	piOptionFWHMNoOfDigits()	Integer	
Read/write property	piOptionFWHMSize()	Integer	
Read/write property	piOptionsROIPrf()	Integer	
Read/write property	piSeqMode()	Integer	Sets or returns the default value of SeqMode for the functions piGetLoadImageFile and piGetLoadImageFile. If SeqMode=TRUE these function offer sequence mode file types as default

Read/write property	plOptionFWHMColor()	Long	
Function	pfLoadImage(Optional sFile As Variant, Optional fShow As Variant, Optional iImgIndex As Integer)	Integer	Loads an image
Function	pfLoadImageNoShow(ByVal sFile As String, ByVal iType As Integer, ByRef iImg As Integer)	Integer	Loads an image without showing it
Function	pfSaveImage(Optional sFile As Variant, Optional sAreDefault As Variant, Optional fSaveROI As Variant)	Integer	Saves an image
Function	pfUpdateImageAfterDataChange(ByVal iImg As Integer, Optional sError As Variant)		A function that updates all relevant modules after the client programmer has changed the image data, its size, data depth or status
Function	piGetLoadImageFile(ByRef sCompleteFile As String, ByRef iDefaultImageType As Integer, ByRef fDoSeqSaveLoad As Integer, ByRef iSeqMode As Integer, ByVal iImageTypes As Integer)	Integer	Calls a file dialog which can be used to get an image file name for load
Function	piGetSaveImageFile(ByRef sCompleteFile As String, ByRef sComment As String, ByRef iDefaultImageType As Integer, ByRef fDoSeqSaveLoad As Integer, ByRef iSeqMode As Integer, ByRef fDisplayToTIFFMode As Integer, ByVal iImageTypes As Integer, ByVal iNumberOfDigitsForSequence As Integer)	Integer	Calls a file dialog which can be used to get an image file name for save

Function	psGetMemoryProperties()	String	Returns as string which contains information about the image. Avoid using this function, it is only used for compatibility reasons
Sub	pAsyncLoadImage (Optional sFile As Variant, Optional fShow As Variant)		Loads an image asynchronously
Sub	pAsyncSaveImage (Optional sFile As Variant, Optional sAreDefault As Variant, Optional fSaveROI As Variant)		Saves an image asynchronously
Sub	pChangeActiveImage(ByVal iImg As Integer, ByVal fDisplayImage As Integer, Optional fDontSetFocus As Variant)		Changes the active image
Sub	pGetFWHMProperties(ByRef fDisplayFWHM As Integer, ByRef lColor As Long, ByRef iFontSize As Integer, ByRef iNoOfDigits As Integer)		Returns all relevant informations for writing the FWHM assoziated with the quick profile
Sub	pPrint(ByVal hwndOwner As Long)		Prints an image

HLutControl

Event	LimitsChanged()		Event which is raised when the limits of the HLut object are changing
Read only property	pHEdnLUTParGamma()	HEditNumber	Returns an object reference to the HEditNumber object associated with LUT Parameters dialogs ""Gamma"" editbox
Read only property	phEdnLUTParLut16xShift()	HEditNumber	Returns an object reference to the HEditNumber object associated with LUT Parameters dialogs ""Shift for 16x LUT"" editbox
Read only property	pHEntLUTParOvlColors()	HEntry	Returns an object reference to the HEntry object associated with the LUT parameters dialogs ""Overlay Colors"" entrybox
Read only property	pHLutLUTContrControl()	HLut	Returns an object reference to the HLut object associated with the LUT Control dialogs ""Control"" entrybox
Read only property	pHRadLUTParBitRange()	HRadios	Returns an object reference to the HRadios object associated with the LUT Parameters dialogs ""Bit Range""
Read only property	pHRadLUTParColor()	HRadios	Returns an object reference to the HRadios object associated with the LUT Parameters dialogs ""Color""
Read only property	pHRadLUTParLUTType()	HRadios	Returns an object reference to the HRadios object associated with the LUT Parameters dialogs ""LUT Type""
Read only property	pHWinLUTContrDlg()	HWindow	Returns an object reference to the HWindow object associated with the LUT Control dialogs main window
Read only property	pHWinLUTParDlg()	HWindow	Returns an object reference to the HWindow object associated with the LUT Parameters dialogs main window
Read/write property	pfRestoreWindow	Integer	Sets or returns a value which defines whether the

	Pos()		window position, its size and window state is restored when the window is displayed again after it has been closed
Read/write property	pfUserIF()	Integer	Sets or returns a value which defines whether the associated window should be shown on screen. A window is only displayed under the following condition: pfUserIF=TRUE, pfVisible=TRUE, pfHideForm=FALSE
Function	psGetStatus()	String	Returns the current LUT status string

HSystemScaling

Event	Message(ByVal sMessage As String)		Event which is raised to inform the user of ongoing operations
Read only property	pHChkScalSetAssgnScalToImg()	HCheck	Returns an object reference to the HCheck object associated with the Scaling Setup dialogs 'Assign Scaling To Image' checkbox
Read only property	pHChkScalSetAssgnScalToPrf()	HCheck	Returns an object reference to the HCheck object associated with the Scaling Setup dialogs 'Assign Scaling To Profile' checkbox
Read only property	pHComCreaPolyCancel()	HCommand	Returns an object reference to the HCommand object associated with the create polynomial dialogs 'Cancel' pushbutton
Read only property	pHComCreatePolOK()	HCommand	Returns an object reference to the HCommand object associated with the create polynomial dialogs 'OK' pushbutton
Read only property	pHComScalEditNextPixel()	HCommand	Returns an object reference to the HCommand object associated with the scaling editor dialogs 'Next Pixel' pushbutton
Read only property	pHComScalEditPreviousPixel()	HCommand	Returns an object reference to the HCommand object associated with the scaling editor dialogs 'Previous Pixel' pushbutton
Read only property	pHComScalSetCancel()	HCommand	Returns an object reference to the HCommand object associated with the scaling setup dialogs 'Cancel' pushbutton
Read only property	pHComScalSetFreeGetFromImage()	HCommand	Returns an object reference to the HCommand object associated with the scaling setup dialogs 'Free Scaling Get From Image' pushbutton
Read only property	pHComScalSetFreeHFileGet()	HCommand	Returns an object reference to the HCommand object associated with the scaling setup dialogs 'Free Scaling Horizontal Get File' pushbutton
Read only property	pHComScalSetFreeHPixelGet()	HCommand	Returns an object reference to the HCommand object associated with the scaling setup dialogs 'Free Scaling Horizontal Get Pixel' pushbutton
Read only property	pHComScalSetFreeVFileGet()	HCommand	Returns an object reference to the HCommand object associated with the scaling setup dialogs 'Free Scaling vertical Get File' pushbutton
Read only property	pHComScalSetFreeVPixelGet()	HCommand	Returns an object reference to the HCommand object associated with the scaling setup dialogs 'Free Scaling vertical Get Pixel' pushbutton
Read only property	pHComScalSetSet()	HCommand	Returns an object reference to the HCommand object associated with the scaling setup dialogs 'Set' pushbutton
Read only property	pHComScalSetSetSpectrographScaling()	HCommand	Returns an object reference to the HCommand object associated with the scaling setup dialogs 'Set Spectrograph Scaling' pushbutton
Read only property	pHComScalSetSquareGetFromImage()	HCommand	Returns an object reference to the HCommand object associated with the scaling setup dialogs 'Square Scaling Get From Image' pushbutton
Read only property	pHComScalSetSquareGetPixel()	HCommand	Returns an object reference to the HCommand object associated with the scaling setup dialogs 'Square Scaling Get Pixel' pushbutton

Read only property	pHComScalSetStreakSetTimeScaling()	HCommand	Returns an object reference to the HCommand object associated with the scaling setup dialogs 'Streak Scaling Set Time Scaling' pushbutton
Read only property	pHComScalSetStreakTFFileGet()	HCommand	Returns an object reference to the HCommand object associated with the scaling setup dialogs 'Streak Scaling Time/Focus Get File' pushbutton
Read only property	pHComScalSetStreakTFPixelGet()	HCommand	Returns an object reference to the HCommand object associated with the scaling setup dialogs 'Streak Scaling Time/Focus Get Pixel' pushbutton
Read only property	pHComScalSetStreakXFileGet()	HCommand	Returns an object reference to the HCommand object associated with the scaling setup dialogs 'Streak Scaling X-Direction Get File' pushbutton
Read only property	pHComScalSetStreakXPixelGet()	HCommand	Returns an object reference to the HCommand object associated with the scaling setup dialogs 'Streak Scaling X-Direction Get Pixel' pushbutton
Read only property	pHComSpecScalAssign()	HCommand	Returns an object reference to the HCommand object associated with the spectrograph scaling dialogs 'Assign' pushbutton
Read only property	pHComSpecScalClose()	HCommand	Returns an object reference to the HCommand object associated with the spectrograph scaling dialogs 'Close' pushbutton
Read only property	pHComSpecScalDefinePosWave1()	HCommand	Returns an object reference to the HCommand object associated with the spectrograph scaling dialogs 'Define Position Wavelength 1' pushbutton
Read only property	pHComSpecScalDefinePosWave2()	HCommand	Returns an object reference to the HCommand object associated with the spectrograph scaling dialogs 'Define Position Wavelength 2' pushbutton
Read only property	pHComSpecScalDispersion()	HCommand	Returns an object reference to the HCommand object associated with the spectrograph scaling dialogs 'Set Dispersion' pushbutton
Read only property	pHComSpecScalGetCenterWavePix()	HCommand	Returns an object reference to the HCommand object associated with the spectrograph scaling dialogs 'Get Center Wavelength' pushbutton
Read only property	pHComTimScalAssignStreakScaling()	HCommand	Returns an object reference to the HCommand object associated with the time scaling dialogs 'Assign Streak Scaling' pushbutton
Read only property	pHComTimScalClose()	HCommand	Returns an object reference to the HCommand object associated with the time scaling dialogs 'Close' pushbutton
Read only property	pHComTimScalGetFromTextFile()	HCommand	Returns an object reference to the HCommand object associated with the time scaling dialogs 'Get From Text File' pushbutton
Read only property	pHComTimScalStreakTFileGet()	HCommand	Returns an object reference to the HCommand object associated with the time scaling dialogs 'Streak Scaling Time/Focus Get File' pushbutton
Read only property	pHComTimScalStreakTPixelGet()	HCommand	Returns an object reference to the HCommand object associated with the time scaling dialogs 'Streak Scaling Time/Focus Get Pixel' pushbutton
Read only property	pHDisScalEditFileName()	HDisp	Returns an object reference to the HDisp object associated with the scaling editor dialogs 'File Name' display area
Read only property	pHDisScalEditMode()	HDisp	Returns an object reference to the HDisp object associated with the scaling editor dialogs 'Mode' display area
Read only property	pHDisScalEditPixel()	HDisp	Returns an object reference to the HDisp object associated with the scaling editor dialogs 'Pixel' display area
Read only property	pHDisScalEditValue()	HDisp	Returns an object reference to the HDisp object associated with the scaling editor dialogs 'Value' display area
Read only property	pHDisScalEditXA	HDisp	Returns an object reference to the HDisp object

	xis()		associated with the scaling editor dialogs 'X Axis' display area
Read only property	pHDisScalEditYAxis()	HDisp	Returns an object reference to the HDisp object associated with the scaling editor dialogs 'Y Axis' display area
Read only property	pHDisScalSetCurrentHUnit()	HDisp	Returns an object reference to the HDisp object associated with the scaling setup dialogs 'Current Horizontal Unit' display area
Read only property	pHDisScalSetCurrentHValue()	HDisp	Returns an object reference to the HDisp object associated with the scaling setup dialogs 'Current Horizontal Value' display area
Read only property	pHDisScalSetCurrentVUnit()	HDisp	Returns an object reference to the HDisp object associated with the scaling setup dialogs 'Current Vertical Unit' display area
Read only property	pHDisScalSetCurrentVValue()	HDisp	Returns an object reference to the HDisp object associated with the scaling setup dialogs 'Current Vertical value' display area
Read only property	pHDisScalSetTFDirection()	HDisp	Returns an object reference to the HDisp object associated with the scaling setup dialogs 'Time/Focus Direction' display area
Read only property	pHDisScalSetXDirection()	HDisp	Returns an object reference to the HDisp object associated with the scaling setup dialogs 'X Direction' display area
Read only property	pHDisSpecScalSpectrograph()	HDisp	Returns an object reference to the HDisp object associated with the spectrograph scaling dialogs 'Spectrograph' display area
Read only property	pHEdnCreaPolyCoefficient1()	HEditNumber	Returns an object reference to the HEditNumber object associated with create polynomial dialogs 'Coefficient 1' editbox
Read only property	pHEdnCreaPolyCoefficient2()	HEditNumber	Returns an object reference to the HEditNumber object associated with create polynomial dialogs 'Coefficient 2' editbox
Read only property	pHEdnCreaPolyCoefficient3()	HEditNumber	Returns an object reference to the HEditNumber object associated with create polynomial dialogs 'Coefficient 3' editbox
Read only property	pHEdnCreaPolyCoefficient4()	HEditNumber	Returns an object reference to the HEditNumber object associated with create polynomial dialogs 'Coefficient 4' editbox
Read only property	pHEdnCreaPolyCoefficient5()	HEditNumber	Returns an object reference to the HEditNumber object associated with create polynomial dialogs 'Coefficient 5' editbox
Read only property	pHEdnCreaPolyOrder()	HEditNumber	Returns an object reference to the HEditNumber object associated with create polynomial dialogs 'Order' editbox
Read only property	pHEdnCreaPolyValidChannels()	HEditNumber	Returns an object reference to the HEditNumber object associated with create polynomial dialogs 'Valid Channel' editbox
Read only property	pHEdnScalSetFreeHPixel()	HEditNumber	Returns an object reference to the HEditNumber object associated with scaling setup dialogs 'Free Scaling Horizontal Pixel' editbox
Read only property	pHEdnScalSetFreeHValue()	HEditNumber	Returns an object reference to the HEditNumber object associated with scaling setup dialogs 'Free Scaling Horizontal Value' editbox
Read only property	pHEdnScalSetFreeVPixel()	HEditNumber	Returns an object reference to the HEditNumber object associated with scaling setup dialogs 'Free Scaling Vertical Pixel' editbox
Read only property	pHEdnScalSetFreeVValue()	HEditNumber	Returns an object reference to the HEditNumber object associated with scaling setup dialogs 'Free Scaling Vertical Value' editbox
Read only property	pHEdnScalSetSquarePixel()	HEditNumber	Returns an object reference to the HEditNumber object associated with scaling setup dialogs 'Square Scaling Pixel' editbox

Read only property	pHEdnScalSetSquareValue()	HEditNumber	Returns an object reference to the HEditNumber object associated with scaling setup dialogs 'Square Scaling Value' editbox
Read only property	pHEdnScalSetStreakTFPixel()	HEditNumber	Returns an object reference to the HEditNumber object associated with scaling setup dialogs 'Streak Scaling Time/Focus Pixel' editbox
Read only property	pHEdnScalSetStreakTFValue()	HEditNumber	Returns an object reference to the HEditNumber object associated with scaling setup dialogs 'Streak Scaling Time/Focus Value' editbox
Read only property	pHEdnScalSetStreakXPixel()	HEditNumber	Returns an object reference to the HEditNumber object associated with scaling setup dialogs 'Streak Scaling X Direction Pixel' editbox
Read only property	pHEdnScalSetStreakXValue()	HEditNumber	Returns an object reference to the HEditNumber object associated with scaling setup dialogs 'Streak Scaling X Direction Value' editbox
Read only property	pHEdnSpecScalCenterWavePix()	HEditNumber	Returns an object reference to the HEditNumber object associated with spectrograph scaling dialogs 'Center Wave Pixel' editbox
Read only property	pHEdnSpecScalDispersion()	HEditNumber	Returns an object reference to the HEditNumber object associated with spectrograph scaling dialogs 'Dispersion' editbox
Read only property	pHEdnSpecScalWave1()	HEditNumber	Returns an object reference to the HEditNumber object associated with spectrograph scaling dialogs 'Wavelength 1' editbox
Read only property	pHEdnSpecScalWave2()	HEditNumber	Returns an object reference to the HEditNumber object associated with spectrograph scaling dialogs 'Wavelength 2' editbox
Read only property	pHEdnTimScalStreakTPixel()	HEditNumber	Returns an object reference to the HEditNumber object associated with time scaling dialogs 'Streak Time Pixel' editbox
Read only property	pHEdnTimScalStreakTValue()	HEditNumber	Returns an object reference to the HEditNumber object associated with time scaling dialogs 'Streak Time Value' editbox
Read only property	pHEntScalSetSquareProfile()	HEntry	Returns an object reference to the HEntry object associated with the scaling setup dialogs 'Square Scaling Get From Profile' entrybox
Read only property	pHEntSpecScalGrating()	HEntry	Returns an object reference to the HEntry object associated with the spectrograph scaling dialogs 'Grating' entrybox
Read only property	pHEntTimScalPlugin()	HEntry	Returns an object reference to the HEntry object associated with the time scaling dialogs 'Plugin' entrybox
Read only property	pHEntTimScalStreakCamera()	HEntry	Returns an object reference to the HEntry object associated with the time scaling dialogs 'Streak Camera' entrybox
Read only property	pHEntTimScalTimeRange()	HEntry	Returns an object reference to the HEntry object associated with the time scaling dialogs 'Time Range' entrybox
Read only property	pHEstScalSetFreeHFile()	HEditString	Returns an object reference to the HEditString object associated with the scaling setup dialogs 'Free Scaling Horizontal File' editbox
Read only property	pHEstScalSetFreeHUnit()	HEditString	Returns an object reference to the HEditString object associated with the scaling setup dialogs 'Free Scaling Horizontal Unit' editbox
Read only property	pHEstScalSetFreeVFile()	HEditString	Returns an object reference to the HEditString object associated with the scaling setup dialogs 'Free Scaling Vertical File' editbox
Read only property	pHEstScalSetFreeVUnit()	HEditString	Returns an object reference to the HEditString object associated with the scaling setup dialogs 'Free Scaling Vertical Unit' editbox
Read only property	pHEstScalSetScalingDirectory()	HEditString	Returns an object reference to the HEditString object associated with the scaling setup dialogs 'Free Scaling

			Horizontal File' editbox
Read only property	pHEstScalSetSquareUnit()	HEditString	Returns an object reference to the HEditString object associated with the scaling setup dialogs 'Square Scaling Unit' editbox
Read only property	pHEstScalSetStreakTFFile()	HEditString	Returns an object reference to the HEditString object associated with the scaling setup dialogs 'Streak Scaling Time/Focus File' editbox
Read only property	pHEstScalSetStreakTFUnit()	HEditString	Returns an object reference to the HEditString object associated with the scaling setup dialogs 'Streak Scaling Time/Focus Unit' editbox
Read only property	pHEstScalSetStreakXFile()	HEditString	Returns an object reference to the HEditString object associated with the scaling setup dialogs 'Streak Scaling X Direction File' editbox
Read only property	pHEstScalSetStreakXUnit()	HEditString	Returns an object reference to the HEditString object associated with the scaling setup dialogs 'Streak Scaling X Direction Unit' editbox
Read only property	pHEstTimScalStreakTFile()	HEditString	Returns an object reference to the HEditString object associated with the time scaling dialogs 'Streak Time Axis File' editbox
Read only property	pHEstTimScalStreakTUnit()	HEditString	Returns an object reference to the HEditString object associated with the time scaling dialogs 'Streak Time Axis Unit' editbox
Read only property	pHFraScalSetFreeScaling()	HFrame	Returns an object reference to the HFrame object associated with the scaling setup dialogs 'Free Scaling' frame
Read only property	pHFraScalSetSquareScaling()	HFrame	Returns an object reference to the HFrame object associated with the scaling setup dialogs 'Square Scaling' frame
Read only property	pHFraScalSetStreakScaling()	HFrame	Returns an object reference to the HFrame object associated with the scaling setup dialogs 'Streak Scaling' frame
Read only property	pHFraSpecScalSetDispersion()	HFrame	Returns an object reference to the HFrame object associated with the spectrograph scaling setup dialogs 'Dispersion' frame
Read only property	pHFraTimScalIndividualAssignment()	HFrame	Returns an object reference to the HFrame object associated with the time scaling dialogs 'Individual Assignment' frame
Read only property	pHFraTimScalScaling()	HFrame	Returns an object reference to the HFrame object associated with the time scaling dialogs 'Scaling' frame
Read only property	pHFraTimScalScalingType()	HFrame	Returns an object reference to the HFrame object associated with the time scaling dialogs 'Scaling Type' frame
Read only property	pHFraTimScalTimeRange()	HFrame	Returns an object reference to the HFrame object associated with the time scaling dialogs 'Time Range' frame
Read only property	pHMenScalEditAdd()	HMenu	Returns an object reference to the HMenu object associated with the scaling editor dialogs 'Add' menu entry
Read only property	pHMenScalEditDifferentialDisplay()	HMenu	Returns an object reference to the HMenu object associated with the scaling editor dialogs 'Differential Display' menu entry
Read only property	pHMenScalEditDiffPolynomial()	HMenu	Returns an object reference to the HMenu object associated with the scaling editor dialogs 'Create Differental Polynomial' menu entry
Read only property	pHMenScalEditExit()	HMenu	Returns an object reference to the HMenu object associated with the scaling editor dialogs 'Exit' menu entry
Read only property	pHMenScalEditFile()	HMenu	Returns an object reference to the HMenu object associated with the scaling editor dialogs 'File' menu
Read only property	pHMenScalEditFunctions()	HMenu	Returns an object reference to the HMenu object associated with the scaling editor dialogs 'Functions'

			menu
Read only property	pHMenScalEditIntegralDisplay()	HMenu	Returns an object reference to the HMenu object associated with the scaling editor dialogs 'Integral Display' menu entry
Read only property	pHMenScalEditIntPolynomial()	HMenu	Returns an object reference to the HMenu object associated with the scaling editor dialogs 'Create Integral Polynomial' menu entry
Read only property	pHMenScalEditLoad()	HMenu	Returns an object reference to the HMenu object associated with the scaling editor dialogs 'Load' menu entry
Read only property	pHMenScalEditMultiply()	HMenu	Returns an object reference to the HMenu object associated with the scaling editor dialogs 'Multiply' menu entry
Read only property	pHMenScalEditOptions()	HMenu	Returns an object reference to the HMenu object associated with the scaling editor dialogs 'Options' menu entry
Read only property	pHMenScalEditPrint()	HMenu	Returns an object reference to the HMenu object associated with the scaling editor dialogs 'Print' menu entry
Read only property	pHMenScalEditSave()	HMenu	Returns an object reference to the HMenu object associated with the scaling editor dialogs 'Save' menu entry
Read only property	pHRadScalSetFreeHType()	HRadios	Returns an object reference to the HRadios object associated with the scaling setup dialogs 'Free Scaling Horizontal Type' radiobutton group
Read only property	pHRadScalSetFreeVType()	HRadios	Returns an object reference to the HRadios object associated with the scaling setup dialogs 'Free Scaling Vertical Type' radiobutton group
Read only property	pHRadScalSetScalingMethod()	HRadios	Returns an object reference to the HRadios object associated with the scaling setup dialogs 'Scaling Method' radiobutton group
Read only property	pHRadScalSetStreakScalingTimeAxis()	HRadios	Returns an object reference to the HRadios object associated with the scaling setup dialogs 'Streak Scaling Time Axis' radiobutton group
Read only property	pHRadScalSetStreakTFType()	HRadios	Returns an object reference to the HRadios object associated with the scaling setup dialogs 'Streak Scaling Time/Focus Type' radiobutton group
Read only property	pHRadScalSetStreakXType()	HRadios	Returns an object reference to the HRadios object associated with the scaling setup dialogs 'Streak Scaling X Direction Type' radiobutton group
Read only property	pHRadTimScalStreakTType()	HRadios	Returns an object reference to the HRadios object associated with the time scaling dialogs 'Streak Time Axis Type' radiobutton group
Read only property	pHWinCreaPolyDlg()	HWindow	Returns an object reference to the HWindow object associated with the create polynomial main window
Read only property	pHWinScalEditDlg()	HWindow	Returns an object reference to the HWindow object associated with the scaling editor main window
Read only property	pHWinScalSetDlg()	HWindow	Returns an object reference to the HWindow object associated with the scaling setup main window
Read only property	pHWinSpecScalDlg()	HWindow	Returns an object reference to the HWindow object associated with the spectrograph scaling main window
Read only property	pHWinTimScalDlg()	HWindow	Returns an object reference to the HWindow object associated with the time scaling main window
Read only property	psScalingDirectory()	String	Returns the scaling directory
Read only property	pssdSystem()	HSystemScalingData	Returns an object reference to the HSystemScalingData object
Read/write property	pfRestoreWindowPos()	Integer	Sets or returns a value which defines whether the window position, its size and window state is restored when the window is displayed again after it has been

			closed
Read/write property	pfUserIF()	Integer	Sets or returns a value which defines whether the associated window should be shown on screen. A window is only displayed under the following condition: pfUserIF=TRUE, pfVisible=TRUE, pfHideForm=FALSE
Function	pfGetFromTextFile(ByVal sTextFile As String, Optional sError As Variant)	Integer	Executes the ""Get Form Text File"" function specifiying a file name for the text file

HSequence

Event	CameraIsStarted()		
Event	EndSequence()		Event which is raised after the sequence acquisition has ended
Event	ErrorEndSequence()		Event which is raised after the sequence acquisition has ended due to an error
Event	Message(ByVal sMessage As String)		Event which is raised to inform the user of ongoing operations
Event	SeqSingleAcqEnded(ByVal lIndex As Long)		
Event	StartSequence()		Event which is raised before the sequence acquisition has started
Read only property	pfSequenceIsInRAM()	Integer	Returns a flag which defines whether the sequence is stored in RAM
Read only property	pHChkOptAutoCorrectAfterSeq()	HCheck	Returns an object reference to the HCheck object associated with the options dialogs 'Auto Correct After Sequence' checkbox
Read only property	pHChkOptDisplayImgDuringSequence()	HCheck	Returns an object reference to the HCheck object associated with the options dialogs 'Display Image During Sequence' checkbox
Read only property	pHChkOptDoAcquireWrap()	HCheck	Returns an object reference to the HCheck object associated with the options dialogs 'Wrap During Sequence Acquisition' checkbox
Read only property	pHChkOptEnableAcquireWrap()	HCheck	Returns an object reference to the HCheck object associated with the options dialogs 'Enable Wrap During Sequence Acquisition' checkbox
Read only property	pHChkOptEnableStop()	HCheck	Returns an object reference to the HCheck object associated with the options dialogs 'Enable Stop' checkbox
Read only property	pHChkOptPromptBeforeStart()	HCheck	Returns an object reference to the HCheck object associated with the options dialogs 'Prompt before Start' checkbox
Read only property	pHChkOptWarning()	HCheck	Returns an object reference to the HCheck object associated with the options dialogs 'Issue Warning' checkbox
Read only property	pHChkSeqAcquireROI()	HCheck	Returns an object reference to the HCheck object associated with the sequence dialogs 'Acquire ROI' checkbox
Read only property	pHChkSeqAutoFixpoint()	HCheck	Returns an object reference to the HCheck object associated with the sequence dialogs 'Auto Fixpoint' checkbox
Read only property	pHChkSeqExcludeSample()	HCheck	Returns an object reference to the HCheck object associated with the sequence dialogs 'Exclude sample' checkbox
Read only property	pHChkSeqNeverLoadToRAM()	HCheck	Returns an object reference to the HCheck object associated with the sequence dialogs 'Never Load to RAM' checkbox

