

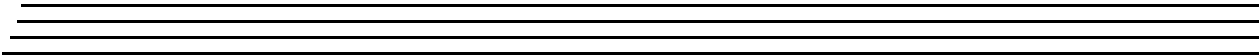
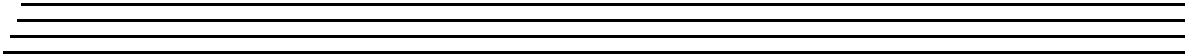
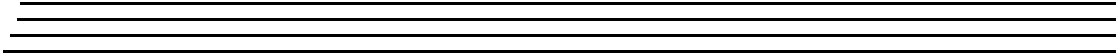
DATA TRANSLATION[®]



UM-17325-C

***DT-Active Open Layers
User's Manual***

for Imaging Boards



Third Edition
May, 2000

**Copyright © 1999, 2000 by Data Translation,
Inc.**

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form by any means, electronic, mechanical, by photocopying, recording, or otherwise, without the prior written permission of Data Translation, Inc.

Information furnished by Data Translation, Inc. is believed to be accurate and reliable; however, no responsibility is assumed by Data Translation, Inc. for its use; nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent rights of Data Translation, Inc.

Use, duplication, or disclosure by the United States Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer software clause at 48 C.F.R. 252.227-7013, or in subparagraph (c)(2) of the Commercial computer Software - Registered Rights clause at 48 C.F.R., 52-227-19 as applicable. Data Translation, Inc., 100 Locke Drive, Marlboro, MA 01752.

Data Translation, Inc.
100 Locke Drive
Marlboro, MA 01752-1192
(508) 481-3700

www.datatranslation.com
Fax: (508) 481-8620
E-mail: info@datx.com

Data Translation is a registered trademark of Data Translation, Inc. DT-Open Layers, DT-Active Open Layers, Frame Grabber SDK, MACH Series, and Sync Sentinel are trademarks of Data Translation, Inc.

All other brand and product names are trademarks or registered trademarks of their respective companies.

Table of Contents

About this Manual	vii
Intended Audience.	vii
What You Should Learn from this Manual	vii
Organization of this Manual.	viii
Conventions Used in this Manual	ix
Related Information.	ix
Where to Get Help	x
Chapter 1: Getting Started	1
What is the DT-Active Open Layers Control?	2
Quick Start.	5
System Requirements	5
Installing the Software	6
Adding the Control to Your Project	7
Visual Basic 6.0	8
Visual C++ 6.0.	8
Distributing an Application Program	9
Using the Example Programs	10
Using the Online Help	11
Removing the Software	12
Chapter 2: Summary of Properties, Methods, and Events	13
General Imaging	15
Input	16
Input Source	16
Color	17
Sync Signal	18
Clock	19

- Look-Up Table..... 19
- Video Area 19
- Trigger..... 20
- Digital Camera 21
- Frame Buffers 21
- Passthru 22
- Overlay..... 23
- Acquisition 24
- Image Processing 25
- Digital I/O..... 26
- Chapter 3: Using the Control 27**
- System Operations 29
 - Initializing a Frame Grabber Board 29
 - Defining Timeouts 30
 - Working with Camera Files..... 30
 - Working with Status Codes..... 31
 - Releasing a Frame Grabber Board 31
- Input Operations 32
 - Configuring the Input Source 32
 - Based Source Mode 32
 - Video Signal Type..... 33
 - Video Input Channel 35
 - Input Filter 35
 - Video Input Signal 37
 - Step 1. Adjust the Offset 39
 - Step 2. Apply the Gain..... 40
 - Step 3. Apply the Reference 41
 - Color Definitions 42
 - Sync Signals 45

Sync Source	46
Sync Threshold	47
Horizontal and Vertical Sync Transition	48
Sync Sentinel	48
Sync Master Mode	51
Pixel Clock.	53
Setting Up the Input Look-Up Table	55
Defining the Active Video Area	57
Horizontal Video Signal	58
Vertical Video Signal	61
Establishing a Frame Within the Active Video Area	62
Setting Up Triggers.	64
Digital Camera Operations	66
Setting the Digital Camera Type.	66
Using the Output Signals	67
Allocating a Frame Buffer	68
Passthru Operations.	72
Performing a Passthru Operation.	72
Adjusting the Source Origin	74
Modifying the Passthru LUT.	75
Taking a Snapshot.	77
Creating Overlays	78
Acquisition Operations	80
Image Processing Operations	82
Digital I/O Operations	85

Chapter 4: Programming Flowcharts	87
Frame Acquisition	89
Passthru to Display without Overlays	93
Passthru to Display with Overlays	96
Chapter 5: Product Support	117
General Checklist	118
Service and Support	119
Telephone Technical Support	119
E-Mail and Fax Support	121
World-Wide Web	121
Appendix A: Reserving System Memory	123
Index	127

About this Manual

This manual describes how to install, set up, and use the DT-Active Open Layers™ control (for imaging) to develop application programs for Data Translation PCI frame grabber boards.

Intended Audience

This manual is intended for engineers, scientists, technicians, or others responsible for developing imaging or machine vision application programs using the Microsoft® Visual Basic® or Visual C++® compiler.

It is assumed that you are a proficient programmer, that you are experienced programming in the Windows® 95, Windows 98, or Windows NT® 4.0 operating environment on the IBM® PC or compatible computer platform, that you have familiarity with imaging principles, and that you have clearly defined your requirements.

What You Should Learn from this Manual

This manual provides installation instructions for Windows 95, Windows 98, and Windows NT; summarizes the properties, methods, and events provided by the DT-Active Open Layers control; and describes how to use the properties, methods, and events to develop an imaging program. Using this manual, you should be able to successfully install the DT-Active Open Layers software and get started writing an application program.

This manual is intended to be used with the online help for the DT-Active Open Layers control. The online help contains all of the specific reference information for each of the properties, methods, events, and error codes.

Organization of this Manual

This manual is organized as follows:

- [Chapter 1, “Getting Started,”](#) describes how to install the DT-Active Open Layers control.
- [Chapter 2, “Summary of Properties, Methods, and Events,”](#) summarizes the properties, methods, and events provided with the DT-Active Open Layers control.
- [Chapter 3, “Using the Control,”](#) describes the operations you can perform using the DT-Active Open Layers control.
- [Chapter 4, “Programming Flowcharts,”](#) provides programming flowcharts for using the properties, methods, and events provided with the DT-Active Open Layers control.
- [Chapter 5, “Product Support,”](#) describes how to get help if you have trouble using the DT-Active Open Layers control.
- [Appendix A, “Reserving System Memory,”](#) lists the amount of system memory you must reserve for each of the supported device drivers.

An index completes this document.

Conventions Used in this Manual

The following conventions are used in this manual:

- Notes provide useful information that requires special emphasis, cautions provide information to help you avoid losing data or damaging your equipment, and warnings provide information to help you avoid catastrophic damage to yourself or your equipment.
- Items that you select or type are shown in **bold**. Property, method, and event names are also shown in bold.
- When navigating the screens, an instruction such as **Configure > Board** means to select “Board” from the drop-down menu under “Configure.”

Related Information

Refer to the following documentation for more information on using the DT-Active Open Layers control:

- DT-Active Open Layers online help. This contains all of the specific reference information for each of the properties, methods, events, and error codes provided by the DT-Active Open Layers control. Refer to [page 11](#) for information on how to open this help file.
- Board-specific documentation.

Some frame grabber boards are shipped with a user’s manual and a device driver manual. The user’s manual describes how to install the board, how to get started using the board, and the features of the board. The device driver manual describes how to install the device driver for the board and the capabilities supported by the device driver.

Some boards are shipped with a getting started manual and a user's manual. The getting started manual describes how to install the board, how to install the device driver for the board, and how to get started using the board. The user's manual describes the features of the board and the capabilities supported by the device driver for the board.

Refer to the following documentation for programming information:

- For Windows NT 4.0 programmers, the *Microsoft Windows NT Training Kit*, ISBN 1-55615-864-5, Microsoft Corporation.
- For Windows 95 and Windows 98 programmers, the *Microsoft Windows User's Guide*, Document Number SY06851-0290, Microsoft Corporation.
- For Visual Basic and Visual C++ programmers, the online help in your programming environment.

Where to Get Help

Should you run into problems installing or using the DT-Active Open Layers control, the Technical Support Department is available to provide technical assistance. Refer to [Chapter 5, "Product Support,"](#) for information on how to contact the Technical Support Department. If you are outside the U.S. or Canada, call your local distributor, whose number is listed in your Data Translation product handbook or contact the Data Translation web site (www.datatranslation.com).



Getting Started

What is the DT-Active Open Layers Control?	2
Quick Start	5
Using the Example Programs	10
Using the Online Help	11
Removing the Software	12

What is the DT-Active Open Layers Control?

The DT-Active Open Layers control for imaging is an ActiveX® control that allows you to develop imaging and machine vision application programs in Visual Basic and Visual C++. An example of how to use DT-Active Open Layers in Visual Java++ is also provided. You use the DT-Active Open Layers control as you would use any other ActiveX or OLE custom control in your Windows application.

The DT-Active Open Layers control provides the following features:

- **Uses the standard and extension functions provided in the Frame Grabber SDK**– The standard functions, are fully compatible with DT-Open Layers™, which is a set of standards for developing integrated, modular application programs under Windows. You can use the standard functions to perform common operations on all or most frame grabber boards. The extension functions allow you to perform unique operations that are specific to a particular frame grabber board or to a small set of frame grabber boards.
- **Provides properties, methods, and/or events for Visual Basic and Visual C++** – The DT-Active Open Layers control has unique properties, methods, and events, described as follows:
 - Properties, available for Visual Basic only, represent the variables that allow you to configure the operation. Properties are accessible through the Properties window and through the control’s custom Property Pages. Some properties are read-only and provide information about your operation.
 - Methods, available for Visual Basic and Visual C++, are tools that you can use in your code to provide run-time control of operations. Some methods are accessible through the control’s custom Property Pages.

- Events, available for Visual Basic and Visual C++, are actions that cause code to be executed when an event occurs.
- **Provides multiple-board support** – The DT-Active Open Layers control is hardware-independent. You can add support for new boards, without altering or recompiling code at the application level, by adding a new DT-Open Layers device driver. You install the device driver separately (in the Windows environment); refer to your board-specific documentation for more information on installing device drivers.

The DT-Active Open Layers control supports the following Data Translation PCI frame grabber boards:

- **DT3155** – an industrial-accuracy monochrome frame grabber board.
- **DT3152** – a high-accuracy variable-scan monochrome frame grabber board.
- **DT3153** – a composite color frame grabber board.
- **DT3154** – a high-performance RGB color frame grabber board.
- **DT3157** – an RS-422 input, 8-bit to 16-bit, single-channel/ dual-channel, monochrome frame grabber board for digital cameras.
- **DT3130 Series** – consists of the DT3131, DT3132, DT3133, DT3131-ISO, DT3132-ISO, and DT3133-ISO boards; these composite color frame grabber boards provide one to three simultaneous input channels, one to three external trigger inputs (which can be either isolated or not isolated), and one to three strobe outputs (which can be either isolated or not isolated).
- **DT3120** – a low-cost, composite color frame grabber board.

Note: The list of supported boards is constantly expanding. Refer to the Data Translation web site (www.datatranslation.com) for information on supported boards and device driver availability.

Quick Start

The following is an overview of the tasks required to install and use the DT-Active Open Layers control:

1. Make sure that your system meets the requirements for installing the DT-Active Open Layers control. For more information, refer to the next section.
2. Install the DT-Active Open Layers control. For more information, refer to [page 6](#).
3. Add the DT-Active Open Layers control to your Visual Basic or Visual C++ project. For more information, refer to [page 7](#).
4. Create your application program. For more information, refer to your programming language documentation.
5. Distribute your application program. For more information, refer to [page 9](#).

If you have problems installing or using the DT-Active Open Layers control, refer to [page 11](#) for information on accessing the online help, or refer to [Chapter 5, "Product Support,"](#) for information on contacting Data Translation's Technical Support Department.

System Requirements

To use the DT-Active Open Layers control, you need the following:

- 100 MHz Pentium (or higher) PC with a CD-ROM drive.
- A minimum of 32 MB of RAM.
- 3 MB of hard disk space.
- One or more supported frame grabber boards with appropriate cables, installed in a PCI slot.

- The latest version of the device driver for each installed frame grabber board. If you purchased a new frame grabber board when you purchased the DT-Active Open Layers control, the latest version of the device driver is included on the software CD-ROM. If you already have a frame grabber board installed in your system, download the latest version of the device driver from the Data Translation web site (www.datatranslation.com).
- Video input source (camera).
- Microsoft Windows 95, Windows 98, or Windows NT 4.0 (or higher).
- Microsoft Visual C++ 6.0 or higher, Microsoft Visual Basic 6.0 or higher, or Microsoft Visual Java++ 6.0 or higher.
- For Windows NT, Service Pack 3.0 (or higher).

Installing the Software

Before you install the DT-Active Open Layers control, make sure that you have installed your frame grabber board and device driver. Refer to your board-specific documentation for installation instructions.

To install the DT-Active Open Layers control, perform the following steps:

1. Insert the DT-Active Open Layers CD-ROM in the CD-ROM drive.
If Auto-Run is enabled, the software installation procedure starts automatically when you insert the CD-ROM; continue at step 4. If Auto-Run is disabled, continue at step 2.
2. Click **Start** from the Task Bar, then click **Run**.
The Run dialog box appears.
3. Enter `x:\setup` (where *x* is your CD-ROM drive letter).
The Welcome screen appears after a brief product screen.

4. Click **Next**, then click **Yes** to accept the license agreement.
5. If you want to install the files in the default directory (C:\Program Files\DT-Active Open Layers\Frame Grabber), click **Next**. If you want to change the installation directory, enter the appropriate directory and then click **Next**.
6. If you want to use the default program folder name (Data Translation, Inc\DT-Active Open Layers) in the Programs menu, click **Next**. If you want to change the program folder name, enter the appropriate name and then click **Next**.
The files are copied to your system. For a complete list of the files installed, refer to the README.TXT file.
7. If you want to read the README.TXT file, click the check box.
8. Click **Finish**.
A program group is created with icons for Visual Basic and Visual C++ example programs; the online help; UninstallShield; and the README.TXT file.

Note: After installing the software, read the README.TXT file for any information not included in this document at release time.

Adding the Control to Your Project


This section describes how to add the DT-Active Open Layers control to a

- Visual Basic 6.0 project (see the next section), and
- Visual C++ 6.0 project (see [page 8](#)).

Visual Basic 6.0

To add the DT-Active Open Layers control to a Visual Basic 6.0 project, perform the following steps:

1. Select **Project > Components**.
The Components dialog box appears.
2. Click the **Controls** tab.
3. Click **Data Translation Active Open Layers Frame Grabber Control**, then click **OK**.

When the DT-Active Open Layers control is loaded, the  icon appears in the toolbox.

Visual C++ 6.0

To add the DT-Active Open Layers control to a Visual C++ 6.0 project, perform the following steps:

1. Start Microsoft Visual C++ 6.0.
2. Create a new MFC project using the application wizard. Make sure that you add support for ActiveX controls.
3. Select **Project > Add to Project > Components and Controls**.
The "Component and Controls Gallery" appears.
4. Double-click the **Registered ActiveX Controls** folder.
5. Click **Data Translation Active Open Layers Frame Grabber Control**, then click **Insert**.
The program prompts you to insert the component and to confirm the creation of a wrapper class for accessing the control through Visual C++.
6. If you intend to use the DT-Active Open Layers events, use the Class Wizard to add the event handlers.

Note: Several DT-Active Open Layers methods use enumerated variables. To ensure that these enumerated variables are recognized in Visual C++, make sure that you add the following line to the beginning of your application program:

```
#import "x:\Path\DTAOLFG.DLL" no_namespace
```

where *x:\Path* is the complete path to the location of the DTAOLFG.DLL file on your system. For example, the following line might be used in a Windows NT system:

```
#import "c:\winnt\system32\DTAOLFG.DLL" no_namespace
```

Distributing an Application Program

If you intend to distribute application programs that use the DT-Active Open Layers control, keep in mind that you will need to perform the following tasks:

- **Install the DTAOLFG.DLL file** – When creating your application's installation program, you can use the Visual Basic Application Setup Wizard or you can create the program yourself. If you are using the wizard, the application automatically installs the DTAOLFG.DLL file. If you are creating the installation program yourself, make sure that the DTAOLFG.DLL file is installed in the user's WINDOWS\SYSTEM directory.
- **Register the DT-Active Open Layers control** – If you are using the Visual Basic Application Setup Wizard, the installation program automatically registers the DT-Active Open Layers control to the target system. If you are creating the installation program yourself, you must register the control by running the Regsvr32.exe program on your DT-Active Open Layers CD-ROM. Make sure that you provide a complete path to the control.

Using the Example Programs

A simple Acquire example program is shipped with the DT-Active Open Layers control for Visual Basic, Visual C++, and Visual Java++. This program allows you to open a frame grabber board, allocate a frame buffer, and acquire image data to the frame buffer.

In addition, a comprehensive example program, called DTAcquire, is shipped for Visual Basic. This example program contains the source code for the DT-Acquire application that is provided on the Imaging OMNI CD-ROM that ships with Data Translation frame grabber boards.

Note: Additional example programs will be provided on the Data Translation web site (www.datatranslation.com) as they become available.

You can modify the example programs to suit your particular needs and then compile the code in your particular development environment.

To use the example programs in Visual Basic, Visual C++, or Visual Java++, perform the following steps:

1. Open Visual Basic, Visual C++, or Visual Java++.
2. For Visual Basic, open the example project Acquire.VBP or DTAcquire.VBP.

For Visual C++, open the example project Acquire.CPP.

For Visual Java++, open the example project Acquire.VJP.

3. Compile the program, then run it.

Using the Online Help

This manual is intended to be used with the online help for the DT-Active Open Layers control. The online help contains all of the specific reference information for each of the properties, methods, events, and error codes.

To open the online help, click **Programs\Data Translation, Inc\DT-Active Open Layers** from the Start menu and then select the DT-Active Open Layers (for imaging) online help. To access context-sensitive help while using the DT-Active Open Layers control, either highlight a property in the Properties window and then press **F1** or click the Help button in one of the tabs in the Property Pages. The help topic associated with that property/tab appears.

Removing the Software

If you are having problems with the DT-Active Open Layers control, you may want to remove the control and then reinstall it. To remove the DT-Active Open Layers control, exit your programming environment, click **Programs \Data Translation, Inc \DT-Active Open Layers** from the Start menu, and then select **UninstallShield**. All applicable files and file settings are removed.

2

Summary of Properties, Methods, and Events

General Imaging	15
Input	16
Digital Camera	21
Frame Buffers	21
Passthru	22
Overlay	23
Acquisition	24
Image Processing	25
Digital I/O	26

This chapter provides a summary of the DT-Active Open Layers control properties, methods, and events. For more detailed information, refer to the DT-Active Open Layers control online help. Refer to [page 11](#) for information on opening the online help.

Note: The properties described in this chapter are appropriate for Visual Basic only. If you are using Visual C++, use the associated Visual C++ methods instead. For example, to set and/or return the pixel clock source, use the Visual C++ **SetClockSource** and **GetClockSource** methods instead of the Visual Basic **ClockSource** property. For more information about Visual C++ methods, refer to the DT-Active Open Layers online help.

General Imaging

Property/Method Name	Description
OpenDevice method	Opens a frame grabber board using its alias.
CloseDevice method	Closes the frame grabber board previously opened using the OpenDevice method.
GetDeviceName method	Returns the alias string for a specified frame grabber board.
DeviceCount property	Returns the number of frame grabber board aliases available.
ActiveDevice property	Sets and returns a frame grabber board alias and opens the specified board.
StatusMessage property	Translates the last DT-Open Layers status code into a corresponding text message.
Timeout property	Sets and returns the timeout period, in seconds, used by the device driver.
DeviceType property	Returns the type of the current frame grabber board.
CameraFile property	Sets and returns the complete path to a camera file.

Input

This section contains a summary of the following input properties, methods, and events:

- Input source (see the next section),
- Color (see [page 17](#)),
- Sync signal (see [page 18](#)),
- Clock (see [page 19](#)),
- Look-up table (see [page 19](#)),
- Video area (see [page 19](#)), and
- Trigger (see [page 20](#)).

Input Source

Property/Method Name	Description
QueryInputCaps method	Returns information about the input capabilities and characteristics of the frame grabber board and device driver.
BasedSourceModeEnable property	Specifies whether you want to use the configuration of a particular base input channel for all input channels or whether you want to use separate configurations for each input channel.
BasedSource property	Sets and returns the base input channel.
VideoSignalType property	Sets and returns the type of video signal (composite, variable-scan, Y/C, or RGB).
VideoInputSource property	Sets and returns the input channel providing the video signal.
InputFilter property	Sets and returns the type of input filter used.
Offset property	For monochrome frame grabber boards, sets and returns the offset voltage.