Read only property	pHChkSeqSaveImages()	HCheck	Returns an object reference to the HCheck object associated with the sequence dialogs 'Save Images' checkbox
Read only property	pHChkSeqSaveProfiles()	HCheck	Returns an object reference to the HCheck object associated with the sequence dialogs 'Save Profiles' checkbox
Read only property	pHChkSeqWrapPlay()	HCheck	Returns an object reference to the HCheck object associated with the sequence dialogs 'Wrap During Play' checkbox
Read only property	pHComOptCancel()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'Cancel' pushbutton
Read only property	pHComOptOK()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'Cancel' pushbutton
Read only property	pHComSeqCopyToNewImg()	HCommand	Returns an object reference to the HCheck object associated with the sequence dialogs 'Copy To New Image' pushbutton
Read only property	pHComSeqFirstSample()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'Cancel' pushbutton
Read only property	pHComSeqGet1stImage()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'Get First Image' pushbutton
Read only property	pHComSeqGet1stProfile()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'Get First Profile' pushbutton
Read only property	pHComSeqGetAreaForFixPoint()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'Get Area For Fix Point' pushbutton
Read only property	pHComSeqGetFixPntManually()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'Get Fix Point Manually' pushbutton
Read only property	pHComSeqGetIntervalForCG()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'Get Interval For Center Of Gravity' pushbutton
Read only property	pHComSeqGetROI()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'Get ROI' pushbutton
Read only property	pHComSeqLastSample()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'Last Sample' pushbutton
Read only property	pHComSeqNextSample()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'Next Sample' pushbutton
Read only property	pHComSeqOptions()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'Options' pushbutton
Read only property	pHComSeqPlayBackward()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'Play Backward' pushbutton
Read only property	pHComSeqPlayForward()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'Play Forward' pushbutton
Read only property	pHComSeqPlayStop()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'Stop Play' pushbutton
Read only property	pHComSeqPreviousSample()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'Previous Sample' pushbutton
Read only property	pHComSeqProceedProcessing()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'Proceed Processing' pushbutton
Read only property	pHComSeqRecallAreaForFixPoint()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'REcall Area For Fix Point' pushbutton

Read only property	pHComSeqRecallIntervalForCG()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'Recall Interval For Center Of Gravity' pushbutton
Read only property	pHComSeqResetProcessing()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'Reset Processing' pushbutton
Read only property	pHComSeqStartAcquisition()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'Start Acquisition' pushbutton
Read only property	pHComSeqStopAcquisition()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'Stop Acquisition' pushbutton
Read only property	pHComSeqStopProcessing()	HCommand	Returns an object reference to the HCheck object associated with the options dialogs 'Stop Processing' pushbutton
Read only property	pHDisSeqBytesPerImage()	HDisp	Returns an object reference to the HDisp object associated with the sequence dialogs 'Bytes Per Image' display area
Read only property	pHDisSeqExcludedSamples()	HDisp	Returns an object reference to the HDisp object associated with the sequence dialogs 'Excluded Samples' display area
Read only property	pHDisSeqFileName()	HDisp	Returns an object reference to the HDisp object associated with the sequence dialogs 'File Name' display area
Read only property	pHDisSeqFixPoint()	HDisp	Returns an object reference to the HDisp object associated with the sequence dialogs 'Fix Point' display area
Read only property	pHDisSeqFreeImages()	HDisp	Returns an object reference to the HDisp object associated with the sequence dialogs 'Free Images' display area
Read only property	pHDisSeqMeanProcessing()	HDisp	Returns an object reference to the HDisp object associated with the sequence dialogs 'Mean Value of Processing' display area
Read only property	pHDisSeqNrSamples()	HDisp	Returns an object reference to the HDisp object associated with the sequence dialogs 'Number Of Samples' display area
Read only property	pHDisSeqRemainingSecs()	HDisp	Returns an object reference to the HDisp object associated with the sequence dialogs 'Remaining Seconds' display area
Read only property	pHDisSeqSDProcessing()	HDisp	Returns an object reference to the HDisp object associated with the sequence dialogs 'Standard Deviation of Processing' display area
Read only property	pHDisSeqSpaceOnDisk()	HDisp	Returns an object reference to the HDisp object associated with the sequence dialogs 'Space On Disk' display area
Read only property	pHDisSeqUndefinedSamples()	HDisp	Returns an object reference to the HDisp object associated with the sequence dialogs 'Undefined Samples' display area
Read only property	pHDisSeqValidSamples()	HDisp	Returns an object reference to the HDisp object associated with the sequence dialogs 'Valid Samples' display area
Read only property	pHEdnSeqCurrentSample()	HEditNumber	Returns an object reference to the HEditNumber object associated with the sequence dialogs 'Current Sample' editbox
Read only property	pHEdnSeqNoOfLoops()	HEditNumber	Returns an object reference to the HEditNumber object associated with the sequence dialogs 'Number Of Loops' editbox
Read only property	pHEdnSeqProfileNo()	HEditNumber	Returns an object reference to the HEditNumber object associated with the sequence dialogs 'Profile Number' editbox
Read only property	pHEstSeqAcquireI	HEditStri	Returns an object reference to the HEditString object

	nterval()	ng	associated with the sequence dialogs 'Acquire Interval' editbox
Read only property	pHEstSeqFirstImgToStore()	HEditString	Returns an object reference to the HEditString object associated with the sequence dialogs 'First Image To Store' editbox
Read only property	pHEstSeqFirstPrfToStore()	HEditString	Returns an object reference to the HEditString object associated with the sequence dialogs 'First Profile To Store' editbox
Read only property	pHEstSeqPlayInterval()	HEditString	Returns an object reference to the HEditString object associated with the sequence dialogs 'Play Interval' editbox
Read only property	pHFraSeqAcquireImages()	HFrame	Returns an object reference to the HFrame object associated with the sequence dialogs 'Acquire Images' frame
Read only property	pHFraSeqAcquireProfiles()	HFrame	Returns an object reference to the HFrame object associated with the sequence dialogs 'Acquire Profiles' frame
Read only property	pHFraSeqCycleFeatures()	HFrame	Returns an object reference to the HFrame object associated with the sequence dialogs 'Cycle Features' frame
Read only property	pHFraSeqCycleStatus()	HFrame	Returns an object reference to the HFrame object associated with the sequence dialogs 'Cycle Status' frame
Read only property	pHFraSeqFixPoint()	HFrame	Returns an object reference to the HFrame object associated with the sequence dialogs 'Fix Point' frame
Read only property	pHFraSeqNeverLoadToRAM()	HFrame	Returns an object reference to the HFrame object associated with the sequence dialogs 'Never Load To RAM' frame
Read only property	pHFraSeqProfileSets()	HFrame	Returns an object reference to the HFrame object associated with the sequence dialogs 'Profile Sets' frame
Read only property	pHFraSeqSpecial()	HFrame	Returns an object reference to the HFrame object associated with the sequence dialogs 'Special' frame
Read only property	pHRadSeqAcqSpeed()	HRadios	Returns an object reference to the HRadios object associated with the sequence dialogs 'Acquisition Speed' radiobutton group
Read only property	pHRadSeqAcquisitionMode()	HRadios	Returns an object reference to the HRadios object associated with the sequence dialogs 'Acquisition Mode' radiobutton group
Read only property	pHRadSeqCorrectionDirection()	HRadios	Returns an object reference to the HRadios object associated with the sequence dialogs 'Correction Dialog' radiobutton group
Read only property	pHRadSeqProcessingMethod()	HRadios	Returns an object reference to the HRadios object associated with the sequence dialogs 'Processing Method' radiobutton group
Read only property	pHRadSeqSampleType()	HRadios	Returns an object reference to the HRadios object associated with the sequence dialogs 'Sample Type' radiobutton group
Read only property	pHRadSeqStoreImages()	HRadios	Returns an object reference to the HRadios object associated with the sequence dialogs 'Store Images' radiobutton group
Read only property	pHTabSeqTopic()	HTab	Returns an object reference to the HTab object associated with the sequence dialogs 'Topic' tab group
Read only property	pHWinOptDlg()	HWindow	Returns an object reference to the HWindow object associated with the option dialogs main window
Read only property	pHWinOptOptions()	HWindow	Returns an object reference to the HWindow object associated with the option dialogs options window
Read only property	pHWinSeqDlg()	HWindow	Returns an object reference to the HWindow object associated with the sequence dialogs main window
Read only property	piSeqProcessing()	Integer	Returns a value which defines which Processing Mode will be selected. Possible values are defined in the enumeration SequenceProcessing
Read/write property	pfRestoreWindow	Integer	Sets or returns a value which defines whether the

	Pos()		window position, its size and window state is restored when the window is displayed again after it has been closed
Read/write property	pfUserIF()	Integer	Sets or returns a value which defines whether the associated window should be shown on screen. A window is only displayed under the following condition: pfUserIF=TRUE, pfVisible=TRUE, pfHideForm=FALSE
Function	pfLoadSequence(Optional sFileOpt As Variant, Optional iSeqModeOpt As Variant)	Integer	Loads a sequence
Function	pfSaveSequence(Optional sFile As Variant, Optional sAreDefault As Variant, Optional fSaveROI As Variant)	Integer	Saves a sequence
Sub	pAsyncLoadSequence(Optional sFileOpt As Variant, Optional iSeqModeOpt As Variant)		Loads a Sequence asynchronously
Sub	pAsyncSaveSequence(Optional sFile As Variant, Optional sAreDefault As Variant, Optional fSaveROI As Variant)		Saves a Sequence
Sub	pAsyncSeqStartAcquisition()		Starts Sequence Acquisition asynchronously
Sub	pCorrectSequence(ByVal fBack As Integer, ByVal fShad As Integer, ByVal fCurvCorr As Integer)		Performs a correction on a sequence (background, shading or curvature correction)
Sub	pRemoveSequence()		Removes a sequence
HJitter			
Event	Message(ByVal sMessage As String)		Event which is raised to inform the user of ongoing operations

HGeneral
HACam

Read only property	pHCh4CamClearFrameBuffer()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Clear frame buffer on start' checkbox
Read only property	pHCh4CamDoRTBacksub()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Do Real Time Backsub' checkbox
Read only property	pHCh4CamFGTriger()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Clear frame buffer on start' checkbox

Read only property	pHCo4CamAcquire()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Acquire' pushbutton
Read only property	pHCo4CamDoPCSetup()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Do Photon Counting Setup' pushbutton
Read only property	pHCo4CamFreeze()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Freeze' pushbutton
Read only property	pHCo4CamGetBGData()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Get Background Data' pushbutton
Read only property	pHCo4CamLive()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Live' pushbutton
Read only property	pHCo4CamSingleExposure()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Single Exposure' pushbutton
Read only property	pHCo4CamStop()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Stop' pushbutton
Read only property	pHComCamDown()	HCommand	Returns an object reference to the HCommand object associated with the camera dialogs 'Down' pushbutton (Small down arrow)
Read only property	pHComCamUp()	HCommand	Returns an object reference to the HCommand object associated with the camera dialogs 'Up' pushbutton (Small up arrow)
Read only property	pHComSetupCancel()	HCommand	Returns an object reference to the HCommand object associated with the camera setup dialogs 'Cancel' pushbutton
Read only property	pHComSetupOK()	HCommand	Returns an object reference to the HCommand object associated with the camera setup dialogs 'Setup' pushbutton
Read only property	pHDi4CamAboveThreshold()	HDisp4Array	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Above Threshold' display area
Read only property	pHDi4CamMessage()	HDisp4Array	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Message' display area
Read only property	pHEd4CamNrExposures()	HEditNumber4Array	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Number of Exposures' editbox
Read only property	pHEd4CamThreshold()	HEditNumber4Array	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Threshold' editbox
Read only property	pHEn4CamIntegrAfterTrig()	HEntry4Array	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Integrate After Trigger' editbox
Read only property	pHFr4CamAction()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Action' frame
Read only property	pHFr4CamFGTrigger()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Frame Grabber Trigger' frame
Read only property	pHFr4CamIntegration()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Integration' frame
Read only property	pHFr4CamPcMode()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Photon Counting Mode' frame
Read only property	pHFr4CamPhotonCounting()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Photon Counting' frame

Read only property	pHFr4CamRTBS()	HFrame4 Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Real Time Backsub' frame
Read only property	pHPr4CamPercent ()	HProgress 4Array	Returns an object reference to the HProgress4Array object associated with the camera dialogs 'Percent' progress bar
Read only property	pHRa4CamPcMod e()	HRadios4 Array	Returns an object reference to the HRadios4Array object associated with the camera dialogs 'Photon Counting Mode' radiobutton group
Read only property	pHRadSetupStartA cq()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Start Acquisition' radiobutton group
Read only property	pHRadSetupSync Mode()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Sync Mode' radiobutton group
Read only property	pHTabCamAcqMo de()	HTab	Returns an object reference to the HTab object associated with the camera dialogs 'Acquisition mode' tab group
Read only property	pHWinCamDlg()	HWindow	Returns an object reference to the HWindow object associated with the camera dialogs main window
Read only property	pHWinSetupAMV SSetting()	HWindow	Returns an object reference to the HWindow object associated with the camera setup dialogs window displaying the AMVS settings
Read only property	pHWinSetupDlg()	HWindow	Returns an object reference to the HWindow object associated with the camera setup dialogs main window
Read only property	piDatTypeForAcq uire()	Integer	Returns the data type in ACQUIRE mode
Read only property	piDatTypeForAI()	Integer	Returns the data type in Analog Integration mode
Read only property	piDatTypeForLive ()	Integer	Returns the data type in LIVE mode
Read only property	piDatTypeForPC()	Integer	Returns the data type in Photon Counting mode
Read only property	psCameraName()	String	Returns the camera name
Read/write property	pfRestoreWindow Pos()	Integer	Sets or returns a value which defines whether the window position, its size and window state is restored when the window is displayed again after it has been closed
Read/write property	pfStreakOperate()	Integer	A value that tells the camera object whether vertical streak operation is switched on. Don't modify this entry otherwise streak trigger handshake may not longer work
Read/write property	pfStreakOperateD TBE()	Integer	A value that tells the camera object whether horizontal streak operation is switched on. Don't modify this entry otherwise streak trigger handshake may not longer work
Read/write property	pfStreakUseDTBE ()	Integer	Tells the camera object that a Dual time base extender is currently used for trigger handshake
Read/write property	pfUserIF()	Integer	Sets or returns a value which defines whether the associated window should be shown on screen. A window is only displayed under the following condition: pfUserIF=TRUE, pfVisible=TRUE, pfHideForm=FALSE
Read/write property	piStreakTriggerMe thod()	Integer	Tells the camera (or acquisition/sequence) object the streak trigger method. Don't modify this entry otherwise Steak trigger handshake may no longer work
Function	pGetStreakRelated CCDCaps(ByRef fCCDCanExtTrig As Integer, ByRef fCCDCanStartStop As Integer, ByRef sExtTrigCCDMeth od As String, ByRef		Returns streak trigger related capabilities of the CCD camera

	sExtTrigCPUMethod As String, ByRef sStartStopCCDMethod As String, ByRef sStartStopCPUMethod As String)		
Function	psGetStatus()	String	Returns the current camera status string
Sub	pStopAcquisition()		Stops a currently running acquisition

HC474295

Read only property	pHCh4CamClearFrameBuffer()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Clear frame buffer on start' checkbox
Read only property	pHCh4CamDoRTBacksub()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Do Real Time Backsub' checkbox
Read only property	pHCh4CamStreakTrigger()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Streak Trigger Active' checkbox
Read only property	pHCo4CamAcquire()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Acquire' pushbuttonpushbutton
Read only property	pHCo4CamDoPCSetup()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Do Photon Counting Setup' pushbutton
Read only property	pHCo4CamFreeze()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Freeze' pushbutton
Read only property	pHCo4CamGetBGData()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Get Background Data' pushbutton
Read only property	pHCo4CamLive()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Live' pushbutton
Read only property	pHCo4CamSingleExposure()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Single Exposure' pushbutton
Read only property	pHCo4CamStop()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Stop' pushbutton
Read only property	pHComCamDown()	HCommand	Returns an object reference to the HCommand object associated with the camera dialogs 'Down' pushbutton (Small down arrow)
Read only property	pHComCamUp()	HCommand	Returns an object reference to the HCommand object associated with the camera dialogs 'Up' pushbutton (Small up arrow)
Read only property	pHComSetupCancel()	HCommand	Returns an object reference to the HCommand object associated with the camera setup dialogs 'Cancel' pushbutton
Read only property	pHComSetupOK()	HCommand	Returns an object reference to the HCommand object associated with the camera setup dialogs 'OK' pushbutton
Read only property	pHDi4CamAboveThreshold()	HDisp4Array	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Above Threshold' display area
Read only property	pHDi4CamMessage()	HDisp4Array	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Message' display area
Read only property	pHDi4CamStreakTrigger()	HDisp4Array	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Streak Trigger' display area

Read only property	pHDisSetupCameraInfo()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Camera Info' display area
Read only property	pHDisSetupScanArea()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Scan Area' display area
Read only property	pHEd4CamNrExposures()	HEditNumber4Array	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Number of Exposures' editbox
Read only property	pHEd4CamNrTrigger()	HEditNumber4Array	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Number of Streak Trigger' editbox
Read only property	pHEd4CamThreshold()	HEditNumber4Array	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Threshold' editbox
Read only property	pHEdnSetupHOffs()	HEditNumber	Returns an object reference to the HEditNumber object associated with the setup dialogs 'Horizontal Offset' (subarray mode) editbox
Read only property	pHEdnSetupHWidth()	HEditNumber	Returns an object reference to the HEditNumber object associated with the setup dialogs 'Horizontal Width' (subarray mode) editbox
Read only property	pHEdnSetupVOffs()	HEditNumber	Returns an object reference to the HEditNumber object associated with the setup dialogs 'Vertical Offset' (subarray mode) editbox
Read only property	pHEdnSetupVWidth()	HEditNumber	Returns an object reference to the HEditNumber object associated with the setup dialogs 'Vertical Width' (subarray mode) editbox
Read only property	pHEn4CamExposure()	HEntry4Array	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Exposure Time' entrybox
Read only property	pHEn4CamGain()	HEntry4Array	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Analog Gain' entrybox
Read only property	pHEn4CamOffset()	HEntry4Array	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Analog Offset' entrybox
Read only property	pHFr4CamAction()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Action' frame
Read only property	pHFr4CamExposureTime()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Exposure Time' frame
Read only property	pHFr4CamGainOffset()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Analog Gain' frame
Read only property	pHFr4CamIntegration()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Integration' frame
Read only property	pHFr4CamPhotonCounting()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Photon Counting' frame
Read only property	pHFr4CamRTBS()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Real Time Backsub' frame
Read only property	pHFr4CamStreakTrigger()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Streak Trigger' frame
Read only property	pHFraSetupSubarray()	HFrame	Returns an object reference to the HFrame object associated with the setup dialogs 'Subarray' frame
Read only property	pHPr4CamPercent()	HProgress4Array	Returns an object reference to the HProgress4Array object associated with the camera dialogs 'Percent' progress bar
Read only property	pHRa4CamPcMode()	HRadios4Array	Returns an object reference to the HRadios4Array object associated with the camera dialogs 'Photon Counting Mode' radiobutton group

Read only property	pHRadSetupBinning()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Binning' radiobutton group
Read only property	pHRadSetupCCDArea()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'CCD Area' radiobutton group
Read only property	pHRadSetupLightMode()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Light Mode' radiobutton group
Read only property	pHRadSetupScanMode()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Scan Mode' radiobutton group
Read only property	pHRadSetupTimingMode()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Timing Mode' radiobutton group
Read only property	pHRadSetupTriggerMode()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Trigger Mode' radiobutton group
Read only property	pHRadSetupTriggerPolarity()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Trigger Polarity' radiobutton group
Read only property	pHRadSetupTriggerSource()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Trigger Source' radiobutton group
Read only property	pHTabCamAcqMode()	HTab	Returns an object reference to the HTab object associated with the camera dialogs 'Acquisition mode' tab group
Read only property	pHWinCamDlg()	HWindow	Returns an object reference to the HWindow object associated with the camera dialogs main window
Read only property	pHWinSetupDlg()	HWindow	Returns an object reference to the HWindow object associated with the camera setup dialogs main window
Read only property	piDatTypeForAcquire()	Integer	Returns the data type in ACQUIRE mode
Read only property	piDatTypeForAI()	Integer	Returns the data type in Analog Integration mode
Read only property	piDatTypeForLive()	Integer	Returns the data type in LIVE mode
Read only property	piDatTypeForPC()	Integer	Returns the data type in Photon Counting mode
Read only property	psCameraName()	String	Returns the camera name
Read/write property	pfRestoreWindowPos()	Integer	Sets or returns a value which defines whether the window position, its size and window state is restored when the window is displayed again after it has been closed
Read/write property	pfStreakOperate()	Integer	A value that tells the camera object whether vertical streak operation is switched on. Don't modify this entry otherwise streak trigger handshake may not longer work
Read/write property	pfStreakOperateDTBE()	Integer	A value that tells the camera object whether horizontal streak operation is switched on. Don't modify this entry otherwise streak trigger handshake may not longer work
Read/write property	pfStreakUseDTBE()	Integer	Tells the camera object that a Dual time base extender is currently used for trigger handshake
Read/write property	pfUserIF()	Integer	Sets or returns a value which defines whether the associated window should be shown on screen. A window is only displayed under the following condition: pfUserIF=TRUE, pfVisible=TRUE, pfHideForm=FALSE
Read/write property	piStreakTriggerMethod()	Integer	Tells the camera (or acquisition/sequence) object the streak trigger method. Don't modify this entry otherwise Steak trigger handshake may no longer work
Function	pGetStreakRelatedCCDCaps(ByRef fCCDCanExtTrig		Returns streak trigger related capabilities of the CCD camera

	As Integer, ByRef fCCDCanStartStop As Integer, ByRef sExtTrigCCDMethod As String, ByRef sExtTrigCPUMethod As String, ByRef sStartStopCCDMethod As String, ByRef sStartStopCPUMethod As String)		
Function	psGetStatus()	String	Returns the current camera status string
Sub	pStopAcquisition()		Stops a currently running acquisition

HC4880

Read only property	pHCh4CamClearFrameBuffer()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Clear frame buffer on start' checkbox
Read only property	pHCh4CamDoRTBacksub()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Do Real Time Backsub' checkbox
Read only property	pHCh4CamStreakTrigger()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Streak Trigger Active' checkbox
Read only property	pHChkSetupCooler()	HCheck	Returns an object reference to the HCheck object associated with the camera setup dialogs 'Cooler' checkbox
Read only property	pHChkSetupPanelSwitch()	HCheck	Returns an object reference to the HCheck object associated with the camera setup dialogs 'Panel Switch' checkbox
Read only property	pHChkSetupShadingControl()	HCheck	Returns an object reference to the HCheck object associated with the camera setup dialogs 'Shading Control' checkbox
Read only property	pHCo4CamAcquire()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Acquire' pushbutton
Read only property	pHCo4CamDoPCSetup()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Do Photon Counting Setup' pushbutton
Read only property	pHCo4CamFreeze()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Freeze' pushbutton
Read only property	pHCo4CamGetBGData()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Get Background Data' pushbutton
Read only property	pHCo4CamLive()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Live' pushbutton
Read only property	pHCo4CamSingleExposure()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Single Exposure' pushbutton
Read only property	pHCo4CamStop()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Stop' pushbutton
Read only property	pHComCamDown()	HCommand	Returns an object reference to the HCommand object associated with the camera dialogs 'Down' pushbutton (Small down arrow)
Read only property	pHComCamUp()	HCommand	Returns an object reference to the HCommand object associated with the camera dialogs 'Up' pushbutton

			(Small up arrow)
Read only property	pHComSetupCancel()	HCommand	Returns an object reference to the HCommand object associated with the camera setup dialogs 'Cancel' pushbutton
Read only property	pHComSetupOK()	HCommand	Returns an object reference to the HCommand object associated with the camera setup dialogs 'OK' pushbutton
Read only property	pHDi4CamAboveThreshold()	HDisp4Array	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Above Threshold' display area
Read only property	pHDi4CamMessage()	HDisp4Array	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Message' display area
Read only property	pHDi4CamStreakTrigger()	HDisp4Array	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Streak Trigger' display area
Read only property	pHDisSetupActualTemperature()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Actual Temperature' display area
Read only property	pHDisSetupCameraType()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Camera Type' display area
Read only property	pHDisSetupChip()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Chip' display area
Read only property	pHDisSetupVersion()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Version' display area
Read only property	pHDisSetupVolGain()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Volume Gain' display area
Read only property	pHDisSetupVolOffset()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Volume Offset' display area
Read only property	pHDisSubLower()	HDisp	Returns an object reference to the HDisp object associated with the camera subarray dialogs 'Lower' display area
Read only property	pHDisSubUpper()	HDisp	Returns an object reference to the HDisp object associated with the camera subarray dialogs 'Upper' display area
Read only property	pHDisSubWidth()	HDisp	Returns an object reference to the HDisp object associated with the camera subarray dialogs 'Width' display area
Read only property	pHEd4CamATN()	HEditNumber4Array	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Acquire Trigger Number' editbox
Read only property	pHEd4CamNrExposures()	HEditNumber4Array	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Number of Exposures' editbox
Read only property	pHEd4CamNrTrigger()	HEditNumber4Array	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Number of Streak Trigger' editbox
Read only property	pHEd4CamThreshold()	HEditNumber4Array	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Threshold' editbox
Read only property	pHEdnSetupTargetTemperature()	HEditNumber	Returns an object reference to the HEditNumber object associated with the camera setup dialogs 'Target Temperature' editbox
Read only property	pHEdnSubVerBinning()	HEditNumber	Returns an object reference to the HEditNumber object associated with the camera subarray dialogs 'Binning' editbox
Read only property	pHEn4CamAMD()	HEntry4A	Returns an object reference to the HEntry4Array object

		rray	associated with the camera dialogs 'Acquire Mode' entrybox
Read only property	pHEn4CamASH()	HEntry4Array	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Acquire Shutter' entrybox
Read only property	pHEn4CamATP()	HEntry4Array	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Acquire Trigger Polarity' entrybox
Read only property	pHEn4CamExposure()	HEntry4Array	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Exposure Time' entrybox
Read only property	pHEn4CamMCP()	HEntry4Array	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'MCP' entrybox
Read only property	pHEn4CamSOP()	HEntry4Array	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Scan Optical Black' entrybox
Read only property	pHEn4CamSPX()	HEntry4Array	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Superpixel' entrybox
Read only property	pHEn4CamTDY()	HEntry4Array	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Trigger Delay' entrybox
Read only property	pHEntSetupGain()	HEntry	Returns an object reference to the HEntry object associated with the camera setup dialogs 'Gain' entrybox
Read only property	pHEntSetupOffset()	HEntry	Returns an object reference to the HEntry object associated with the camera setup dialogs 'Offset' entrybox
Read only property	pHEntSubHorBinning()	HEntry	Returns an object reference to the HEntry object associated with the camera setup dialogs 'Binning' entrybox
Read only property	pHEntSubLower()	HEntry	Returns an object reference to the HEntry object associated with the camera subarray dialogs 'Lower' entrybox
Read only property	pHEntSubUpper()	HEntry	Returns an object reference to the HEntry object associated with the camera subarray dialogs 'Upper' entrybox
Read only property	pHFr4CamAction()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Action' frame
Read only property	pHFr4CamAmpGain()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Amp Gain' frame
Read only property	pHFr4CamExposureTime()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Exposure Time' frame
Read only property	pHFr4CamIntegration()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Integration' frame
Read only property	pHFr4CamPhotonCounting()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Photon Counting' frame
Read only property	pHFr4CamRTBS()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Real Time Backsub' frame
Read only property	pHFr4CamScanMode()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Scan Mode' frame
Read only property	pHFr4CamSpecial()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Special' frame
Read only property	pHFr4CamStreakTrigger()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Streak Trigger' frame
Read only property	pHFraSetupCameraInfo()	HFrame	Returns an object reference to the HFrame4Array object associated with the camera setup dialogs 'Camera Info' frame
Read only property	pHFraSetupContrastEnhancement()	HFrame	Returns an object reference to the HFrame4Array object associated with the camera setup dialogs 'Contrast

			Enhacement' frame
Read only property	pHFraSetupTemperature()	HFrame	Returns an object reference to the HFrame4Array object associated with the camera setup dialogs 'Temperature' frame
Read only property	pHPr4CamPercent()	HProgress4Array	Returns an object reference to the HProgress4Array object associated with the camera dialogs 'Percent' progress bar
Read only property	pHRa4CamAmpGain()	HRadios4Array	Returns an object reference to the HRadios4Array object associated with the camera dialogs 'Amp Gain' radiobutton group
Read only property	pHRa4CamPcMode()	HRadios4Array	Returns an object reference to the HRadios4Array object associated with the camera dialogs 'Photon Counting Mode' radiobutton group
Read only property	pHRa4CamSMD()	HRadios4Array	Returns an object reference to the HRadios4Array object associated with the camera dialogs 'Scan Mode' radiobutton group
Read only property	pHRadSetupContrastEnhancement()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Contast Enhancement'
Read only property	pHRadSubHorSubarray()	HRadios	Returns an object reference to the HRadios object associated with the camera subarray dialogs 'Horizontal Subarray' radiobutton group
Read only property	pHTabCamAcqMode()	HTab	Returns an object reference to the HTab object associated with the camera dialogs 'Acquisition mode' tab group
Read only property	pHWinCamDlg()	HWindow	Returns an object reference to the HWindow object associated with the camera dialogs main window
Read only property	pHWinSetupDlg()	HWindow	Returns an object reference to the HWindow object associated with the camera setup dialogs main window
Read only property	pHWinSubDlg()	HWindow	Returns an object reference to the HWindow object associated with the camera subarry dialogs main window
Read only property	piDatTypeForAcquire()	Integer	Returns the data type in ACQUIRE mode
Read only property	piDatTypeForAI()	Integer	Returns the data type in Analog Integration mode
Read only property	piDatTypeForLive()	Integer	Returns the data type in LIVE mode
Read only property	piDatTypeForPC()	Integer	Returns the data type in Photon Counting mode
Read only property	psCameraName()	String	Returns the camera name
Read/write property	pfRestoreWindowPos()	Integer	Sets or returns a value which defines whether the window position, its size and window state is restored when the window is displayed again after it has been closed
Read/write property	pfStreakOperate()	Integer	A value that tells the camera object whether vertical streak operation is switched on. Don't modify this entry otherwise streak trigger handshake may not longer work
Read/write property	pfStreakOperateDTBE()	Integer	A value that tells the camera object whether horizontal streak operation is switched on. Don't modify this entry otherwise streak trigger handshake may not longer work
Read/write property	pfStreakUseDTBE()	Integer	Tells the camera object that a Dual time base extender is currently used for trigger handshake
Read/write property	pfUserIF()	Integer	Sets or returns a value which defines whether the associated window should be shown on screen. A window is only displayed under the following condition: pfUserIF=TRUE, pfVisible=TRUE, pfHideForm=FALSE
Read/write property	piStreakTriggerMethod()	Integer	Tells the camera (or acquisition/sequence) object the streak trigger method. Don't modify this entry otherwise Steak trigger handshake may no longer work
Function	pGetStreakRelatedCCDCaps(ByRef fCCDCanExtTrig		Returns streak trigger related capabilities of the CCD camera

	As Integer, ByRef fCCDCanStartStop As Integer, ByRef sExtTrigCCDMethod As String, ByRef sExtTrigCPUMethod As String, ByRef sStartStopCCDMethod As String, ByRef sStartStopCPUMethod As String)		
Function	psGetStatus()	String	Returns the current camera status string
Sub	pStopAcquisition()		Stops a currently running acquisition

HC488080

Read only property	pHCh4CamClearFrameBuffer()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Clear frame buffer on start' checkbox
Read only property	pHCh4CamDoRTBacksub()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Do Real Time Backsub' checkbox
Read only property	pHCh4CamExternalTrigger()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Streak Trigger Active' checkbox
Read only property	pHCh4CamStreakTrigger()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Streak Trigger Active' checkbox
Read only property	pHCo4CamAcquire()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Acquire' pushbuttonpushbutton
Read only property	pHCo4CamDoPCSetup()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Do Photon Counting Setup' pushbutton
Read only property	pHCo4CamFreeze()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Freeze' pushbutton
Read only property	pHCo4CamGetBGData()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Get Background Data' pushbutton
Read only property	pHCo4CamLive()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Live' pushbutton
Read only property	pHCo4CamSingleExposure()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Single Exposure' pushbutton
Read only property	pHCo4CamStop()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Stop' pushbutton
Read only property	pHComCamDown()	HCommand	Returns an object reference to the HCommand object associated with the camera dialogs 'Down' pushbutton (Small down arrow)
Read only property	pHComCamUp()	HCommand	Returns an object reference to the HCommand object associated with the camera dialogs 'Up' pushbutton (Small up arrow)
Read only property	pHComSetupCancel()	HCommand	Returns an object reference to the HCommand object associated with the camera setup dialogs 'Cancel' pushbutton
Read only property	pHComSetupOK()	HCommand	Returns an object reference to the HCommand object associated with the camera setup dialogs 'OK' pushbutton

Read only property	pHDi4CamAbove Threshold()	HDisp4Ar ray	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Above Threshold' display area
Read only property	pHDi4CamFrmSht ()	HDisp4Ar ray	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Frames/Shutter' display area
Read only property	pHDi4CamMessag e()	HDisp4Ar ray	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Message' display area
Read only property	pHDi4CamStreakT rigger()	HDisp4Ar ray	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Streak Trigger' display area
Read only property	pHDisSetupChip()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Chip' display area
Read only property	pHDisSetupROM( )	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'ROM' display area
Read only property	pHDisSetupType()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Type' display area
Read only property	pHDisSetupVersio n()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Version' display area
Read only property	pHDisSetupVolGa in()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Volume Gain' display area
Read only property	pHDisSetupVolOff set()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Volume Offset' display area
Read only property	pHEd4CamNrExp osures()	HEditNu mber4Arr ay	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Number of Exposures' editbox
Read only property	pHEd4CamNrTrig ger()	HEditNu mber4Arr ay	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Number of Streak Trigger' editbox
Read only property	pHEd4CamThresh old()	HEditNu mber4Arr ay	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Threshold' editbox
Read only property	pHEn4CamExposu re()	HEntry4A rray	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Exposure Time' entrybox
Read only property	pHEntSetupGain()	HEntry	Returns an object reference to the HEntry object associated with the camera setup dialogs 'Gain' entrybox
Read only property	pHEntSetupOffset( )	HEntry	Returns an object reference to the HEntry object associated with the camera setup dialogs 'Offset' entrybox
Read only property	pHEs4CamSMDE xtended()	HEditStri ng4Array	Returns an object reference to the HEditString4Array object associated with the camera dialogs 'Scan Mode Extended' editbox
Read only property	pHFr4CamAction( )	HFrame4 Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Action' frame
Read only property	pHFr4CamExposu reTime()	HFrame4 Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Exposure Time' frame
Read only property	pHFr4CamExterna lTrigger()	HFrame4 Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Streak Trigger' frame
Read only property	pHFr4CamIntegrat ion()	HFrame4 Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Integration' frame
Read only property	pHFr4CamPhoton Counting()	HFrame4 Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Photon Counting'

			frame
Read only property	pHFr4CamRTBS()	HFrame4 Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Real Time Backsub' frame
Read only property	pHFr4CamScanMode()	HFrame4 Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Scan Mode' frame
Read only property	pHFr4CamStreakTrigger()	HFrame4 Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Streak Trigger'
Read only property	pHFraSetupCameraInfo()	HFrame	Returns an object reference to the HFrame4Array object associated with the camera setup dialogs 'Camera Info' frame
Read only property	pHFraSetupContrastEnhancement()	HFrame	Returns an object reference to the HFrame4Array object associated with the camera setup dialogs 'Contrast Enhacement' frame
Read only property	pHPr4CamPercent()	HProgress4Array	Returns an object reference to the HProgress4Array object associated with the camera dialogs 'Percent' progress bar
Read only property	pHRa4CamAmpGain()	HRadios4 Array	Returns an object reference to the HRadios4Array object associated with the camera dialogs 'Amp Gain' radiobutton group
Read only property	pHRa4CamPcMode()	HRadios4 Array	Returns an object reference to the HRadios4Array object associated with the camera dialogs 'Photon Counting Mode' radiobutton group
Read only property	pHRa4CamSMD()	HRadios4 Array	Returns an object reference to the HRadios4Array object associated with the camera dialogs 'Scan Mode' radiobutton group
Read only property	pHRadSetupContrastEnhancement()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Contast Enhancement'
Read only property	pHTabCamAcqMode()	HTab	Returns an object reference to the HTab object associated with the camera dialogs 'Acquisition mode' tab group
Read only property	pHWinCamDlg()	HWindow	Returns an object reference to the HWindow object associated with the camera dialogs main window
Read only property	pHWinSetupDlg()	HWindow	Returns an object reference to the HWindow object associated with the camera setup dialogs main window
Read only property	piDatTypeForAcquire()	Integer	Returns the data type in ACQUIRE mode
Read only property	piDatTypeForAI()	Integer	Returns the data type in Analog Integration mode
Read only property	piDatTypeForLive()	Integer	Returns the data type in LIVE mode
Read only property	piDatTypeForPC()	Integer	Returns the data type in Photon Counting mode
Read only property	psCameraName()	String	Returns the camera name
Read/write property	pfRestoreWindowPos()	Integer	Sets or returns a value which defines whether the window position, its size and window state is restored when the window is displayed again after it has been closed
Read/write property	pfStreakOperate()	Integer	A value that tells the camera object whether vertical streak operation is switched on. Don't modify this entry otherwise streak trigger handshake may not longer work
Read/write property	pfStreakOperateDTBE()	Integer	A value that tells the camera object whether horizontal streak operation is switched on. Don't modify this entry otherwise streak trigger handshake may not longer work
Read/write property	pfStreakUseDTBE()	Integer	Tells the camera object that a Dual time base extender is currently used for trigger handshake
Read/write property	pfUserIF()	Integer	Sets or returns a value which defines whether the associated window should be shown on screen. A window is only displayed under the following condition: pfUserIF=TRUE, pfVisible=TRUE, pfHideForm=FALSE
Function	pGetStreakRelated		Tells the camera (or acquisition/sequence) object the

	CCDCaps(ByRef fCCDCanExtTrig As Integer, ByRef fCCDCanStartStop As Integer, ByRef sExtTrigCCDMethod As String, ByRef sExtTrigCPUMethod As String, ByRef sStartStopCCDMethod As String, ByRef sStartStopCPUMethod As String)		streak trigger method. Don't modify this entry otherwise Steak trigger handshake may no longer work
Function	psGetStatus()	String	Returns the current camera status string
Sub	pStopAcquisition()		Stops a currently running acquisition

HC474298

Read only property	pHCh4CamClearFrameBuffer()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Clear frame buffer on start' checkbox
Read only property	pHCh4CamDoRTBacksub()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Do Real Time Backsub' checkbox
Read only property	pHCh4CamStreakTrigger()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Streak Trigger Active' checkbox
Read only property	pHChkSetupCooler()	HCheck	
Read only property	pHCo4CamAcquire()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Acquire' pushbutton
Read only property	pHCo4CamDoPCSetup()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Do Photon Counting Setup' pushbutton
Read only property	pHCo4CamFreeze()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Freeze' pushbutton
Read only property	pHCo4CamGetBGData()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Get Background Data' pushbutton
Read only property	pHCo4CamLive()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Live' pushbutton
Read only property	pHCo4CamSingleExposure()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Single Exposure' pushbutton
Read only property	pHCo4CamStop()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Stop' pushbutton
Read only property	pHComCamDown()	HCommand	Returns an object reference to the HCommand object associated with the camera dialogs 'Down' pushbutton (Small down arrow)
Read only property	pHComCamUp()	HCommand	Returns an object reference to the HCommand object associated with the camera dialogs 'Up' pushbutton (Small up arrow)
Read only property	pHComSetupCancel()	HCommand	Returns an object reference to the HCommand object associated with the camera setup dialogs 'Cancel' pushbutton
Read only property	pHComSetupOK()	HComma	Returns an object reference to the HCommand object

		nd	associated with the camera setup dialogs 'OK' pushbutton
Read only property	pHDi4CamAbove Threshold()	HDisp4Ar ray	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Above Threshold' display area
Read only property	pHDi4CamMessag e()	HDisp4Ar ray	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Message' display area
Read only property	pHDi4CamStreakT rigger()	HDisp4Ar ray	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Streak Trigger' display area
Read only property	pHDisSetupActual Temperature()	HDisp	
Read only property	pHDisSetupCamer aInfo()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Camera Info' display area
Read only property	pHDisSetupConfig uration()	HDisp	
Read only property	pHDisSetupVolGa in()	HDisp	
Read only property	pHDisSetupVolOff set()	HDisp	
Read only property	pHDisSubLower()	HDisp	Returns an object reference to the HDisp object associated with the camera subarray dialogs 'Lower' display area
Read only property	pHDisSubUpper()	HDisp	Returns an object reference to the HDisp object associated with the camera subarray dialogs 'Upper' display area
Read only property	pHDisSubWidth()	HDisp	Returns an object reference to the HDisp object associated with the camera subarray dialogs 'Width' display area
Read only property	pHEd4CamNrExp osures()	HEditNu mber4Arr ay	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Number of Exposures' editbox
Read only property	pHEd4CamNrTrig ger()	HEditNu mber4Arr ay	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Number of Streak Trigger' editbox
Read only property	pHEd4CamThresh old()	HEditNu mber4Arr ay	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Threshold' editbox
Read only property	pHEdnSetupTarget Temperature()	HEditNu mber	
Read only property	pHEn4CamExposu re()	HEntry4A rray	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Exposure Time' entrybox
Read only property	pHEn4CamGain()	HEntry4A rray	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Analog Gain' entrybox
Read only property	pHEn4CamOffset( )	HEntry4A rray	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Analog Offset' entrybox
Read only property	pHEn4CamSPX()	HEntry4A rray	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Superpixel' entrybox
Read only property	pHEntSubLower()	HEntry	Returns an object reference to the HEntry object associated with the camera subarray dialogs 'Lower' entrybox
Read only property	pHEntSubUpper()	HEntry	Returns an object reference to the HEntry object associated with the camera subarray dialogs 'Upper' entrybox
Read only property	pHFr4CamAction( )	HFrame4 Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Action' frame
Read only property	pHFr4CamExposu	HFrame4	Returns an object reference to the HFrame4Array object

	reTime()	Array	associated with the camera dialogs 'Exposure Time' frame
Read only property	pHFr4CamGainOffset()	HFrame4Array	an object reference to the HFrame4Array object associated with the camera dialogs 'Analog Gain' frame
Read only property	pHFr4CamIntegration()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Integration' frame
Read only property	pHFr4CamPhotonCounting()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Photon Counting' frame
Read only property	pHFr4CamRTBS()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Real Time Backsub' frame
Read only property	pHFr4CamScanMode()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Scan Mode' frame
Read only property	pHFr4CamStreakTrigger()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Streak Trigger' frame
Read only property	pHFraSetupCameraInfo()	HFrame	
Read only property	pHFraSetupTemperature()	HFrame	
Read only property	pHPr4CamPercent()	HProgress4Array	Returns an object reference to the HProgress4Array object associated with the camera dialogs 'Percent' progress bar
Read only property	pHRa4CamPcMode()	HRadios4Array	Returns an object reference to the HRadios4Array object associated with the camera dialogs 'Photon Counting Mode' radiobutton group
Read only property	pHRa4CamSMD()	HRadios4Array	Returns an object reference to the HRadios4Array object associated with the camera dialogs 'Scan Mode' radiobutton group
Read only property	pHRadSetupLightMode()	HRadios	
Read only property	pHRadSetupShutterAction()	HRadios	
Read only property	pHRadSetupTimingMode()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Timing Mode' radiobutton group
Read only property	pHRadSetupTriggerMode()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Trigger Mode' radiobutton group
Read only property	pHRadSetupTriggerPolarity()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Trigger Polarity' radiobutton group
Read only property	pHRadSetupTriggerSource()	HRadios	
Read only property	pHRadSubHorSubarray()	HRadios	Returns an object reference to the HRadios object associated with the camera subarray dialogs 'Horizontal Subarray' radiobutton group
Read only property	pHTabCamAcqMode()	HTab	Returns an object reference to the HTab object associated with the camera dialogs 'Acquisition mode' tab group
Read only property	pHWinCamDlg()	HWindow	Returns an object reference to the HWindow object associated with the camera dialogs main window
Read only property	pHWinSetupDlg()	HWindow	Returns an object reference to the HWindow object associated with the camera setup dialogs main window
Read only property	pHWinSubDlg()	HWindow	Returns an object reference to the HWindow object associated with the camera subarry dialogs main window
Read only property	piDatTypeForAcquire()	Integer	Returns the data type in ACQUIRE mode
Read only property	piDatTypeForAI()	Integer	Returns the data type in Analog Integration mode
Read only property	piDatTypeForLive	Integer	Returns the data type in LIVE mode

	()		
Read only property	piDatTypeForPC()	Integer	Returns the data type in Photon Counting mode
Read only property	psCameraName()	String	Returns the camera name
Read/write property	pfRestoreWindowPos()	Integer	Sets or returns a value which defines whether the window position, its size and window state is restored when the window is displayed again after it has been closed
Read/write property	pfStreakOperate()	Integer	A value that tells the camera object whether vertical streak operation is switched on. Don't modify this entry otherwise streak trigger handshake may not longer work
Read/write property	pfStreakOperateDTBE()	Integer	A value that tells the camera object whether horizontal streak operation is switched on. Don't modify this entry otherwise streak trigger handshake may not longer work
Read/write property	pfStreakUseDTBE()	Integer	Tells the camera object that a Dual time base extender is currently used for trigger handshake
Read/write property	pfUserIF()	Integer	Sets or returns a value which defines whether the associated window should be shown on screen. A window is only displayed under the following condition: pfUserIF=TRUE, pfVisible=TRUE, pfHideForm=FALSE
Read/write property	piStreakTriggerMethod()	Integer	Tells the camera (or acquisition/sequence) object the streak trigger method. Don't modify this entry otherwise Steak trigger handshake may no longer work
Function	pGetStreakRelatedCCDCaps(ByRef fCCDCanExtTrig As Integer, ByRef fCCDCanStartStop As Integer, ByRef sExtTrigCCDMethod As String, ByRef sExtTrigCPUMethod As String, ByRef sStartStopCCDMethod As String, ByRef sStartStopCPUMethod As String)		Returns streak trigger related capabilities of the CCD camera
Function	psGetStatus()	String	Returns the current camera status string
Sub	pStopAcquisition()		Stops a currently running acquisition

HC7300

Read only property	pHCh4CamClearFrameBuffer()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Clear frame buffer on start' checkbox
Read only property	pHCh4CamDoRTBacksub()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Do Real Time Backsub' checkbox
Read only property	pHCh4CamStreakTrigger()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Streak Trigger Active' checkbox
Read only property	pHCo4CamAcquire()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Acquire' pushbuttonpushbutton
Read only property	pHCo4CamDoPCSetup()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Do Photon Counting Setup' pushbutton
Read only property	pHCo4CamFreeze()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Freeze' pushbutton

Read only property	pHCo4CamGetBGData()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Get Background Data' pushbutton
Read only property	pHCo4CamLive()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Live' pushbutton
Read only property	pHCo4CamSingleExposure()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Single Exposure' pushbutton
Read only property	pHCo4CamStop()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Stop' pushbutton
Read only property	pHComCamDown()	HCommand	Returns an object reference to the HCommand object associated with the camera dialogs 'Down' pushbutton (Small down arrow)
Read only property	pHComCamUp()	HCommand	Returns an object reference to the HCommand object associated with the camera dialogs 'Up' pushbutton
Read only property	pHComSetupCancel()	HCommand	Returns an object reference to the HCommand object associated with the camera setup dialogs 'Cancel' pushbutton
Read only property	pHComSetupOK()	HCommand	Returns an object reference to the HCommand object associated with the camera setup dialogs 'OK' pushbutton
Read only property	pHDi4CamAboveThreshold()	HDisp4Array	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Above Threshold' display area
Read only property	pHDi4CamMessage()	HDisp4Array	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Message' display area
Read only property	pHDi4CamStreakTrigger()	HDisp4Array	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Streak Trigger' display area
Read only property	pHDisSetupCameraInfo()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Camera Info' display area
Read only property	pHEd4CamNrExposures()	HEditNumber4Array	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Number of Exposures' editbox
Read only property	pHEd4CamNrTriger()	HEditNumber4Array	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Number of Streak Trigger' editbox
Read only property	pHEd4CamThreshold()	HEditNumber4Array	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Threshold' editbox
Read only property	pHEn4CamExposure()	HEntry4Array	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Exposure Time' entrybox
Read only property	pHEn4CamGain()	HEntry4Array	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Analog Gain' entrybox
Read only property	pHEn4CamOffset()	HEntry4Array	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Analog Offset' entrybox
Read only property	pHEntSetupOffset()	HEntry	Returns an object reference to the HEntry object associated with the camera setup dialogs 'Offset' entrybox
Read only property	pHEntSetupWidth()	HEntry	Returns an object reference to the HEntry object associated with the camera setup dialogs 'Width' entrybox
Read only property	pHFr4CamAction()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Action' frame
Read only property	pHFr4CamExposu	HFrame4	Returns an object reference to the HFrame4Array object

	reTime()	Array	associated with the camera dialogs 'Exposure Time' frame
Read only property	pHFr4CamGainOffset()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Analog Gain' frame
Read only property	pHFr4CamIntegration()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Integration' frame
Read only property	pHFr4CamPhotonCounting()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Photon Counting' frame
Read only property	pHFr4CamRTBS()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Real Time Backsub' frame
Read only property	pHFr4CamStreakTrigger()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Streak Trigger' frame
Read only property	pHFraSetupScanMode()	HFrame	Returns an object reference to the HFrame object associated with the camera setup dialogs 'Scan Mode' frame
Read only property	pHPr4CamPercent()	HProgress4Array	Returns an object reference to the HProgress4Array object associated with the camera dialogs 'Percent' progress bar
Read only property	pHRa4CamPcMode()	HRadios4Array	Returns an object reference to the HRadios4Array object associated with the camera dialogs 'Photon Counting Mode' radiobutton group
Read only property	pHRadSetupScanMode()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Scan Mode' radiobutton group
Read only property	pHRadSetupTimingMode()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Timing Mode' radiobutton group
Read only property	pHRadSetupTriggerMode()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Trigger Mode' radiobutton group
Read only property	pHRadSetupTriggerPolarity()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Trigger Polarity' radiobutton group
Read only property	pHRadSetupTriggerSource()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Trigger Source' radiobutton group
Read only property	pHRadSetupXDirection()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'X Direction' radiobutton group
Read only property	pHTabCamAcqMode()	HTab	Returns an object reference to the HTab object associated with the camera dialogs 'Acquisition mode' tab group
Read only property	pHWinCamDlg()	HWindow	Returns an object reference to the HWindow object associated with the camera dialogs main window
Read only property	pHWinSetupDlg()	HWindow	Returns an object reference to the HWindow object associated with the camera setup dialogs main window
Read only property	piDatTypeForAcquire()	Integer	Returns the data type in ACQUIRE mode
Read only property	piDatTypeForAI()	Integer	Returns the data type in Analog Integration mode
Read only property	piDatTypeForLive()	Integer	Returns the data type in LIVE mode
Read only property	piDatTypeForPC()	Integer	Returns the data type in Photon Counting mode
Read only property	psCameraName()	String	Returns the camera name
Read/write property	pfRestoreWindowPos()	Integer	Sets or returns a value which defines whether the window position, its size and window state is restored when the window is displayed again after it has been closed
Read/write property	pfUserIF()	Integer	Sets or returns a value which defines whether the associated window should be shown on screen. A

			window is only displayed under the following condition: pfUserIF=TRUE, pfVisible=TRUE, pfHideForm=FALSE
Function	psGetStatus()	String	Returns the current camera status string
Sub	pStopAcquisition()		Stops a currently running acquisition

HC800010

Read only property	pHCh4CamClearFrameBuffer()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Clear frame buffer on start' checkbox
Read only property	pHCh4CamDoRTBacksub()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Do Real Time Backsub' checkbox
Read only property	pHCo4CamAcquire()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Acquire' pushbuttonpushbutton
Read only property	pHCo4CamDoPCSetup()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Do Photon Counting Setup' pushbutton
Read only property	pHCo4CamFreeze()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Freeze' pushbutton
Read only property	pHCo4CamGetBGData()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Get Background Data' pushbutton
Read only property	pHCo4CamLive()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Live' pushbutton
Read only property	pHCo4CamSingleExposure()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Single Exposure' pushbutton
Read only property	pHCo4CamStop()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Stop' pushbutton
Read only property	pHComCamDown()	HCommand	Returns an object reference to the HCommand object associated with the camera dialogs 'Down' pushbutton (Small down arrow)
Read only property	pHComCamUp()	HCommand	Returns an object reference to the HCommand object associated with the camera dialogs 'Up' pushbutton (Small up arrow)
Read only property	pHComSetupCancel()	HCommand	Returns an object reference to the HCommand object associated with the camera setup dialogs 'Cancel' pushbutton
Read only property	pHComSetupOK()	HCommand	Returns an object reference to the HCommand object associated with the camera setup dialogs 'OK' pushbutton
Read only property	pHDi4CamAboveThreshold()	HDisp4Array	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Above Threshold' display area
Read only property	pHDi4CamMessage()	HDisp4Array	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Message' display area
Read only property	pHDisSetupCameraInfo()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Camera Info' display area
Read only property	pHDisSetupScanArea()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Scan Area' display area
Read only property	pHEd4CamNrExposures()	HEditNumber4Array	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Number of Exposures' editbox
Read only property	pHEd4CamThresh	HEditNu	Returns an object reference to the HEditNumber4Array

	old()	mber4Arr ay	object associated with the camera dialogs 'Threshold' editbox
Read only property	pHEdnSetupHOffs ()	HEditNu mber	Returns an object reference to the HEditNumber object associated with the setup dialogs 'Horizontal Offset' (subarray mode) editbox
Read only property	pHEdnSetupHWid th()	HEditNu mber	Returns an object reference to the HEditNumber object associated with the setup dialogs 'Horizontal Width' (subarray mode) editbox
Read only property	pHEdnSetupVOffs ()	HEditNu mber	Returns an object reference to the HEditNumber object associated with the setup dialogs 'Vertical Offset' (subarray mode) editbox
Read only property	pHEdnSetupVWid th()	HEditNu mber	Returns an object reference to the HEditNumber object associated with the setup dialogs 'Vertical Width' (subarray mode) editbox
Read only property	pHEn4CamExposu re()	HEntry4A rray	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Exposure Time' entrybox
Read only property	pHEn4CamGain()	HEntry4A rray	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Analog Gain' entrybox
Read only property	pHEn4CamOffset( )	HEntry4A rray	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Analog Offset' entrybox
Read only property	pHFr4CamAction( )	HFrame4 Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Action' frame
Read only property	pHFr4CamExposu reTime()	HFrame4 Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Exposure Time' frame
Read only property	pHFr4CamGainOf fset()	HFrame4 Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Analog Gain' frame
Read only property	pHFr4CamIntegrat ion()	HFrame4 Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Integration' frame
Read only property	pHFr4CamPhoton Counting()	HFrame4 Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Photon Counting' frame
Read only property	pHFr4CamRTBS()	HFrame4 Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Real Time Backsub' frame
Read only property	pHFraSetupSubarr ay()	HFrame	Returns an object reference to the HFrame object associated with the setup dialogs 'Subarray' frame
Read only property	pHPr4CamPercent ()	HProgress 4Array	Returns an object reference to the HProgress4Array object associated with the camera dialogs 'Percent' progress bar
Read only property	pHRa4CamPcMod e()	HRadios4 Array	Returns an object reference to the HRadios4Array object associated with the camera dialogs 'Photon Counting Mode' radiobutton group
Read only property	pHRadSetupBinni ng()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Binning' radiobutton group
Read only property	pHRadSetupScan Mode()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Scan Mode' radiobutton group
Read only property	pHRadSetupScanS peed()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'CCD Area' radiobutton group
Read only property	pHRadSetupTimin gMode()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Timing Mode' radiobutton group
Read only property	pHRadSetupTrigge rMode()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Trigger Mode' radiobutton group