Input Source (cont.)

Property/Method Name	Description
Gain property	For monochrome frame grabber boards, sets and returns the gain.
Reference property	For monochrome frame grabber boards, sets and returns the reference voltage.

2

Color

Property/Method Name	Description
ColorQueryInterface method	Determines which color interfaces are supported by the frame grabber board.
ColorImageParameters method	Returns information about the color capabilities and characteristics of the color frame grabber board and device driver.
Brightness property	Sets and returns the brightness level.
Contrast property	Sets and returns the contrast level.
VSaturation property	Sets and returns the V-saturation level.
USaturation property	Sets and returns the U-saturation level.
Hue property	Sets and returns the hue level.
RedOffset property	Sets and returns the red offset value.
GreenOffset property	Sets and returns the green offset value.
BlueOffset property	Sets and returns the blue offset value.
RedReference property	Sets and returns the red reference value.
GreenReference property	Sets and returns the green reference value.
BlueReference property	Sets and returns the blue reference value.

Sync Signal

Property/Method Name	Description
SyncSource property	Sets and returns the sync source for composite video signals.
SyncThresh property	Sets and returns the sync threshold for composite video signals.
HorizontalTransition property	Sets and returns the edge at which the horizontal sync signal occurs for variable-scan video signals.
VerticalTransition property	Sets and returns the edge at which the vertical sync signal occurs for variable-scan video signals.
SyncSentinel property	Enables and disables the Sync Sentinel.
HSyncSearchPos property	Sets and returns the horizontal sync search position.
HSyncInsertPos property	Sets and returns the horizontal sync insert position.
VSynSearchPos property	Sets and returns the vertical sync search position.
VSynInsertPos property	Sets and returns the vertical sync insert position.
SyncMasterEnabled property	Enables and disables Sync Master mode.
HorizontalPulseWidth property	Sets and returns the width of the horizontal sync signal for Sync Master mode.
VerticalPulseWidth property	Sets and returns the width of the vertical sync signal for Sync Master mode.
HorizontalFrequency property	Sets and returns the frequency of the horizontal sync signal for Sync Master mode.
VerticalFrequency property	Sets and returns the frequency of the vertical sync signal for Sync Master mode.
Phase property	Sets and returns the phase between the horizontal and vertical sync signals for Sync Master mode.

Clock

Property/Method Name	Description
ClockSource property	Sets and returns the pixel clock source.
ClockTransition property	Sets and returns the edge at which an external pixel clock pulse occurs.
ClockFrequency property	Sets and returns the frequency of the internal pixel clock.

Look-Up Table

Property/Method Name	Description
InputLUT property	Sets and returns the ILUT to use.
ReadInputLUT method	Returns a range of values from the specified ILUT.
WriteInputLUT method	Loads a set of values into the specified ILUT.

Video Area

Property/Method Name	Description
TotalPixelsPerLine property	Sets and returns the size of the entire area between two consecutive horizontal sync signals.
TotalLinesPerField property	Sets and returns the size of the entire area between two consecutive vertical sync signals.
FirstActivePixel property	Sets and returns the left-most pixel in the active video area.
FirstActiveLine property	Sets and returns the top-most pixel in the active video area.
ActivePixelCount property	Sets and returns the width of the active video area.
ActiveLineCount property	Sets and returns the height of the active video area.

Video Area (cont.)

Property/Method Name	Description
BackPorchStart property	Sets and returns the start of the back porch.
ClampStart property	Sets and returns the clamp start position.
ClampEnd property	Sets and returns the clamp end position.

Trigger

Property/Method Name	Description
TriggerType property	Sets and returns the trigger type (software or external) for a single-frame acquisition.
TriggerTransition property	Sets and returns the edge at which an external trigger occurs for a single-frame acquisition.
MultiTriggerType property	Sets and returns the trigger type (software or external) for a multiple-frame acquisition.
MultiTriggerTransition property	Sets and returns the edge at which an external trigger occurs for a multiple-frame acquisition.
MultiTriggerMode property	Sets and returns the trigger mode for a multiple-frame acquisition.

Digital Camera

Property/Method Name	Description
DigitalCameraType property	Sets and returns the type of the currently attached digital camera.
EnableExposePulse property	Enables and disables the generation of an exposure pulse to a digital camera.
ExposePulseDuration property	Sets and returns the exposure pulse width.
ExposePulsePolarity property	Sets and returns the active state of the exposure pulse (active high or active low).
ResetCamera method	Resets the currently attached digital camera.

2

Frame Buffers

Property/Method Name	Description
AllocateFrame method	Allocates a frame buffer for an image acquisition and returns the frame index associated with the frame buffer.
DestroyFrame method	Destroys a frame buffer previously created by the AllocateFrame method and releases any resources associated with the frame index.
FrameCount property	Returns the number of allocated frame buffers.
ActiveFrame property	Sets and returns the frame index of the current frame buffer.
FrameBaseAddress property ^a	Returns the base address of the current frame buffer.
FrameTop property	Returns the position of the first line in the active frame buffer.
FrameLeft property	Returns the position of the first pixel in the active frame buffer.

Frame Buffers (cont.)

Property/Method Name	Description
FrameWidth property	Returns the number of pixels per line in the active frame buffer.
FrameHeight property	Returns the number of lines in the active frame buffer.
FrameType property	Returns the type of frame (or field) that you want to acquire in the active frame buffer.
FrameHorizontalScale property	Returns the percentage that the active frame buffer has been scaled horizontally.
FrameVerticalScale property	Returns the percentage that the active frame buffer has been scaled vertically.
FrameStorageMode property	Returns the way in which the data in the active frame buffer is stored in memory.
FrameBytesPerPixel property	Returns the number of bytes in each pixel in the current frame buffer.

a. Used in Visual C++ only.

Passthru

Property/Method Name	Description
QueryPassthruCaps method	Returns information about the passthru capabilities and characteristics of the frame grabber board and device driver.
StartPassthru method	Starts a passthru from the image source to the specified window handle.
StopPassthru method	Stops an asynchronous passthru that is currently in progress.
XPassthruOrigin property	Sets and returns the x-coordinate for the source origin.

Passthru (cont.)

Property/Method Name	Description
YPassthruOrigin property	Sets and returns the y-coordinate for the source origin.
PassthruSnapshot method	Copies the display memory contents to a frame buffer without stopping the passthru operation.
LoadDefaultPassthruLUT method	Loads the passthru LUT with default grayscale values.
LoadPassthruLUT method	Loads entries into the passthru LUT.
ExtendPassthruPalette method	Adds colors to system palette for display during a passthru operation.
OnPassthruNotify event	Generated by the control each time a frame is acquired/displayed during an asynchronous passthru operation.

Overlay

Property/Method Name	Description
AllocateSurface method	Creates an overlay surface.
EnableOverlays method	Turns on overlays for the specified device.
DisableOverlays method	Turns off overlays for the specified device.
DestroySurface method	Destroys an overlay surface that was created with AllocateSurface .
GetSurfaceDC method	Gets a surface device context (DC) so that you can draw and image in the overlay surface.
EraseSurface method	Erases an overlay surface so that you can change the current contents.
ReleaseSurfaceDC method	Releases a surface DC that is retrieved by GetSurfaceDC .

Overlay (cont.)

Property/Method Name	Description
SurfaceCount property	Returns the number of surfaces.
ActiveSurface property	Specifies which surface to use as the active overlay surface.
Translucent property	Sets and returns the state (translucent or opaque) of the active overlay surface.
ColorKey property	Sets the source keying color for the active surface (the color that is replaced by the passthru image).
ImageFile property	Sets and returns a bitmap (.BMP) file to use as an overlay.
OverlaySurfaceOnFrame method	Copies the contents of an overlay surface into the currently acquired frame.

Acquisition

Property/Method Name	Description
AcquireFrames method	Acquires one or more frames, either synchronously or asynchronously.
IsAsyncAcquireFramesDone method	Determines whether an asynchronous acquisition has completed.
StopAsyncAcquireFrames method	Stops an asynchronous acquisition.
OnAcquireNotify event	Generated by the control when an asynchronous acquisition has completed.

Image Processing

Property/Method Name	Description
DrawAcquiredFrame method	Draws an acquired frame on a window.
DrawRGBPlane method	For the DT3154 frame grabber board only, draws RGB planes that were separated using the ExtractRGBPlanesEx method to a window.
SaveAcquiredFrame method	Save the acquired frame to a bitmap image file.
CopyFrameRect method	Copies pixel data from a rectangular region of a frame buffer to a rectangular region of another (or the same) frame buffer.
ReadFrameEx method ^a	Reads the pixel data from a rectangular region of a frame buffer, allocates memory for a user array, and places the data into the user array.
WriteFrame method	Writes pixel data from a user array into a rectangular region within a frame buffer.
ExtractRGBPlanesEx method ^b	For the DT3154 frame grabber board only, separates an RGB image into three independent monochrome frame buffers, one for each color plane.

- a. This method replaces the **ReadFrame** method that was provided in earlier versions of this software package. **ReadFrame** has been left in the software for compatibility with existing applications; however, it is recommended that you use the **ReadFrameEx** method instead.
- b. This method replaces the **ExtractRGBPlanes** method that was provided in earlier versions of this software package. **ExtractRGBPlanes** has been left in the software for compatibility with existing applications; however, it is recommended that you use the **ExtractRGBPlanesEx** method instead.

Digital I/O

Property/Method Name	Description
DigIOConfig property	Sets and returns the digital I/O configuration.
DigIOPattern property	Sets the digital output lines to the specified pattern and/or returns the value of the digital input lines.
DigOutCaps property	Returns the number and location of the digital output lines on the frame grabber board.
DigInCaps property	Returns the number and location of the digital input lines on the frame grabber board.



Using the Control

System Operations	29
Input Operations	32
Digital Camera Operations	66
Allocating a Frame Buffer	68
Passthru Operations	72
Acquisition Operations	80
Image Processing Operations	82
Digital I/O Operations	85

This chapter provides conceptual information that describes the operations provided by the DT-Active Open Layers control. Use this information with the reference information provided in the DT-Active Open Layers online help when programming your frame grabber board; refer to [page 11](#) for more information on opening the help file.

Note: The properties described in this chapter are appropriate for Visual Basic only. If you are using Visual C++, use the associated Visual C++ methods instead. For example, to set and/or return the pixel clock source, use the Visual C++ **SetClockSource** and **GetClockSource** methods instead of the Visual Basic **ClockSource** property. For more information about Visual C++ methods, refer to the DT-Active Open Layers online help.

System Operations

The DT-Active Open Layers control allows you to

- Initialize a frame grabber board (see the next section),
- Define timeouts (see [page 30](#)),
- Work with camera files (see [page 30](#)),
- Work with status codes (see [page 31](#)), and
- Release a frame grabber board (see [page 31](#)).

3

Initializing a Frame Grabber Board

For most operations, you must initialize the device driver for the specified frame grabber board using either the **OpenDevice** method or the **ActiveDevice** property. (Note that **ActiveDevice** closes any open boards before initializing the specified board.)

OpenDevice and **Active Device** require an alias, which is the name you gave the frame grabber board when you configured its device driver. You need one alias for each board. An alias allows you to access more than one board of the same type in your system.

To determine the number of frame grabber boards installed in your system, use the **DeviceCount** property. This is useful if you want to generate a list of devices for an “Open Device” dialog.

Each instance of the DT-Active Open Layers control is associated with one frame grabber board at a time. You must open the board you want to use before you can configure the board’s properties. When you add the control to the project, the properties are set to the default settings.

Note: A slight delay occurs between the time the **OpenDevice** method is called and the time the device driver is initialized and the frame grabber board is actually available to perform operations. If you are programming in Visual Basic, it is recommended that you either include a **DoEvents** procedure after the **OpenDevice** method or put the **OpenDevice** method in the **FormLoad** procedure. If you are programming in Visual C++, it is recommended that you include a delay loop in your code.

Defining Timeouts

By default, the device driver for a frame grabber board uses a timeout period of 10 seconds. This means that once you start an acquisition, the software waits 10 seconds for some kind of action (such as an acquisition or trigger) to occur. If the action it expects does not complete within that time, the DT-Active Open Layers control terminates the process, returns an error, and returns control to the application.

To change the timeout period, set the **Timeout** property to the amount of time, in seconds, that the frame grabber board should wait before terminating a process. To disable timeouts, set the **Timeout** property to 0.

Working with Camera Files

After you set up your frame grabber board to work with your camera, you can save the current camera settings in a camera file (.CAM). Use the **CameraFile** property to specify the complete path to the camera file. When you save your form/project, the current camera settings are saved to the specified file. When you open your form/project, you can use the **CameraFile** property to specify the camera file you want to use. By default, no camera file is specified.

Note that only the camera settings are saved in the camera file; however, other configuration information is saved when you save your form/project.

Working with Status Codes

Status codes are associated with DT-Active Open Layers control properties and methods. Usually, if the status is equal to `OLC_STS_NORMAL` or `S_OK`, the operation was successful. Other status codes can indicate the status of an operation, such as whether it has completed, or an error condition. It is good practice to check the status code regularly for an error condition. You can retrieve the text string associated with the last status code using the **StatusMessage** property.

For information about a specific status code/message, refer to the DT-Active Open Layers control online help. See [page 11](#) for more information about opening the online help.

For information about how to perform error checking in your application program, refer to the example programs.

Releasing a Frame Grabber Board

When you finish acquiring images, release the frame grabber board with the **CloseDevice** method. Make sure that you close all open boards before you exit the program. Boards that are not explicitly closed can continue to use system resources (such as memory), making the resources unavailable until the system is restarted. Also, boards that are not explicitly closed may not be marked as available until the system is restarted.

Input Operations

This section describes how to

- Configure the input source (see the next section),
- Set up the input look-up table (see [page 55](#)),
- Establish the active video area (see [page 57](#)),
- Establish a frame within the active video area (see [page 62](#)), and
- Set up triggers (see [page 64](#)).

Configuring the Input Source

This section describes how to configure the following parameters:

- Based source mode (see the next section),
- Video signal type (see [page 33](#)),
- Video input channel (see [page 35](#)),
- Input filter (see [page 35](#)),
- Video input signal (see [page 37](#)),
- Color definitions (see [page 42](#)),
- Sync signals (see [page 45](#)), and
- Pixel clock (see [page 53](#)).

Based Source Mode

Many frame grabber boards acquire image data from more than one video input source, such as a camera, simultaneously. However, the board can process incoming data from only one source at a time. If your board supports input operations, you must specify which input source (channel) to use to acquire image data.

Typically, whenever you select a new input channel, you must specify input settings for the new channel. If you want to use input settings that you have specified for another video input channel for the new channel, set the **BasedSourceModeEnable** property to TRUE and use the **BasedSource** property to indicate the base (the channel whose input settings you want to use). The frame grabber board will use the input settings of the base for any subsequent channels you select.

Note: As long as based source mode is enabled, you can access the input controls for the base input channel only. Attempting to read or modify the input controls for any other input channel results in an error until you disable based source mode.

Video Signal Type

The video signal type is typically one of the following:

- **Composite** – For monochrome frame grabber boards, horizontal and vertical syncs are combined into one signal. For color frame grabber boards, red, green, and blue image data are combined in one signal.
- **Variable-scan** – For monochrome frame grabber boards only. Horizontal and vertical syncs reside on separate signals.
- **Y/C** – For color frame grabber boards only. Luminance (Y) and color (C) data are stored separately.
- **RGB** – For color frame grabber boards only. Red, green, and blue image data reside on separate signals.

To determine whether your frame grabber board supports programming the video signal type, you can query the board. [Table 1](#) lists the information about the video signal type that you can query for, the method to use to perform the query, and the key to use in the query method.

Table 1: Getting Information About the Video Signal Type

Information	Method	Key
For monochrome boards, whether the board supports video signal types.	QueryInputCaps	OLC_FG_IC_DOES_VIDEO_SELECT
For monochrome boards, the video signal types the board supports.	QueryInputCaps	OLC_FG_IC_VIDEO_TYPE_LIMITS
For color boards, whether the board supports video signal types.	ColorQueryInterface	OLC_QUERY_COLOR_INTERFACE_SIGNAL_TYPE

The video signal types returned by the **QueryInputCaps** method are defined in [Table 2](#).

Table 2: Video Signal Types Returned

Video Signal Type	Description
OLC_FG_VID_COMPOSITE	Composite
OLC_FG_VID_VARSCAN	Variable-scan

To set the video input type, use the **VideoSignalType** property with the appropriate setting. If you specify a composite video signal type for a monochrome board, you must also specify the sync source (see [page 46](#)) and the sync threshold (see [page 47](#)). If you specify a variable-scan video input source, you can also modify the edge of the horizontal and vertical sync signals (see [page 48](#)).

Video Input Channel

The number of video input channels available depends on your frame grabber board. To determine the number of input channels supported, use the **QueryInputCaps** method with the key `OLC_FG_IC_INPUT_SOURCE_COUNT`. To specify the input channel providing the video signal, use the **VideoInputSource** property.

If based source mode is disabled when you select a new video input channel, the frame grabber board reloads the default input settings. If based source mode is enabled when you select a new video input channel, the board uses the input settings of the specified base for the new input channel. For more information about based source mode, refer to [page 32](#).

3

Input Filter

While it is possible to acquire monochrome images from color signals, the color content of these signals can cause interference patterns that degrade the image. Most monochrome frame grabber boards provide a chrominance (or notch) input filter that removes color information for cleaner acquisition and more accurate analysis.

If you are using an AC-coupled video signal that has chrominance information on it, as is the case with the NTSC and PAL video formats, you can apply the input filter to remove the chrominance information. For frame grabber boards that support filtering, the input filter for 60 Hz is set to approximately 3.5 MHz, while the input filter for 50 Hz is set to approximately 4.28 MHz. Check your board-specific documentation for the exact values.

If you are using a DC-coupled video signal, you must use the DC Filter - None setting.

To determine whether your frame grabber board supports programming the input filter, you can use the **QueryInputCaps** method to query the board. [Table 3](#) lists the information about the input filter that is returned by **QueryInputCaps** and the key to use in **QueryInputCaps** to get the information.

Table 3: Getting Information About the Input Filter

Information	Key
Whether the frame grabber board supports input filters.	OLC_FG_IC_DOES_INPUT_FILTER
The input filters the board supports.	OL_FG_IC_INPUT_FILTER_LIMITS

The input filter types returned by the **QueryInputCaps** method are defined in [Table 4](#).

Table 4: Input Filter Types Returned

Input Filter Type	Description
OLC_FG_FILT_AC_NONE	AC coupled, no filter.
OLC_FG_FILT_AC_50	AC coupled, 50 Hz filter.
OLC_FG_FILT_AC_60	AC coupled, 60 Hz filter.
OLC_FG_FILT_DC_NONE	DC coupled, no filter.

To select an input filter (or to turn filtering off), use the **InputFilter** property with the appropriate setting.

Video Input Signal

For monochrome frame grabber boards, you can improve the resolution of your acquired image by specifying which portion of the video input signal you want to digitize. You do this by specifying an offset, gain, and reference value.

To determine whether your board supports programming the offset, gain, and reference, use the **QueryInputCaps** method with the key `OLC_FG_IC_DOES_PROG_A2D`. If **QueryInputCaps** indicates that the board supports programming black and white levels, it also supports programming the offset, gain, and reference.

Offset, gain, and reference are described as follows:

- **Offset** – The voltage you apply to the minimum value of the video input signal ($-V_{\min}$) to zero it. All data below 0 V is digitized as black pixels. Use the **Offset** property to specify the offset voltage.
- **Gain** – The value you use to multiply the amplitude of the video input signal, thereby increasing or decreasing the overall range of the signal. You apply the gain after you apply the offset voltage. Use the **Gain** property to specify the gain (0.5, 1, 2, or 4).
- **Reference** – The maximum voltage (after offset and gain have been applied) that you want to digitize. All data above the reference voltage is digitized as white pixels. Use the **Reference** property to specify the reference voltage.

Note: Some frame grabber boards, such as the DT3155, support a gain of 1 only. For these boards, the board-specific documentation refers to offset and reference in terms of a black level and a white level. Black level is defined as the voltage below which all other voltages are digitized to black. White level is defined as the voltage above which all other voltages are digitized to white.

To determine the range of offset and reference values for these boards, you must convert the black level and white level values to the appropriate offset and reference values, using the following equations:

$$\text{Offset}_{\min} = -\text{BlackLevel}_{\max}$$

$$\text{Offset}_{\max} = -\text{BlackLevel}_{\min}$$

$$\text{Reference}_{\min} = \text{WhiteLevel}_{\min} - \text{BlackLevel}_{\max}$$

$$\text{Reference}_{\max} = \text{WhiteLevel}_{\max} - \text{BlackLevel}_{\min}$$

For example, assume that the minimum black level supported by your frame grabber board is 1.275 mV. The maximum offset that you can specify for this board is -1.275 mV.