Read only property	pHRadSetupTriggerPolarity()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Trigger Polarity' radiobutton group
Read only property	pHTabCamAcqMode()	HTab	Returns an object reference to the HTab object associated with the camera dialogs 'Acquisition mode' tab group
Read only property	pHWinCamDlg()	HWindow	Returns an object reference to the HWindow object associated with the camera dialogs main window
Read only property	pHWinSetupDlg()	HWindow	Returns an object reference to the HWindow object associated with the camera setup dialogs main window
Read only property	piDatTypeForAcquire()	Integer	Returns the data type in ACQUIRE mode
Read only property	piDatTypeForAI()	Integer	Returns the data type in Analog Integration mode
Read only property	piDatTypeForLive()	Integer	Returns the data type in LIVE mode
Read only property	piDatTypeForPC()	Integer	Returns the data type in Phton Counting mode
Read only property	psCameraName()	String	Returns the camera name
Read/write property	pfRestoreWindowPos()	Integer	Sets or returns a value which defines whether the window position, its size and window state is restored when the window is displayed again after it has been closed
Read/write property	pfUserIF()	Integer	Sets or returns a value which defines whether the associated window should be shown on screen. A window is only displayed under the following condition: pfUserIF=TRUE, pfVisible=TRUE, pfHideForm=FALSE
Function	psGetStatus()	String	Returns the current camera status string
Sub	pStopAcquisition()		Stops a currently running acquisition

HC800020

Read only property	pHCh4CamBGSub()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Background subtraction (camera feature)' checkbox
Read only property	pHCh4CamClearFrameBuffer()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Clear frame buffer on start' checkbox
Read only property	pHCh4CamDoRTBacksub()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Real Time Background Subtraction' checkbox
Read only property	pHCh4CamHighVoltage()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'High Voltage' checkbox
Read only property	pHCh4CamRecurFilter()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Recursive Filter' checkbox
Read only property	pHCo4CamAcquire()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Acquire' pushbuttonpushbutton
Read only property	pHCo4CamDoPCSetup()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Do Photon Counting Setup' pushbutton
Read only property	pHCo4CamFreeze()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Freeze' pushbutton
Read only property	pHCo4CamGetBGData()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Get Background Data' pushbutton
Read only property	pHCo4CamLive()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Live' pushbutton
Read only property	pHCo4CamSingle	HComma	Returns an object reference to the HCommand4Array

	Exposure()	nd4Array	object associated with the camera dialogs 'Single Exposure' pushbutton
Read only property	pHCo4CamStop()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Stop' pushbutton
Read only property	pHCo4CamStoreBGSub()	HCommand4Array	
Read only property	pHComCamDown()	HCommand	Returns an object reference to the HCommand object associated with the camera dialogs 'Down' pushbutton (Small down arrow)
Read only property	pHComCamUp()	HCommand	Returns an object reference to the HCommand object associated with the camera dialogs 'Up' pushbutton (Small up arrow)
Read only property	pHComSetupOK()	HCommand	Returns an object reference to the HCommand object associated with the camera setup dialogs 'OK' pushbutton
Read only property	pHDi4CamAboveThreshold()	HDisp4Array	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Above Threshold' display area
Read only property	pHDi4CamMessage()	HDisp4Array	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Message' display area
Read only property	pHDisSetupCameraInfo()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Camera Info' display area
Read only property	pHEd4CamBGOffset()	HEditNumber4Array	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Background Offset (camera feature)' checkbox
Read only property	pHEd4CamNrExposures()	HEditNumber4Array	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Number of Exposures' editbox
Read only property	pHEd4CamThreshold()	HEditNumber4Array	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Threshold' editbox
Read only property	pHEn4CamGain()	HEntry4Array	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Analog Gain' entrybox
Read only property	pHEn4CamHVoltage()	HEntry4Array	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'High Voltage' entrybox
Read only property	pHEn4CamOffset()	HEntry4Array	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Analog Offset' entrybox
Read only property	pHEn4CamRecurNumber()	HEntry4Array	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Recursive Number' entrybox
Read only property	pHFr4CamAction()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Action' frame
Read only property	pHFr4CamControllerFeatures()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Exposure Time' frame
Read only property	pHFr4CamIntegration()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Integration' frame
Read only property	pHFr4CamPhotonCounting()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Photon Counting' frame
Read only property	pHFr4CamRTBS()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Real Time Backsub' frame
Read only property	pHPr4CamPercent()	HProgress4Array	Returns an object reference to the HProgress4Array object associated with the camera dialogs 'Percent' progress bar

Read only property	pHRa4CamPcMode()	HRadios4Array	Returns an object reference to the HRadios4Array object associated with the camera dialogs 'Photon Counting Mode' radiobutton group
Read only property	pHTabCamAcqMode()	HTab	Returns an object reference to the HTab object associated with the camera dialogs 'Acquisition mode' tab group
Read only property	pHWinCamDlg()	HWindow	Returns an object reference to the HWindow object associated with the camera dialogs main window
Read only property	pHWinSetupDlg()	HWindow	Returns an object reference to the HWindow object associated with the camera setup dialogs main window
Read only property	piDatTypeForAcquire()	Integer	Returns the data type in ACQUIRE mode
Read only property	piDatTypeForAI()	Integer	Returns the data type in Analog Integration mode
Read only property	piDatTypeForLive()	Integer	Returns the data type in LIVE mode
Read only property	piDatTypeForPC()	Integer	Returns the data type in Photon Counting mode
Read only property	psCameraName()	String	Returns the camera name
Read/write property	pfRestoreWindowPos()	Integer	Sets or returns a value which defines whether the window position, its size and window state is restored when the window is displayed again after it has been closed
Read/write property	pfUserIF()	Integer	Sets or returns a value which defines whether the associated window should be shown on screen. A window is only displayed under the following condition: pfUserIF=TRUE, pfVisible=TRUE, pfHideForm=FALSE
Function	psGetStatus()	String	Returns the current camera status string
Sub	pStopAcquisition()		Stops a currently running acquisition
HFlatPanel			
Read only property	pHCh4CamClearFrameBuffer()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Clear frame buffer on start' checkbox
Read only property	pHCh4CamDoRTBacksub()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the camera dialogs 'Do Real Time Backsub' checkbox
Read only property	pHCo4CamAcquire()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Acquire' pushbuttonpushbutton
Read only property	pHCo4CamDoPCSetup()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Do Photon Counting Setup' pushbutton
Read only property	pHCo4CamFreeze()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Freeze' pushbutton
Read only property	pHCo4CamGetBGData()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Get Background Data' pushbutton
Read only property	pHCo4CamLive()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Live' pushbutton
Read only property	pHCo4CamSingleExposure()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Single Exposure' pushbutton
Read only property	pHCo4CamStop()	HCommand4Array	Returns an object reference to the HCommand4Array object associated with the camera dialogs 'Stop' pushbutton
Read only property	pHComCamDown()	HCommand	Returns an object reference to the HCommand object associated with the camera dialogs 'Down' pushbutton (Small down arrow)
Read only property	pHComCamUp()	HComma	Returns an object reference to the HCommand object

		nd	associated with the camera dialogs 'Up' pushbutton (Small up arrow)
Read only property	pHComSetupCancel()	HCommand	Returns an object reference to the HCommand object associated with the camera setup dialogs 'Cancel' pushbutton
Read only property	pHComSetupOK()	HCommand	Returns an object reference to the HCommand object associated with the camera setup dialogs 'OK' pushbutton
Read only property	pHDi4CamAboveThreshold()	HDisp4Array	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Above Threshold' display area
Read only property	pHDi4CamMessage()	HDisp4Array	Returns an object reference to the HDisp4Array object associated with the camera dialogs 'Message' display area
Read only property	pHDisSetupCameraInfo()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Camera Info' display area
Read only property	pHDisSetupScanArea()	HDisp	Returns an object reference to the HDisp object associated with the camera setup dialogs 'Scan Area' display area
Read only property	pHEd4CamNrExposures()	HEditNumber4Array	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Number of Exposures' editbox
Read only property	pHEd4CamNrTrigger()	HEditNumber4Array	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Number of Streak Trigger' editbox
Read only property	pHEd4CamThreshold()	HEditNumber4Array	Returns an object reference to the HEditNumber4Array object associated with the camera dialogs 'Threshold' editbox
Read only property	pHEn4CamExposure()	HEntry4Array	Returns an object reference to the HEntry4Array object associated with the camera dialogs 'Exposure Time' entrybox
Read only property	pHFr4CamAction()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Action' frame
Read only property	pHFr4CamExposureTime()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Exposure Time' frame
Read only property	pHFr4CamIntegration()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Integration' frame
Read only property	pHFr4CamPhotonCounting()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Photon Counting' frame
Read only property	pHFr4CamRTBS()	HFrame4Array	Returns an object reference to the HFrame4Array object associated with the camera dialogs 'Real Time Backsub' frame
Read only property	pHPr4CamPercent()	HProgress4Array	Returns an object reference to the HProgress4Array object associated with the camera dialogs 'Percent' progress bar
Read only property	pHRa4CamPcMode()	HRadios4Array	Returns an object reference to the HRadios4Array object associated with the camera dialogs 'Photon Counting Mode' radiobutton group
Read only property	pHRadSetupBinning()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Binning' radiobutton group
Read only property	pHRadSetupSubtype()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Subtype' radiobutton group
Read only property	pHRadSetupTimingMode()	HRadios	Returns an object reference to the HRadios object associated with the camera setup dialogs 'Timing Mode' radiobutton group
Read only property	pHTabCamAcqMode()	HTab	Returns an object reference to the HTab object associated with the camera dialogs 'Acquisition mode'

			tab group
Read only property	pHWinCamDlg()	HWindow	Returns an object reference to the HWindow object associated with the camera dialogs main window
Read only property	pHWinSetupDlg()	HWindow	Returns an object reference to the HWindow object associated with the camera setup dialogs main window
Read only property	piDatTypeForAcquire()	Integer	Returns the data type in ACQUIRE mode
Read only property	piDatTypeForAI()	Integer	Returns the data type in Analog Integration mode
Read only property	piDatTypeForLive()	Integer	Returns the data type in LIVE mode
Read only property	piDatTypeForPC()	Integer	Returns the data type in Photon Counting mode
Read only property	psCameraName()	String	Returns the camera name
Read/write property	pfRestoreWindowPos()	Integer	Sets or returns a value which defines whether the window position, its size and window state is restored when the window is displayed again after it has been closed
Read/write property	pfUserIF()	Integer	Sets or returns a value which defines whether the associated window should be shown on screen. A window is only displayed under the following condition: pfUserIF=TRUE, pfVisible=TRUE, pfHideForm=FALSE
Function	psGetStatus()	String	Returns the current camera status string
Sub	pStopAcquisition()		Stops a currently running acquisition
HDCam			
Read only property	pHCh4CamClearFrameBuffer()	HCheck4Array	
Read only property	pHCh4CamDoRTBacksub()	HCheck4Array	
Read only property	pHCh4CamStreakTrigger()	HCheck4Array	
Read only property	pHCo4CamAcquire()	HCommand4Array	
Read only property	pHCo4CamDoPCSetup()	HCommand4Array	
Read only property	pHCo4CamFreeze()	HCommand4Array	
Read only property	pHCo4CamGetBGData()	HCommand4Array	
Read only property	pHCo4CamLive()	HCommand4Array	
Read only property	pHCo4CamSingleExposure()	HCommand4Array	
Read only property	pHCo4CamStop()	HCommand4Array	
Read only property	pHComCamDown()	HCommand	
Read only property	pHComCamUp()	HCommand	
Read only property	pHComSetupCancel()	HCommand	
Read only property	pHComSetupOK()	HCommand	
Read only property	pHDi4CamAboveThreshold()	HDisp4Array	
Read only property	pHDi4CamMessage()	HDisp4Array	
Read only property	pHDi4CamStreakTrigger()	HDisp4Array	
Read only property	pHDisSetupCameraInfo()	HDisp	

Read only property	pHDisSetupScanArea()	HDisp	
Read only property	pHEd4CamNrExposures()	HEditNumber4Array	
Read only property	pHEd4CamThreshold()	HEditNumber4Array	
Read only property	pHEdnSetupHOffs()	HEditNumber	
Read only property	pHEdnSetupHWidth()	HEditNumber	
Read only property	pHEdnSetupTemperature()	HEditNumber	
Read only property	pHEdnSetupVOffs()	HEditNumber	
Read only property	pHEdnSetupVWidth()	HEditNumber	
Read only property	pHEn4CamExposure()	HEntry4Array	
Read only property	pHEn4CamGain()	HEntry4Array	
Read only property	pHEn4CamOffset()	HEntry4Array	
Read only property	pHEntSetupIIGain()	HEntry	
Read only property	pHEntSetupLightMode()	HEntry	
Read only property	pHEntSetupScanSpeed()	HEntry	
Read only property	pHEntSetupTemperatureSwitch()	HEntry	
Read only property	pHFr4CamAction()	HFrame4Array	
Read only property	pHFr4CamExposureTime()	HFrame4Array	
Read only property	pHFr4CamGainOffset()	HFrame4Array	
Read only property	pHFr4CamIntegration()	HFrame4Array	
Read only property	pHFr4CamPcMode()	HFrame4Array	
Read only property	pHFr4CamPhotonCounting()	HFrame4Array	
Read only property	pHFr4CamRTBS()	HFrame4Array	
Read only property	pHFr4CamStreakTrigger()	HFrame4Array	
Read only property	pHFraSetupSubarray()	HFrame	
Read only property	pHPr4CamPercent()	HProgress4Array	
Read only property	pHRa4CamPcMode()	HRadios4Array	
Read only property	pHRadSetupBinning()	HRadios	
Read only property	pHRadSetupScanMode()	HRadios	
Read only property	pHRadSetupTriggerMode()	HRadios	

Read only property	pHRadSetupTriggerPolarity()	HRadios	
Read only property	pHRadSetupTriggerSource()	HRadios	
Read only property	pHTabCamAcqMode()	HTab	
Read only property	pHWinCamDlg()	HWindow	
Read only property	pHWinSetupDlg()	HWindow	
Read only property	piDatTypeForAcquire()	Integer	
Read only property	piDatTypeForAI()	Integer	
Read only property	piDatTypeForLive()	Integer	
Read only property	piDatTypeForPC()	Integer	
Read only property	psCameraName()	String	
Read/write property	pfRestoreWindowPos()	Integer	
Read/write property	pfStreakOperate()	Integer	
Read/write property	pfStreakOperateDTBE()	Integer	
Read/write property	pfStreakUseDTBE()	Integer	
Read/write property	pfUserIF()	Integer	
Read/write property	piStreakTriggerMethod()	Integer	
Read/write property	plHandle()	Long	
Read/write property	plModel()	Long	
Function	pGetStreakRelatedCCDCaps(ByRef fCCDCanExtTrig As Integer, ByRef fCCDCanStartStop As Integer, ByRef sExtTrigCCDMethod As String, ByRef sExtTrigCPUMethod As String, ByRef sStartStopCCDMethod As String, ByRef sStartStopCPUMethod As String)		
Function	psGetStatus()	String	
Sub	pStopAcquisition()		

HExternalDevices

Event	DevicesBuilt()		Event which is raised when the streak devices objects have been setup up properly
Event	Message(ByVal sMessage As String)		Event which is raised to inform the user of ongoing operations
Event	TASetupExecuted()		Event which is raised when the TA setup has been executed
Event	TrigSetExecuted()		Event which is raised when the Trigger Setup has been executed
Read only property	pcItem(ByVal Index As Integer)	HExternalDevice	Returns a reference to the specified device
Read only property	pfDualTimeBaseExist()	Integer	Returns a flag which indicates whether a dual time base extender is used as a streak camera option

Read only property	pfStreakUseDTBE()	Integer	Tells the camera object that a Dual time base extender is currently used for trigger handshake
Read only property	pHCh4SetupOptionDelay()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the device control setup dialogs 'Use Delay Option' checkbox
Read only property	pHCh4SetupOptionDelay2()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the device control setup dialogs 'Use Delay2 Option' checkbox
Read only property	pHCh4SetupOptionSpec()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the device control setup dialogs 'Use Spectrograph Option' checkbox
Read only property	pHCh4SetupOptionTD()	HCheck4Array	Returns an object reference to the HCheck4Array object associated with the device control setup dialogs 'Use Streak Option' checkbox
Read only property	pHChkAutoDelParameter00()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 00' checkbox
Read only property	pHChkAutoDelParameter01()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 01' checkbox
Read only property	pHChkAutoDelParameter02()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 02' checkbox
Read only property	pHChkAutoDelParameter03()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 03' checkbox
Read only property	pHChkAutoDelParameter04()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 04' checkbox
Read only property	pHChkAutoDelParameter05()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 05' checkbox
Read only property	pHChkAutoDelParameter06()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 07' checkbox
Read only property	pHChkAutoDelParameter07()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 07' checkbox
Read only property	pHChkAutoDelParameter08()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 08' checkbox
Read only property	pHChkAutoDelParameter09()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 09' checkbox
Read only property	pHChkAutoDelParameter10()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 10' checkbox
Read only property	pHChkAutoDelParameter11()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 11' checkbox
Read only property	pHChkAutoDelParameter12()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 12' checkbox
Read only property	pHChkAutoDelParameter13()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 13' checkbox
Read only property	pHChkAutoDelParameter14()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 14' checkbox
Read only property	pHChkAutoDelParameter15()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto

			Delay parameter 15' checkbox
Read only property	pHChkAutoDelParameter16()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 16' checkbox
Read only property	pHChkAutoDelParameter17()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 17' checkbox
Read only property	pHChkAutoDelParameter18()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 18' checkbox
Read only property	pHChkAutoDelParameter19()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 19' checkbox
Read only property	pHChkAutoDelParameter20()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 20' checkbox
Read only property	pHChkAutoDelParameter21()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 21' checkbox
Read only property	pHChkAutoDelParameter22()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 22' checkbox
Read only property	pHChkAutoDelParameter23()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 23' checkbox
Read only property	pHChkAutoDelParameter24()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 24' checkbox
Read only property	pHChkAutoDelParameter25()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 25' checkbox
Read only property	pHChkAutoDelParameter26()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 26' checkbox
Read only property	pHChkAutoDelParameter27()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 27' checkbox
Read only property	pHChkAutoDelParameter28()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 28' checkbox
Read only property	pHChkAutoDelParameter29()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay parameter 29' checkbox
Read only property	pHChkOptionsAutoDelay2Delay()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay2 Delay' checkbox
Read only property	pHChkOptionsAutoDelayDelay()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Delay Delay' checkbox
Read only property	pHChkOptionsAutoMCP()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto MCP' checkbox
Read only property	pHChkOptionsAutoStreakDelay()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Streak Delay' checkbox
Read only property	pHChkOptionsAutoStreakShutter()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Auto Streak Shutter' checkbox
Read only property	pHChkSetupA6538Connected()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'A6538 Connected To Status Port' checkbox

Read only property	pHChkSetupCounterBoardInstalled()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Dt2819 Installed' checkbox
Read only property	pHChkSetupGPIBConnectedToDELAY()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'GP-IB connected To Delay' checkbox
Read only property	pHChkSetupGPIBConnectedToDELAY2()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'GP-IB connected To Delay2' checkbox
Read only property	pHChkSetupGPIBConnectedToSPEC()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'GP-IB connected To Spectrograph' checkbox
Read only property	pHChkSetupGPIBConnectedToTD()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'GP-IB connected To Streak Camera' checkbox
Read only property	pHChkSetupGPIBInstalled()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'GP-IB installed' checkbox
Read only property	pHChkSetupTDStatusCableConnected()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Status Cable Connected To Streak Camera' checkbox
Read only property	pHChkSetupUseDeviceDELAY()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Use Delaybox' checkbox
Read only property	pHChkSetupUseDeviceDELAY2()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Use Delaybox2' checkbox
Read only property	pHChkSetupUseDeviceSPEC()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Use Spectrograph' checkbox
Read only property	pHChkSetupUseDeviceTD()	HCheck	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Use Streakcamera' checkbox
Read only property	pHChkTrigSetConnectMonitorOut()	HCheck	Returns an object reference to the HCheck object associated with the trigger setup dialogs 'Connect Monitor Out' checkbox
Read only property	pHChkTrigSetConnectResetIn()	HCheck	Returns an object reference to the HCheck object associated with the trigger setup dialogs 'Conect Reset In' checkbox
Read only property	pHChkTrigSetUseDTBE()	HCheck	Returns an object reference to the HCheck object associated with the Trigger setup dialogs 'Use dual time base extender' checkbox
Read only property	pHComAutoDelCancel()	HCommand	Returns an object reference to the HCommand object associated with the auto delay setup dialogs 'Cancel' command button
Read only property	pHComAutoDelOK()	HCommand	Returns an object reference to the HCommand object associated with the auto delay setup dialogs 'OK' command button
Read only property	pHComAutoDelResetStoredValues()	HCommand	Returns an object reference to the HCommand object associated with the auto delay setup dialogs 'Reset Stored Values' command button
Read only property	pHComOptionsOK()	HCommand	Returns an object reference to the HCommand object associated with the device control setup dialogs 'Cancel' command button
Read only property	pHComOptionsSetupAutoDelay()	HCommand	Returns an object reference to the HCommand object associated with the device control setup dialogs 'Setup Auto Delay' command button
Read only property	pHComOptionsSetupAutoDelay2()	HCommand	Returns an object reference to the HCommand object associated with the device control setup dialogs 'Setup Auto Delay2' command button
Read only property	pHComProcAbort(	HComma	Returns an object reference to the HCommand object

	)	nd	associated with the device control processing dialogs 'abort' command button
Read only property	pHComSetupCancel()	HCommand	Returns an object reference to the HCheck object associated with the device control setup dialogs 'Cancel' checkbox
Read only property	pHComSetupSetup()	HCommand	Returns an object reference to the HCommand object associated with the device control setup dialogs 'Setup' command button
Read only property	pHComTrigSetCancel()	HCommand	Returns an object reference to the HCommand object associated with the trigger setup dialogs 'Cancel' command button
Read only property	pHComTrigSetOK()	HCommand	Returns an object reference to the HCommand object associated with the trigger setup dialogs 'OK' command button
Read only property	pHComTrigSetShowTiming()	HCommand	Returns an object reference to the HCommand object associated with the trigger setup dialogs 'Show Timing' command button
Read only property	pHDisAutoDelIntroduction()	HDisp	Returns an object reference to the HDisp object associated with the Auto delay dialogs 'Introduction' display area
Read only property	pHDisProcMessage()	HDisp	Returns an object reference to the HDisp object associated with the Processing dialogs 'Message' display area
Read only property	pHDisTrigSetCCDMode()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'CCD mode' display area
Read only property	pHDisTrigSetComputerExposureControl()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Computer exposure control' display area
Read only property	pHDisTrigSetComputerGPIB()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Computer GPIB' display area
Read only property	pHDisTrigSetComputerMonitorIn()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Computer Monitor In' display area
Read only property	pHDisTrigSetComputerResetOut()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Reset Out' display area
Read only property	pHDisTrigSetConfiguration()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Configuation' display area
Read only property	pHDisTrigSetDetectMonitorByGPIB()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Detect Monitor By GPIB' display area
Read only property	pHDisTrigSetDTBE()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Dual time base extender' display area
Read only property	pHDisTrigSetHMonitorOut()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'H-Monitor Out' display area
Read only property	pHDisTrigSetHResetIn()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'H-Reset In' display area
Read only property	pHDisTrigSetHTrigIn()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'H-Trigger In' display area
Read only property	pHDisTrigSetMainUnitType()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Main Unit Type' display area
Read only property	pHDisTrigSetPluginMode()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Plugin Mode'

			display area
Read only property	pHDisTrigSetPluginType()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Plugin Type' display area
Read only property	pHDisTrigSetResetByGPIB()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Reset by GPIB' display area
Read only property	pHDisTrigSetStreakCameraGPIB()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Streak camera GPIB' display area
Read only property	pHDisTrigSetTrigIn()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'Trigger In' display area
Read only property	pHDisTrigSetVMonitorOut()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'V-Monitor Out' display area
Read only property	pHDisTrigSetVResetIn()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'V-Reset In' display area
Read only property	pHDisTrigSetVTrigIn()	HDisp	Returns an object reference to the HDisp object associated with the Trigger Setup dialogs 'V-Trigger In' display area
Read only property	pHEdnSetupCounterBoardIObase()	HEditNumber	Returns an object reference to the HEditNumber object associated with the device control setup dialogs 'DT 2819 IO Base' editbox
Read only property	pHEdnSetupGPIBBaseDELAY()	HEditNumber	Returns an object reference to the HEditNumber object associated with the device control setup dialogs 'Delaybox GP-IB address' editbox
Read only property	pHEdnSetupGPIBBaseDELAY2()	HEditNumber	Returns an object reference to the HEditNumber object associated with the device control setup dialogs 'Delaybox2 GP-IB address' editbox
Read only property	pHEdnSetupGPIBBaseSPEC()	HEditNumber	Returns an object reference to the HEditNumber object associated with the device control setup dialogs 'Delaybox Spectrograph address' editbox
Read only property	pHEdnSetupGPIBBaseTD()	HEditNumber	Returns an object reference to the HEditNumber object associated with the device control setup dialogs 'Streakcamera GP-IB address' editbox
Read only property	pHEdnSetupGPIBIOBase()	HEditNumber	Returns an object reference to the HEditNumber object associated with the device control setup dialogs 'GP-IB board IO address' editbox
Read only property	pHEdnTrigSetExposure()	HEditNumber	
Read only property	pHEdnTrigSetPostTriggerTime()	HEditNumber	Returns an object reference to the HEditNumber object associated with the Trigger setup dialogs 'Post Trigger Time' editbox
Read only property	pHEdnTrigSetResetDelay()	HEditNumber	Returns an object reference to the HEditNumber object associated with the Trigger setup dialogs 'Reset Delay' editbox
Read only property	pHEntSetupDeviceDELAY()	HEntry	Returns an object reference to the HEntry object associated with the device control setup dialogs 'Delaybox device' entrybox
Read only property	pHEntSetupDeviceDELAY2()	HEntry	Returns an object reference to the HEntry object associated with the device control setup dialogs 'Delaybox2 device' entrybox
Read only property	pHEntSetupDeviceSPEC()	HEntry	Returns an object reference to the HEntry object associated with the device control setup dialogs 'Spectrograph device' entrybox
Read only property	pHEntSetupDeviceTD()	HEntry	Returns an object reference to the HEntry object associated with the device control setup dialogs 'Streakcamera device' entrybox
Read only property	pHEntSetupPlugin	HEntry	Returns an object reference to the HEntry object

	DELAY()		associated with the device control setup dialogs 'Delaybox Plugin' entrybox
Read only property	pHEntSetupPluginDELAY2()	HEntry	Returns an object reference to the HEntry object associated with the device control setup dialogs 'Delaybox2 Plugin' entrybox
Read only property	pHEntSetupPluginSPEC()	HEntry	Returns an object reference to the HEntry object associated with the device control setup dialogs 'Spectrograph Plugin' entrybox
Read only property	pHEntSetupPluginTD()	HEntry	Returns an object reference to the HEntry object associated with the device control setup dialogs 'Streakcamera Plugin' entrybox
Read only property	pHFraSetupExternalDevices()	HFrame	Returns an object reference to the HFrame object associated with the device control setup dialogs 'External Devices' frame
Read only property	pHFraSetupIFSetup()	HFrame	Returns an object reference to the HFrame object associated with the device control setup dialogs 'IF Setup' frame
Read only property	pHProProcProgress()	HProgress	Returns an object reference to the HProgress object associated with the Processing dialogs 'Progress' bar
Read only property	pHRadSetupCounterBoardType()	HRadios	Returns an object reference to the HRadios object associated with the device control dialogs 'Counter board type' radiobutton group
Read only property	pHRadTrigSetTriggerMethod()	HRadios	Returns an object reference to the HRadios object associated with the device control dialogs 'Trigger method' radiobutton group
Read only property	pHWinAutoDelDlg()	HWindow	Returns an object reference to the HWindow object associated with the Auto delay setup dialogs main window
Read only property	pHWinOptionsDlg()	HWindow	Returns an object reference to the HWindow object associated with the options dialogs main window
Read only property	pHWinProcDlg()	HWindow	
Read only property	pHWinSetupDlg()	HWindow	Returns an object reference to the HWindow object associated with the device control setup dialogs main window
Read only property	pHWinTimingDiagram()	HWindow	Returns an object reference to the HWindow object associated with the Timing dialogs timing diagram window
Read only property	pHWinTimingDlg()	HWindow	Returns an object reference to the HWindow object associated with the Timing dialogs main window
Read only property	pHWinTrigSetDlg()	HWindow	Returns an object reference to the HWindow object associated with the Trigger setup dialogs main window
Read only property	piA6538Connected()	Integer	Returns a flag which defines whether an A6538 is connected to the trigger status adapter
Read only property	piCounterBoardInstalled()	Integer	Returns a flag which defines whether a Dt2819 board is installed
Read only property	piCounterBoardIOBase()	Integer	Returns the DT2819 board IO base adress
Read only property	piGPIBInstalled()	Integer	Returns a flag which defines whether a GP-IB board is installed in the system
Read only property	piGPIBIOBase()	Integer	Returns the GP-IB board IO base address
Read only property	piMaxIndex()	Integer	Returns the maximum index for pcItem
Read only property	piMinIndex()	Integer	Returns the minimum index for pcItem
Read only property	piPostTriggerTime()	Integer	Returns the Post Trigger Time
Read only property	piStreakTriggerMethod()	Integer	Returns the currently selected streak trigger method
Read only property	piTDStatusCableConnected()	Integer	Returns a flag which defines whether a status cable is connected to the streak camera
Read only property	plTriggerDelay()	Long	Returns the trigger delay
Read/write property	pfAutoMCP()	Integer	Returns a flag which defines whether Auto MCP is

			active
Read/write property	pfAutoStreakShutter()	Integer	Returns a flag which defines whether Auto Streak Shutter is active
Read/write property	pfDoStatusRegularly()	Integer	Sets or returns a flag which defines whether the status should inquired regularly (Typically once per second)
Read/write property	pfRestoreWindowPos()	Integer	Sets or returns a value which defines whether the window position, its size and window state is restored when the window is displayed again after it has been closed
Read/write property	pfUserIF()	Integer	Sets or returns a value which defines whether the associated window should be shown on screen. A window is only displayed under the following condition: pfUserIF=TRUE, pfVisible=TRUE, pfHideForm=FALSE
Function	pDoGetStatus()	Integer	Executes status read
Function	plIEEE488_AttachDevice(ByVal iBoardNo As Long, ByVal iIEEE488Address As Long, ByVal lpszOutEOS As String, ByVal lpszInEOS As String, ByVal fDeviceIsSlow As Long, ByVal uiTimeout As Long)	Long	Low Level IEEE 488 function: Attaches a device
Function	plIEEE488_Close(ByVal iBoardNo As Long)	Long	Low Level IEEE 488 function: Closes the GP-IB session
Function	plIEEE488_DetachDevice(ByVal lDeviceHandle As Long)	Long	Low Level IEEE 488 function: Detaches a device
Function	plIEEE488_IsSRQPending(ByVal iBoardNo As Long)	Long	Low Level IEEE 488 function: Inquires pending SRQs
Function	plIEEE488_Open(ByVal iIOAddress As Long, ByVal iMyIEEE488Address As Long, ByVal fController As Long)	Long	Low Level IEEE 488 function: Opens a GP-IB session
Function	plIEEE488_ParallelPoll(ByVal lDeviceHandle As Long, ByVal lpszPollStatus As String)	Long	Low Level IEEE 488 function: Executes a parallel poll
Function	plIEEE488_ReadBinary(ByVal lDeviceHandle As Long, bByte()	Byte, ByVal iCount As Long) As Long	Low Level IEEE 488 function: Reads GP-IB data binary
Function	plIEEE488_ReadString(ByVal lDeviceHandle As	Long	Low Level IEEE 488 function: Reads GP-IB data as string

	Long, lpszString As String, ByVal iCount As Long)		
Function	plIEEE488_SendBinary(ByVal lDeviceHandle As Long, lpData As Byte, ByVal iCount As Long, ByVal fEOI As Long)	Long	Low Level IEEE 488 function: Sends GB-IB data binary
Function	plIEEE488_SendString(ByVal lDeviceHandle As Long, ByVal lpszString As String)	Long	Low Level IEEE 488 function: Sends GB-IB data as string
Function	plIEEE488_SerialPoll(ByVal lDeviceHandle As Long, ByVal lpszPollStatus As String)	Long	Low Level IEEE 488 function: Executes serial poll
Function	plIEEE488_SetGotoLocal(ByVal lDeviceHandle As Long)	Long	Low Level IEEE 488 function: Sends Goto Local command
Function	plIEEE488_SetRemoteEnable(ByVal iBoardNo As Long)	Long	Low Level IEEE 488 function: Sends Set remote enable command
Function	plIEEE488_TA_IdentGPIBBoard()	Long	Returns the board handle in the case the HPDTA has already opened a GB-IB session
Function	psGetStatus()	String	Returns the current devices status string
Function	rfIsDualTimeBaseExtender(ByVal s As String)		Returns True if the specified plugin name is a dual time base extender
Function	rTA_IdentOptionName(ByVal iDeviceType As Integer, ByVal iOptionNo As Integer, ByRef sOption As String, ByVal iCount As Integer)	Integer	Returns the name of the specified option
Function	rTA_IdentOptionNo(ByVal iDeviceType As Integer, ByRef iOptionNo As Integer)	Integer	Returns the number of available options
Function	rTA_IdentParEntryName(ByVal iDeviceType As Integer, ByVal iParameter As Integer, ByVal iEntry As Integer, ByRef sEntry As String, ByVal	Integer	Returns the name of the specified parameter

	iCount As Integer)		
Function	rTA_IdentParEntryNo(ByVal iDeviceType As Integer, ByVal iParameter As Integer, ByRef iEntryNo As Integer)	Integer	Returns the number of available entries
Function	rTA_IdentPluginName(ByVal iDeviceType As Integer, ByVal iPluginNo As Integer, ByRef sPlugin As String, ByVal iCount As Integer)	Integer	Returns the name of the specified plugin
Function	rTA_IdentPluginNo(ByVal iDeviceType As Integer, ByRef iPluginNo As Integer)	Integer	Returns the number of plugins
Function	rTA_IdentProperty(ByVal iDeviceType As Integer, ByVal iPropertyID As Integer, varReturn As Variant)	Integer	Returns a property value of the specified device
Function	rTA_Setup()	Integer	Executes TA setup
Function	rTA_SetupDevice(ByVal iDeviceType As Integer, ByVal iDeviceNo As Integer)	Integer	Executes Setup Device
Function	rTA_SetupOption(ByVal iDeviceType As Integer, ByVal iOptionNo As Integer, ByVal fInstalled As Integer)	Integer	Executes Setup Option
Function	rTA_SetupPlugin(ByVal iDeviceType As Integer, ByVal iPluginNo As Integer)	Integer	Executes Setup Plugin
Function	rTA_TrigSet()	Integer	
Sub	pEmergencyOff()		executes emergency off
Sub	pSetCCDCaps(fCCDCanExtTrig As Integer, fCCDCanStartStop As Integer, sExtTrigCCDMethod As String,		Sets CCD capabilities related to streak trigger handshake

	sExtTrigCPUMethod As String, sStartStopCCDMethod As String, sStartStopCPUMethod As String)		
Sub	pShowDialogs()		Shows all external devices status/control dialogs
Sub	pTA_HandleSRQs()		Gives the device DLLs the opportunity to handle SRQs (if any)
Sub	rGetDeviceList(ByVal iDT As Integer, HEntry As HEntry)		Gets a list of the supported devices
Sub	rGetDualTimeBaseInfo(ByRef fDualTimeBaseExist As String, ByRef sDualTimeBase As String, ByRef sDualTimeBaseRange As String)		Gets informations about the dual time base extender (if any)
Sub	rTA_SetupCloseSession()		Executes device close session

HDevPar

Event	ChangeEnabled()		Event which is raised when the pfEnabled property changes
Event	ChangeLimits()		Event which is raised when either pvMinValue or pvMaxValue changes
Event	ChangeType()		Event which is raised when the piDevParType property changes
Event	ChangeValue(ByVal dbOldValue As Double, ByVal fFromDevice As Integer)		Event which is raised when the pfValue property changes
Event	ChangeVisible()		Event which is raised when the pfVisible property changes
Read only property	pcParent()	Object	Returns a reference to the parent object of this object. If this reference is Nothing this object is the topmost object within the object hierarchy (normally the HWindow object of the associated dialog)
Read only property	pdbStep()	Double	Returns a value which defines the step width of two neighbouring values. If it is zero all values between pvMinValue and pvMaxValue are possible
Read only property	pfControlAvail()	Integer	Returns a value which specifies whether the object can be controlled by the client programmer (This is dependent on the pfVisible and pfEnabled properties and those of its parent)
Read only property	pfControlAvailable()	Integer	Returns a value which defines whether the associated parameter on the external device can be controlled by the software
Read only property	pfEnabled()	Integer	Returns a value which specifies whether the object is enabled
Read only property	pfKnownParameter()	Integer	Returns a value which defines whether the parameter is a known one. The values for allknown parameters are described in the enumerations StreakParameter, SpectrographParameter and DelayParameter
Read only property	pfStatusAvailable()	Integer	Returns a value which defines whether status information of the associated parameter can be provided by the external device

Read only property	pfVisible()	Integer	Returns a value which specifies whether the object is visible
Read only property	piDevParType()	Integer	Returns a value which defines the ParameterType of the parameter. Possible values are described in the enumeration DevParType
Read only property	piMode()	Integer	Returns a value which defines the ParameterMode of the parameter. Possible values are described in the enumeration DevParMode
Read only property	piParameter()	Integer	Returns a value identifying the parameter. Possible values of known parameters are described in the enumerations StreakParameter, SpectrographParameter and DelayParameter
Read only property	psName()	String	Returns the name of the object
Read only property	psParameterName( )	String	Returns a value defining the parameter name
Read only property	psSetKeyValue(By Val iIndex As Integer)	String	Returns the key value of a specified entry index (Not necessarily the current one)
Read only property	pvMaxValue()	Variant	Maximum value limiting the range for pvValue
Read only property	pvMinValue()	Variant	Minimum value limiting the range for pvValue
Read/write property	piExponent()	Integer	Sets or returns the exponent used to display a value. If this parameter is zero no exponent is diplayed
Read/write property	psKeyValue()	String	Sets or returns the key value of the object never always raising the ChangeValue event.
Read/write property	pvValue()	Variant	Returns the value of the object
Sub	pRegEvent(ByVal sEvent As String, ByVal fBasicEvent As Integer)		Should not be used by clients! (Registers an event to the error handler)

HExternalDevice

Event	ReleaseParameters ()		Event which is raised when the streak parameters are released and built up newly. THe clien program must release all object references to individual parameters in this moment
Read only property	pfControlAvaileable()	Integer	Returns a flag which defines whether Control is available for the device
Read only property	pfStatusAvaileable ()	Integer	Returns a flag which defines whether status information is available of the device
Read only property	pfStatusOutExist()	Integer	Returns a flag which defines whether the status out connector exits for the device (streak cameras only)
Read only property	pfTriggerConnectorExist()	Integer	Returns a flag which defines whether trigger connectors exits for the device (streak cameras only)
Read only property	pHDevPars()	HDevPars	Returns an object reference to the HDevPars object of the device
Read only property	pHWinDevDlg()	HWindow	Returns an object reference to the HWindow object associated with the devices dialogs main window
Read only property	piDeviceIndex()	Integer	Returns the current devices index
Read only property	piGPIBBase()	Integer	Returns the GP-IB base address of the device (if connected)
Read only property	piGPIBCableConnected()	Integer	Returns a flag which defines whether the device is connected to the GP-IB board
Read only property	piGPIBCableConnectionSuccess()	Integer	Returns a flag which defines whether the device is connected to the GP-IB board successfully
Read only property	piNoOfControl()	Integer	Returns the number of controls of the device
Read only property	piNoOfOptions()	Integer	Returns the number of installed options
Read only property	piNoOfUnknownPar()	Integer	Returns the number of unknown parameters
Read only property	piOptionInstalled(ByVal iIndex As Integer)	Integer	Returns a flag which defines whether the specified option is installed in the device

Read only property	piPluginIndex()	Integer	Returns the current plugin index
Read only property	piUseDevice()	Integer	Returns a flag which indicates whether the devce is used
Read only property	psDeviceName()	String	Returns the device name
Read only property	psDevType()	String	Returns a string describing the device type
Read only property	psOptionName(ByVal iIndex As Integer)	String	Returns the specified option name
Read only property	psPluginName()	String	Returns the plugin name
Read/write property	pfRestoreWindowPos()	Integer	Sets or returns a value which defines whether the window position, its size and window state is restored when the window is displayed again after it has been closed
Read/write property	pfUserIF()	Integer	Sets or returns a value which defines whether the associated window should be shown on screen. A window is only displayed under the following condition: pfUserIF=TRUE, pfVisible=TRUE, pfHideForm=FALSE
Function	pfGetParameter(ByVal iParameter As Integer, ByRef dbValue As Double)	Integer	Returns the value of a specified parameter
Function	pfGetParameterByString(ByVal sParameterName As String, ByRef sValue As String)	Integer	Returns the value of a parameter specified by string
Function	pfParameter2Control(ByVal iParameter As Integer, ByRef iControl As Integer)	Integer	Returns the control index of the specified parameter. If the parameter does not exits the returns value is false
Function	pfParameterName2Control(ByVal sParameterName As String, ByRef iControl As Integer)	Integer	Returns the control index of parameter specified by name. If the parameter does not exits the returns value is false
Function	pfSetParameter(ByVal iParameter As Integer, ByVal dbNewValue As Double)	Integer	Sets the value of a specified parameter
Function	pfSetParameterByString(ByVal sParameterName As String, ByVal sNewValue As String)	Integer	Sets the value of a parameter specified by string
Function	pfSetParameterToMaximum(ByVal iParameter As Integer)	Integer	Sets the specified parameter to the maximum value
Function	pfSetParameterToMinimum(ByVal iParameter As Integer)	Integer	Sets the specified parameter to the minimum value
Function	psGetStatus()	String	Returns the current device status string

HDevPars

Event	ChangeEnabled(B		Event which is raised when the pfEnabled property

	yVal Index As Integer)		changes
Event	ChangeLimits(By Val Index As Integer)		Event which is raised when either pvMinValue or pvMaxValue changes
Event	ChangeType(ByVa l Index As Integer)		Event which is raised when the piDevParType property changes
Event	ChangeValue(ByV al Index As Integer, ByVal dbOldValue As Double, ByVal fFromDevice As Integer)		Event which is raised when the pfValue property changes
Event	ChangeVisible(By Val Index As Integer)		Event which is raised when the pfVisible property changes
Read only property	pcItem(ByVal Index As Integer)	HDevPar	Returns an object reference to the specified HDevPar object
Read only property	psName()	String	Returns the name of the object
Read/write property	piNrValidItems()	Integer	Returns a value which specifies the number of valid HDevPar objects. These are numbered from 0 to piNrValidItems-1

HAcq

Event	CameraIsStarted()		
Event	CameraTemperatur e(ByVal sTemp As String)		Event which is raised when the camera temperature changes
Event	ChangeAcqMode(ByVal iAcqMode As Integer)		Event which is raised when the Acquisiton mode changes
Event	EndAcquisition()		Event which is raised when an acquisition has ended. In the case of an acquisition which consists of several individual images (Live, Analog integratio, photon counting) it is only called once at the end of the complete acquisition
Event	EndBacksub()		Event which is raised when an backsub acquisition has ended. In the case of an acquisition which consists of several individual images (Analog integratio, photon counting) it is only called once at the end of the complete acquisition
Event	Message(ByVal sMessage As String)		Event which is raised to inform the user of ongoing operations
Event	NewLiveImage()		Event which is raised when a new image lin live mode is acquired and displayed
Event	NewPartImage(By Val lPartImage As Long)		Event which is raised when a new part of an image is acquired. E.g. a new part image is a newly acquired image which is added to the image in Analog Integration mode
Event	StartAcquisition()		Event which is raised before an acquisition starts. In the case of an acquisition which consists of several individual images (Live, Analog integratio, photon counting) it is only called once at the start of the complete acquisition
Event	StartBacksub()		Event which is raised before a backsub acquisition has been started. In the case of an acquisition which consists of several individual images (Analog integratio, photon counting) it is only called once at the start of the complete acquisition
Read only property	pfAcqPending()	Integer	Returns a flag which indicates whether an acquisition is

			currently running
Read only property	pfGrbIs4MType()	Integer	Returns a flag which defines whether the current frame grabber is a 4M type
Read only property	pfOptionAutoBacksub()	Integer	
Read only property	pfProcPending()	Integer	Returns a flag which indicates whether an image processing is currently running
Read only property	pfProcStopped()	Integer	Rreturns a flag which defines that a currently pending processing should be stopped
Read only property	pftDefaultExposureTime()	Single	Returns the default exposure time when the application starts. In the case the camera cannot set this value precisely the nearest possible value is set
Read only property	pHChkCorrGetRTBSFromCamera()	HCheck	Returns an object reference to the HCheck object associated with the correction setup dialogs 'Get RTBS form Camera' checkbox
Read only property	pHChkCorrSpectSensitivityCorr()	HCheck	Returns an object reference to the HCheck object associated with the correction setup dialogs 'Spectral Sensitivity Correction' checkbox
Read only property	pHChkCorrSubtractWithOpenShutter()	HCheck	Returns an object reference to the HCheck object associated with the correction setup dialogs 'Subtract With Open Shutter' checkbox
Read only property	pHChkOpt32BitInAI()	HCheck	
Read only property	pHChkOptAutoBacksub()	HCheck	Returns an object reference to the HCheck object associated with the options dialogs 'Auto Backsub' checkbox
Read only property	pHChkOptAutoCurvature()	HCheck	Returns an object reference to the HCheck object associated with the options dialogs 'Auto Curvature' checkbox
Read only property	pHChkOptAutoShading()	HCheck	Returns an object reference to the HCheck object associated with the options dialogs 'Auto Shading' checkbox
Read only property	pHChkOptClipZero()	HCheck	Returns an object reference to the HCheck object associated with the options dialogs Clip To Zero 'During backsub' checkbox
Read only property	pHChkOptDefectCorrection()	HCheck	
Read only property	pHChkOptSlowDisplay()	HCheck	
Read only property	pHChkOptWriteDPCFile()	HCheck	Returns an object reference to the HCheck object associated with the options dialogs 'Write DPC file' checkbox
Read only property	pHComBackAbort()	HCommand	Returns an object reference to the HCommand object associated with the backsub dialogs 'Abort' pushbutton
Read only property	pHComBackStop()	HCommand	Returns an object reference to the HCommand object associated with the backsub dialogs 'Stop' pushbutton
Read only property	pHComCorrCancel()	HCommand	Returns an object reference to the HCommand object associated with the Correction Setup dialogs 'Cancel' pushbutton
Read only property	pHComCorrGetAcquireModeFile()	HCommand	Returns an object reference to the HCommand object associated with the Correction Setup dialogs 'Get Acquire Mode Backsub File' pushbutton
Read only property	pHComCorrGetAnalogIntegrationFile()	HCommand	Returns an object reference to the HCommand object associated with the Correction Setup dialogs 'Get Analog Integration Mode Backsub File' pushbutton
Read only property	pHComCorrGetLampFile()	HCommand	Returns an object reference to the HCommand object associated with the Correction Setup dialogs 'Get Lamp File' pushbutton
Read only property	pHComCorrGetLiveModeFile()	HCommand	Returns an object reference to the HCommand object associated with the Correction Setup dialogs 'Get Live Mode Backsub File' pushbutton

Read only property	pHComCorrGetShadingFile()	HCommand	Returns an object reference to the HCommand object associated with the Correction Setup dialogs 'Get Shading File' pushbutton
Read only property	pHComCorrOK()	HCommand	Returns an object reference to the HCommand object associated with the Correction Setup dialogs 'OK' pushbutton
Read only property	pHComCurvcorrCancel()	HCommand	Returns an object reference to the HCommand object associated with the Curvature Correction Setup dialogs 'Cancel' pushbutton
Read only property	pHComCurvcorrDrawCurve()	HCommand	Returns an object reference to the HCommand object associated with the Curvature Correction Setup dialogs 'Draw Curve' pushbutton
Read only property	pHComCurvcorrHorPoint1()	HCommand	Returns an object reference to the HCommand object associated with the Curvature Correction Setup dialogs 'Define Horizontal Point 1' pushbutton
Read only property	pHComCurvcorrHorPoint2()	HCommand	Returns an object reference to the HCommand object associated with the Curvature Correction Setup dialogs 'Define Horizontal Point 2' pushbutton
Read only property	pHComCurvcorrHorPoint3()	HCommand	Returns an object reference to the HCommand object associated with the Curvature Correction Setup dialogs 'Define Horizontal Point 3' pushbutton
Read only property	pHComCurvcorrHorRecall1()	HCommand	Returns an object reference to the HCommand object associated with the Curvature Correction Setup dialogs 'Recall Horizontal Point 1' pushbutton
Read only property	pHComCurvcorrHorRecall2()	HCommand	Returns an object reference to the HCommand object associated with the Curvature Correction Setup dialogs 'Recall Horizontal Point 2' pushbutton
Read only property	pHComCurvcorrHorRecall3()	HCommand	Returns an object reference to the HCommand object associated with the Curvature Correction Setup dialogs 'Recall Horizontal Point 3' pushbutton
Read only property	pHComCurvcorrLoad()	HCommand	Returns an object reference to the HCommand object associated with the Curvature Correction Setup dialogs 'Load' pushbutton
Read only property	pHComCurvcorrSave()	HCommand	Returns an object reference to the HCommand object associated with the Curvature Correction Setup dialogs 'Save' pushbutton
Read only property	pHComCurvcorrSet()	HCommand	Returns an object reference to the HCommand object associated with the Curvature Correction Setup dialogs 'Set' pushbutton
Read only property	pHComCurvcorrVerPoint1()	HCommand	Returns an object reference to the HCommand object associated with the Curvature Correction Setup dialogs 'Define Vertical Point 1' pushbutton
Read only property	pHComCurvcorrVerPoint2()	HCommand	Returns an object reference to the HCommand object associated with the Curvature Correction Setup dialogs 'Define Vertical Point 2' pushbutton
Read only property	pHComCurvcorrVerPoint3()	HCommand	Returns an object reference to the HCommand object associated with the Curvature Correction Setup dialogs 'Define Vertical Point 3' pushbutton
Read only property	pHComCurvcorrVerRecall1()	HCommand	Returns an object reference to the HCommand object associated with the Curvature Correction Setup dialogs 'Recall Vertical Point 1' pushbutton
Read only property	pHComCurvcorrVerRecall2()	HCommand	Returns an object reference to the HCommand object associated with the Curvature Correction Setup dialogs 'Recall Vertical Point 2' pushbutton
Read only property	pHComCurvcorrVerRecall3()	HCommand	Returns an object reference to the HCommand object associated with the Curvature Correction Setup dialogs 'Recall Vertical Point 3' pushbutton
Read only property	pHComDefPixCalculateDefectPixels()	HCommand	
Read only property	pHComDefPixClo	HComma	

	se()	nd	
Read only property	pHComDefPixGetImageDeadPixels()	HCommand	
Read only property	pHComDefPixGetImageHotPixels()	HCommand	
Read only property	pHComDefPixReCalculateDeadPixels()	HCommand	
Read only property	pHComDefPixReCalculateHotPixels()	HCommand	
Read only property	pHComDefPixSaveToINIFile()	HCommand	
Read only property	pHComOptCancel()	HCommand	Returns an object reference to the HCommand object associated with the Options dialogs 'Cancel' pushbutton
Read only property	pHComOptGetDefectPixelFile()	HCommand	
Read only property	pHComOptOK()	HCommand	Returns an object reference to the HCommand object associated with the Options dialogs 'OK' pushbutton
Read only property	pHDisBackMessage()	HDisp	Returns an object reference to the HDisp object associated with the Backsubs dialogs 'Message' display area
Read only property	pHDisDefPixAverageDeadPixel()	HDisp	
Read only property	pHDisDefPixAverageHotPixel()	HDisp	
Read only property	pHDisDefPixNrDefectColumns()	HDisp	
Read only property	pHDisDefPixNrDefectLines()	HDisp	
Read only property	pHDisDefPixNrDefectPixels()	HDisp	
Read only property	pHDisDefPixStandDevDeadPixel()	HDisp	
Read only property	pHDisDefPixStandDevHotPixel()	HDisp	
Read only property	pHDisDefPixStatus()	HDisp	
Read only property	pHEdnCorrAddConstant()	HEditNumber	Returns an object reference to the HEditNumber object associated with the Correction setup dialogs 'Add Constant During Backsub' editbox
Read only property	pHEdnDefPixLineColumnsPercentage()	HEditNumber	
Read only property	pHEdnDefPixThresholdDeadLines()	HEditNumber	
Read only property	pHEdnDefPixThresholdDeadPixels()	HEditNumber	
Read only property	pHEdnDefPixThresholdHotLines()	HEditNumber	
Read only property	pHEdnDefPixThresholdHotPixels()	HEditNumber	
Read only property	pHEdnOptAdditionalTimeout()	HEditNumber	Returns an object reference to the HEditNumber object associated with the Correction setup dialogs 'Additional Timeout' editbox
Read only property	pHEstCorrBackAcqFile()	HEditString	Returns an object reference to the HEditString object associated with the Correction Setup dialogs 'Acquire Mode Background File' editbox
Read only property	pHEstCorrBackAIntFile()	HEditString	Returns an object reference to the HEditString object associated with the Correction Setup dialogs 'Analog Integration Mode Background File' editbox

Read only property	pHEstCorrBackLiveFile()	HEditString	Returns an object reference to the HEditString object associated with the Correction Setup dialogs 'Live Mode Background File' editbox
Read only property	pHEstCorrLampFile()	HEditString	Returns an object reference to the HEditString object associated with the Correction Setup dialogs 'Lamp File' editbox
Read only property	pHEstCorrShadingFile()	HEditString	Returns an object reference to the HEditString object associated with the Correction Setup dialogs 'Acquire Mode Shading File' editbox
Read only property	pHEstDefPixImageDeadPixel()	HEditString	
Read only property	pHEstDefPixImageHotPixel()	HEditString	
Read only property	pHEstOptDefectPixelFile()	HEditString	
Read only property	pHFraCorrBackgroundSubtraction()	HFrame	Returns an object reference to the HFrame object associated with the Correction Setup dialogs 'Background Subtraction' frame
Read only property	pHFraCorrShadingCorrection()	HFrame	Returns an object reference to the HFrame object associated with the Correction Setup dialogs 'Shading Correction' frame
Read only property	pHFraDefPixDeadPixels()	HFrame	
Read only property	pHFraDefPixHotPixels()	HFrame	
Read only property	pHFraDefPixResults()	HFrame	
Read only property	pHProBackProgress()	HProgress	Returns an object reference to the HProgress object associated with the backsub dialogs 'Progress' progress bar
Read only property	pHRadCorrBackgroundSource()	HRadios	Returns an object reference to the HRadios object associated with the Correction setup dialogs 'Background Source' radiobutton group
Read only property	pHRadCurvcorrCurvatureDirection()	HRadios	Returns an object reference to the HRadios object associated with the Correction setup dialogs 'Curvature Direction' radiobutton group
Read only property	pHRadDefPixMethod()	HRadios	
Read only property	pHWinBackDlg()	HWindow	Returns an object reference to the HWindow object associated with the Backsub dialogs main window
Read only property	pHWinCorrDlg()	HWindow	Returns an object reference to the HWindow object associated with the Correction Setup dialogs main window
Read only property	pHWinCurvcorrDlg()	HWindow	Returns an object reference to the HWindow object associated with the Curvature Correction dialogs main window
Read only property	pHWinDefPixDlg()	HWindow	
Read only property	pHWinOptDlg()	HWindow	Returns an object reference to the HWindow object associated with the Options dialogs main window
Read only property	pHWinOptOptions()	HWindow	Returns an object reference to the HWindow object associated with the options dialogs options picture box
Read only property	piAmMod()	Integer	Returns the acquisition module. Possible values are described in the enumeration AcquisitionModule
Read/write property	pfDisableAutoAction()	Integer	Sets or returns a flag which defines whether automatic actions should be executed
Read/write property	pfDisableAutoInquiry()	Integer	Sets or returns a flag which defines whether an inquiry is shown to the user before the autoaction takes place
Read/write property	pfOption32BitInAI()	Integer	