You can convert the nominal values using the equations

$\text{Offset}_{\text{nominal}} = -\text{BlackLevel}_{\text{nominal}}$ and $\text{Reference}_{\text{nominal}} = \text{WhiteLevel}_{\text{nominal}} - \text{BlackLevel}_{\text{nominal}}$. The granularity is unchanged.

Figure 1 illustrates a signal in which part of the image is below 0.0 V and part is above 1.0 V.

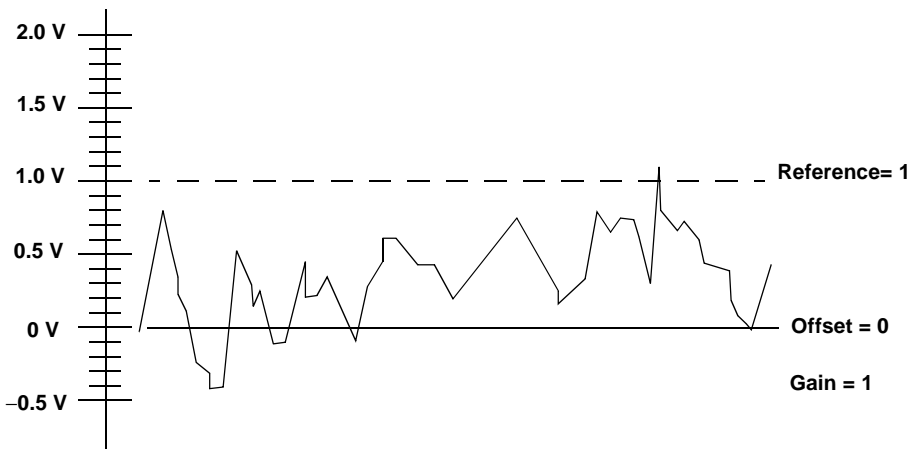


Figure 1: Original Signal

To adjust the video input signal so that the frame grabber board digitizes the signal between -0.4 V and 1.1 V, perform the steps described in the following sections.

Step 1. Adjust the Offset

To digitize the data below 0.0 V, shift the signal up by setting the offset to $-V_{\min}$. In the example, $V_{\min} = -0.4$ V; therefore, $-V_{\min} = 0.4$ V.

Note that the range of valid offset voltages is not linear. When setting the offset, the **Offset** property uses the supported voltage closest to the voltage you specify. The value returned is the actual offset.

[Figure 2](#) shows the effect of using an actual offset of 0.4032 V.

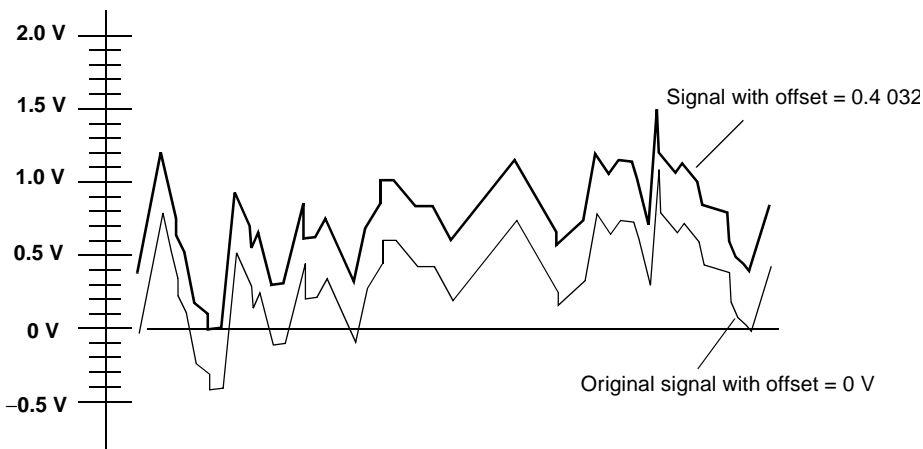


Figure 2: Use of Offset Voltage

Note that since the image has been shifted up by 0.4032 V, a voltage level of -0.4032 V, relative to the camera's output, is now digitized as black.

Step 2. Apply the Gain

With an offset of 0.4032 V, the maximum point on the signal is now 1.5032 V. Assume that your frame grabber board can digitize voltages between 0.0 V and 1.28 V only. To digitize the part of the signal between 1.28 V and 1.5032 V, you must decrease the amplitude of the video signal. To decrease the amplitude of the signal, use a gain of 0.5. [Figure 3](#) shows the effect of the gain on the video signal.

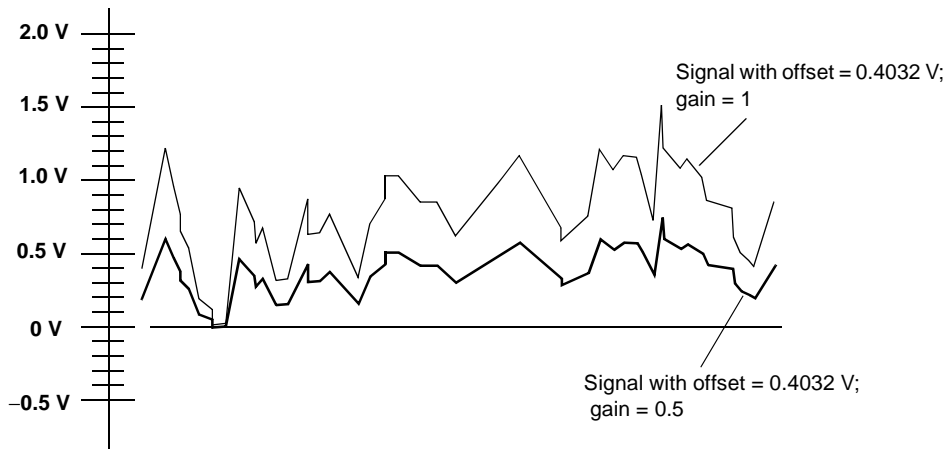


Figure 3: Use of Gain to Adjust the Amplitude of the Signal

Step 3. Apply the Reference

To determine the reference, use the equation:

$$reference = gain (V_{max} + offset)$$

For example, if the offset is 0.4032 V, V_{max} is 1.1 V, and the gain is 0.5, *reference* is determined as follows:

$$reference = 0.5 (1.1 \text{ V} + 0.4032 \text{ V})$$

$$reference = 0.7516 \text{ V}$$

Note that the range of valid reference voltages is not linear. When setting the reference, the **Reference** property uses the supported voltage closest to the voltage you specify. The value returned is the actual reference.

Figure 4 shows the effect of an offset of 0.4032 V, a gain value of 0.5, and an actual reference value of 0.75 V. The frame grabber board can now digitize the entire video signal from -0.4 V to 1.1 V, relative to the camera output.

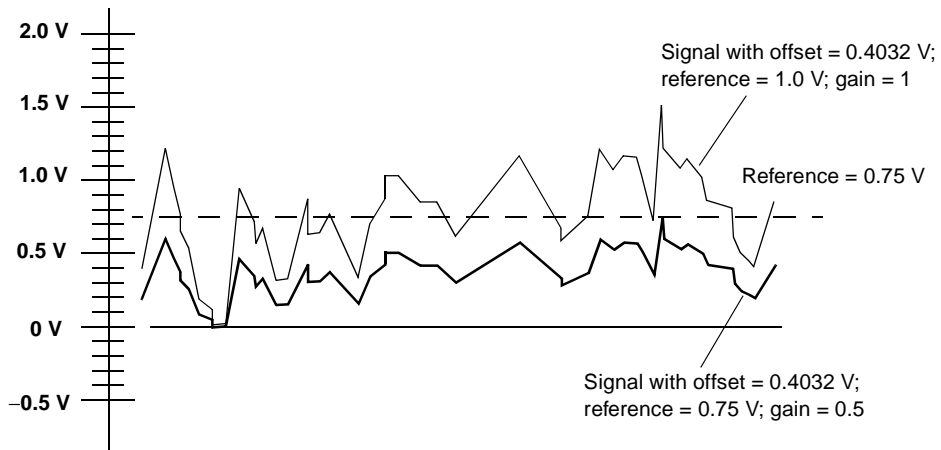


Figure 4: Use of Reference, Offset, and Gain

Color Definitions

Some frame grabber boards allow you to adjust one or more color settings. To determine whether you can adjust color settings, use the **ColorQueryInterface** method with the key `OLC_QUERY_COLOR_INTERFACE_IMAGE_PARAMETER`. If **ColorQueryInterface** indicates that you can adjust color settings, use the **ColorImageParameters** method with the control `OLC_QUERY_CAPABILITY` and the appropriate parameter setting to determine whether a particular color settings is supported.

Note: In addition to determining whether you can adjust color settings, the **ColorQueryInterface** method allows you to determine whether other features, such as digital I/O and Sync Master mode, are supported for color frame grabber boards. For more information, refer to the DT-Active Open Layers online help.

In addition to determining which color settings are supported, the **ColorImageParameters** method provides other operations that you can perform, such as returning the maximum, minimum, nominal, and granularity values for a color setting. For more information, refer to the DT-Active Open Layers online help.

The following color settings are supported:

- For frame grabber boards that support the HSI (Hue/Saturation/Intensity) color model:
 - *Brightness* – The brightness level is a value associated with a pixel that represents its gray value. Use the **Brightness** property to adjust the brightness.
 - *Contrast* – The contrast level is the overall range of the monochrome signal of an image. For example, a high contrast image has a large range between black and white values; a low contrast image has a small range between black and white values. Use the **Contrast** property to adjust the contrast.
 - *V-saturation* – The V-saturation level is the purity of the blue and green primary colors in an image. For example, if a particular pixel has a value of 0 for green, but a value of 256 for blue, the pixel is saturated in blue. Use the **VSaturation** property to adjust the V-saturation.

- *U-saturation* – The U-saturation level is the purity of the green and red primary colors in an image. For example, if a particular pixel has a value of 0 for green, but a value of 256 for red, the pixel is saturated in red. Use the **USaturation** property to adjust the U-saturation.
- *Hue* – The hue level is the intensity or shade of the color. Use the **Hue** property to adjust the hue.
- For frame grabber boards that support RGB color images:
 - *Offset* – The offset is the voltage that is added to the minimum full-scale voltage of your video input signal. For example, if the negative full-scale voltage of your video input signal is -53.86 mV and you specify an offset of $+53.86$ mV, the resulting digitized value is 0 V. Use the **RedOffset**, **GreenOffset**, and **BlueOffset** properties to specify the offset values for an RGB signal.
 - *Reference* – The reference is the maximum full-scale value of your video input signal. Use the **RedReference**, **GreenReference**, and **BlueReference** properties to specify the reference values for an RGB signal.

Once adjusted for offset, the frame grabber board digitizes each signal (red, green, and blue) between the adjusted negative full-scale voltage and the reference voltage.

Sync Signals

The video input signal includes both image data and timing information. To control timing, the frame grabber board requires both horizontal and vertical sync signals. [Figure 5](#) illustrates the process of generating the horizontal and vertical sync signals.

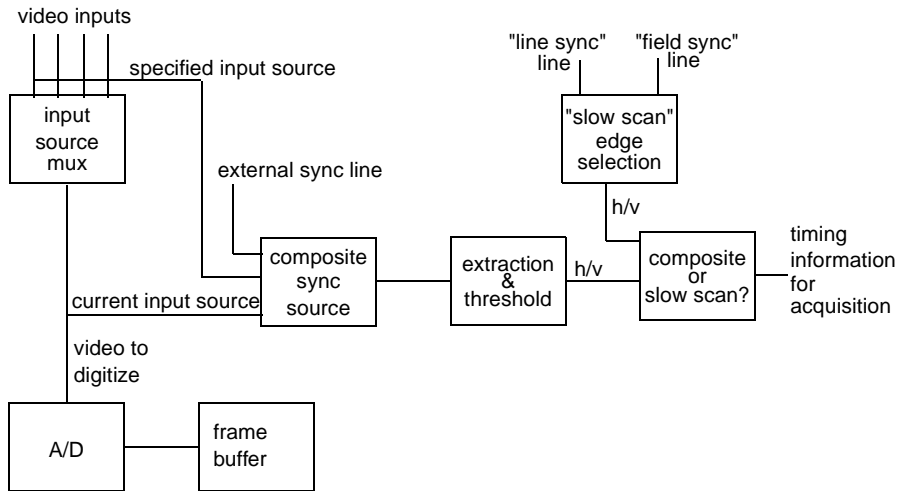


Figure 5: Horizontal and Vertical Timing

The following sections describe how to configure the following sync signal parameters:

- Sync source for composite video input signal (see the next section),
- Sync threshold for composite video input signals (see [page 47](#)),
- Horizontal and vertical sync signal transitions for variable-scan video input signals (see [page 48](#)),
- Sync Sentinel (see [page 48](#)), and
- Sync Master mode (see [page 51](#)).

Sync Source

For composite video signals only, you can obtain the video sync signal from one of the following sync sources:

- The composite sync from the current video input channel (the channel being digitized) only. In this case, the sync signal is stripped from the video signal and fed into the sync circuitry.
- The composite sync from any video input channel. For example, you could use camera 1 on channel 0 as your video input source and camera 2 on channel 1 as your composite sync source. In this case, the sync signal from the selected channel is fed directly into the sync circuitry.
- The composite sync from an external sync source, such as a clock. In this case, the external sync signal is fed directly into the sync circuitry.

The sync period is defined as the portion of the sync signal that falls below the sync threshold. The sync is used to generate the horizontal and vertical timing for the input section of the frame grabber board.

To determine whether your frame grabber board supports programming the sync source, you can use the **QueryInputCaps** method to query the board. [Table 5](#) lists the information about the sync source that is returned by **QueryInputCaps** and the key to use in **QueryInputCaps** to get the information.

Table 5: Getting Information About the Sync Source

Information	Key
The composite sync sources your frame grabber board supports.	OLC_FG_IC_CSOURCE_SOURCE_LIMITS
If your frame grabber board supports a composite sync source from any video input channel, the number of input channels supported.	OL_FG_IC_INPUT_SOURCE_COUNT

The composite sync sources returned by the `QueryInputCaps` method are defined in [Table 6](#).

Table 6: Composite Sync Sources Returned

Composite Sync Source	Description
OLC_FG_CSYSYNC_CURRENT_SRC	Take the composite sync from the current video input source only.
OLC_FG_CSYSYNC_SPECIFIC_SRC	Take the composite sync from any input video source.
OLC_FG_CSYSYNC_EXTERNAL_LINE	Take the composite sync from a source other than one of the available video input sources.

To set the composite sync source, use the `SyncSource` property with the appropriate setting.

Sync Threshold

For composite video signals only, you must specify the sync threshold — the point at which the frame grabber board acknowledges the sync signal. To determine the upper and lower threshold limits that your board supports, use the `QueryInputCaps` method with the key `OLC_FG_IC_CSYSYNC_THRESH_LIST`.

To set the threshold level, use the `SyncThresh` property with the appropriate setting.

Horizontal and Vertical Sync Transition

For variable-scan video signals only, the line (horizontal) and field (vertical) sync are taken directly from the input device and are used to generate the horizontal and vertical timing for the input section of the frame grabber board.

By default, both the horizontal and vertical sync signals occur on a high-to-low transition. To modify the horizontal and/or vertical sync signal transitions, use the **HorizontalTransition** and/or **VerticalTransition** properties with one of the following settings:

- **OnHighToLow** – transitions occur on a falling edge.
- **OnLowToHigh** – transitions occur on a rising edge.

Sync Sentinel

Some video signals, including those from VCRs, can be noisy or low quality, resulting in blurry acquisition due to missing or extraneous sync signals. The Sync Sentinel circuit (when available) corrects for missing, extraneous, or low-level sync signals by allowing for programmable control over the input sync circuitry of the frame grabber board.

When enabled, the Sync Sentinel provides a window in which a sync signal can be detected. Some frame grabber boards allow you to adjust the size of this window. On such boards, the horizontal sync search position defines the pixel location within a line after which the board begins to search for the horizontal sync. The board searches for the horizontal sync from that point until the horizontal sync insert position is reached. If the horizontal sync is not detected by that point, the Sync Sentinel inserts a horizontal sync to synchronize to the video signal.

The vertical sync search position defines the line location within a field after which the frame grabber board begins to search for the vertical sync. The board searches for the vertical sync from that point until the vertical sync insert position is reached. If the vertical sync is not detected before the vertical sync insert position is reached, the Sync Sentinel inserts a vertical sync to stay in sync with the video signal.

By setting the sync search position immediately before the falling edge of the sync signal and setting the sync insert position immediately after the falling edge of the sync signal, you can prevent the frame grabber board from searching for syncs except where they are expected.

If you are switching among multiple input sources that are not synchronized with one another or if the sync signals occur at random intervals, you can disable the Sync Sentinel. This allows the frame grabber board to locate the sync from a new video signal as soon as it occurs.

Note: Instead of disabling the Sync Sentinel, you can set the sync search position as far before the falling edge of the sync as possible and set the sync insert position as far after the falling edge of the sync as possible. This allows the board to search for sync for almost the entire line and/or field.

To determine whether your frame grabber board supports programming the Sync Sentinel, you can use the **QueryInputCaps** method to query the board. [Table 7](#) lists the information about Sync Sentinel that is returned by **QueryInputCaps** and the key to use in **QueryInputCaps** to get the information.

Table 7: Getting Information About the Sync Sentinel

Information	Key
Whether the frame grabber board supports the Sync Sentinel.	OLC_FG_IC_DOES_SYNC_SENTINEL
Whether the search and insert positions are fixed or variable.	OLC_FG_IC_SYNC_SENTINEL_TYPE_LIMITS

The information returned by the **QueryInputCaps** method is defined in [Table 8](#).

Table 8: Sync Sentinel Values Returned

Sync Sentinel Value	Description
OLC_FG_SYNC_SENTINEL_FIXED	The search and insert settings for the Sync Sentinel are fixed and cannot be altered through software.
OLC_FG_SYNC_SENTINEL_VARIABLE	The search and insert settings for the Sync Sentinel are variable and can be altered through software.

To enable and disable the Sync Sentinel, use the **SyncSentinel** property.

If **QueryInputCaps** indicates that you can alter the horizontal and vertical search and insert positions, use the properties listed in [Table 9](#).

Table 9: Sync Sentinel Properties

Property	Description
HSyncInsertPos	Sets and returns the horizontal sync insert position, specified as a percentage of the total pixels per line.
HSyncSearchPos	Sets and returns the horizontal sync search area, specified as a percentage of the total pixels per line.
VSynInsertPos	Sets and returns the vertical sync insert position, specified as a percentage of the total lines per field.
VSynSearchPos	Sets and returns the vertical sync search area, specified as a percentage of the total lines per field.

Note: Enabling Sync Master mode automatically disables the Sync Sentinel. For more information about Sync Master mode, refer to the next section.

Sync Master Mode

Note: Enabling Sync Master mode automatically disables the Sync Sentinel. For more information about the Sync Sentinel, refer to [page 48](#).

For some frame grabber boards, you can enable Sync Master mode. Sync Master allows you to use the sync signals generated by the board to drive one or more cameras, if desired. The video signal from the camera is then digitized as usual, using the syncs generated by the board as the sync basis (gen-locking).

If your frame grabber board supports Sync Master mode, use the properties in [Table 10](#) to adjust the appropriate settings.

Table 10: Sync Master Mode Properties

Property	Description
SyncMasterEnabled	Enables and disables Sync Master mode.
HorizontalPulseWidth	Sets and returns the width of the horizontal sync signal.
VerticalPulseWidth	Sets and returns the width of the vertical sync signal.
HorizontalFrequency	Sets and returns the horizontal sync frequency.
VerticalFrequency	Sets and returns the vertical sync frequency.
Phase	Sets and returns the phase between the horizontal sync and vertical sync signals, specified as a percentage of the horizontal period multiplied by 100.

With interlaced cameras that accept horizontal and vertical syncs, you can adjust the phase to determine the field to output. By setting up the board as noninterlaced with a sync phase of 50%, the even field is continuously selected. If the phase is 1%, the odd field is continuously selected.

You can alternate the phase at each vertical sync by setting the phase at 50% and setting up the board as interlaced. The even and odd fields are then alternately selected for a true interlaced image. Whenever the board is set up as interlaced, the phase alternates between 1% and whatever phase is specified. These different phase adjustments are illustrated in [Figure 6](#).