Read/write property	pfOptionAutoCurvature()	Integer	Sets or returns a value which defines whether Auto curvature function should be executed
Read/write property	pfOptionAutoShading()	Integer	Sets or returns a value which defines whether Auto shading function should be executed
Read/write property	pfOptionClipZero()	Integer	Sets or returns a value which defines whether clip to Zero function should be executed during background subtraction
Read/write property	pfOptionDefectCorrection()	Integer	
Read/write property	pfOptionSlowDisplay()	Integer	Sets or returns a value which defines whether slow display function should be executed during LIVE mode
Read/write property	pfOptionWriteDPCFile()	Integer	Sets or returns a value which defines whether DPC files should be written during photon counting
Read/write property	pfRestoreWindowPos()	Integer	Sets or returns a value which defines whether the window position, its size and window state is restored when the window is displayed again after it has been closed
Read/write property	pfStartImagesIndividually()	Integer	
Read/write property	pfUserIF()	Integer	Sets or returns a value which defines whether the associated window should be shown on screen. A window is only displayed under the following condition: pfUserIF=TRUE, pfVisible=TRUE, pfHideForm=FALSE
Read/write property	piOptionAdditionalTimeout()	Integer	Sets or returns the value for additional timeout
Read/write property	psOptionDefectPixelFile()	String	
Function	piValidAcqMode(ByVal iAcqMode As Integer)	Integer	Returns a flag which defines whether the specified AcqMode is valid
Function	psGetStatus()	String	Returns the current acquisition status string
Sub	pAsyncAcquire()		Starts Acquire mode asynchronously
Sub	pAsyncAnalogIntegration()		Starts Analog Intergration mode asynchronously
Sub	pAsyncLive(ByVal fSingleImage As Integer)		Starts Live mode asynchronously
Sub	pAsyncPhotonCounting()		Starts Photon Counting mode asynchronously
Sub	pGetAcqDim(iX As Integer, iY As Integer, iDX As Integer, iDY As Integer, iBytesPerPixel As Integer)		Gets the acquisition dimensions for the current camera settings
Sub	pStopAcquisition()		Stops the currently running acquisition
Sub	pUpdateCameraParms()		

HGrb

Read only property	pfGrbIs4MType()	Integer	Return whether or not the framebuffer has 4 MB
Read only property	piFrameBytesPerPixel()	Integer	Returns the number of bytes per pixel within the frame grabbers frame
Read/write property	pGrbDataValid()	Integer	Returns a value whether or not the data in the frame grabber is valid
Read/write property	pVideoExtSync()	Integer	Sets or returns whether external sync is used for a video camera
Read/write property	pVideoStartField()	Integer	Sets the start field for a video camera
Function	pAcqAverageMulti	Integer	Average a series of 8 bit images acquired from video

	pleFrames(ByVal fICPAverageMultipleFrames As Integer, ByVal iICPAverageMultipleFramesCount As Integer, ByVal lpvBufferArea As Long, ByVal areBufferiX As Integer, ByVal areBufferiY As Integer, ByVal areBufferiDX As Integer, ByVal areBufferiDY As Integer, ByVal iBufferBytesPerPixel As Integer, ByVal lBuffer16Handle As Long)		camera with external trigger
Function	pAcqGrbStartTrigSeqSnap(ByVal fICPAverageMultipleFrames As Integer, ByVal iICPAverageMultipleFramesCount As Integer, ByVal miAdditionalTimeout As Integer, ByRef IHCall2AcqControl As IHCall2AcqControl)	Integer	Start a triggered sequential snap
Function	pCheckVideoCamera()	Integer	Check operation of a video camera
Function	pCreateFrame(ByVal iBits As Integer, ByVal fSeqFrame As Integer, ByVal iNrSeqFrames As Integer)	Integer	Creates a frame on the frame grabber
Function	pfAcqEnterVB(ByVal fTargetField As Integer, ByVal ftTimeout As Single)	Integer	Wait until the camera enters the vertical blank
Function	pfAcqPassLine(ByVal lStartLine As Long, ByVal iVact As Integer, ByVal ftTimeout As Single)	Integer	Waits until the camera passes a specified line. The line number can alos have the meaning of a byte counter
Function	pfCanAcquireLargeFrames()	Integer	Returns a value which indicates that the current frame grabber configuration can acquire large frames (larger than the frame grabbers memory)
Function	pfGrbCheckEndOf	Integer	Return the state whether or not an acquisition has ended

	Acquisition()		
Function	pfGrbCopyFBToBuffer(ByVal lpvBufferArea As Long, ByRef areBufferiX As Integer, ByRef areBufferiY As Integer, ByRef areBufferiDX As Integer, ByRef areBufferiDY As Integer, ByRef iBufferBytesPerPixel As Integer)	Integer	Copies the content of the frame buffer into a computers memory buffer
Function	pfGrbCopyFBToBufferInParts(ByVal lpvBufferArea As Long, ByRef areBufferiX As Integer, ByRef areBufferiY As Integer, ByRef areBufferiDX As Integer, ByRef areBufferiDY As Integer, ByRef iBufferBytesPerPixel As Integer, ByVal iParts As Integer, ByVal fCheckValid As Integer)	Integer	Copies the content of the frame buffer into a computers memory buffer in parts
Function	pfGrbCopyFBToMem(ByRef HImage As HImage)	Integer	Copies the content of the frame buffer into an image memory
Function	pfGrbCopyFBToMemInParts(ByRef HImage As HImage, ByVal iParts As Integer, ByVal fCheckValid As Integer)	Integer	Copies the content of the frame buffer into an image memory in parts
Function	pfGrbSetInputLUTforPC(ByVal iThreshold As Integer, ByVal iDat As Integer)	Integer	Sets the Input LUT for photon counting
Function	pGetVideoCameraInfo(ByRef iCameraDX As Integer, ByRef iCameraDY As Integer, ByRef ftVideoFrameTime As Single)	Integer	Get info about the cameras frame
Function	pGetWindow(ByRef hOff%, ByRef hAct%, ByRef vOff%, ByRef	Integer	Gets the hardware window parameters for a given ROI

	vAct%)		
Function	pgrbacqbits(ByVal itxmode As Integer)	Integer	Sets the frame grabbers acquisition bits
Function	pgrbfreeze()	Integer	Freezes the current acquisition
Function	pGrbGetImgAttrib (ByRef areSourceiX As Integer, ByRef areSourceiY As Integer, ByRef areSourceiDX As Integer, ByRef areSourceiDY As Integer, ByRef iBytesPerPixel As Integer)	Integer	Gets the attributes of the image in frame grabber
Function	pGrbGetScanArea(ByVal iCameraDX As Integer, ByVal iCameraDY As Integer, ByRef pntDummyX As Integer, ByRef pntDummyY As Integer, ByRef pntCamWidthX As Integer, ByRef pntCamWidthY As Integer, ByRef areGRBScaniX As Integer, ByRef areGRBScaniY As Integer, ByRef areGRBScaniDX As Integer, ByRef areGRBScaniDY As Integer, ByRef areValidIMGiX As Integer, ByRef areValidIMGiY As Integer, ByRef areValidIMGiDX As Integer, ByRef areValidIMGiDY As Integer, ByRef pntOrigFBX As Integer, ByRef pntOrigFBY As Integer)	Integer	Get the scan area
Function	pGrbGetWindowForSubArea(ByVal iX%, ByVal iY%, ByVal iDX%, ByVal iDY%, ByRef hOff%, ByRef hAct%, ByRef vOff%, ByRef vAct%, ByRef sError$)	Integer	Get the window generator hardware parameters for a given subarrea
Function	pgrbgrab()	Integer	Starts grabbing

Function	pGrbSetDefaultSettings(ByVal iFrameDX As Integer, ByVal iFrameDY As Integer)	Integer	Set default settings
Function	pgrbsnap()	Integer	Snaps a single image
Function	pGrbWaitEndOfImgReadout(ByVal ftTimeout As Single)	Integer	Wait unti end of image readout
Function	pGrbWriteLinearInputLUT(ByVal iDat As Integer)	Integer	Writes a linear input LUT
Function	pGrbWriteReducedLinLUT(ByVal iDat As Integer, ByVal iThreshold As Integer)	Integer	Write a linear LUT which reduces the image to 8 bit
Function	piInit(ByVal sConfigFile As String, ByVal iGRB As Integer, ByVal iGrabberSysNo As Integer, ByVal iAmMod As Integer, ByVal iComPort As Integer, ByRef sError As String, ByRef HError As HError, ByVal iModelNr As Long, lHandle As Long)	Integer	Initializes the frame grabber
Function	plAcqAmDigStartLine(ByRef areACQiX As Integer, ByRef areACQiY As Integer, ByRef areACQiDX As Integer, ByRef areACQiDY As Integer, ByVal lpvBufferArea As Long, ByRef iBufferBytesPerPixel As Integer, ByRef ftTimeForReadout As Single, ByRef ftFrameTime As Single, ByVal ftTimeoutFromCamera As Single, ByRef lLastLine As Long)	Long	Calculates the starting line (or byte counter) used for the CatCo strategy
Function	plCurrenLine(ByVal iDY As Integer)	Long	Returns the current line (or byte counter)

Function	pPassVB()	Integer	Passes the next vertical blank
Function	pRSeqDoABSequence(ByVal areGRBScaniX As Integer, ByVal areGRBScaniY As Integer, ByVal areGRBScaniDX As Integer, ByVal areGRBScaniDY As Integer, ByVal iBytesPerPixel As Integer, ByVal fWrap As Integer, lRSeqPtr()	Long, ByRef sISeqTime() As String, ByRef dbRSeqStartTime As Double, ByRef HEdnSeqCurrentSample As HEditNumber, ByVal fEnableStopinSequence As Integer, ByVal lNrSamples As Long, ByRef IHCall2AcqControl As IHCall2AcqControl, ByVal ftTimeout As Single) As Integer	Executes an AB sequence
Function	pRSeqDoAmvsCatcoSequence(ByVal fWrap As Integer, ByRef iErrors As Integer, ByRef HEdnSeqCurrentSample As HEditNumber, ByRef lRSeqPtr()	Long, ByRef dbRSeqStartTime As Double, ByRef sISeqTime() As String, ByVal fEnableStopinSequence As Integer, ByVal lNrSamples As Long, ByRef IHCall2AcqControl As IHCall2AcqControl	Executes CatCo sequence for video module

		) As Integer	
Function	pSetWindow(ByVal hOff%, ByVal hAct%, ByVal vOff%, ByVal vAct%)	Integer	Sets the window generator harware parameters
Function	psGetStatus()	String	Returns a string defining the current status
Function	Test()		
Sub	pCleanUp()		Clean up all references
Sub	pSetHMsgBox(HMsgBox As HMsgBox)		
Sub	pSetIHCall2Sequence(ByVal IHCall2Sequence As IHCall2Sequence)		Sets the target object for Call2Sequence calls
Sub	pSetIHCall2UserFunc(ByVal IHCall2UserFunc As IHCall2UserFunc)		Sets the target object for Call2userFunc calls

HLicence

Read only property	pfApplicationKeyFound()	Integer	Returns a flag defining whether the correct application key could be found
Read only property	pfGetCustomKey(iIndex As Integer)	Integer	Returns an information whether the Custom Key with the specified index exists
Read only property	pfLicenceAcquire()		Returns a flag defining whether a key for acquiring images could be found
Read only property	pfLicenceFitting()		Returns a flag defining whether a key for fitting could be found
Read only property	pfLicenceRCOnly()		Returns a flag defining whether a key for RC only could be found
Read only property	pfLicenceSave()		Returns a flag defining whether a key for saving images and profiles could be found
Read only property	pfLicenceSequence()		Returns a flag defining whether a key for sequence acquisition could be found
Read only property	pfLicenceTransAbs()		Returns a flag defining whether a key for transien absorption could be found
Read only property	piLicenceDemoMode()	Integer	Returns a value defining which demo mode is used in the case no licence could be found.

HUtils

Event	CloseComm()		Event which is raised when the RS232 communication is closed
Event	CommPortChanged()		Event which is raised when the com prt has been changed
Event	EndDetected()		Event which is raised when an END string is detected in the communication from the camera
Event	GeneralMessage(ByVal sMessage As String)		
Read only property	pfRS232DlgVisible()	Integer	Returns a flag which defines whether the RS232 dialog is visible

HAsyncCommand

Event	AfterCommand(ByVal sCommand As String)		Event which is raised after the execution of an asynchronous command
Event	BeforeCommand(ByVal sCommand		Event which is raised before the execution of an asynchronous command

	As String)		
Event	DoCommand(ByVal sCommand As String, ByVal vParam0 As Variant, ByVal vParam1 As Variant, ByVal vParam2 As Variant, ByVal vParam3 As Variant, ByVal vParam4 As Variant)		Event which is raised to execute an asynchronous command
Read only property	pfAsyncCommandActive()	Integer	Returns a value which defines whether an acync command is currently running. Another async command will not start is such case
Read only property	psActiveAsyncCommand()	String	Returns a string indication the currently active async command (if an async command is actice)

HMsgBox

Event	MsgBox(ByVal iID As Integer, ByVal sPrompt As String, ByVal sTitle As String, ByVal Style As Integer, ByVal Buttons As Integer, ByRef default As Integer)		Event which is raised in remote control mode when the pfNoDialogs property is set to true instead of showing a messagebox. This event can be use for the client program to react on such messages
Read/write property	pfNoDialogs()	Integer	Sets or returns a value which defines whether a visible MessageBox should be explizitly supressed

HError

Read only property	psErrorFile()	String	Sets or returns a string defining the filename to which the erro report is written in case of an error
Sub	pErrRegEvent(ByVal sName As String, ByVal fBasicEvent As Integer)		Registers an event to the error handler
Sub	pEventError(ByVal sModuleEvent As String)		Starts the event handler when an error has been raised in an event. This is the ending point of the error handling
Sub	pFunctionError(ByVal sModuleFunction As String, Optional RaiseError)		Starts the event handler when an error has been occurred in a function/sub. At the end such error handler raises again an error which returns control one level up in the calling order. The last level in the calling order is an event (See pEventError)

HCheck

Event	ChangeEnabled()		Event which is raised when the pfEnabled property changes
Event	ChangeValue()		Event which is raised when the pfValue property changes
Event	ChangeVisible()		Event which is raised when the pfVisible property changes
Read only property	pcParent()	Object	Returns a reference to the parent object of this object. If this reference is Nothing this object is the topmost object within the object hierarchy (normally the HWindow object of the associated dialog)

Read only property	pfControlAvail()	Integer	Returns a value which specifies whether the object can be controlled by the client programmer (This is dependent on the pfVisible and pfEnabled properties and those of its parent)
Read only property	pfEnabled()	Integer	Returns a value which specifies whether the object is enabled
Read only property	pfVisible()	Integer	Returns a value which specifies whether the object is visible
Read only property	psName()	String	Returns the name of the object
Read/write property	pfValue(Optional DoEvent As Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer)	Integer	Sets or returns the value of the object (true if checked and false if unchecked)
Read/write property	psCaption()	String	Sets or returns the text which is used to label the associated control
Sub	pRegEvent(ByVal sEvent As String, ByVal fBasicEvent As Integer)		Should not be used by clients! (Registers an event to the error handler)

HCheck4Array

Event	ChangeEnabled(ByVal Index As Integer)		Event which is raised when the pfEnabled property changes
Event	ChangeValue(ByVal Index As Integer)		Event which is raised when the pfValue property changes
Event	ChangeVisible(ByVal Index As Integer)		Event which is raised when the pfVisible property changes
Read only property	pcItem(ByVal Index As Integer)	HCheck	Returns an object reference to the specified HCheck object
Read only property	psName()	String	Returns the name of the object

HCommand

Event	ChangeEnabled()		Event which is raised when the pfEnabled property changes
Event	ChangeVisible()		Event which is raised when the pfVisible property changes
Event	Click()		Event which is raised when the pClick method is executed
Read only property	pcParent()	Object	Returns a reference to the parent object of this object. If this reference is Nothing this object is the topmost object within the object hierarchy (normally the HWindow object of the associated dialog)
Read only property	pfControlAvail()	Integer	Returns a value which specifies whether the object can be controlled by the client programmer (This is dependent on the pfVisible and pfEnabled properties and those of its parent)
Read only property	pfEnabled()	Integer	Returns a value which specifies whether the object is enabled
Read only property	pfVisible()	Integer	Returns a value which specifies whether the object is visible
Read only property	psName()	String	Returns the name of the object
Read/write property	psCaption()	String	Sets or returns the text which is used to label the associated control
Sub	pClick()		Method which raises the Click-Event and executes the associated command
Sub	pRegEvent(ByVal		Should not be used by clients! (Registers an event to the

	sEvent As String, ByVal fBasicEvent As Integer)		error handler)

HCommand4Array

Event	ChangeEnabled(ByVal Index As Integer)		Event which is raised when the pfEnabled property changes
Event	ChangeVisible(ByVal Index As Integer)		Event which is raised when the pfVisible property changes
Event	Click(Index As Integer)		
Read only property	pcItem(ByVal Index As Integer)	HCommand	Returns an object reference to the specified HCommand object
Read only property	psName()	String	Returns the name of the object

HDisp

Event	ChangeMessage()		Event which is raised when the psMessage property changes
Event	ChangeVisible()		Event which is raised when the pfVisible property changes
Read only property	pcParent()	Object	Returns a reference to the parent object of this object. If this reference is Nothing this object is the topmost object within the object hierarchy (normally the HWindow object of the associated dialog)
Read only property	pfControlAvail()	Integer	Returns a value which specifies whether the object can be controlled by the client programmer (This is dependent on the pfVisible and pfEnabled properties and those of its parent)
Read only property	pfEnabled()	Integer	Returns a value which specifies whether the object is enabled
Read only property	pfVisible()	Integer	Returns a value which specifies whether the object is visible
Read only property	psMessage()	String	Returns a text containing the message which is displayed on the corresponding control
Read only property	psName()	String	Returns the name of the object
Read/write property	psCaption()	String	Sets or returns the text which is used to label the associated control
Sub	pRegEvent(ByVal sEvent As String, ByVal fBasicEvent As Integer)		Should not be used by clients! (Registers an event to the error handler)

HDisp4Array

Event	ChangeMessage(Index As Integer)		Event which is raised when the psMessage property of the specified HDisp object changes
Event	ChangeVisible(Index As Integer)		Event which is raised when the pfVisible property of the specified HDisp object changes
Read only property	pcItem(ByVal Index As Integer)	HDisp	Returns an object reference to the specified HDisp object
Read only property	psName()	String	Returns the name of the object

HEditNumber

Event	ChangeEnabled()		Event which is raised when the pfEnabled property changes
Event	ChangeLimits()		Event which is raised when either pvMinValue or pvMaxValue changes
Event	ChangeValue()		Event which is raised when the pfValue property changes
Event	ChangeVisible()		Event which is raised when the pfVisible property changes
Read only property	pcParent()	Object	Returns a reference to the parent object of this object. If this reference is Nothing this object is the topmost object

			within the object hierarchy (normally the HWindow object of the associated dialog)
Read only property	pfControlAvail()	Integer	Returns a value which specifies whether the object can be controlled by the client programmer (This is dependent on the pfVisible and pfEnabled properties and those of its parent)
Read only property	pfEnabled()	Integer	Returns a value which specifies whether the object is enabled
Read only property	pfVisible()	Integer	Returns a value which specifies whether the object is visible
Read only property	piNumberType()	Integer	Returns the number type of the object (Possible values are described in the enumeration NumberType)
Read only property	psName()	String	Returns the name of the object
Read only property	pvMaxValue()	Variant	Maximum value limiting the range for pvValue
Read only property	pvMinValue()	Variant	Minimum value limiting the range for pvValue
Read/write property	psCaption()	String	Sets or returns the text which is used to label the associated control
Read/write property	pvValue(Optional DoEvent As Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer)	Variant	Returns the value of the object
Sub	pRegEvent(ByVal sEvent As String, ByVal fBasicEvent As Integer)		Should not be used by clients! (Registers an event to the error handler)

HEditNumber4Array

Event	ChangeEnabled(ByVal Index As Integer)		Event which is raised when the pfEnabled property changes
Event	ChangeLimits(By Val Index As Integer)		
Event	ChangeValue(ByVal Index As Integer)		Event which is raised when the pfValue property changes
Event	ChangeVisible(By Val Index As Integer)		Event which is raised when the pfVisible property changes
Read only property	pcItem(ByVal Index As Integer)	HEditNumber	Returns an object reference to the specified HEditNumber object
Read only property	psName()	String	Returns the name of the object

HEditString

Event	ChangeEnabled()		Event which is raised when the pfEnabled property changes
Event	ChangeText()		Event which is raised when the psText property changes
Event	ChangeVisible()		Event which is raised when the pfVisible property changes
Read only property	pcParent()	Object	Returns a reference to the parent object of this object. If this reference is Nothing this object is the topmost object within the object hierarchy (normally the HWindow object of the associated dialog)
Read only property	pfControlAvail()	Integer	Returns a value which specifies whether the object can be controlled by the client programmer (This is dependent on the pfVisible and pfEnabled properties and those of its parent)
Read only property	pfEnabled()	Integer	Returns a value which specifies whether the object is enabled

Read only property	pfVisible()	Integer	Returns a value which specifies whether the object is visible
Read only property	psName()	String	Returns the name of the object
Read/write property	psCaption()	String	Sets or returns the text which is used to label the associated control
Read/write property	psText(Optional DoEvent As Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer)	String	Sets or returns the text of the object
Sub	pRegEvent(ByVal sEvent As String, ByVal fBasicEvent As Integer)		Should not be used by clients! (Registers an event to the error handler)

HEntry

Event	ChangeEnabled()		Event which is raised when the pfEnabled property changes
Event	ChangeNoEntries()		Event which is raised when the piNoEntries property changes
Event	ChangeValue()		Event which is raised when the selected entry changes
Event	ChangeVisible()		Event which is raised when the pfVisible property changes
Read only property	EntryType()	Integer	Returns the entry type of the object. Possible values are defined in the enumeration EntryType
Read only property	pcParent()	Object	Returns a reference to the parent object of this object. If this reference is Nothing this object is the topmost object within the object hierarchy (normally the HWindow object of the associated dialog)
Read only property	pfControlAvail()	Integer	Returns a value which specifies whether the object can be controlled by the client programmer (This is dependent on the pfVisible and pfEnabled properties and those of its parent)
Read only property	pfEnabled()	Integer	Returns a value which specifies whether the object is enabled
Read only property	pfVisible()	Integer	Returns a value which specifies whether the object is visible
Read only property	piNoEntries()	Integer	Returns the number of entries of the object
Read only property	psName()	String	Returns the name of the object
Read only property	psSetKeyValue(By Val Index As Integer)	String	Returns the key value of a specified entry index (Not necessarily the current one)
Read only property	psSetTag(ByVal Index As Integer)	String	Rerturns the tag value of a specified entry index (Not necessarily the current one)
Read only property	psTag()	String	Rerturns the tag value of the object (This is an additional string entry associated to every entry)
Read/write property	pfSearchNext()	Integer	Sets or returns a value which defines whether the nearest neighbour should be selected if no exact match is found (Only valid in combination EntryTypeTime orEntryTypeNumber)
Read/write property	pftValue(Optional DoEvent As Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer)	Single	Sets or returns a numerical value associated to the entry type. Time values in ms, numerical values or entry indexes are used depending on the EntryType
Read/write property	piEntry(Optional DoEvent As	Integer	Sets or returns the entry index (Entries are numbered from piEntry=0 to piNrEntries-1)

	Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer)		
Read/write property	psCaption()	String	Sets or returns the text which is used to label the associated control
Read/write property	psKeyValue(Optional DoEvent As Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer)	String	Sets or returns the key value of the object. This is a string value of the entry. When this value is set the all entries are searched whether the contain the specified string.
Sub	pRegEvent(ByVal sEvent As String, ByVal fBasicEvent As Integer)		Should not be used by clients! (Registers an event to the error handler)

HEntry4Array

Event	ChangeEnabled(ByVal Index As Integer)		Event which is raised when the pfEnabled property changes
Event	ChangeLimits(By Val Index As Integer)		
Event	ChangeValue(ByVal Index As Integer)		Event which is raised when the pfValue property changes
Event	ChangeVisible(By Val Index As Integer)		Event which is raised when the pfVisible property changes
Read only property	pcItem(ByVal Index As Integer)	HEntry	Returns an object reference to the specified HEntry object
Read only property	psName()	String	Returns the name of the object

HFrame

Event	ChangeEnabled()		Event which is raised when the pfEnabled property changes
Event	ChangeVisible()		Event which is raised when the pfVisible property changes
Read only property	pcParent()	Object	Returns a reference to the parent object of this object. If this reference is Nothing this object is the topmost object within the object hierarchy (normally the HWindow object of the associated dialog)
Read only property	pfControlAvail()	Integer	Returns a value which specifies whether the object can be controlled by the client programmer (This is dependent on the pfVisible and pfEnabled properties and those of its parent)
Read only property	pfEnabled()	Integer	Returns a value which specifies whether the object is enabled
Read only property	pfVisible()	Integer	Returns a value which specifies whether the object is visible
Read only property	psName()	String	Returns the name of the object
Read/write property	psCaption()	String	Sets or returns the text which is used to label the associated control
Sub	pRegEvent(ByVal sEvent As String, ByVal fBasicEvent As Integer)		Should not be used by clients! (Registers an event to the error handler)

HFrame4Array

Event	ChangeEnabled(B		Event which is raised when the pfEnabled property

	yVal Index As Integer)		changes
Event	ChangeVisible(By Val Index As Integer)		Event which is raised when the pfVisible property changes
Read only property	pcItem(ByVal Index As Integer)	HFrame	Returns an object reference to the specified HFrame object
Read only property	psName()	String	Returns the name of the object

HMenu

Event	ChangeChecked()		Event which is raised when the pfChecked property changes
Event	ChangeEnabled()		Event which is raised when the pfEnabled property changes
Event	ChangeVisible()		Event which is raised when the pfVisible property changes
Event	Click()		Event which is raised when the pClick method is executed
Read only property	pcParent()	Object	Returns a reference to the parent object of this object. If this reference is Nothing this object is the topmost object within the object hierarchy (normally the HWindow object of the associated dialog)
Read only property	pfChecked()	Integer	Returns a value which specifies whether the object is checked
Read only property	pfControlAvail()	Integer	Returns a value which specifies whether the object can be controlled by the client programmer (This is dependent on the pfVisible and pfEnabled properties and those of its parent)
Read only property	pfEnabled()	Integer	Returns a value which specifies whether the object is enabled
Read only property	pfVisible()	Integer	Returns a value which specifies whether the object is visible
Read only property	psName()	String	Returns the name of the object
Read/write property	psCaption()	String	Sets or returns the text which is used to label the associated control
Sub	pClick()		Method which raises the Click-Event and executes the associated command
Sub	pRegEvent(ByVal sEvent As String, ByVal fBasicEvent As Integer)		Should not be used by clients! (Registers an event to the error handler)

HProgress

Event	ChangePercent()		Event which is raised when the pfPercent property changes
Event	ChangeVisible()		Event which is raised when the pfVisible property changes
Read only property	pcParent()	Object	Returns a reference to the parent object of this object. If this reference is Nothing this object is the topmost object within the object hierarchy (normally the HWindow object of the associated dialog)
Read only property	pfControlAvail()	Integer	Returns a value which specifies whether the object can be controlled by the client programmer (This is dependent on the pfVisible and pfEnabled properties and those of its parent)
Read only property	pfEnabled()	Integer	Returns a value which specifies whether the object is enabled
Read only property	pftPercent()	Single	Returns a value indication the displayed percentage
Read only property	pfVisible()	Integer	Returns a value which specifies whether the object is visible
Read only property	psMessage()	String	Returns a string indicating the displayed message
Read only property	psName()	String	Returns the name of the object

Read/write property	psCaption()	String	Sets or returns the text which is used to label the associated control
Sub	pRegEvent(ByVal sEvent As String, ByVal fBasicEvent As Integer)		Should not be used by clients! (Registers an event to the error handler)

HProgress4Array

Event	ChangePercent(ByVal Index As Integer)		Event which is raised when the pfPercent property changes
Event	ChangeVisible(ByVal Index As Integer)		Event which is raised when the pfVisible property changes
Read only property	pcItem(ByVal Index As Integer)	HProgress	Returns an object reference to the specified HCommand object
Read only property	psName()	String	Returns the name of the object

HRadios

Event	ChangeEnabled()		Event which is raised when the pfEnabled property changes
Event	ChangeNoEntries()		Event which is raised when the piNoEntries property changes
Event	ChangeRadio()		Event which is raised when the piRadio property changes
Event	ChangeRadioEnabled(ByVal iRadio As Integer)		Event which is raised when the pfRadioEnabled property changes
Event	ChangeRadioVisible(ByVal iRadio As Integer)		Event which is raised when the pfRadioVisible property changes
Event	ChangeVisible()		Event which is raised when the pfVisible property changes
Read only property	pcParent()	Object	Returns a reference to the parent object of this object. If this reference is Nothing this object is the topmost object within the object hierarchy (normally the HWindow object of the associated dialog)
Read only property	pfControlAvail()	Integer	Returns a value which specifies whether the object can be controlled by the client programmer (This is dependent on the pfVisible and pfEnabled properties and those of its parent)
Read only property	pfEnabled()	Integer	Returns a value which specifies whether the object is enabled
Read only property	pfRadioEnabled(ByVal iRadio As Integer, Optional DoEvent As Variant)	Integer	Returns a value which specifies the enabled status of the indicated radiobutton
Read only property	pfRadioVisible(ByVal iRadio As Integer, Optional DoEvent As Variant)	Integer	Returns a value which specifies the enabled status if the indicated radiobutton
Read only property	pfVisible()	Integer	Returns a value which specifies whether the object is visible
Read only property	piNoEntries()	Integer	Returns the number of entries of the object
Read only property	piRadioFromValue(ByVal iValue As Integer)	Integer	Returns the radiobutton index from a specified value
Read only property	piSetValue(ByVal iRadio As Integer)	Integer	Returns the value of a specified radiobutton index (Not necessarily the current one)
Read only property	psName()	String	Returns the name of the object
Read only property	psSetKeyValue(By	String	Returns the key value of a specified radiobutton index

	Val iRadio As Integer)		(Not necessarily the current one)
Read/write property	piRadio(Optional DoEvent As Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer)	Integer	Sets or returns the index of the currenly selected radiobutton. It is numbered from (piRadio=0 to piNoEntries-1)
Read/write property	piValue(Optional DoEvent As Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer)	Integer	Returns the value which is associated with the currently selected radiobutton
Read/write property	psCaption()	String	Sets or returns the text which is used to label the associated control
Read/write property	psKeyValue(Optional DoEvent As Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer)	String	Sets or returns the key value of the object. When this value is set the radiobutton with the specified key value is selected
Sub	pRegEvent(ByVal sEvent As String, ByVal fBasicEvent As Integer)		Should not be used by clients! (Registers an event to the error handler)

HRadios4Array

Event	ChangeEnabled(ByVal Index As Integer)		Event which is raised when the pfEnabled property changes
Event	ChangeNoEntries(ByVal Index As Integer)		Event which is raised when the piNoEntries property changes
Event	ChangeRadio(ByVal Index As Integer)		
Event	ChangeRadioEnabled(ByVal iIndex As Integer, ByVal Index As Integer)		Event which is raised when the pfRadioEnabled property changes
Event	ChangeRadioVisible(ByVal iIndex As Integer, ByVal Index As Integer)		Event which is raised when the pfRadioVisible property changes
Event	ChangeVisible(ByVal Index As Integer)		Event which is raised when the pfVisible property changes
Read only property	pcItem(ByVal Index As Integer)	HRadios	Returns an object reference to the specified HRadios object
Read only property	psName()	String	Returns the name of the object

HSingleTab

Event	ChangeEnabled()		Event which is raised when the pfEnabled property changes
Event	ChangeVisible()		Event which is raised when the pfVisible property changes
Read only property	pcParent()	Object	Returns a reference to the parent object of this object. If this reference is Nothing this object is the topmost object

			within the object hierarchy (normally the HWindow object of the associated dialog)
Read only property	pfControlAvail()	Integer	Returns a value which specifies whether the object can be controlled by the client programmer (This is dependent on the pfVisible and pfEnabled properties and those of its parent)
Read only property	pfEnabled()	Integer	Returns a value which specifies whether the object is enabled
Read only property	pfVisible()	Integer	Returns a value which specifies whether the object is visible
Read only property	psName()	String	Returns the name of the object
Read/write property	psCaption()	String	Sets or returns the text which is used to label the associated control
Sub	pRegEvent(ByVal sEvent As String, ByVal fBasicEvent As Integer)		Should not be used by clients! (Registers an event to the error handler)

HTab

Event	ChangeEnabled()		Event which is raised when the pfEnabled property changes
Event	ChangeNoEntries()		Event which is raised when the piNoEntries property changes
Event	ChangeTab()		Event which is raised when the piTab property changes
Event	ChangeTabEnabled(ByVal iTab As Integer)		Event which is raised when the pfTabEnabled property changes
Event	ChangeTabVisible(ByVal iTab As Integer)		Event which is raised when the pfTabVisible property changes
Event	ChangeVisible()		Returns a value which specifies whether the object is visible
Read only property	pcParent()	Object	Returns a reference to the parent object of this object. If this reference is Nothing this object is the topmost object within the object hierarchy (normally the HWindow object of the associated dialog)
Read only property	pcSingleTab(ByVal iTab As Integer)	HSingleTab	a reference to the HSingleTab object associated to the specified Tab
Read only property	pfControlAvail()	Integer	Returns a value which specifies whether the object can be controlled by the client programmer (This is dependent on the pfVisible and pfEnabled properties and those of its parent)
Read only property	pfEnabled()	Integer	Returns a value which specifies whether the object is enabled
Read only property	pfTabEnabled(ByVal iTab As Integer, Optional DoEvent As Variant)	Integer	Returns a value which specifies whether the specified HSingleTab object is enabled
Read only property	pfTabVisible(ByVal iTab As Integer, Optional DoEvent As Variant)	Integer	Returns a value which specifies whether the specified HSingleTab object is visible
Read only property	pfVisible()	Integer	Returns a value which specifies whether the object is visible
Read only property	piNoEntries()	Integer	Returns the number of entries of the object
Read only property	piSetValue(ByVal iTab As Integer)	Integer	Returns the value of a specified option index (Not necessarily the current one)
Read only property	piTabFromValue(ByVal iValue As Integer)	Integer	Returns the Tab index from a specified value

Read only property	psCaption()	String	Sets or returns the text which is used to label the associated control
Read only property	psName()	String	Returns the name of the object
Read only property	psSetKeyValue(ByVal iTab As Integer)	String	Returns the key value of a specified Tab index (Not necessarily the current one)
Read/write property	piTab(Optional DoEvent As Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer)	Integer	Sets or returns the index of the currently selected Tab
Read/write property	piValue(Optional DoEvent As Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer)	Integer	Returns the value which is associated with the currently selected Tab
Read/write property	psKeyValue(Optional DoEvent As Variant, Optional IgnoreDisable As Variant, Optional NoError As Integer)	String	Sets or returns the key value of the object. When this value is set the Tab with the specified key value is selected
Sub	pRegEvent(ByVal sEvent As String, ByVal fBasicEvent As Integer)		Should not be used by clients! (Registers an event to the error handler)

HWindow

Event	ActivateWindow()		Event which is raised when the associated window is activated
Event	BeforeWindowVisible()		Event which is raised before the associated Window is shown on the screen
Event	ChangeEnabled()		Event which is raised when the pfEnabled property changes
Event	ChangeUserIF()		Event which is raised when the pfUserIF property changes
Event	ChangeVisible()		Event which is raised when the pfVisible property changes
Event	CloseWindow(ByRef Cancel As Integer, ByVal Mode As Integer)		Event which is raised when the window is closed
Event	KeyDownWindow(ByVal KeyCode As Integer, ByVal Shift As Integer)		Event which is raised when a key is pressed on the window
Event	MouseDown(ByVal Button As Integer, ByVal Shift As Integer, ByVal iX As Integer, ByVal iY As Integer)		Event which is raised when the mouse is pressed on the window
Event	MouseMove(ByVal Button As Integer, ByVal Shift As Integer,		Event which is raised when the mouse moves on the window

	ByVal iX As Integer, ByVal iY As Integer)		
Event	MouseUp(ByVal Button As Integer, ByVal Shift As Integer, ByVal iX As Integer, ByVal iY As Integer)		Event which is raised when the mouse is released on the window
Event	NowWindowVisible()		Event which is raised after the associated Window is shown on the screen
Event	ResizeWindow()		Event which is raised when the window is resized
Read only property	pcParent()	Object	Returns a reference to the parent object of this object. If this reference is Nothing this object is the topmost object within the object hierarchy (normally the HWindow object of the associated dialog)
Read only property	pfControlAvail()	Integer	Returns a value which specifies whether the object can be controlled by the client programmer (This is dependent on the pfVisible and pfEnabled properties and those of its parent)
Read only property	pfEnabled()	Integer	Returns a value which specifies whether the object is enabled
Read only property	pfWindowSizable()	Integer	Returns a value which defines whether the associated window is sizeable
Read only property	piClientAreaXOffset()	Integer	Returns a value which defines the offset of the windows client from the window in x-direction in pixels
Read only property	piClientAreaYOffset()	Integer	Returns a value which defines the offset of the windows client from the window in y-direction in pixels
Read only property	plHWnd()	Long	Returns the associated windows window handle. If there is no associated window this handle is zero
Read only property	psName()	String	Returns the name of the object
Read only property	psOwnerClass()	String	Returns the name of the class to which the window belongs
Read/write property	pfHideForm()	Integer	Sets or retuns a value which defines whether the associated window should be hided. the associated window is neither unloaded nor the event CloseWindow is raised. This property can be used to temporarily hide a window without unloding it
Read/write property	pfRestoreWindowPos()	Integer	Sets or returns a value which defines whether the window position, its size and window state is restored when the window is displayed again after it has been closed
Read/write property	pfUserIF()	Integer	Sets or returns a value which defines whether the associated window should be shown on screen. A window is only displayed under the following condition: pfUserIF=TRUE, pfVisible=TRUE, pfHideForm=FALSE
Read/write property	pfVisible()	Integer	Returns a value which specifies whether the object is visible
Read/write property	plColor()	Long	Sets or returns the clolor of the associated window
Read/write property	plHWndParentWindow()	Long	Returns the window handle of the parent window of this window. If this window is a standalone window this handle is zero.
Read/write property	psCaption()	String	Sets or returns the text which is used to label the associated control
Read/write property	psPicture()	String	Sets or returns the file name of a bitmap image which is displayed on this window. The followinf formats are supported: BMP, ICO, CUR, RLE, WMF, EMF, GIF, JPG
Read/write property	psTag()	String	Sets or returns the Tag property of this object. The Tag property can be used to communicate with the associated

			window (internal use only)
Function	piGetWindowPosition(ByRef iX As Integer, ByRef iY As Integer, ByRef iDX As Integer, ByRef iDY As Integer, ByRef iWindowState As Integer)	Integer	A method which returns the associated windows position, size and window state. Only valid when pfVisible=TRUE and pfUserIF=TRUE
Function	piSetWindowPosition(ByVal iX As Integer, ByVal iY As Integer, ByVal iDX As Integer, ByVal iDY As Integer, ByVal iWindowState As Integer, ByVal fCheckPosition As Integer)	Integer	A method which sets the associated windows position, size and window state. Only valid when pfVisible=TRUE and pfUserIF=TRUE
Sub	pActivateMainWindow()		Method which activates the associated window
Sub	pClose(ByRef Cancel As Integer, ByVal iMode As Integer)		Method which closes the associated window
Sub	pRegEvent(ByVal sEvent As String, ByVal fBasicEvent As Integer)		Should not be used by clients! (Registers an event to the error handler)
Sub	pSetFocus()		Method which sets the focus to the associated window. Works only if the associated window is visible on screen

HLut

Event	AutoLUT(ByRef Cancel As Integer)		Event which is raised when the user presses the AutoLUT key
Event	ChangeEnabled()		Event which is raised when the pfEnabled property changes
Event	ChangeVisible()		Event which is raised when the pfVisible property changes
Event	CursorsChanged(ByVal LowerCursor As Long, ByVal UpperCursor As Long)		Event which is raised when either of the two LUt cursors have been changed
Event	LimitsChanged(ByVal iSize As Integer)		Event which is raised when the the LUT limits changes
Event	LowerCursorChanged(ByVal LowerCursor As Long)		Event which is raised when the lower cursor changes
Event	Resize()		Event which is raised when the LUT control is resized
Event	UpperCursorChanged(ByVal UpperCursor As Long)		Event which is raised when the upper cursor changes
Read only property	pcParent()	Object	Returns a reference to the parent object of this object. If this reference is Nothing this object is the topmost object within the object hierarchy (normally the HWindow object of the associated dialog)

Read only property	pfControlAvail()	Integer	Returns a value which specifies whether the object can be controlled by the client programmer (This is dependent on the pfVisible and pfEnabled properties and those of its parent)
Read only property	pfEnabled()	Integer	Returns a value which specifies whether the object is enabled
Read only property	pfVisible()	Integer	Returns a value which specifies whether the object is visible
Read only property	piLUTSize()	Integer	Returns a value which specifies the LUTSize. Possible values are defined in the enumeration LUTSize
Read only property	plMinCursorDiff()	Long	Returns the minimum difference between lower and upper value (cursor)
Read only property	plMult()	Long	For 16x LUT types the data specified and displayed with the LUT tool have to bee multiplied by plMult to get the values valid for image data
Read only property	psName()	String	Returns the name of the object
Read/write property	piAutoOption()	Integer	
Read/write property	plLowerValue()	Long	Sets or returns a value which defines the lower value (lower cursor) of the object
Read/write property	plUpperValue()	Long	Sets or returns a value which defines theupper value (upper cursor) of the object
Read/write property	psCaption()	String	Sets or returns the text which is used to label the associated control
Sub	pRegEvent(ByVal sEvent As String, ByVal fBasicEvent As Integer)		Should not be used by clients! (Registers an event to the error handler)
Sub	pSetAuto()		Method which executes the Auto LUT function
Sub	pSetCursors(ByVal lLowerValue As Long, ByVal lUpperValue As Long)		Method which sets both values, upper and lower value (cursor)

HImageArea

Event	ChangeCenter()		Changes the zoom center of the image
Event	ChangeEnabled()		Event which is raised when the pfEnabled property changes
Event	ChangeVisible()		Event which is raised when the pfVisible property changes
Event	ChangeZoom()		Event which is raised when the zoom factor of the image changes
Read only property	pcParent()	Object	Returns a reference to the parent object of this object. If this reference is Nothing this object is the topmost object within the object hierarchy (normally the HWindow object of the associated dialog)
Read only property	pfControlAvail()	Integer	Returns a value which specifies whether the object can be controlled by the client programmer (This is dependent on the pfVisible and pfEnabled properties and those of its parent)
Read only property	pfEnabled()	Integer	Returns a value which specifies whether the object is enabled
Read only property	pfVisible()	Integer	Returns a value which specifies whether the object is visible
Read only property	piPRFDirection()	Integer	Sets or returns a value which defines the quick profile direction. Possible values are HORINTEGRPROFILE and VERINTEGRPROFILE
Read only property	plHWndImageArea()	Long	Window handle of the window where the image is displayed
Read only property	psName()	String	Returns the name of the object
Read/write property	pftZoom()	Single	Sets or returns the current zooming factor of the image

Read/write property	piROIType()	Integer	Sets or returns a value which defines the ROI Type. Possible values are defined in the enumeration ROIType
Read/write property	psCaption()	String	Sets or returns the text which is used to label the associated control
Sub	pEnlargeROI(ByVal fDirection As Integer)		Enlarges the ROi in the specified direction. Possible values are defined in the enumeration EnlargeDirection
Sub	pGetareImgToDisplay(ByRef iX As Integer, ByRef iY As Integer, ByRef iDX As Integer, ByRef iDY As Integer)		Returns the area of the displayed image
Sub	pGetareROI(ByRef iX As Integer, ByRef iY As Integer, ByRef iDX As Integer, ByRef iDY As Integer)		Returns the ROI area
Sub	pGetftpntCenter(ByRef X As Single, ByRef Y As Single)		Returns the zooming center point
Sub	pGetpntPicWindow(ByRef X As Integer, ByRef Y As Integer)		Returns the size of the window which is used to display the image
Sub	pGetpntWindow(ByRef X As Integer, ByRef Y As Integer)		Returns the size of the window which is used to display the image including scroll bars and empty area (if any)
Sub	pGetwndROI(ByRef Bottom As Integer, ByRef Left As Integer, ByRef Right As Integer, ByRef Top As Integer)		Returns the window size and location which corresponds the the ROI
Sub	pLetareROI(ByVal iX As Integer, ByVal iY As Integer, ByVal iDX As Integer, ByVal iDY As Integer)		Sets the ROI area
Sub	pLetftpntCenter(ByVal X As Single, ByVal Y As Single)		Sets the zooming center point
Sub	pRegEvent(ByVal sEvent As String, ByVal fBasicEvent As Integer)		Should not be used by clients! (Registers an event to the error handler)

HEditString4Array

Event	ChangeEnabled(ByVal Index As Integer)		Event which is raised when the pfEnabled property changes
Event	ChangeText(ByVal Index As Integer)		Event which is raised when the psText property changes

Event	ChangeVisible(ByVal Index As Integer)		Event which is raised when the pfVisible property changes
Read only property	pcItem(ByVal Index As Integer)	HEditString	Returns an object reference to the specified HCommand object
Read only property	psName()	String	Returns the name of the object

HWindow20Array

Event	ActivateWindow(ByVal Index As Integer)		Event which is raised when a window is activated
Event	BeforeWindowVisible(ByVal Index As Integer)		Event which is raised before window is visible
Event	ChangeEnabled(ByVal Index As Integer)		Event which is raised when the pfEnable property changes
Event	ChangeUserIF(ByVal Index As Integer)		Event which is raised when the pfUserIF property changes
Event	ChangeVisible(ByVal Index As Integer)		Event which is raised when the pfVisible property changes
Event	CloseWindow(ByVal Index As Integer, ByRef Cancel As Integer, ByVal Mode As Integer)		Event which is raised when the Window is closed
Event	KeyDownWindow(ByVal Index As Integer, ByVal KeyCode As Integer, ByVal Shift As Integer)		Event which is raised when a key is pressed on the window
Event	MouseDown(ByVal Index As Integer, ByVal Button As Integer, ByVal Shift As Integer, ByVal iX As Integer, ByVal iY As Integer)		Event which is raised when the mouse is pressed on the window
Event	MouseMove(ByVal Index As Integer, ByVal Button As Integer, ByVal Shift As Integer, ByVal iX As Integer, ByVal iY As Integer)		Event which is raised when the mouse is moved on the window
Event	MouseUp(ByVal Index As Integer, ByVal Button As Integer, ByVal Shift As Integer, ByVal iX As Integer, ByVal iY As Integer)		Event which is raised when the mouse is released on the window
Event	NowWindowVisible(ByVal Index As Integer)		Event which is raised when the window is getting visible

Event	ResizeWindow(ByVal Index As Integer)		Event which is raised when the window is resized
Read only property	pcItem(ByVal Index As Integer)	HWindow	Returns an object reference to the specified HWindow object
Read only property	psName()	String	Returns the name of the object

IHCall2Appl

Function	lGetMainColor()	Long	Returns the applications main color
Function	lhWndClientWindow()	Long	Returns the handle to the applications client area window
Function	lhWndMainWindow()	Long	Returns the handle to the applications main window
Sub	AutoActionAfterAcq()		Executes auto action after acquisition
Sub	AutoActionBeforeAcq(ByVal fDisableAutoInquiry As Integer)		Executes auto action before acquisition
Sub	DisablePanelMessage(ByVal fDisablePanelMessage As Integer)		Enables or disables the panel message on the applications window
Sub	EnableMainWindow(ByVal fEnabled As Integer)		Enables or disables the applications main window
Sub	EnableSequenceControls(ByVal fEnable As Integer)		Enables or disables the sequence controls on the applications window
Sub	GetImageScaling(ByVal iImg As Integer)		Assigns image scaling information to the current image according the current system scaling
Sub	GetImageStatus(ByVal iImg As Integer, ByVal fIncludeCameraStatus As Integer)		Assigns image status information to the current image according the current status
Sub	SetMainColor(ByVal lMainColor As Long)		Sets the applications main window color

IHCall2Cam

Function	CameraHasMechanicShutter()	Integer	Returns an flag which defines whether the camera has a mechanical shutter
Function	fAcqCanBeStopped()	Integer	
Function	iAcqModeForPCTest()	Integer	
Sub	AbortCamera()		Aborts the current camera image acquisiton
Sub	AbortCameraSingleExp()		Aborts a single exposure of the camera
Sub	AboveThresholdCalculated(ByVal iAcqMode As Integer, ByVal sString As String)		Indicates that the value above threshold has been calculated
Sub	CameraSuitableForBacksub()		Returns an flag which defines whether the camera is suitable for background subtraction
Sub	ChangeAcqMode(ByVal iAcqMode As Integer)		Indicates that the current AcqMode has changed

Sub	EnableCameraEndResponse(ByVal fEnable As Integer)		Enables or disables the cameras end response
Sub	ExternalTriggerElapsed(ByVal iAcqMode As Integer, ByVal lElapsed As Long)		Indicates that an external trigger ha s been elapsed
Sub	GetStopStatus(ByRef ftExpTime As Single, ByRef iTrigger As Integer, ByRef iExpStatus As Integer)		Gets the stop status of the camera
Sub	IndicateProgress(ByVal iAcqMode As Integer, ByVal ftPercent As Single, ByVal sOut As String)		Indicates the progress of an acquisition
Sub	Message(ByVal iAcqMode As Integer, ByVal sMessage As String)		Indicates a message
Sub	RestoreAcqControls(ByVal iAcqMode As Integer)		Restore the live mode controls to its initial values (Which had been present before SetLiveControlsToPC)
Sub	SetAcqControlsToPC(ByVal iAcqMode As Integer)		Sets the live mode controls to the values which exists in photon counting mode
Sub	SetBackgroundParms(ByVal iSet As Integer, ByVal sCameraSection As String, ByVal lNrExposures As Long, ByVal ftExposureTime As Single, ByVal fDoCloseMechanicShutter As Integer)		Set the camera to values which are used for background subtraction
Sub	SetCamDlgVisible(ByRef iPrevState As Integer)		Sets the camera dialog to visible
Sub	SetClearFBOnStart(ByVal fClearFBOnStart As Integer, ByVal iAcqMode As Integer)		
Sub	StartCamera(ByVal iMode As Integer, ByVal fWait As Integer)		Starts the camera
Sub	StopCamera(ByVal fWait As Integer)		Stops the camera

Sub	SwitchOffRTBS()		Switched off real time backsub
Sub	SynchronizeCamera(ByVal ftExposureTime As Single)		Synchronize the camera. As a result a new image should be started now
Sub	UpdateCameraParms()		Informs the HAcq object about the current camera parameters relevant for acquisition

IHCall2Exttrig

Sub	DoExtExposure(ByVal ftExpTime As Single)		Executes an extrenal exposure
Sub	EndTrig()		Ends a trigger
Sub	SetupExtTrig(ByVal iMode As Integer, ByVal fPolarityPositiv As Integer, ByVal ftInterval As Single)		Setup external triggering
Sub	StartTrig()		Starts an extrnal trigger

IHCall2Licence

Function	fLicenceForAcquisition()	Integer	Returns a flag which defines whether a licence for acquisiton is present
Function	fLicenceForSave()	Integer	Returns a flag which defines whether a licence for Save is present

IHCall2UserFunc

Sub	UserFunction(ByVal iIndex As Integer, ByVal fAppendOutString As Integer, ByVal fGetMemoryProperties As Integer, Optional fStop As Variant, Optional sOutBack As Variant)		Executes user function

IHCall2LUTControl

Sub	GetLUTSizeAndColor(ByRef iLutSize As Integer, ByRef iLutColor As Integer)		Returns the LUT size and color
Sub	GetLUTValues(ByRef lLower As Long, ByRef lUpper As Long, ByRef lMult As Long)		Returns the LUT values
Sub	SetAuto()		Executes Auto LUT
Sub	SetHistogram(ByRef lHistogram()	Long, ByVal iHistogramSize As Integer, ByVal iHistogramBinsize As	Sets the histogram

		Integer, ByVal iHistogramDivideFactor As Integer, ByVal lNrHistogramPixels As Long)	
Sub	SetZeroLower(ByVal fZeroLower)		
Sub	UpdateLUTFromImage(ByVal iImg As Integer, ByVal fDisplay As Integer)		Sets the LUT parameters valid in the current image

IHCall2Sequence

Function	fPSeqGetMask(ByVal sFile As String, ByRef sMask As String, ByRef lNr As Long, ByRef iPrf As Integer)	Integer	Gets the mask string for the specified filename
Function	fSeqDisplaySamplePlus1()	Integer	Displays the previous sample
Function	iCorrDirection()	Integer	Returns the correction direction
Function	iProfileNo()	Integer	Returns the profile number
Function	iSeqProcessing()	Integer	Returns the sequence processing mode
Function	SeqGetCurrentSample()	Long	Returns the current sample
Sub	CheckAutoFixpoint(ByVal fAutoFixpoint As Integer)		Sets the checkbox value for Auto Fixpoint
Sub	CheckExcludeSample(ByVal fExcludeSample As Integer)		Sets the checkbox value for Exclude Sample
Sub	PSeqCheckAndLoad1stProfile(ByVal sFullName As String)		Executes check and load first profile
Sub	SeqDisplayFirstSample()		Displays the first sample
Sub	SeqDisplaySample(ByVal lNr As Long)		Displays the specified sample
Sub	SeqSingleAcqEnded(lC As Long)		
Sub	SetProfileAnalysis()		Shows the profile analysis window
Sub	ValueExcludedSamples(ByVal lExcludedSamples As Long)		Returns the number of excluded samples
Sub	ValueFixPoint(ByVal ftFixPoint As Single)		Returns the number of fix points
Sub	ValueMeanProcess		Returns the mean value calculated during processing

	ing(ByVal ftMean As Single)		
Sub	ValueSDProcessing(ByVal ftSD As Single)		Returns the standard deviation calculated during processing
Sub	ValueUndefinedSamples(ByVal lUndefinedSamples As Long)		Returns the number of undefined samples
Sub	ValueValidSamples(ByVal lValidSamples As Long)		Returns the number if valid samples

IHCall2AcqControl

Function	iAcqControl()	Integer	Returns a flag which defines the current acquisiton control. Possible values are defined in the enumeration AcqControl
Sub	SetAcqControl(ByVal iAcqControl As Integer)		Sets a flag which defines the current acquisiton control. Possible values are defined in the enumeration AcqControl