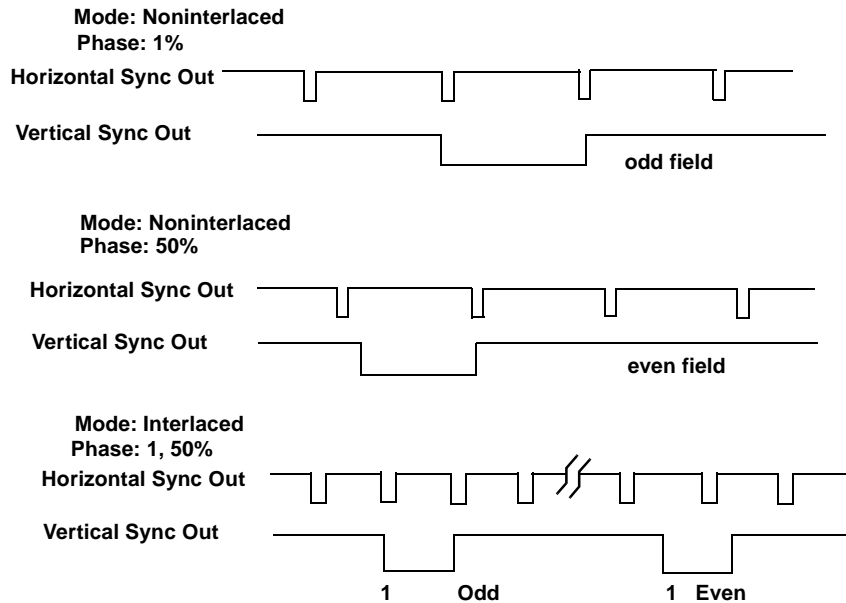


Figure 6: Sync Out Phase Adjustment

Pixel Clock

The data input rate of your frame grabber board is set by the pixel clock. Pixels are available to the board at increments of PixelPeriod, which is equal to $1 / \text{pixel clock frequency}$.

Many cameras provide an external pixel clock, but most frame grabber boards are equipped with an onboard, internal pixel clock as well. By default, the frame grabber board assumes that you are using the internal pixel clock.

If you are using an internal pixel clock, you can change the frequency, if required.

If you are using an external pixel clock, the clock pulse occurs on a high-to-low transition by default. You can modify the clock transition, if required.

To determine whether your frame grabber board supports programming the pixel clock, you can use the **QueryInputCaps** method to query the board. [Table 11](#) lists the information about the pixel clock that is returned by **QueryInputCaps** and the key to use in **QueryInputCaps** to get the information.

Table 11: Getting Information About the Pixel Clock

Information	Key
Whether the frame grabber board supports a programmable pixel clock.	OLC_FG_IC_DOES_PIXEL_CLOCK
The pixel clock sources supported.	OLC_FG_IC_CLOCK_SOURCE_LIMITS
For an internal pixel clock, the range of frequencies supported.	OLC_FG_IC_CLOCK_FREQ_LIMITS

The pixel clock sources returned by the **QueryInputCaps** method are defined in [Table 12](#).

Table 12: Pixel Clock Sources Returned

Pixel Clock Source	Description
OLC_FG_CLOCK_INTERNAL	The frame grabber board generates the timing for acquisition.
OLC_FG_CLOCK_EXTERNAL	An external source (such as a camera) generates the timing for acquisition.

To set the pixel clock source, use the **ClockSource** property with the appropriate setting.

To set the pixel clock frequency for an internal pixel clock, use the **ClockFrequency** property. Note that the range of clock frequency values is not linear. When setting the value, the **ClockFrequency** property uses the supported frequency closest to the value you specify. The value returned by **ClockFrequency** is the actual frequency.

To set the clock transition for an external pixel clock, use the **ClockTransition** property with one of the following settings:

- **OnHighToLow** – clock pulse occurs on a falling edge.
- **OnLowToHigh** – clock pulse occurs on a rising edge.

Setting Up the Input Look-Up Table

An input look-up table (ILUT) allows you to change the value of an incoming pixel. When the ILUT gets an input pixel value, it retrieves the output value for that particular pixel from the ILUT and passes the output value to the region of interest. Pixel values range from 0 to 255 and can generally be defined for each video signal.

You can specify the relationship between the pixel input value and the ILUT output value by loading the ILUT with different processing setups. For example, you can pass an image unaltered (the default setting, known as identity), or you can perform pixel point operations, such as image multiplication and division, intensity correction, and reverse-video, before passing the image on.

As an example, assume that the ILUT is loaded with the identity pattern. An input value of 0 (black in monochrome mode) has an output value of 0 (black in monochrome mode). An input value of 1 has an output value of 1. An input value of 2 has an output value of 2, and so on, up to an input value of 255 (which has an output value of 255 or white in monochrome mode).

As another example, if you load the ILUT with an inverse or negative pattern, an input value of 0 has an output value of 255, an input value of 1 has an output value of 254, and so on, up to an input value of 255 (which has an output value of 0 or black in monochrome mode).

To determine whether your frame grabber board supports programming an ILUT, you can use the **QueryInputCaps** method to query the board. [Table 13](#) lists the information about ILUTs that is returned by **QueryInputCaps** and the key to use in **QueryInputCaps** to get the information.

Table 13: Getting Information About ILUTs

Information	Key
Whether the frame grabber board supports reviewing and modifying ILUTs.	OLC_FG_IC_DOES_INPUT_FILTER
The number of ILUTs the board provides.	OLC_FG_IC_ILUT_COUNT
The maximum number of entries allowed in a given ILUT.	OLC_FG_IC_MAX_ILUT_INDEX
The maximum value allowed in a given ILUT.	OLC_FG_IC_MAX_ILUT_VALUE

To determine the contents (full or partial) of an ILUT, use the **ReadInputLUT** method. To change an ILUT, use the **WriteInputLUT** method. To specify the ILUT to use for future acquisitions, use the **InputLUT** property.

Note that in addition to ILUTs, passthru operations make use of passthru LUTs. For more information about passthru LUTs, refer to [page 75](#).

Defining the Active Video Area

The total video area is a complete set of horizontal and vertical input lines from which you extract the active video area and the frame within the active video area. The total video area includes all parts of the signal, including nonvisual portions such as horizontal and vertical blanking information and sync information.

The total video area is as wide as the total pixels per line (the entire area between two consecutive horizontal sync signals) and as tall as the total lines per field (the entire area between two consecutive vertical sync signals).

The total pixels per line can be calculated as follows:

$$\text{Total pixels per line} = \frac{\text{pixel clock frequency (MHz)}}{\text{horizontal frequency (kHz)}}$$

The total lines per field can be calculated as follows:

$$\text{Total lines per field} = \frac{\text{horizontal frequency (kHz)}}{\text{vertical frequency (Hz)}}$$

To determine whether your frame grabber board supports programming the total video area, you can use the **QueryInputCaps** method to query the board. [Table 14](#) lists the information about the total video area that is returned by **QueryInputCaps** and the key to use in **QueryInputCaps** to get the information.

Table 14: Getting Information About the Total Video Area

Information	Key
Whether the frame grabber board supports programming the total pixels per line and total lines per field.	OLC_FG_IC_DOES_ACTIVE_VIDEO
The range for total pixels per line. ^a	OLC_FG_IC_TOTAL_PIX_PER_LINE_LIMITS
The range for total lines per field. ^a	OLC_FG_IC_TOTAL_LINES_PER_FLD_LIMITS

a. Includes the upper and lower limits, granularity, and nominal value.

To set the total pixels per line, use the **TotalPixelsPerLine** property.
To set the total lines per field, use the **TotalLinesPerField** property.

The active video area floats in the total video area. The active video area is defined as that part of the incoming signal that contains valid video data (not blanking or sync information). Therefore, the active video area consists of the visible portion of those lines containing visible pixel data. Its top is set by the first active line, its left side is set by the first active pixel, it is as wide as the active pixel count, and it is as tall as the active line count.

For more information about the horizontal and vertical signals that comprise the total video area and the parameters you can set to specify the active video area, refer to the following sections.

Horizontal Video Signal

A horizontal line of video is measured in pixels and is identified by the falling edge of the horizontal sync. Each line contains horizontal sync information, blanking information, and active video.

Figure 7 shows the components of a single horizontal line of video. Pixel measurements are relative to the horizontal reference point, which is defined as the beginning of the horizontal sync.

Note that the frame is an area that you establish within the active video area. For more information about frames, refer to [page 62](#).

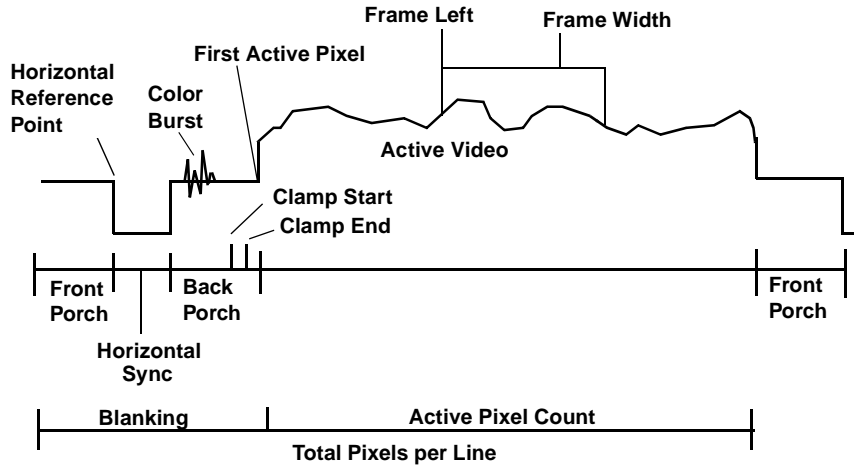


Figure 7: Horizontal Video Signal

In the horizontal video signal, blanking occurs during the horizontal sync and image border periods, which are defined by the front porch (before the horizontal sync) and back porch (after the horizontal sync).

Some frame grabber boards allow you to specify when your camera's back porch period starts, when to expect the first pixel of active video (first active pixel), and how long the camera will receive active video before the next blanking period (active pixel count).

In addition, some frame grabber boards allow you to define a period during the back porch when the board can adjust the black level from the camera. This is called a clamp or DC restore. The board shifts the video signal so that the signal region during the clamp is at 0 V. The active video following the blanking period is then referenced to that shift.

To determine whether your frame grabber board supports programming the horizontal video signal, you can use the **QueryInputCaps** method to query the board. [Table 15](#) lists the information about the horizontal video signal that is returned by **QueryInputCaps** and the key to use in **QueryInputCaps** to get the information.

Table 15: Getting Information About the Horizontal Video Signal

Information	Key
Whether the frame grabber board supports programming the horizontal video signal.	OLC_FG_IC_DOES_ACTIVE_VIDEO
The range for back porch start. ^a	OLC_FG_IC_BACK_PORCH_START_LIMITS
The range for first active pixel. ^a	OLC_FG_IC_ACTIVE_PIXEL_LIMITS
The range for active pixel count. ^a	OLC_FG_IC_ACTIVE_WIDTH_LIMITS
The range for the clamp start position. ^a	OLC_FG_IC_CLAMP_START_LIMITS
The range for the clamp end position. ^a	OLC_FG_IC_CLAMP_END_LIMITS

a. Includes the upper and lower limits, granularity, and nominal value.

To set the back porch start position, use the **BackPorchStart** property. To set the first active pixel, use the **FirstActivePixel** property. To set the active pixel count, use the **ActivePixelCount** property. To set the clamp start position, use the **ClampStart** property. To set the clamp end position, use the **ClampEnd** property.

Vertical Video Signal

A field is composed of a collection of horizontal lines defined by the active line count. Each field contains vertical sync information, blanking information, and lines of active video.

Figure 8 shows the components of a noninterlaced frame or a field of an interlaced frame. Line measurements are relative to the vertical reference point, which is defined as the beginning of the vertical sync.

Note that the frame is an area that you establish within the active video area. For more information about frames, refer to [page 62](#).

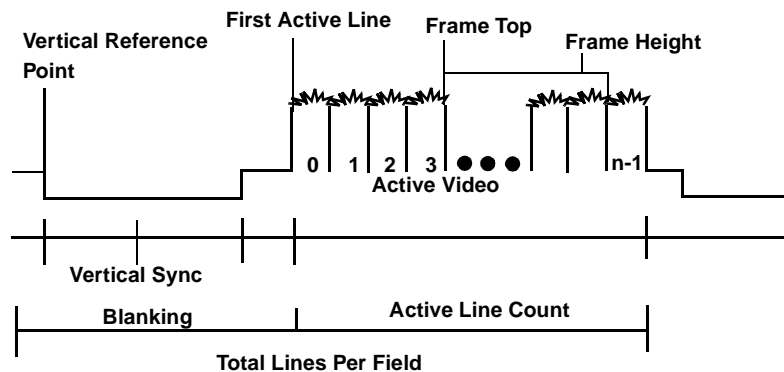


Figure 8: Vertical Video Signal

In the vertical video signal, blanking occurs during the vertical sync. Some frame grabber boards allow you to specify when your camera should expect the first line of active video (the beginning of the active video signal within the field, as a line offset from the beginning of the vertical sync) and how long the camera will receive active video before the next blanking period (active line count).

To determine whether your frame grabber board supports programming the vertical video signal, you can use the **QueryInputCaps** method to query the board. [Table 16](#) lists the information about the vertical video signal that is returned by **QueryInputCaps** and the key to use in **QueryInputCaps** to get the information.

Table 16: Getting Information About the Vertical Video Signal

Information	Key
Whether the frame grabber board supports programming the vertical video signal.	OLC_FG_IC_DOES_ACTIVE_VIDEO
The range for first active line. ^a	OLC_FG_IC_ACTIVE_LINE_LIMITS
The range for active line count. ^a	OLC_FG_IC_ACTIVE_HEIGHT_LIMITS

a. Includes the upper and lower limits, granularity, and nominal value.

To set the first active line, use the **FirstActiveLine** property. To set the active line count, use the **ActiveLineCount** property.

Establishing a Frame Within the Active Video Area

The frame is the portion of the active video area that you want to digitize. The top of the frame is the first line of video relative to the active video area. The left side of the frame is the first pixel of video relative to the active video area. The width of the frame is the number of pixels per line of video. The height of the frame is the number of lines per field. [Figure 9](#) illustrates these relationships.

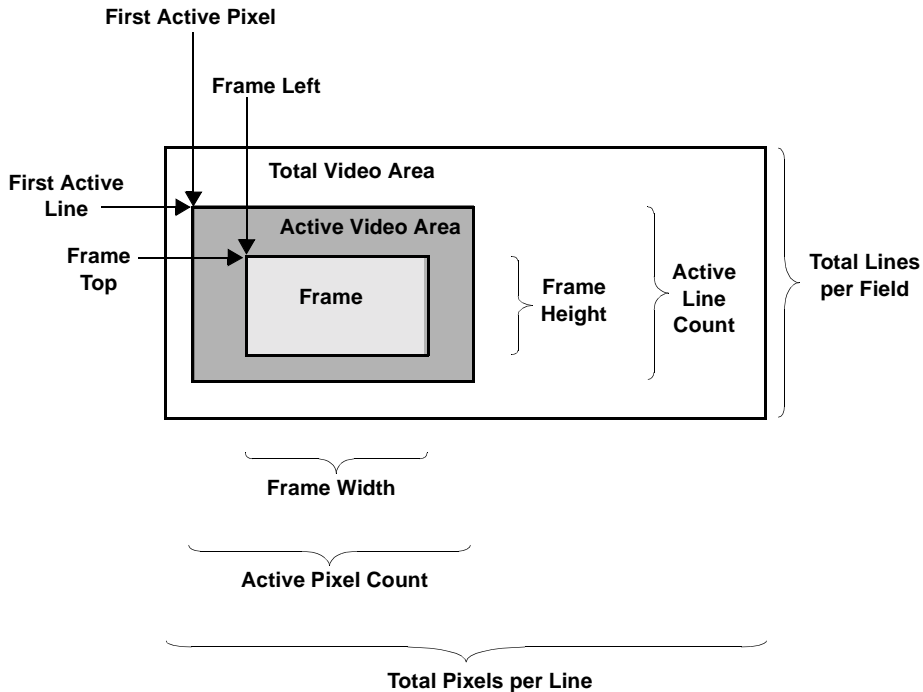


Figure 9: Spatial Relationship of Video Signal

A frame can be either interlaced or noninterlaced. An interlaced frame consists of two consecutive fields, where the start of each field is identified by the falling edge of the vertical sync. These two fields are acquired to create the complete frame. The even field contains lines 0, 2, 4, and so on; the odd field contains lines 1, 3, 5, and so on. A noninterlaced frame consists of a single field, where the start of the field is identified by the falling edge of the vertical sync.

You establish the frame that you want to digitize when you allocate a frame buffer for the frame using the **AllocateFrame** method. For more information about allocating a frame buffer, refer to [page 68](#).

Setting Up Triggers

A trigger is an external event that starts an acquisition. The external event is a signal received from a dedicated external line. To determine if your frame grabber board supports external triggering, you can use the `QueryInputCaps` method to query the board. [Table 17](#) lists the information about the external trigger that is returned by the `QueryInputCaps` and the keys to use to get the information.

Table 17: Getting Information About the External Trigger

Information	Query Key
Whether the frame grabber board supports external triggering.	OLC_FG_IC_DOES_TRIGGER
The types of triggering supported for single-frame acquisitions.	OLC_FG_IC_TRIGGER_TYPE_LIMITS
The types of triggering supported for multiple-frame acquisitions.	OLC_FG_IC_MULT_TRIGGER_TYPE_LIMITS
The trigger modes supported for multiple-frame acquisitions.	OLC_FG_IC_MULT_TRIGGER_MODE_LIMITS

The `OLC_FG_IC_TRIGGER_TYPE_LIMITS` and `OLC_FG_IC_MULT_TRIGGER_TYPE_LIMITS` queries return a mask of bit flags, as defined in [Table 18](#).

Table 18: Trigger Type Bit Flags

Flag	Description
OLC_FG_TRIG_EXTERNAL_LINE	Externally triggered acquisition.
OLC_FG_TRIG_EVENT	Event-triggered acquisition.
OLC_FG_TRIG_ONE_EVENT_DELAY	Delayed event-triggered acquisition.

The `OLC_FG_IC_MULT_TRIGGER_MODE_LIMITS` query returns a mask of bit flags, as defined in [Table 19](#).

Table 19: Trigger Mode Bit Flags

Flag	Description
<code>OLC_FG_MODE_START</code>	Use a single trigger to start the acquisition of a series of multiple frames.
<code>OLC_FG_MODE_EACH</code>	Trigger each frame in a series of multiple frames by a separate trigger.

3

Use the properties in [Table 20](#) to set up triggers for single-frame and multiple-frame acquisitions.

Table 20: Trigger Properties

Type of Acquisition	Property	Description
Single-frame	TriggerType	Enables or disables external triggering.
	TriggerTransition	Indicates whether you want an external trigger to occur on a low-to-high transition or a high-to-low transition.
Multiple-frame	MultTriggerType	Enables or disables external triggering.
	MultTriggerTransition	Indicates whether you want an external trigger to occur on a low-to-high transition or a high-to-low transition.
	MultTriggerMode	Indicates whether you want to use a single trigger to start the acquisition of a series of multiple frames (<code>TriggerToStart</code>) or whether you want to trigger each frame in a series of multiple frames with a separate trigger (<code>TriggerForEach</code>).

Digital Camera Operations

The following sections describe the operations that are specific to digital cameras. These operation include

- Setting the digital camera type (see the next section), and
- Using the output signals (see [page 67](#)).

Setting the Digital Camera Type

Frame grabber boards that support digital cameras can accept 8-bit, 10-bit, 12-bit, 14-bit, or 16-bit monochrome video from one RS-422 differential digital input (single-channel mode) or 8-bit monochrome video from two RS-422 differential digital inputs (dual-channel mode). Use the **DigitalCameraType** property to specify the type of input provided by your camera.

When using single-channel mode, the size of the image can be 2K pixels x 2K pixels, 4K pixels x 1K pixels, or 1K pixels x 4K pixels. When using dual-channel mode, the size of the image is always 1K pixels x 1K pixels. The image size is the size of the frame you want to digitize (maximum 4M pixels). For more information on establishing the size of the frame, refer to [page 62](#).

Using the Output Signals

Some frame grabber boards that support digital cameras can generate the following kinds of output signals:

- **Exposure output** – Use the **EnableExposePulse** property to enable and disable the frame grabber board to generate an exposure pulse to the camera. Use the **ExposePulseDuration** property to specify the exposure pulse width, in microseconds. Use the **ExposePulsePolarity** property to specify the active polarity of the exposure pulse (active high or active low).
- **Reset output** – Use the **ResetCamera** method to reset the digital camera.

Allocating a Frame Buffer

Before you can perform an acquisition, you must have a frame buffer available to hold the acquired data. To allocate the frame buffer, use the **AllocateFrame** method. For each frame buffer that you allocate, **AllocateFrame** returns a unique frame index. Frame buffers use zero-based numbering; that is, the first frame index is number 0, the second frame index is number 1, and so on.

[Table 21](#) lists the parameters of the **AllocateFrame** method. These parameters provide information about the frame that you want to digitize and are used to determine the size of the frame buffer.

Table 21: AllocateFrame Parameters

Parameter	Description
Height	Number of lines in the frame.
Width	Number of pixels per line in the frame.
Top	Position of the first line in the frame.
Left	Position of the first pixel in a line in the frame.
Type	Type of frames (or fields) that you want to acquire (interlaced frames, fields, or noninterlaced frames).

Table 21: AllocateFrame Parameters (cont.)

Parameter	Description
HorizontalScale ^a	Percentage that the frame has been scaled horizontally.
VerticalScale ^a	Percentage that the frame has been scaled vertically.
StorageMode	Format in which to store the image in memory (monochrome, YUV (native, unfiltered RGB format), RGB (32-bit RGB format), or RGB16 (16-bit RGB format)).

- a. For monochrome boards, not all percentages are valid. When you specify a scaling percentage, **AllocateFrame** uses the supported percentage closest to the percentage you specify. To determine the actual percentage used, use the **FrameHorizontalScale** or **FrameVerticalScale** property.

To specify the current frame buffer (the frame buffer you want to acquire to), use the **ActiveFrame** property. If you want to view information about the current frame buffer, use the associated read-only property. For example, to view the number of lines allotted in the current frame buffer, use the **FrameHeight** property. In addition, to determine the number of bytes per pixel for the storage mode you are using, use the **FrameBytesPerPixel** property.

To determine whether your frame grabber board supports programming the size of the frame, you can query the board. [Table 22](#) lists the information about the frame that you can query for, the method to use to perform the query, and the key to use in the query method.

Table 22: Getting Information About the Frame

Information	Method	Key
Whether the frame grabber board supports programming the frame.	QueryInputCaps	OLC_FG_IC_DOES_FRAME_SELECT
The range for frame left. ^a	QueryInputCaps	OLC_FG_IC_FRAME_LEFT_LIMITS
The range for frame width. ^a	QueryInputCaps	OLC_FG_IC_FRAME_WIDTH_LIMITS
The range for frame top. ^a	QueryInputCaps	OLC_FG_IC_FRAME_TOP_LIMITS
The range for frame height. ^a	QueryInputCaps	OLC_FG_IC_FRAME_HEIGHT_LIMITS
For monochrome frame grabber boards, the range for horizontal scaling. ^{a, b}	QueryInputCaps	OLC_FG_IC_FRAME_HINC_LIMITS
For monochrome frame grabber boards, the range for vertical scaling. ^{a, b}	QueryInputCaps	OLC_FG_IC_FRAME_VINC_LIMITS
For color frame grabber boards, whether the board supports hardware scaling.	ColorQueryInterface	OLC_QUERY_COLOR_INTERFACE_HARDWARE_SCALING
The maximum number of pixels allowed in a frame. ^c	QueryInputCaps	OLC_FG_IC_MAX_FRAME_SIZE
The number of bytes in a pixel.	QueryInputCaps	OLC_FG_IC_PIXEL_DEPTH

a. Includes the upper and lower limits, granularity, and nominal value.

b. For monochrome boards, the range returned is in pixel increments, not in percentages. To determine the percentage associated with each pixel increment in the range, use the following equation: $percentage = 100 / pixel\ increment$.

c. The frame height times the frame width must be less than or equal to the value returned.

Notes: If you modify any of the parameters in **AllocateFrame** after you acquire a frame, the data in the frame buffer may become invalid. If you have already acquired data and are preparing to initialize the system for the next acquisition, make sure that you first process and store any frame data that you want to keep from the previous acquisition.

The number of frame buffers that you can allocate depends on the amount of system memory you reserved for use by the device driver. Because memory is allocated on 4 MB boundaries, you may not be able to allocate the number of frame buffers you want even if you have reserved a sufficient amount of system memory. For information on how much system memory you must reserve for a particular number of frame buffers, refer to [Appendix A](#). For information on reserving system memory, refer to your Windows documentation.

Once you have created a frame buffer, you can perform an acquisition. For more information, refer to [page 80](#). After you perform an acquisition, you can move the data for processing. For more information, refer to [page 82](#).

When you finish acquiring images, release each frame buffer with the **DestroyFrame** method. After you destroy a frame buffer, do not use its frame index again in subsequent calls unless you create a new frame buffer (with **AllocateFrame**) using that particular frame index. If you do create a new frame buffer with the same frame index as a previously-destroyed frame buffer, you can make no assumptions about the initial contents of the frame buffer.

Passthru Operations

This section describes the following:

- Performing a passthru operation (see the next section),
- Adjusting the source origin (see [page 74](#)),
- Modifying the passthru LUT (see [page 75](#)),
- Taking a snapshot (see [page 77](#)), and
- Creating overlays (see [page 78](#)).

Performing a Passthru Operation

In a passthru operation, data is continually acquired to display memory (memory on your video display adapter) until some event occurs. Typically, you use passthru to view the image data (in as close to real time as possible) for the purpose of focusing or positioning the camera.

The passthru image is displayed in the client region of a window. If the passthru image is larger than the window's client region, only the part of the image that is the size of the client region is displayed. If the passthru image is smaller than the window's client region, the leftover area of the window has no image displayed on it and remains unchanged.

You can choose to devote all your system resources to the passthru operation (synchronous passthru), or you can start the process and then free your system to perform other tasks while the passthru operation occurs (asynchronous passthru). You cannot interrupt a synchronous passthru operation.

Before starting a passthru operation, you must allocate a single frame buffer with the **AllocateFrame** method. For more information, refer to [page 68](#).

The device drivers of some frame grabber boards also support continuous-acquire passthru operations. Continuous-acquire passthru operations allow you to continually display passthru data while you acquire one or more frames of passthru data into a circular buffer. If your board supports continuous-acquire passthru, you must allocate multiple frame buffers with the **AllocateFrame** method before starting the passthru operation.

To start a passthru operation, use the **StartPassthru** method. To stop an asynchronous passthru operation, use the **StopPassthru** method.

For asynchronous passthru operations only, the DT-Active Open Layers control provides the **OnPassthruNotify** event. This event is generated by the DT-Active Open Layers control each time a passthru frame is acquired/displayed.

To determine if your frame grabber board supports passthru, use the **QueryPassthruCaps** method with the key `OLC_FG_PC_PASSTHRU_MODE_LIMITS`.

The information returned by the **QueryPassthruCaps** method is defined in [Table 23](#).

Table 23: Passthru Values Returned

Passthru Value	Description
<code>OLC_FG_PASSTHRU_ASYNC_BITMAP</code>	Asynchronous bitmap passthru operations are supported.
<code>OLC_FG_PASSTHRU_SYNC_BITMAP</code>	Synchronous bitmap passthru operations are supported.
<code>OLC_FG_PASSTHRU_ASYNC_BITMAP_EXTENDED</code>	Continuous-acquire passthru operations are supported.

Adjusting the Source Origin

Typically, the upper left corner of the display and the upper left corner of the acquired image are identical. During a passthru operation, you can select any other point in the acquired image to become the upper left corner of the display (source origin). This allows you to pan and scroll the image on the display to display part of the acquired image.

For example, if the passthru image is 640 x 480 and the source origin is 300 x 300, the point at 300 x 300 in the passthru image is placed in the upper left corner of the window. The size of the displayed image is 340 x 180; the leftover area of the window has no image displayed on it and remains unchanged.

Figure 10 illustrates adjusting the source origin.

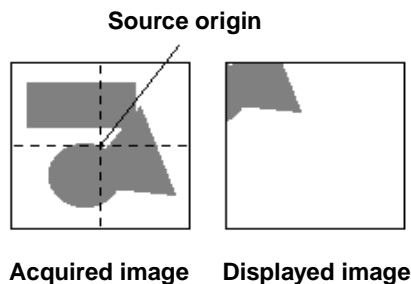


Figure 10: Source Origin Example

To determine whether your frame grabber board supports programming the source origin, you can use the **QueryPassthruCaps** method to query the board. [Table 24](#) lists the information about the source origin that is returned by **QueryPassthruCaps** and the key to use in **QueryPassthruCaps** to get the information.

Table 24: Getting Information About the Source Origin

Information	Key
Whether the frame grabber board supports setting the source origin.	OLC_FG_PC_DOES_SOURCE_ORIGIN
The range for the x-coordinate. ^a	OLC_FG_PC_SRC_ORIGIN_X_LIMITS
The range for the y-coordinate. ^a	OLC_FG_PC_SRC_ORIGIN_Y_LIMITS

a. Includes the upper and lower limits, granularity, and nominal value.

To set the x-coordinate and y-coordinate of the source origin, use the **XPassthruOrigin** and **YPassthruOrigin** properties.

Notes: Use a Windows POINT structure to define the pixels to set as the source origin before calling **XPassthruOrigin** and **YPassthruOrigin**.

Some frame grabber boards require alignment of pixels to 4-pixel, 8-pixel, or 32-pixel boundaries. The typical alignment is 4-pixel boundaries (right and left edges). Refer to your board-specific documentation for more information.

Modifying the Passthru LUT

In addition to the ILUT (described on [page 55](#)), you can use the passthru LUT to affect the displayed image during passthru. By using the passthru LUT with the ILUT, you can change the display image without altering the ILUT itself. This allows you to display reversed or otherwise enhanced images without disrupting the underlying color settings.

When using the passthru LUT, the values in the ILUT are applied to the image first. Then the values in the passthru LUT are applied.

By default, passthru operations load the Windows system palette with 128 grayscale RGB values for display and use the default passthru LUT of 256 monotonically increasing grayscales. If you want to use these monochrome values in the passthru LUT, use the **LoadDefaultPassthruLUT** method. If you want to modify the passthru LUT so that the frame grabber board uses false coloring, use the **LoadPassthruLUT** method.

For each entry in the passthru LUT, the index of the closest matching RGB value in the Windows system palette is used. If you want an exact color to display during passthru, use the **ExtendPassthruPalette** method to add that color to the system palette. This guarantees that the color is available when the frame grabber board attempts to find the closest match in the Windows system palette.

To determine whether your frame grabber board supports programming the passthru LUT, you can use the **QueryPassthruCaps** method to query the board. [Table 25](#) lists the information about the passthru LUT that is returned by **QueryPassthruCaps** and the key to use in **QueryPassthruCaps** to get the information.

Table 25: Getting Information About the Passthru LUT

Information	Key
Whether the frame grabber board supports programming the passthru LUT.	OLC_FG_PC_DOES_PASSTHRU_LUT
The maximum number of entries in the passthru LUT.	OLC_FG_PC_MAX_PLUT_INDEX
The maximum value for a red, green, or blue element in the passthru LUT.	OLC_FG_PC_MAX_PLUT_VALUE

Table 25: Getting Information About the Passthru LUT (cont.)

Information	Key
The maximum number of entries that can be added to the system palette array.	OLC_FG_PC_MAX_PALETTE_INDEX
The maximum value for a red, green, or blue element in the system palette.	OLC_FG_PC_MAX_PALETTE_VALUE

Taking a Snapshot

If your frame grabber board supports it, you can capture one frame of an asynchronous passthru operation without stopping the passthru operation. This is called taking a snapshot. To determine if your board supports snapshots, use the **QueryPassthruCaps** method with the key `OLC_FG_DOES_PASSTHRU_SNAPSHOT`.

Before taking a snapshot, use the **AllocateFrame** method to allocate a frame buffer that is large enough to hold the snapshot.

While the asynchronous passthru is in progress, use the **PassthruSnapshot** method to capture the single passthru image. The frame grabber board writes the captured image to the specified frame buffer. Note that you cannot use **PassthruSnapshot** to capture overlays on the passthru display. To capture both the display and the overlay, call the **OverlaySurfaceOnFrame** method immediately after calling **PassthruSnapshot**.

After you acquire the snapshot, you can process the data as you would any other acquired frame. For more information, refer to [page 82](#).

Creating Overlays

Overlays allow you to place or overlay an image on top of another image. You can create and work with overlays both before and during the passthru operation.

The first step in creating an overlay is to allocate a surface buffer (or surface) with the **AllocateSurface** method. The surface must be large enough to overlay the entire frame area. An overlay surface cannot be smaller than its entire underlying image. Each overlay must have its own surface.

Use the **SurfaceCount** property to determine the number of surfaces you have allocated. Use the **ActiveSurface** property to specify the overlay to make visible.

After you allocate a surface buffer, get a handle to a device context (DC) for the surface buffer with the **GetSurfaceDC** method. DCs are Windows Graphical Device Interface (GDI) objects. Refer to your Windows SDK documentation for more information about DCs and GDIs.

After getting a handle to a DC, use standard GDI functions to draw onto the DC as you would in a window or in another DC (such as a printer DC). This provides you with a graphical interface (Windows API) to render shapes onto overlay surfaces. You can also use an existing bitmap (.BMP) file as an overlay. Use the **ImageFile** property to specify the file to use. You can erase the contents of a surface at any time with the **EraseSurface** method.

When you have finished working with the DC, release the surface DC with the **ReleaseSurfaceDC** method. Repeat the **GetSurfaceDC/edit/ReleaseSurfaceDC** process for each surface buffer you allocate.

Notes: Do not retain a handle to a DC for any longer than necessary. When you use the **GetSurfaceDC** method, the frame grabber board locks the surface buffer and other functions cannot create new DCs for the surface buffer. In addition, a deadlock condition can occur if one function retains a handle while another function is waiting for it to be released.

Overlays that you can see and that allow you to see the image underneath at the same time are called translucent. Translucent overlays are useful when you want to align overlays with the underlying image. To specify whether you want the current overlay to be translucent or opaque, use the **Translucent** property.

You can also use the **ColorKey** property to specify the source keying color (the color that is replaced by the passthru image) for the active surface.

Before calling the passthru operation, you must enable overlays, turning them on for the specified board, by using the **EnableOverlays** method.

For animation, you must create two surface buffers. One surface buffer is used as the front buffer (the displayed buffer); the other surface buffer is the back buffer (the buffer you are drawing to). To specify the front buffer (the surface to overlay on the next refresh), use the **ActiveSurface** property.

When you are finished with the passthru operation using overlays, disable overlays, turning them off for the specified board, by using the **DisableOverlays** method. Then, use the **DestroySurface** method to destroy each surface buffer that you allocated to free as much memory as possible.

Acquisition Operations

Once you have established the frames that you want to digitize (see [page 62](#)) and allocated frame buffers to store the frames (see [page 68](#)), you can acquire the frames.

You can choose to devote all your system resources to the acquisition (a synchronous acquisition), or you can start the process and then free your system to perform other tasks while the acquisition occurs (an asynchronous acquisition). You cannot interrupt a synchronous acquisition.

To acquire one or more frames, use the **AcquireFrames** method. You specify the following:

- The number of frame buffers that you allocated with the **AllocateFrame** method,
- Whether you want to acquire data synchronously or asynchronously, and
- The location in which the frame buffer indexes are stored.

The frame grabber board acquires and stores the first image in the first frame buffer you allocated and then repeats the process until all the frame buffers have been filled.

During the acquisition, use the **IsAsyncAcquireFramesDone** method to determine whether the acquisition has completed. If the acquisition has completed, **IsAsyncAcquireFramesDone** returns `TRUE`; otherwise, **IsAsyncAcquireFramesDone** returns `OLC_STS_PENDING` if the operation has not yet started or `OLC_STS_ACTIVE` if the operation has started but is not yet complete.

In addition, for asynchronous acquisitions only, the DT-Active Open Layers control provides the **OnAcquireNotify** event. This event is generated by the DT-Active Open Layers control when the acquisition has completed.

To stop an asynchronous acquisition, use the **StopAsyncAcquireFrames** method.

Note: If you want to acquire frames while displaying the images on the screen, you can perform a continuous-acquire passthru operation. For more information, refer to [page 72](#).

Once the acquisition is complete, you can display the image data in a window or process the image data. For more information, refer to the next section.

Image Processing Operations

After performing an acquisition, you can display an acquired image in a window using the **DrawAcquiredFrame** method or save the acquired frame to a bitmap image file using the **SaveAcquiredFrame** method. Note that for the DT3154 frame grabber board only, you can separate the red, green, and blue planes of the image using the **ExtractRGBPlanesEx** method and draw them to a window using the **DrawRGBPlane** method.

Note: **ExtractRGBPlanesEx** automatically allocates buffers for the red, green, and blue signals. When you are done using the buffers, your application must free the memory.

In addition, you can move the image data for processing, as follows:

- **Read operation** – Allows you to move data from a frame buffer to a user array.

Use the **ReadFrameEx** method to transfer data from a rectangular region of a frame buffer to a user array. This method automatically allocates the memory for the user array. The size of the array depends on the width and height of the frame and on the image storage mode. For example, if the image storage mode is monochrome (1 byte per pixel), then the user array will contain two-dimensional data consisting of (1*Frame Width) x (1*Frame Height) pixels. If the image storage mode is RGB (4 bytes per pixel), then the user array will contain two-dimensional data consisting of (4*Frame Width) x (1*Frame Height) pixels.

Specify the upper left pixel of the rectangle and the rectangle's width and height. The frame grabber board transfers the data contiguously.

Note: When you are done using the user array, your application must free the memory.

- **Write operation** – Allows you to move data from a user array to a frame buffer.

Before performing a write operation, make sure that you have allocated a frame buffer and that the frame buffer is large enough to hold all the data you want to transfer. For more information, refer to [page 68](#).

Use the **WriteFrame** method to transfer data from a rectangular region of a user array to a frame buffer. Specify the upper left pixel of the rectangle and the rectangle's width and height. The frame grabber board transfers the data contiguously.

- **Copy operation** – Allows you to move data either from one frame buffer to another or from one part of a frame buffer to another part of the same frame buffer.

Use the **CopyFrameRect** method to transfer data directly from one frame buffer to another or from one part of a frame buffer to another part of the same frame buffer. Define the rectangle to copy by specifying the upper left pixel of the rectangle and the rectangle's width and height, and specify the pixel location where you want the upper left corner of the rectangle to be placed in the destination frame. The frame grabber board transfers the data contiguously.

For example, you can copy a 500 x 300 rectangle from the upper left part of one frame buffer and transfer it to a 500 x 300 rectangle in the center of another frame buffer.

Make sure that if you place more than one rectangle in a single frame, the rectangles do not overlap; if they do, the board may return an error. Even if the operation completes successfully, the data in the rectangles may be incorrect.

Make sure that the entire rectangle fits fully within the target frame; if it does not, the board returns an error. The board does not clip the rectangle to make it fit within the frame if the rectangle exceeds the frame's edges.

Digital I/O Operations

Some frame grabber boards provide digital input lines and/or digital output lines. The digital I/O signals are simple register-driven, TTL-level signals that you can use for any purpose, such as controlling or actuating external devices.

You can use the **DigInCaps** property to determine the number and location of the digital input lines supported by the board. You can use the **DigOutCaps** property to determine the number and location of the digital output lines supported by the board.

If your board supports programmable digital I/O, use the **DigIOConfig** property to configure the lines. Create a bit mask in which each bit represents a different line, interpreted from the low order bit (bit 0 represents line 0, bit 1 represents line 1, and so on). When a bit is set to 1, the corresponding line is configured as an output. When a bit is set to 0, the corresponding line is configured as an input.

To write data to a digital output line, use the **DigIOPattern** property. Create a bit mask correlating to the high/low settings of each of the output lines. Set the bit to 1 to write a high-level TTL signal; set the bit to 0 to write a low-level TTL signal. For example, assume that you set **DigIOConfig** to Ah (1010), where lines 0 and 2 are configured for input and lines 1 and 3 are configured for output. If you set **DigIOPattern** to 0h (0000), low-level signals are written to output lines 1 and 3. Note that values written to input lines are ignored.

To read data from a digital input line, use the **DigIOPattern** property. The return value correlates to the high/low values of each of the input lines. If a bit is set to 1, the input is high; if a bit is set to 0, the input is low. Using the previous example, if the return value is 0h (0000), the input at input lines 0 and 2 is low. Note that the value returned from an output line is the last value written to that line.