IhCall2ExtDevs

Function	fSystemCanCloseAutomatically(ByRef sDevicesClosed As String)	Integer	Returns a value which defines whether the system can be closed automatically
Function	iPostTriggerTime()	Integer	Returns the post trigger time
Function	TriggerSetup(ByVal lTriggerCount As Long)	Integer	Executes trigger setup
Function	TriggerStatus(ByRef fFinished As Integer, ByRef lCountsExecuted As Long)	Integer	Returns the trigger status
Function	TriggerStop()	Integer	Stops the current triggering
Sub	CloseSystem(ByVal iClose As Integer)		Closes the whole system. This can include mechnical shutter spectrograph slit and MCP gain

IhCall2Shutdown

Sub	ShutDown()		Executes shutdown of the system

IHCall2TransAbs

Sub	CloseShutters()		Closes the shutters
Sub	OpenShutters()		Opens the shutters

IHCall2MC

Function	SetMC(ByVal iMC As Integer)	Integer	

HProfileScalingData

Read only property	pftScale()	Single	Returns the scaling factor
Read only property	pftValue(iEntry As Integer, Optional sError As Variant)	Single	Returns the value of a given index
Read only property	piType()	Integer	Returns the type (linear/table)
Read only property	psOrigin()	String	Returns the scaling origin
Read only property	psScalingFile()	String	Returns the scaling file name
Read only property	psUnit()	String	Returns the unit
Function	gfCheckSwapScaling()	Integer	Swaps the scaling and profile data in case the scaling is descending
Function	pfCreateDiffPoyno	Single,	Create differential polynomial

	mial(ByVal iOrder As Integer, ByRef ftCoeff()	ByVal iNrValidChannels As Integer, ByVal sUnit As String, Optional sError As Variant) As Integer	
Function	pfCreateIntegPoynomial(ByVal iOrder As Integer, ByRef ftCoeff()	Single, ByVal iNrValidChannels As Integer, ByVal sUnit As String, Optional sError As Variant) As Integer	Create integral polynomial
Function	pfGetArrayOfValues(ByRef ftArray()	Single, ByVal iEntries As Integer, Optional sError As Variant) As Integer	Returns an array of values
Function	pfReadScalingTable(ByVal sFileName As String, ByVal sOffset As String, ByVal fCheck As Integer, ByVal sUnit As String, Optional sError As Variant, Optional varCheckFromChannel As Variant)	Integer	Reads a scaling table from file
Function	pfSetLinearData(ByVal ftScale As Single, ByVal sUnit As String, Optional sError As Variant)	Integer	Sets linear scaling
Function	pfSetScaling(ByRef psd As HProfileScalingData, Optional sError As Variant)		Sets the scaling data of this object identical to the data of the specified object
Function	pfSetTableData(ByRef ftTable()	Single, ByVal iEntries As	Sets table scaling

		Integer, ByVal sUnit As String, Optional sError As Variant, Optional varCheckFromChannel As Variant) As Integer	
Function	pftProfileLocationFT(ByVal fScaled As Integer, ByVal ft As Single)	Single	Returns the profile location (floating point)
Function	pfValueExist(ByVal iEntry As Integer)	Integer	Returns a flag whether or not the value at the specified location exist
Function	piMaxEntries()	Integer	Returns the maximum number of entries
Function	pSaveScalingFile(ByVal sFile As String, Optional sError As Variant)	Integer	Saves the scaling data to file
Function	psScalingInfo()	String	Returns general scaling info
Sub	pGetMinAndMax(ByVal iLower As Integer, ByVal iUpper As Integer, ByRef ftMin As Single, ByRef ftMax As Single)		gets minimum and maximum value
Sub	pLocationToIndex(ByVal ftLocation As Single, ByRef ftIndex As Single, ByVal fDisplayScaled As Integer)		Returns the index corresponding to the given location

HSystemScalingData

Read only property	X()	HProfileScalingData	X Scaling
Read only property	Y()	HProfileScalingData	Y Scaling

HImageScalingData

Read only property	X()	HProfileScalingData	X Scaling
Read only property	Y()	HProfileScalingData	Y Scaling
Function	pfSetScaling(isd As HImageScalingData, Optional sError As Variant)	Integer	Sets all scaling data of this object to the values of the specified object

HPRFParametersData

Read only property	pfDisplay()	Integer	Defines whether the specified profile is displayed
Read only property	pfExist()	Integer	Defines whether the specified profile exists
Read only property	pftMinGreaterZero()	Single	Returns the smalles value > 0
Read only property	piNrData()	Integer	Returns the number of data
Read only property	piType()	Integer	Returns the type of a profile
Read only property	plColor()	Long	Returns the color of a profile
Read only property	ppsdScaling()	HProfileScalingData	Returns the associated HProfileScalingData
Read only property	psImgStatusShort()	String	Returns the associated status
Read/write property	pftData(ByVal iData As Integer, Optional sError As Variant)	Single	Returns a single data value
Function	pfGetArrayOfData(ByRef ftData()	Single, ByVal iNrData As Integer, Optional sError As Variant) As Integer	Returns an array of data
Function	pfLoadPRF(ByRef sCompleteFile As String, ByVal fInquireFileName As Integer, ByRef sMessage As String, ByVal fReadData As Integer)	Integer	Loads a profile from file
Function	pfSearchValue(ByVal fScaled As Integer, ByVal iStartIndex As Integer, ByVal iEndIndex As Integer, ByVal iStep As Integer, ByVal ftSearchVal As Single, ByRef ftScaledValue As Single)	Integer	Searches a value within a profile
Function	pftProfileLocationFT(ByVal fScaled As Integer, ByVal ft As Single)	Single	Returns the location within a profile (floating point)
Function	pftProfileLocationINT(ByVal fScaled As Integer, ByVal i As Integer)	Single	Returns the location within a profile (integer)
Sub	pCalcProfileDef(ByVal fDisplayScaled As Integer)		Calculate profile default values
Sub	pCheckSwapProfile()		Check whether the profile and the scaling data has to be swapped
Sub	pFitGetPDWDefau		Get the profile display window

	lt(ByVal fAbs As Integer)		
Sub	pfSetHPRFParametersData(ByRef ppd As HPRFParametersData)		Set all data associated with this profile to identical to the specified profile
Sub	pGetMinAndMax(ByVal iLower As Integer, ByVal iUpper As Integer, ByRef ftMin As Single, ByRef ftMax As Single)		Returns minimum and maximum values
Sub	pGetSingleMinMaxPos(ByVal iFirst As Integer, ByVal iLast As Integer, ByRef iMinPos As Integer, ByRef iMaxPos As Integer)		get the minimum and maximum positions
Sub	pProfileLocationToValue(ByVal ftLocation As Single, ByRef iIndex As Integer, ByRef ftIndex As Single, ByRef ftValue As Single, ByVal fDisplayScaled As Integer)		Returns the Value at a specified location

Sample programs

The following samples are instructive programs to show how the HiPic/HPD-TA remote control can be operated in principle. One intention of these sample programs was to provide a running client program performing some demonstrating tasks from the beginning. When creating these samples there was no intention to solve any individual tasks optimal. The client programmer is encouraged to improve the functions according to his needs. Many of the functions do not function under all conditions. The samples are not intended to "replace" the full functionality of the standalone products. Any client program made for a real application may add smaller or larger portions of code and modify existing functions. It may be recommendable after using the client program to learn how to program the component to start a new project and writing every line newly. The samples are written in Visual Basic 6.0. Though client programs can be written in every language it is especially easy to write them in Visual Basic and it is recommended to do so at least when learning how to operate the component.
A feature described in one sample program which appears in another sample is not described a second time. Therefore the client programmer should read the explanation of all samples to understand the meaning of all features.

The most common case is to use an ActiveX-DLL. Therefore the first two samples show how to use the components for the HiPic and the HPD-TA. The project names of both client programs are CltHiDll and CltTaDll.

Before starting the client program it is recommended to start the standalone program and set all parameters correctly to run it (Especially the frame grabber and camera should be specified correctly). Do not run standalone program and client program at the same time (To do so special knowledge is necessary). The sample programs are made in a way so that they use the standard ini file where all these settings are stored.

If you use ActiveX components the working directory of the client program is important to find other DLLs. Most of the DLLs the component needs however are located in the Components application directory (This is the directory where the HiPic32u.exe, HiPic32u.dll or HPDTA32u.exe, HPDTA32u.dll are located). To avoid any problem it is recommended to copy the sample code into this directory. Also the working directory for the Visual Basic 6.0 should be changed to this directory. Then the program will also run in debug mode. Alternatively you can use the PATH statement to make sure that all DLLs can be located correctly.

Sample program CltHiDll and CltTaDll

CltHiDll and CltTaDll are sample programs showing how to use the functionality of the HiPic or HPD-TA within another application running on the same computer with the freedom of using or not using the User I/F provided by the HiPic/HPD-TA.
The sample programs CltHiDll and CltTaDll are using the ActiveX-DLL components of the HiPic or the HPD-TA. They can be run as soon as the HiPic or HPD-TA is installed (Version 6.3 is necessary). The installation registers the ActiveX–DLL into the system registry. If -for some reason- this has not been done properly you can do this with the regsvr32.exe utility which should exist in you systems directory but which is also provided with the HiPic/HPDTA setup for savety.

Basic structure of CltHiDll and CltTaDll

The sample programs CltHiDll and CltTaDll contain two forms, one standard module and two class modules. The forms are used during initialization (FormInit) and program execution (FormMain). The standard module contains the main routine Sub Main which start the component and sets up the client program. The class module HiInit or TaInit handles the operation of startup, the class modules HiMain or TaMain handles the execution of code during program operation.

Startup

The startup is done in the routine Sub Main

```
Sub Main()
    Dim HiInit As HiInit
    Dim fSuccess As Integer
    Dim sError As String

    Set HiInit = New HiInit
    HiInit.pfClientEvent = True
    Set gHAppHiPic = HiInit.pcStartHiPic(fSuccess, sError)
    HiInit.pfClientEvent = False

    If (Not fSuccess) Then
        MsgBox "Error loading the ActiveX component HiPic32u. " + sError
        Set HiInit = Nothing
        Exit Sub
    End If
    Set HiInit = Nothing

    Set gHiMain = New HiMain
    gHiMain.pfClientEvent = True
    gHiMain.Init gHAppHiPic
    gHiMain.pfClientEvent = False

End Sub
```

The Routine Sub Main creates an object of type HiInit (or TaInit) (see the respective class modules) and calls the function pcStartHiPic (or pcStartHPDTA). In case of success an object of type HiMain (or TaMain) (see again the respective class modules) is created and its Init method is called.
The function pcStartHiPic (or pcStartHPDTA) does the following:

```
Public Function pcStartHiPic(fSuccess As Integer, sError As String) As HAppHiPic
    Dim iError As Integer

    FormInit.Show

    'Create Init object
    Set mHInitHi = New HInitHi
    If (mHInitHi.piStartStatus <> StartStatusNone) Then
        sError = "Component is already in use. Aborted"
        Set mHInitHi = Nothing
        fSuccess = False
        Exit Function
    End If

    'Show init screen virtually: default ini-file, no init UseI/F, no application User I/F, No
Dialogs
    'By specifying an ini file other than "" the client program can initialize the program
with its own data
mHInitHi.piInit "", False, False, True

    'Specify your initializing setting here (in case you didn't do this by your specific ini
file in the last command)
    'The next three lines would set AMD-DIG + VS module and Analog0 camera
    'mHInitTa.pHRadInitFrameGrabber.piValue = IniGrabberIcPciAmVS
    'mHInitTa.pHRadInitCamera.piValue = ANALOG0
```

```
    'mHInitTa.pHEstInitConfiFile.psText="c:\ccir.cnf"

    'Start program
    mHInitHi.pHComInitOK.pClick

    'Wait until application created or failed
    Do
If (mHInitHi.pfInitStatus >= 0) Then Exit Do
        mfClientEvent = False
        DoEvents
        mfClientEvent = True
    Loop While True

    'Return success status and Application reference
    fSuccess = (mHInitHi.pfInitStatus = InitStatusInitialized)
    If (fSuccess) Then
        Set pcStartHiPic = mHInitHi.pHAppHipic
    Else
        sError = msError
    End If
    'Release object reference to HInitHi
    Set mHInitHi = Nothing

    Unload FormInit

End Function
```

The first thing the pcStartHiPic (or pcStartHPDTA) does is to create an object of the type HInitHi (or HInitTa). See:

```
Set mHInitHi = New HInitHi
```

This is the only object of the HiPic or HPD-TA ActiveX component can be created by the client. All other objects are created inside the component and only an object reference is returned as a public reference. This object returns information whether the is already running, whether it is currently in the state of being initialized or whether it is newly created now. See

```
If (mHInitHi.piStartStatus <> StartStatusNone) Then
...
```

Most probably the client program does not like the fact that the program is already running and will exit in such case. In more complicated cases one can program several client programs using the same component. It is not possible to run several instances of the component (lets say with different hardware). The next step is to call

```
mHInitHi.piInit
```

This is equivalent to showing the init dialog on the computers screen (If the parameter fInitUserIF is set to true this really happens). The parameters specified here defines the general behavior of the program. The first parameter specifies the ini file which should be used for data save and restore. If one specifies a null string ("") the default ini file is used (Currently HiPic32r.ini and HPDTA32 r.ini in the windows directory). Therefore parameters set by the standalone application are used by the client program and vice versa (This is recommended for the beginning). Later on you can specify your own ini file which you have to provide by yourself (The easiest way to do so is to start the standalone application, set all parameters, quit it and copy the default ini file to the desired filename). The next two parameters define whether the init dialog and the dialogs of the main program should be shown per default. The last parameter defined whether dialogs should be omitted or not. This parameter should be set to false generally. During development it is recommended to set the parameters fInitUserIF and fApplUserIF to TRUE. Then the dialogs appear on screen and you

can check whether everything you program is executed correctly. After some time you may want to hide the dialogs by setting fInitUserIF and fApplUserIF to FALSE.
To Start the program you just have to click on the init dialogs OK pushbutton. Within a client program this is done by writing:

```
mHInitHi.pHComInitOK.pClick
```

As the startup is an Asynchronous Command we have to wait until it is finished (either succesful or not). This is done by the loop:

```
    Do
          If (mHInitHi.pfInitStatus >= 0) Then Exit Do DoEvents
    Loop While True
```

Finally the property

```
mHInitHi.pHAppHipic
```

contains a reference to the application object of the HiPic/HPDTA ActiveX component. Pls. note that this object has been created by the HInitHi (or HInitTa) object and not by the client program.

Initializing of the client program

After the component has been started successfully the client program can be initialized. This is performed in the Init routine of the HiMain object (which has been created previously by the client program). The HiMain object contains many references to objects of the component. One example is the Open menu object. The reference is made by writing:

```
    Set mHMenMainOpen = mHAppHiPic.pHMenMainOpen
```

another is the cameras live mode pushbutton object:

```
    Set mHComCamLive = mHAppHiPic.pobjHCamera.pHCo4CamLive.pcItem(LIVE)
```

To place a dialog into a window within the client program the plHWndParentWindow is used. In the following code all image windows dialogs are placed as a child window within the PicClientArea picture box of the client program.

```
    mHAppHPDTA.pHImages.pfUserIF = True
    For iC = 0 To mHAppHPDTA.pHImages.piItemsMax - 1
        mHAppHPDTA.pHImages.pcItem(iC).pHWinImgDlg.plHWndParentWindow =
FormMain.PicClientArea.hWnd
    Next iC
```

Depending on the configuration sometimes an object property returns NOTHING or does not exist at all. The function mfExistSubObject returns an information whether a depend object exist for a given object. In our case this is used to find out whether the camera object has an exposure time property. See

```
        mfExposureExist = mfExistSubObject(mHAppHPDTA.pobjHCamera,
"pHEn4CamExposure")
```

Ending the component

The component can be ended easily by
1.) Freeing all object references except to the application object
2.) Executing

`mHAppHiPic.piEndProg (or mHAppHPDTA.piEndProg)`

3.) Releasing the reference to the application object

Executing and availability of commands

Executing of a command is simply done by calling its pClick method. See:

```
mHComCamLive.pClick
```

This executes Live mode. Under certain circumstances a command (or any other control) is not available (this is equivalent to disabled or invisible control within the standalone program). To be sure whether a control is available the sample program calls the mfCheckControlAvail function which uses the pfControlAvail property of the object.

As an example we quote the DoLive function:

```
Public Sub DoLive()

    miAcqMode = LIVE

    'Change to correct TAB
    If Not mfCheckControlAvail(mHAppHiPic.pobjHCamera.pHTabCamAcqMode, "Camera acquisition
tab") Then Exit Sub
    mHAppHiPic.pobjHCamera.pHTabCamAcqMode.piValue = LIVE

    'Set exposure time to LIVE mode because our sample handles only one exposure time
    If (mfExposureExist) Then
        If Not mfCheckControlAvail(mHAppHiPic.pobjHCamera.pHEn4CamExposure.pcItem(LIVE), "Live
mode exposure control") Then Exit Sub
        mHAppHiPic.pobjHCamera.pHEn4CamExposure.pcItem(LIVE).psKeyValue =
FormMain.EditExposureTime.Text
    End If

    'Disable other acquisition commands which are not automatically disabled by the component
events
    FormMain.CommandAcquire.Enabled = False
    FormMain.CommandSeqStart.Enabled = False

    If Not mfCheckControlAvail(mHComCamLive, "Camera live mode button") Then Exit Sub
    mHComCamLive.pClick

    'Note: The mHComCamLive.pClick should be the last command before exiting the current event
    '      You should not wait for some event in a loop here (Pls. read the comment on
AsyncCommands in the programmers handbook for details)!

End Sub
```

This function first select the proper Tab on the camera dialog, sets the exposure time (if available) and executes the live mode. All accessed control are checked upon their availability prior to executing them.

Setting the exposure time

The exposure time can be set by a HEntry type object. The exposure time simply can be set by its psKeyValue property. In this case the HEntry object understands and evaluates time units

(like "ms", "s" or "m" which stands for miliseconds, seconds or minutes) and sets the nearest possible value (Normally cameras only accepts distince values). In such cases is is recommended to read back the real value like in the following code example:

```
mHAppHiPic.pobjHCamera.pHEn4CamExposure.pcItem(miAcqMode).psKeyValue =
FormMain.EditExposureTime.Text
FormMain.EditExposureTime.Text =
mHAppHiPic.pobjHCamera.pHEn4CamExposure.pcItem(miAcqMode).psKeyValue
```

Using events

If an object reference is declared with the WithEvents qualifier the client program can get events. See the following declaration:

```
Private WithEvents mHComCamAcquire As HCommand
```

One of the events is the

```
Private Sub mHComCamAcquire_ChangeEnabled()
    FormMain.CommandAcquire.Enabled = mHComCamAcquire.pfEnabled
End Sub
```

This event is used to disable the acquire pushbutton on the clients main window (Be careful: a control is available if the control and all its parent controls are visible and enabled. Therefore there might be a case that the parent control gets disabled but the control itself has not raised the Change_Enabled event).

Saving and loading images specifying the filename and type

When a client program wants to save or load images by specifying the filename and its fileype the functions pfSaveImage and pfLoadImage of the HImage Object can be used. The piDefaultImageType property of the HImages object can be used to specify the file type in case of file save (see mSaveImages routine).

Executing image acquisitions and image analysis in one function

Sometimes the client program wants to acquire image data and analyze it then acquire new data change some settings and so on. To perform such kind of operations the sample function mDoMeasurement is provided.

The function does

1.) Start image acquisition. The code is same as for other acquisition functions already explained
2.) Wait for the completion of the acquisition. Be careful: Basically the component needs all the time to do the acquisition process here. Do not place any other code in such waiting loop. Especially do not output message boxes or other user interaction. If you want to stop the action you can do this from another event (like a timer or pushbutton event).
3) Check whether the acquisition operation has been succeeded.
4) Get Image, Scaling or profile data and analyze it. The function pGetareSource gets the valid area of the image. The pfGetImageData2Byte function gets 2 Byte data (only useful of the image contains 2 byte data, otherwise you have to use pfGetImageData1Byte or

pfGetImageData4Byte). pfGetArrayOfValues gets scaling data. pfMemGetPRFData gets profile data.

The above can be repeated and parameters like exposure time can be changed at the end of each cycle (this would be item 5.).

Using the HExternalDevices object (HPD-TA only)

The HExternalDevices objects controls the behaviour of all general features of the connected streak camera and related devices. It contains all dialogs related to these dialogs and User I/F objects belonging to these dialogs. It also contains general features like GP-IB base address of the GB-IB board and such general properties. As these properties are defined in the INI file the sample program does not access any of these properties within HExternalDevices (Except the object references to the individual ExternalDevice objects).

Using the HExternalDevice object (HPD-TA only)

The HExternalDevice object contains all features of the individual device. To get a list of all parameters associated to the device the following code is used:

```
iNoOfControl = mHExternalDevices.pcItem(miDevice).piNoOfControl
For iC = 0 To iNoOfControl - 1
      FormMain.ComboParameter.AddItem
mHExternalDevices.pcItem(miDevice).pHDevPars.pcItem(iC).psParameterName
Next iC
```

Using the HDevPar object (HPD-TA only)

To get the value of an individual Parameter (HDevPar object) the

```
mHDevPar.psKeyValue
```

can be used. It returns the value in string format (see the GetDevParValue function).

The HDevPar object has different ways of containing data (e.g. a list of entries or an integer number or a floating point number). The sample program uses a simplified way of displaying all these features. It distinguishes between

miParameterStyle = ParameterStyleDisp (Values which can be displayed only)
miParameterStyle = ParameterStyleCombo (List values) and
miParameterStyle = ParameterStyleEdit (Numerical values which can be entered)

Depending on this style a different code is executed.

Sample program CltHiExe and CltTaExe

The sample programs CltHiExe and CltTaExe are used to control the features of the HiPic or HPD-TA from another application. The HiPic or HPD-TA is running in a different address space on the same or another computer. If the HiPic or HPD-TA is running on a different computer the user I/F provided by the HiPic/HPD-TA cannot be used on the local computer (because the HiPic/HPD-TA can only output dialog windows on the remote computer). All user I/F and other features which access the local computer (like file save on the local computer) must be programmed by the client programmer.

Image display

To display images in live mode the event

```
Private Sub mhAcq_NewLiveImage()
```

of the HAcq object is used. Other events like the mHSequence_EndSequence (end of sequence mode) can be used as well to display updated images.

The function

```
gGetAndDisplayCurrentImage
```

is transferring and displaying the current image. The methods pGetareSource (get the image size) and pfGetDisplayData (get display data) are used to get the image data. It is the clients programmers responsability to output the image. In the sample program a function provided by the Hipic/HPD-TA is used for the propose (WING4VB_DrawImage. Strictly speaking this function is not part of the developers kit but is used here for easiness). The function mResizePicImage resizes the picturebox control where to display the image to provide correct aspect ratio.

Measurement cycles

The function mDoMeasurement gives an example how to perform measurent cycles. In a first step the acquisition is started. In a next step the program waits until completion of the acquisition. The following code is performing this wait operation:

```
    mfCommandEnded = False
    Do
        mfClientEvent = False
        DoEvents
        mfClientEvent = True
    Loop Until (mfCommandEnded = True)
```

The flag mfCommandEnded is set to TRUE in an event raised by the HAsyncCommand object. See the following code:

```
Private Sub mHAsyncCommand_AfterCommand(sCommand As String)
    mfCommandEnded = True
End Sub
```

The function DoSaveAs gets the data of the current image and saves it in ITEX format. This function is given as an example to show how image data is saved generally. It is assumed that the user wants to use his own file formats under this circumstance.

The function mfGetFileName uses the common dialog to get the file name where to save image file. A function for file open is not provided. The client programmer could realize such function for training purposes.

Sample program CltHiExe2 and CltTaExe2

The sample program CltHiExe2 and CltTaExe2 is using two components: One is running on the remote computer and is acquiring data another is running on the local computer and is used for display and handling images. The component on the local computer is an ActiveX DLL.

The handling of the local ActiveX-DLL is similar to the handling in the sample programs CltHiDll and CltTaDll.

The two components are initialized with the functions

```
pcStartRemoteHPDTA and
pcStartLocalHPDTA
```

and separate object references are held on both components (Distinguished by the ending Rem or Loc like mHInitTaRem and mHInitTaLoc for the Init objects).

Copying data and display with the local component

To display the images acquired on remote component an empty window is created, the image data is transferred and copied to the image buffer. Only one image window is used at a time. Whenever the image size changes (in x, y or pixel depth direction) the function to create an empty window is used:

```
f = mHImagesLoc.pcItem(0).pfCreateEmptyImage(iX, iY, iDX, iDY,
iBytesPerPixel, "Image", SHOWADJUSTSIZE)
```

To copy image data the functions pfGetImageData1Byte, pfGetImageData2Byte or pfGetImageData2Byte is used depending on the piBytesPerPixel property of the image. These functions are expecting an array which can be sized freely within the component. Therefore it should be dimensioned without any bounderies like:

```
Private mbData() As Byte
```

The functions to get image and display data (pfGetImageData1Byte, pfGetImageData2Byte, pfGetImageData2Byte and pfGetDisplayData) are adjusting the buffers transferred to the component correctly therefore the client program does not need to do this. As the image data is transferred with a Marshalling mechanism the data is first copied to the component, then copied back to the client program. To improve transfer speed it is an advantage to ReDim the arrays to a small size before calling such data transfer functions. In such case the data is transferred quickly to the component and only the data transfer process back to the client program uses a considerable amount of time. Therefore we find the lines:

```
ReDim mbData(0, 0) As Byte before the command
ItemCurrent.pfGetImageData1Byte ...
```

The component on the local computer can be used to easily display images, control zoom, ROI, LUT and similar things and save and load images in all formats provided by the HiPic/HPD-TA.

As it runs two components of the HiPic/HPD-TA it requires two licenses (dongles): one on every computer.

TaMonit and HiMonit

Up to now the concept of the client program always was: The Client program controls the operation of the component on the remote computer. In principle there is also another application of the remote control feature: The client program establishes a connection to the component on the remote computer to monitor certain activity but does not actively control it. The samples TaMonit and HiMonit are made for this purpose. A timer inquires every 3 seconds whether the component on the remote computer has been started. If so it pops up and displays all images displayed on the program. In this configuration normally the standalone application is started on the remote computer.

To check whether the program on the remote computer is running it is checked whether

```
mHInitTaStart.piStartStatus = StartStatusRunning
```

If this is TRUE the ActiveX-EXE on the remote computer is running. Subsequently object references are established and events are handled.

TaStart_U and HiStart_U

This sample is the smalles of all samples and has two functions:
1) It start the ActiveX-EXE component HiPic or HPD-TA by pushing the Start button. The effect is almost similar to clicking to the standalone program. The sample does not keep an object reference to the ActiceX component. This function can be used to verify whether the ActiveX-EXE has been correctly installed and registered.
2) It has a property HPDTA_HInitTa or HiPic_HInitHi which creates a new instance of the HInitHi or HInitTa object. This property can be used in the following case:
Under Windows 95 and 98 a client program cannot launch an application by DCom on another computer. It can create objects when the application is already launched but it cannot launch them. In such case the user can place the HiStartS.EXE or TaStartS.EXE in the Autostart folder which launches these small applications at Windows Startup. A client program can now create an object within TaStart_U or HiStart_U. This object can now create an object HInitHi or HInitTa even if the application is not jet launched (because TaStart_U or HiStart_U is running on the same computer). The object reference is handled to the client program and no other reference is kept to the object.

If in the previous samples the flag fUseStarter is set to true the Client program first of all creates an instance of the ClsHiStart or ClsTAStart:

```
Set mClsTAStart = CreateObject("TAStart_u.ClsTAStart", sRemoteComputer)
```

Then it gets the reference to the HInitHi or HInitTa object. During execution of the property the ActiveX-EXE application HiPic or HPD-TA is launched and a new HInitHi or HInitTa object is created.

```
Set mHInitTaStart = mClsTAStart.HPDTA_HInitTa
```

Then the ClsHiStart or ClsTAStart is destroyed:

```
Set mClsTAStart = Nothing
```

The object reference mHInitHiStart or mHInitTaStart can then be used as if the object has been created directly.