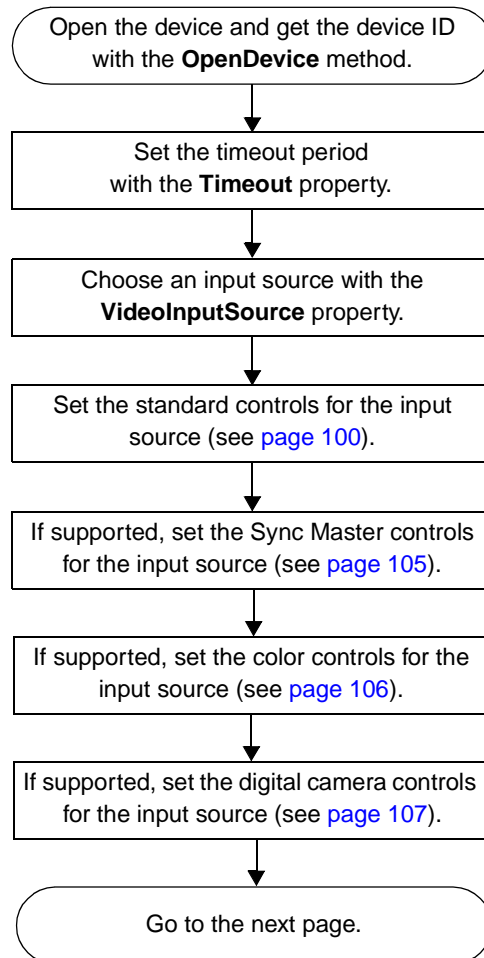
Programming Flowcharts

Frame Acquisition	89
Passthru to Display without Overlays	93
Passthru to Display with Overlays	96

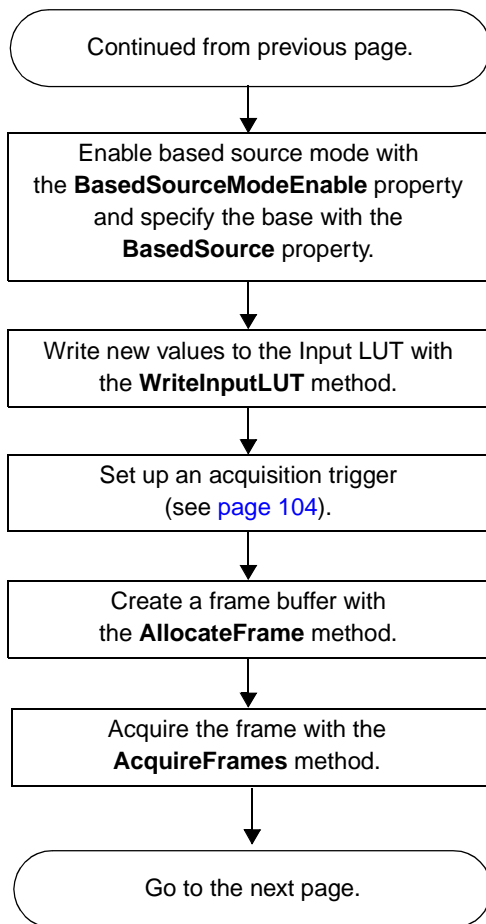
This chapter provides a series of flowcharts showing an overview of the steps required to perform each of the operations supported by the DT-Active Open Layers control. Although the flowcharts do not show error/status checking, it is recommended that you check for error/status messages after using each property, method, or event.

Note: The properties described in this chapter are appropriate for Visual Basic only. If you are using Visual C++, use the associated Visual C++ methods instead. For example, to set and/or return the pixel clock source, use the Visual C++ **SetClockSource** and **GetClockSource** methods instead of the Visual Basic **ClockSource** property. For more information about Visual C++ methods, refer to the DT-Active Open Layers online help.

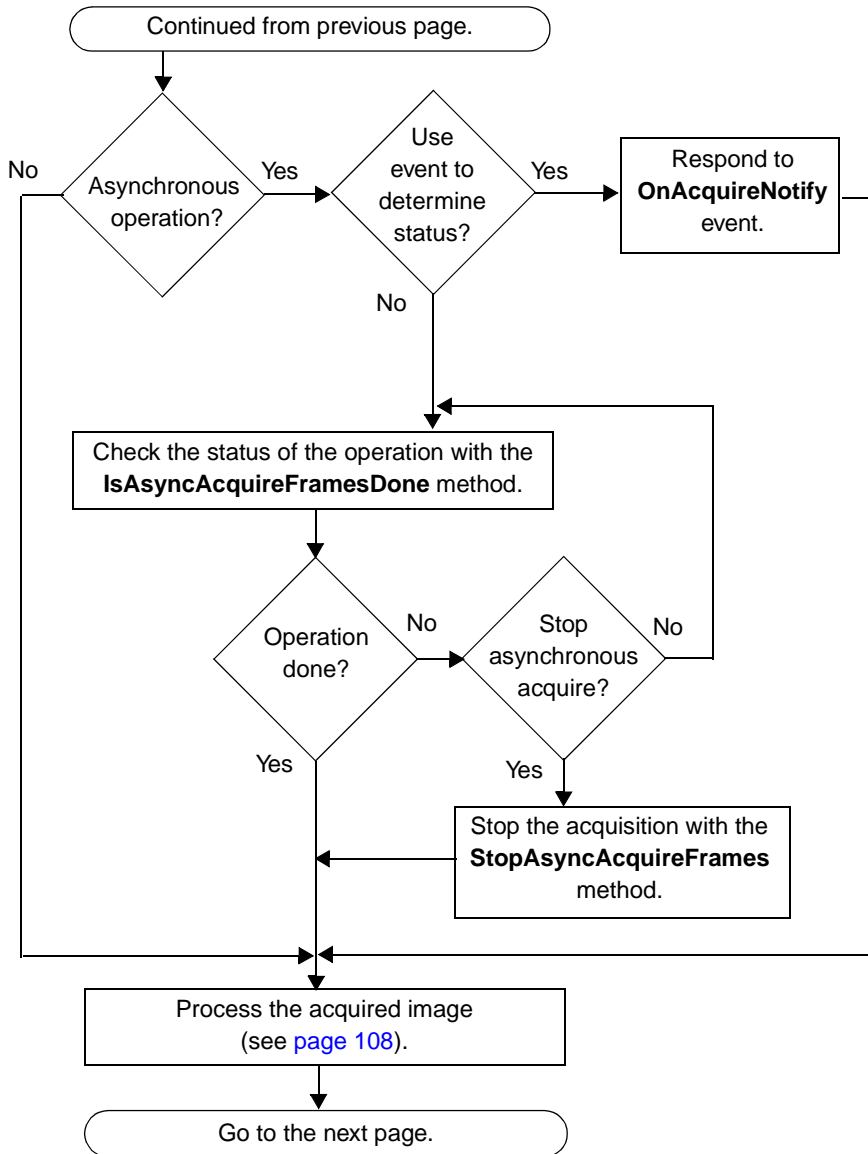
Frame Acquisition



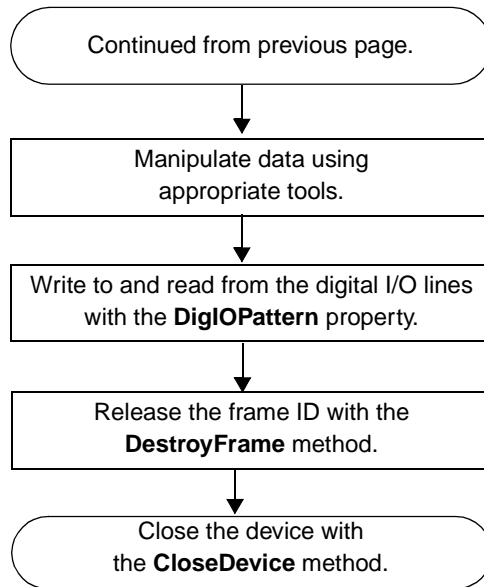
Frame Acquisition (cont.)



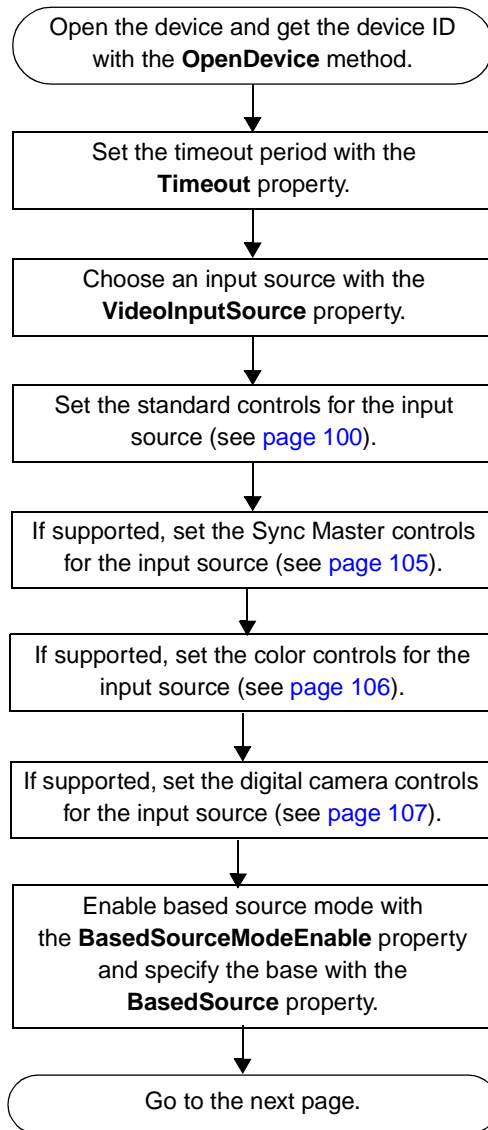
Frame Acquisition (cont.)



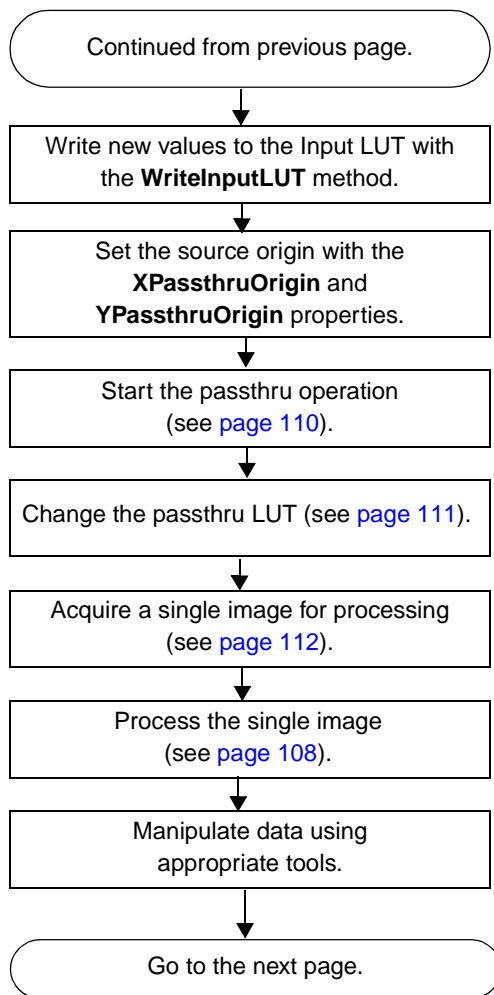
Frame Acquisition (cont.)



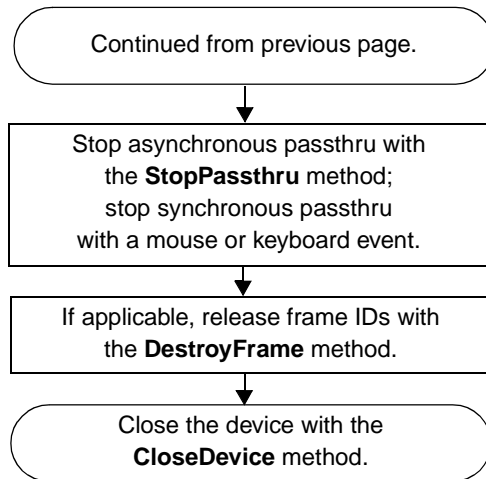
Passthru to Display without Overlays



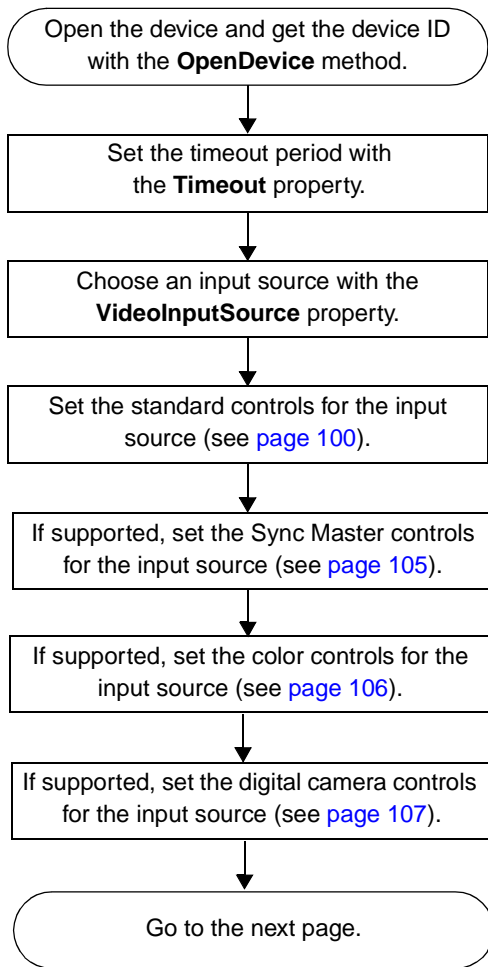
Passthru without Overlays (cont.)



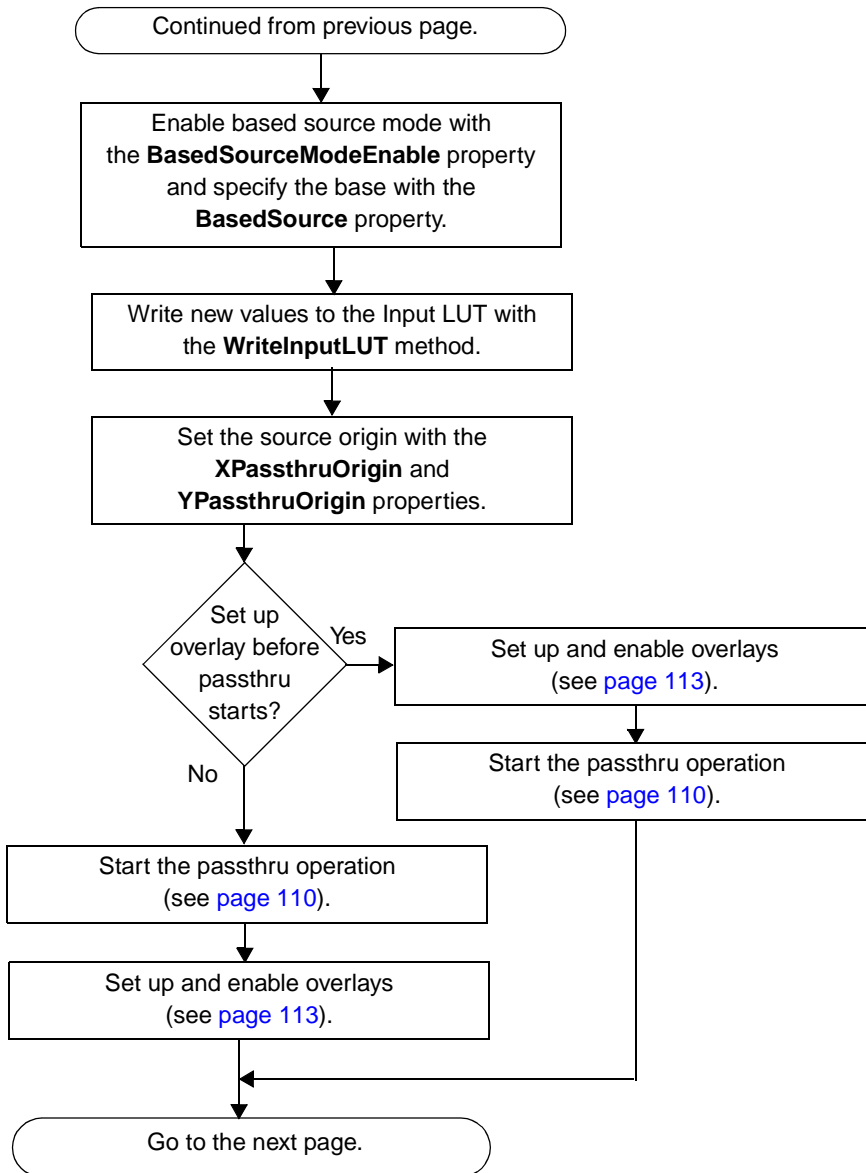
Passthru without Overlays (cont.)



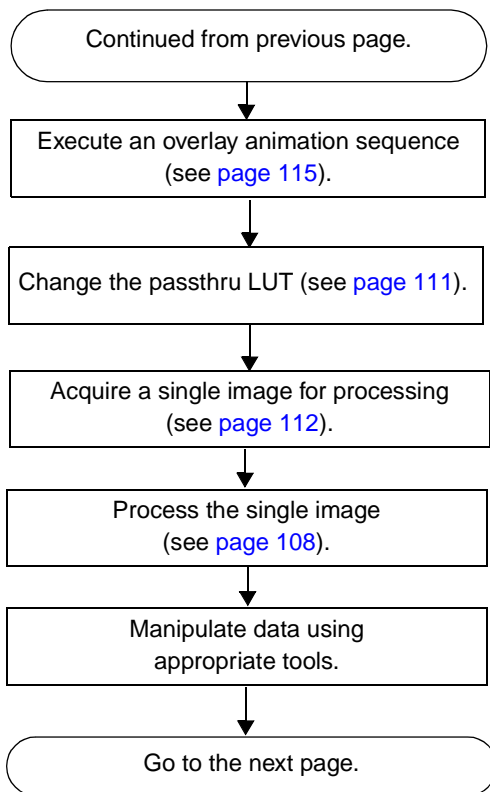
Passthru to Display with Overlays



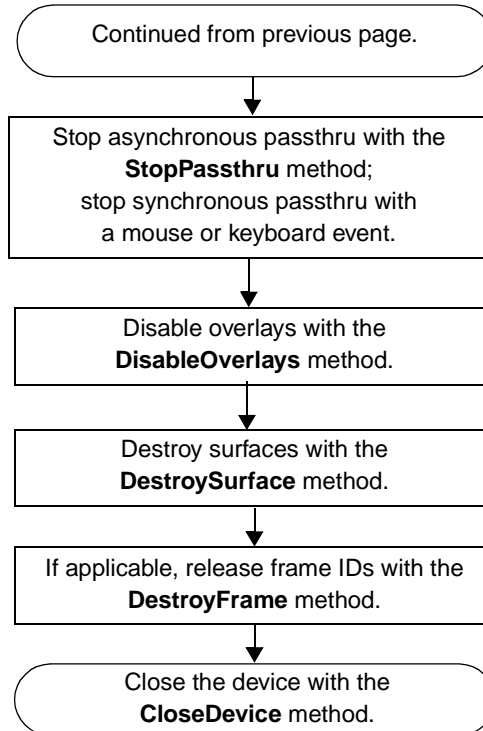
Passthru to Display with Overlays (cont.)



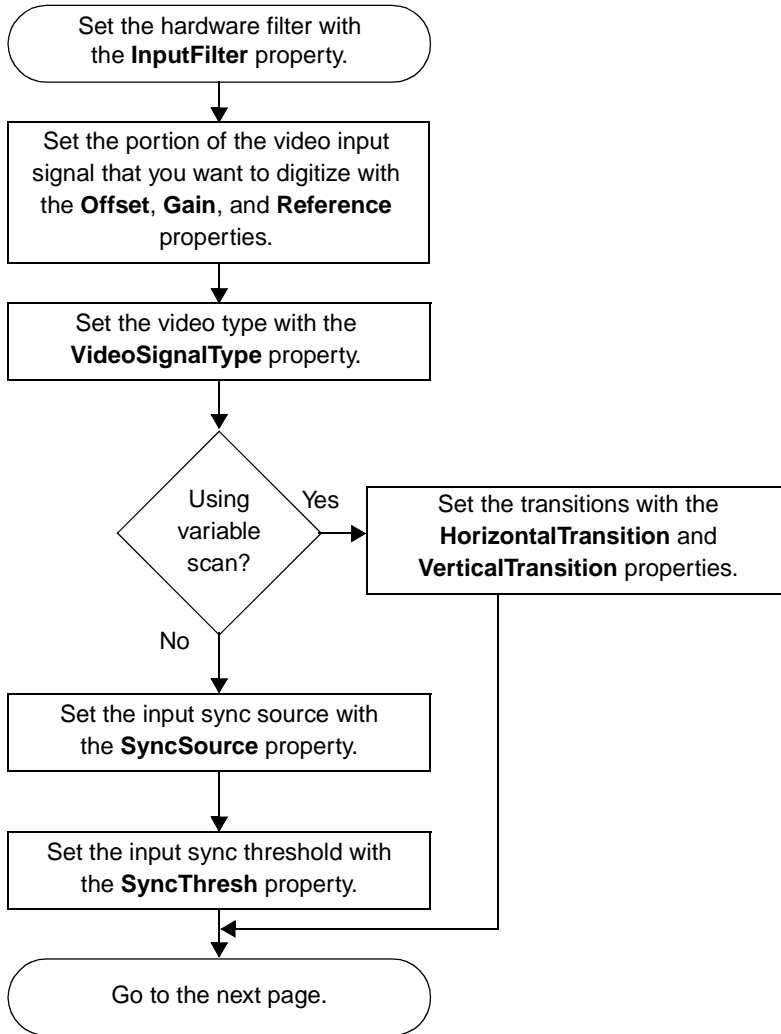
Passthru to Display with Overlays (cont.)

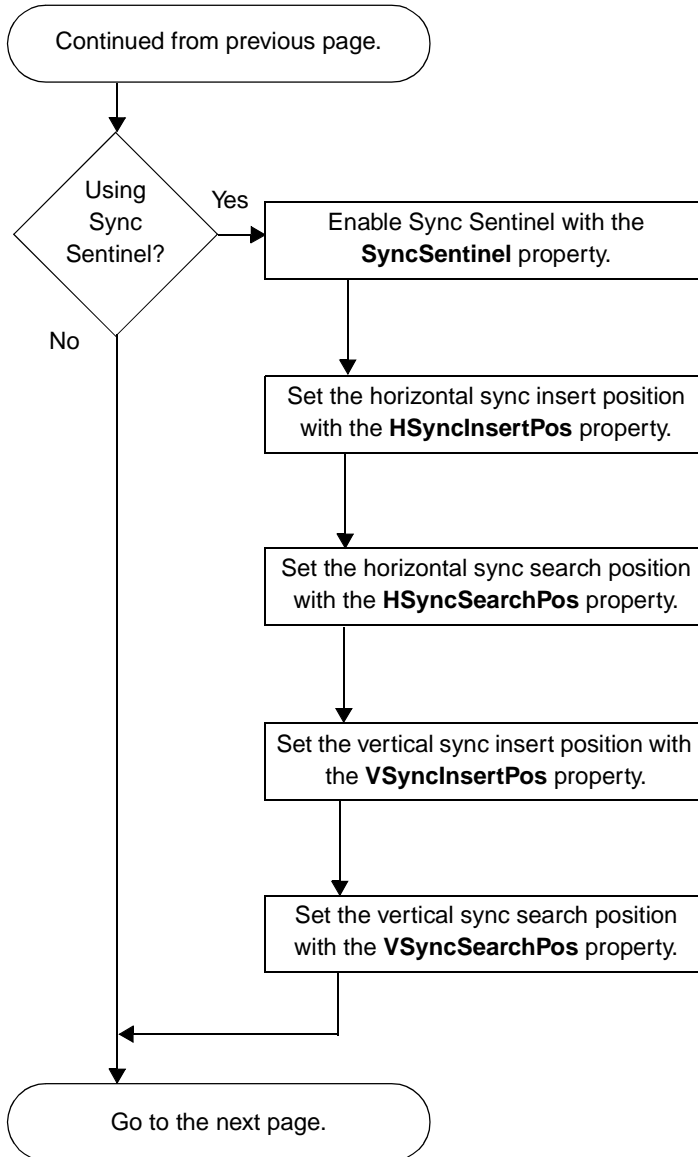


Passthru to Display with Overlays (cont.)

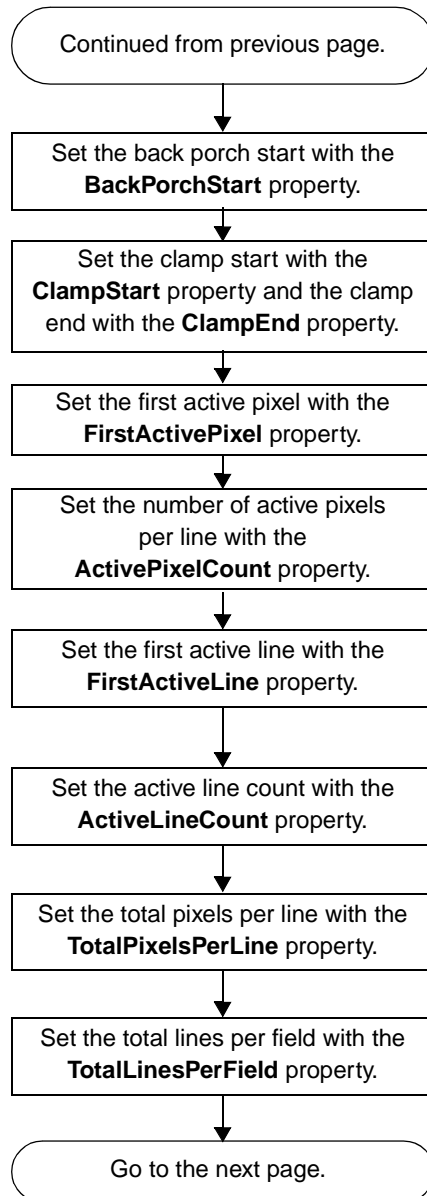


Set the Standard Controls for the Input Source

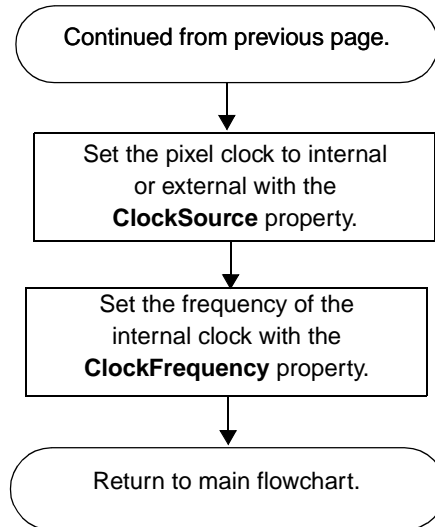


Set the Standard Controls for the Input Source (cont.)

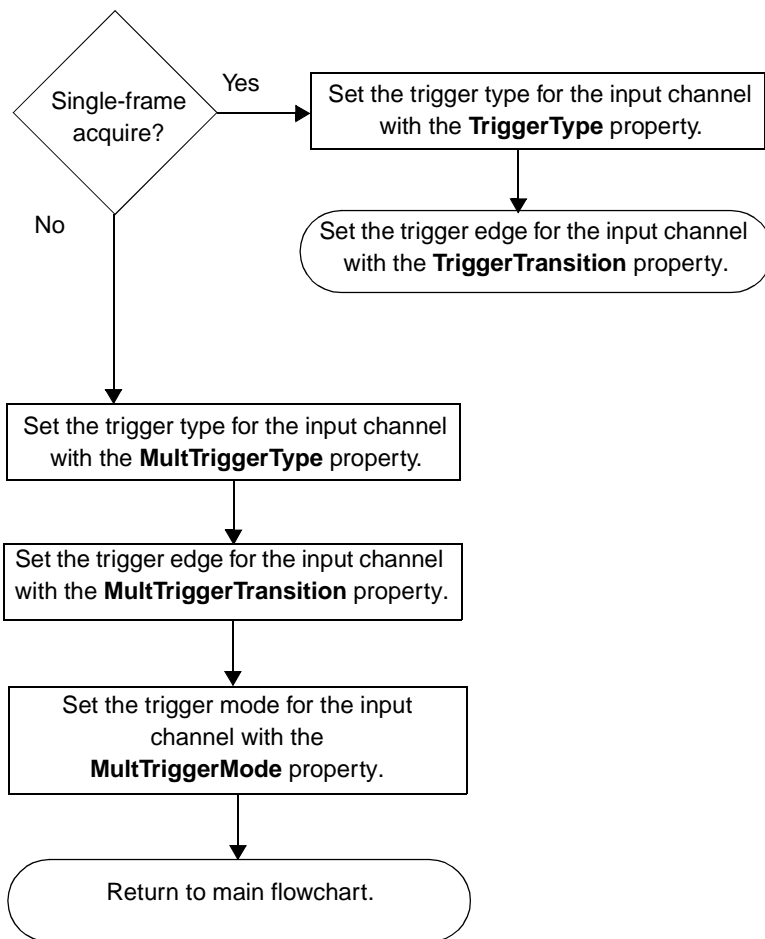
Set the Standard Controls for the Input Source (cont.)



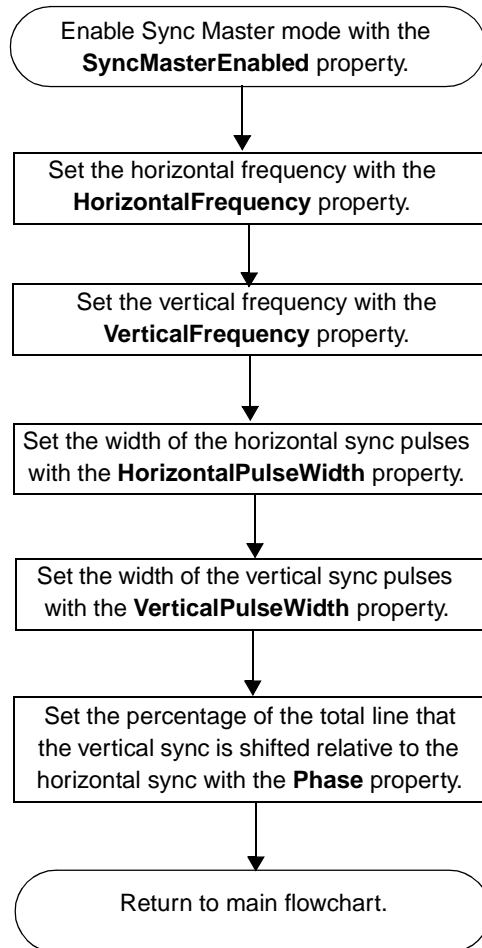
Set the Standard Controls for the Input Source (cont.)



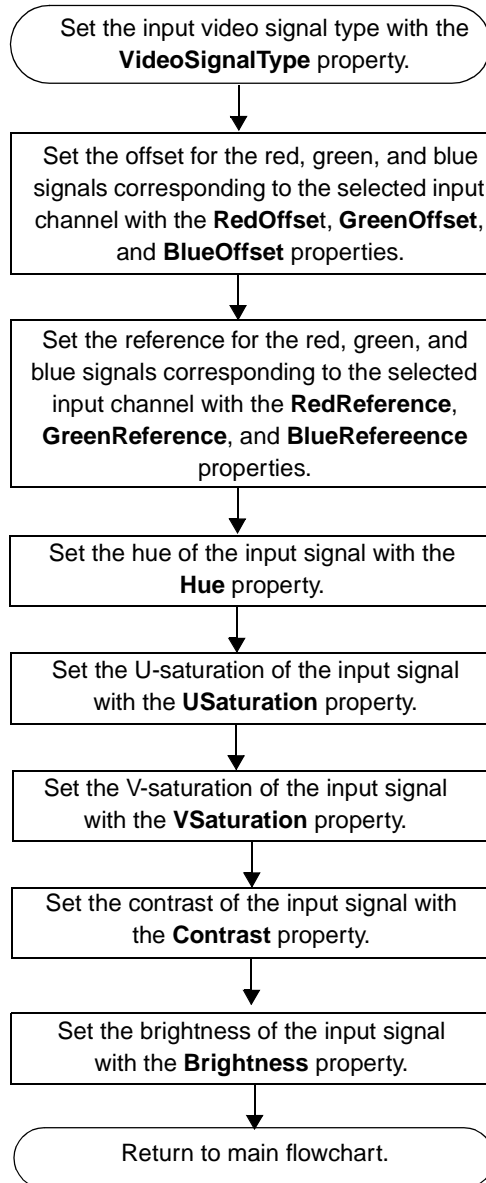
Set the Acquisition Trigger for the Input Source



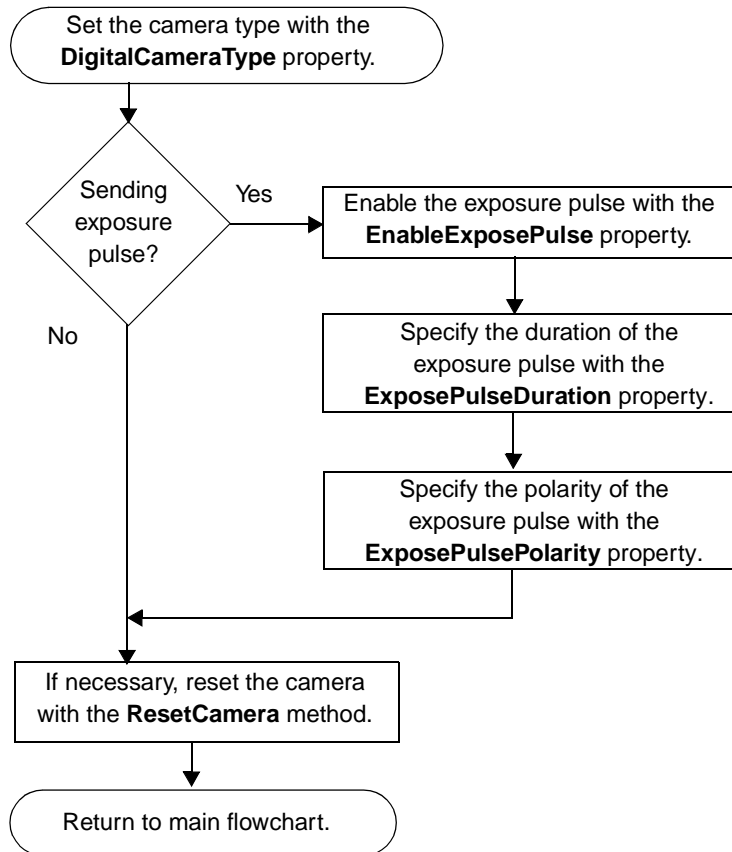
Set the Sync Master Controls for the Input Source



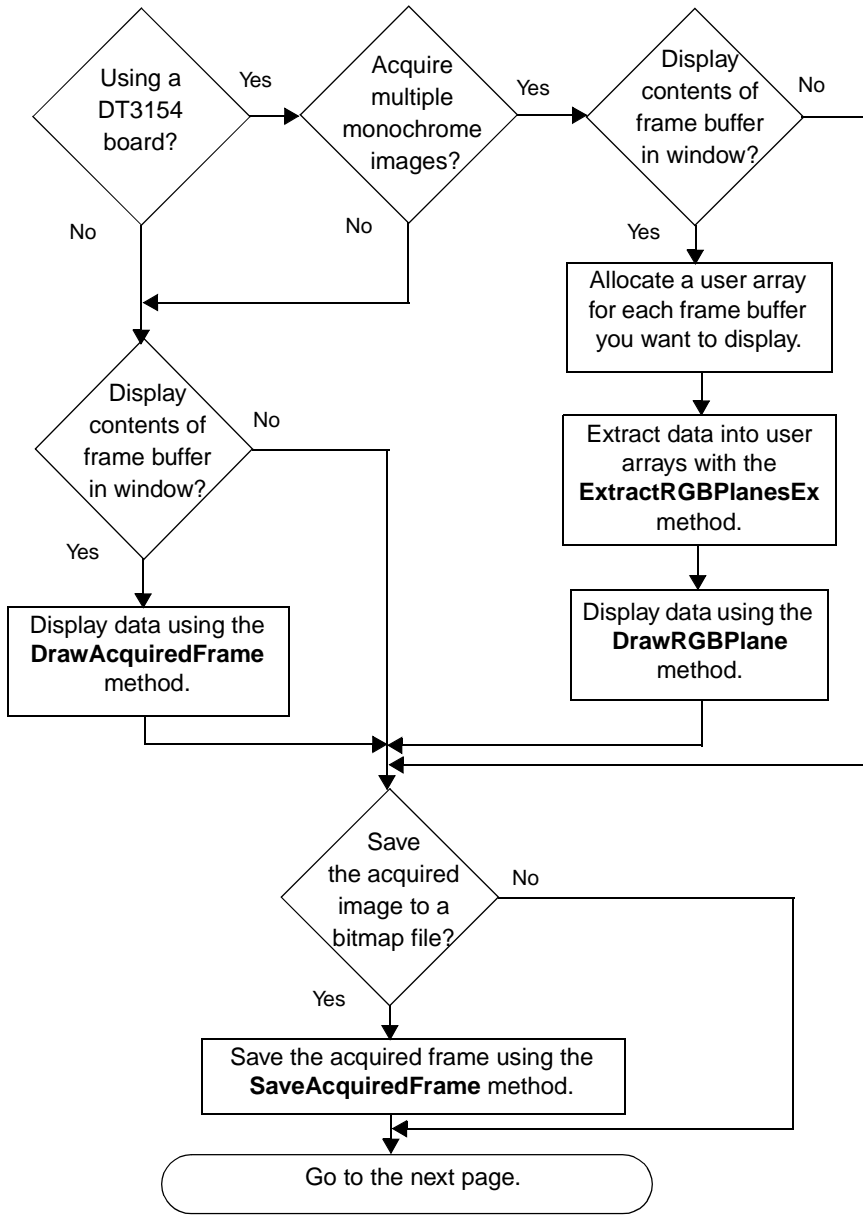
Set the Color Controls for the Input Source



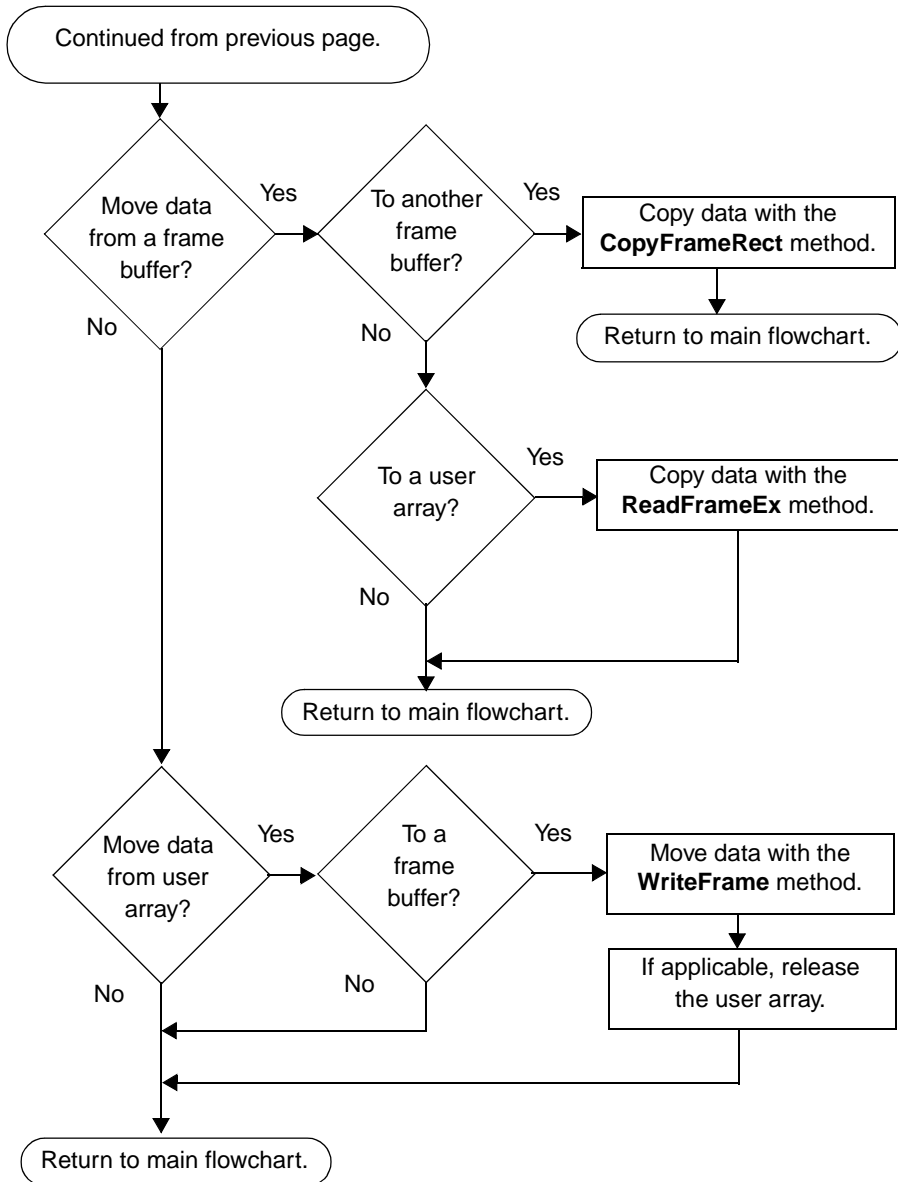
Set the Digital Camera Controls for the Input Source



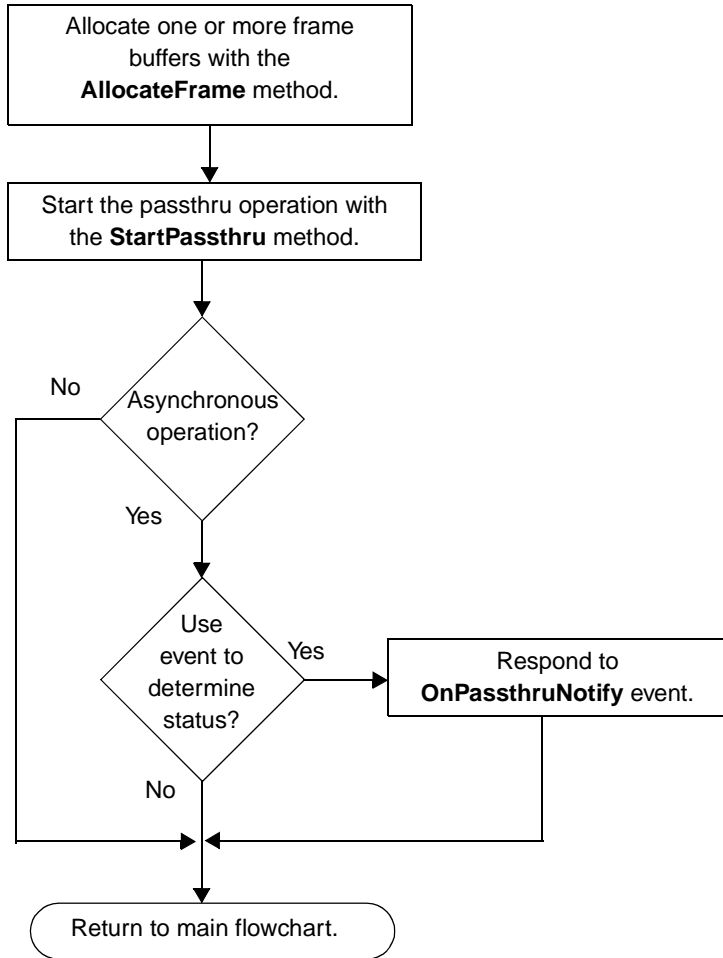
Process an Acquired Image



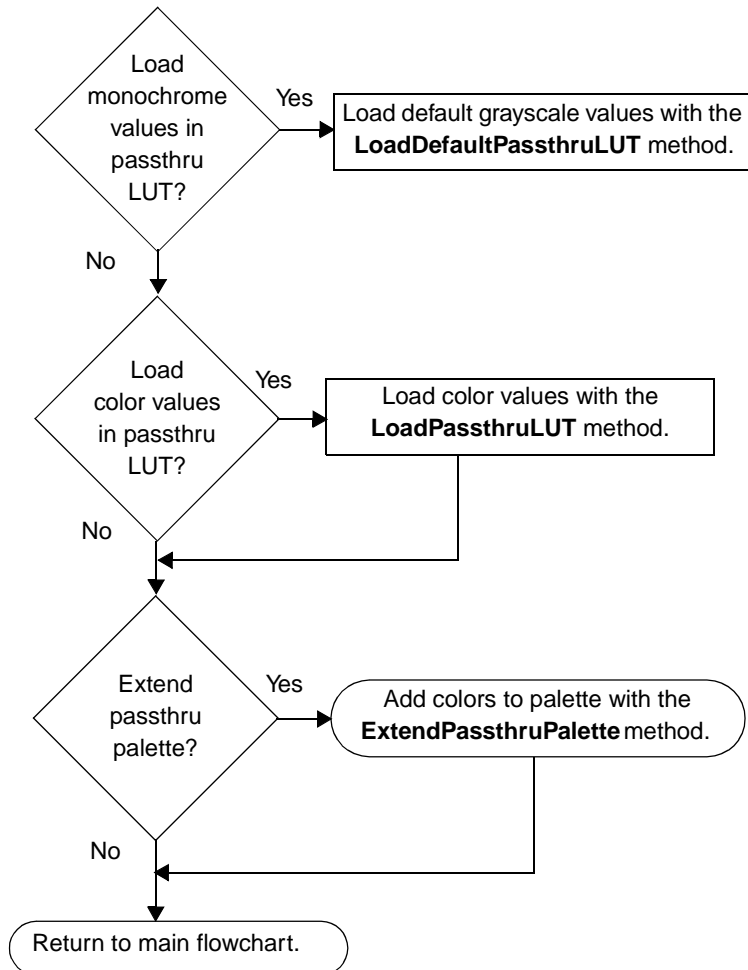
Process an Acquired Image (cont.)



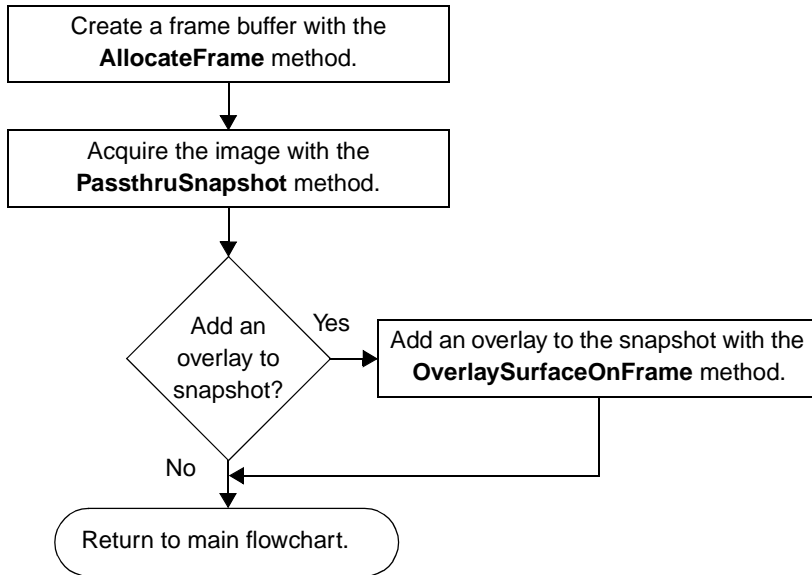
Start the Passthru Operation



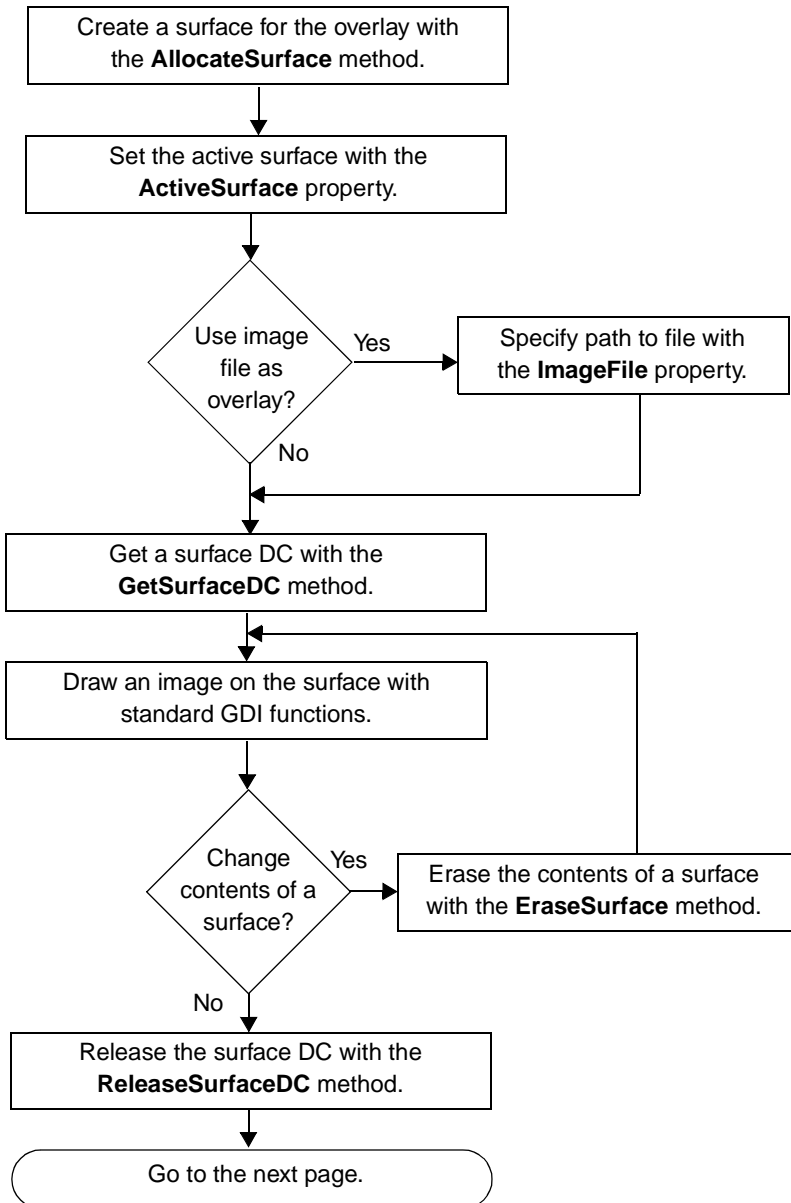
Change the Passthru LUT



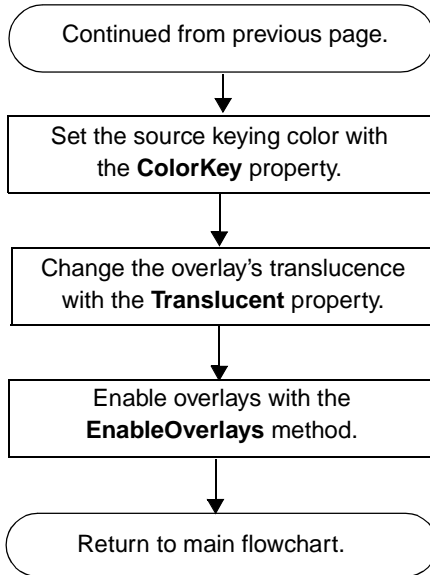
Take a Snapshot



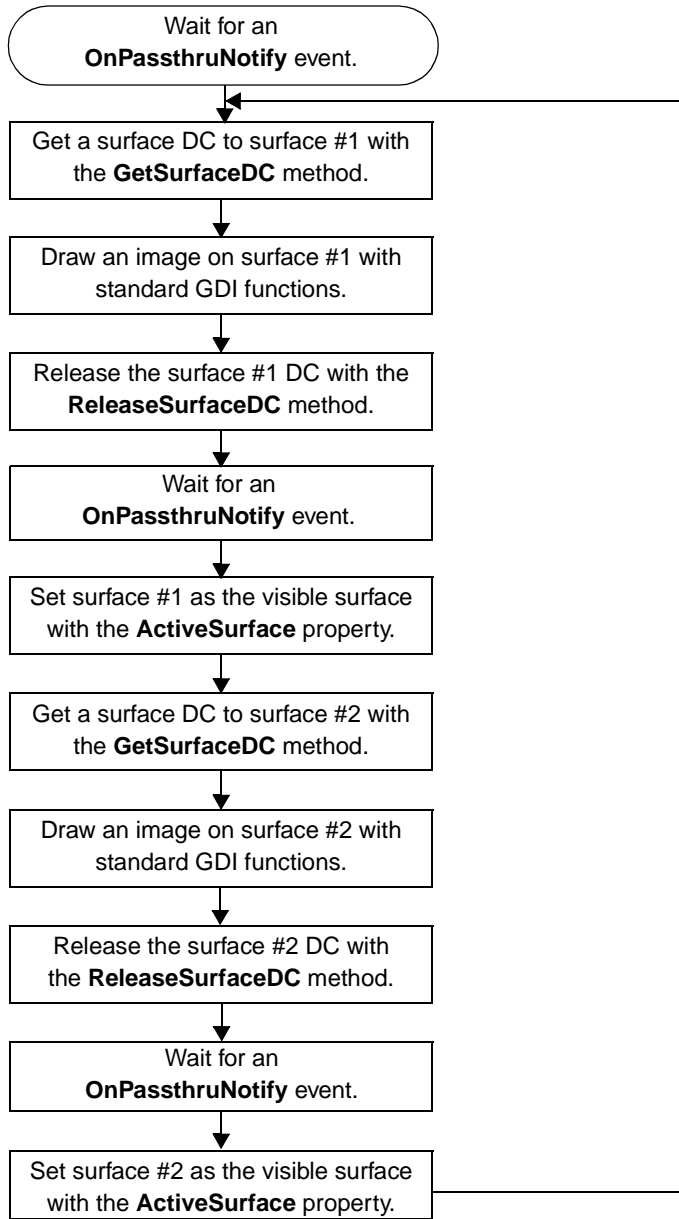
Set up an Overlay



Set up an Overlay (cont.)



Execute an Overlay Animation Sequence





Product Support

General Checklist	118
Service and Support	119

General Checklist

Should you experience problems using the DT-Active Open Layers control, please follow these steps:

1. Read all the appropriate sections of this manual. Make sure that you have added any “Read This First” information to your manual and that you have used this information.
2. Check your DT-Active Open Layers CD-ROM for a README file. If present, read this file for the latest installation and configuration information.
3. Check that you have installed and configured your software properly. For information, refer to the instructions on [page 6](#).
4. Check that you have installed your frame grabber board properly. For information, refer to your board-specific documentation.
5. Check that you have installed the device driver for your board properly. For information, refer to your board-specific documentation.

Note: If you are still having problems, please follow the instructions provided in the next section.

Service and Support

If you have difficulty using the DT-Active Open Layers control, the Data Translation Technical Support Department is available to provide technical assistance. Support upgrades, technical information, and software are also available.

You can always obtain the support needed. The first 90 days are complimentary, as part of the product's original warranty, to help you get your system running. If you call outside of this time frame, you can either purchase a support contract or pay a nominal fee (charged on a per-incident basis).

For "priority support," purchase a support contract. Support contracts guarantee prompt response and are very affordable; contact your local sales office for details.

Telephone Technical Support

Telephone support is normally reserved for original-warranty and support-contract customers. Support requests from noncontract or out-of-warranty customers are processed after requests from original-warranty and support-contract customers.

For the most efficient service, please complete the form on [page 120](#) and be at your computer when you call for technical support. The information on the form helps to identify specific system and configuration-related problems and to replicate the problem in house, if necessary.

You can reach the Technical Support Department by calling (508) 481-3700 x1401. Please have your contract number and serial number available when you call.

If you are located outside the USA, call your local distributor. The name and telephone number of the nearest distributor are provided in the Data Translation catalog.

Information Required for Technical Support

Name: _____ Phone _____

Contract Number: _____

Address: _____

Data Translation hardware product(s): _____

serial number: _____

configuration: _____

Data Translation device driver - SPO number: _____

version: _____

Data Translation software - SPO number: _____

serial number: _____ version: _____

PC make/model: _____

operating system: _____ version: _____

Windows version: _____

processor: _____ speed: _____

RAM: _____ hard disk space: _____

network/number of users: _____ disk cache: _____

graphics adapter: _____ data bus: _____

I have the following boards and applications installed in my system: _____

I am encountering the following problem(s): _____

and have received the following error messages/codes: _____

I have run the board diagnostics with the following results: _____

You can reproduce the problem by performing these steps:

1. _____

2. _____

3. _____

E-Mail and Fax Support

You can also get technical support by e-mailing or faxing the Technical Support Department, as follows:

- **E-mail** – You can reach Technical Support at the following address: tsupport@datx.com
- **Fax** – Photocopy and complete the form on [page 120](#), then fax the form to Technical Support at (508) 481-8620

Support requests from noncontract and out-of-warranty customers are processed with the same priority as telephone support requests.

World-Wide Web

For the latest tips, software fixes, and other product information, you can reach the Technical Support Department from the Data Translation web site. You can access the web site at <http://www.datatranslation.com>.



Reserving System Memory

Because system memory is allocated on 4 MB boundaries, you may not be able to allocate the number of frame buffers you want even if you have reserved a sufficient amount of system memory. [Table 26](#) lists the amount of system memory you must reserve for the device driver (in megabytes) for up to 15 default-size frame buffers. You specify the amount of system memory to reserve for the device driver in the Driver Configuration dialog box in the Control Panel.

Note: You can try to reserve less memory than specified in the table, if desired. However, to guarantee that you will be able to acquire the specified number of frames, you must reserve the amount shown in the table.

For information on reserving system memory, refer to your Windows documentation.

Table 26: Reserving System Memory

Frame Grabber Board	Number of Frame Buffers Required	System Memory to Reserve	
		50 Hz	60 Hz
DT3153 DT3154 DT3130 Series DT3120	1	4 MB	3 MB
	2	6 MB	4 MB
	3	8 MB	5 MB
	4	10 MB	7 MB
	5	12 MB	8 MB
	6	14 MB	9 MB
	7	16 MB	11 MB
	8	18 MB	12 MB
	9	20 MB	13 MB

Table 26: Reserving System Memory (cont.)

Frame Grabber Board	Number of Frame Buffers Required	System Memory to Reserve	
		50 Hz	60 Hz
DT3153 DT3154 DT3130 Series DT3120 (cont.)	10	22 MB	15 MB
	11	24 MB	16 MB
	12	26 MB	17 MB
	13	28 MB	19 MB
	14	30 MB	20 MB
	15	32 MB	21 MB
DT3152 DT3155	1	1 MB	1 MB
	2	2 MB	1 MB
	3	2 MB	2 MB
	4	3 MB	2 MB
	5	3 MB	2 MB
	6	4 MB	3 MB
	7	4 MB	3 MB
	8	4 MB	3 MB
	9	5 MB	4 MB
	10	5 MB	4 MB
	11	6 MB	4 MB
	12	6 MB	4 MB
	13	7 MB	5 MB
	14	7 MB	5 MB
	15	8 MB	5 MB

A

Index

A

AC-coupled video signals 35
AcquireFrames method 24, 80, 90
acquisition (of frames) 80
 asynchronous 80
 flowcharts 89
 properties, methods, and events 24, 81
 stopping 81
 synchronous 80
active line count 62
active pixel count 59
active video area, *see* video area
ActiveDevice property 15, 29
ActiveFrame property 21, 69
ActiveLineCount property 19, 62, 102
ActivePixelCount property 19, 61, 102
ActiveSurface property 24, 78, 79, 113, 115
ActiveX control, *see* DT-Active Open Layers control
adding DT-Active Open Layers control to project 7
address, returning (Visual C++) 21
aliases 29
 returning 15
AllocateFrame method 21, 63, 68, 71, 72, 73, 77, 80, 90, 110, 112
AllocateSurface method 23, 78, 113
allocating
 frame buffers 63, 68, 124
animation 79

application program, distributing 9
asynchronous
 acquisition 80
 passthru 72

B

back porch 59
BackPorchStart property 20, 61, 102
base address, returning (Visual C++) 21
based source mode 32, 35
BasedSource property 16, 33, 90, 93, 97
BasedSourceModeEnable property 16, 33, 90, 93, 97
bitmap files 78
black level 38
 see also offset (monochrome boards), gain, reference (monochrome boards)
BlueOffset property 17, 44, 106
BlueReference property 17, 44, 106
boards, *see* frame grabber boards
brightness level 43
Brightness property 17, 43, 106
buffers, *see* frame buffers, user arrays

C

camera files 30
camera, *see* digital camera
CameraFile property 15, 30

- capabilities
 - color [34](#), [42](#), [69](#)
 - input [34](#), [36](#), [46](#), [49](#), [54](#), [56](#), [57](#), [60](#), [62](#), [69](#)
 - passthru [73](#), [74](#), [76](#)
- channel, *see* video input channel
- chrominance filter, *see* input filter
- clamp [60](#)
- ClampEnd** property [20](#), [61](#), [102](#)
- ClampStart** property [20](#), [61](#), [102](#)
- clock [53](#)
 - frequency [55](#)
 - properties [19](#)
 - source [54](#)
 - transition [55](#)
- ClockFrequency** property [19](#), [55](#), [103](#)
- ClockSource** property [19](#), [54](#), [103](#)
- ClockTransition** property [19](#), [55](#)
- CloseDevice** method [15](#), [31](#), [92](#), [95](#), [99](#)
- closing a frame grabber board [29](#), [31](#)
- color
 - properties and methods [17](#)
 - settings [43](#)
 - used to replace passthru image [79](#)
- ColorImageParameters** method [17](#), [42](#)
- ColorKey** property [24](#), [79](#), [114](#)
- ColorQueryInterface** method [17](#), [34](#), [42](#), [70](#)
- composite video type [33](#)
 - specifying the sync source [46](#)
 - specifying the sync threshold [47](#)
- configuring digital I/O lines [85](#)
- continuous-acquire passthru [73](#)
- contrast level [43](#)
- Contrast** property [17](#), [43](#), [106](#)
- conventions used [ix](#)
- CopyFrameRect** method [25](#), [83](#), [109](#)

- custom control, *see* DT-Active Open Layers control

D

- DC restore [60](#)
- DC, getting handle [78](#)
- DC-coupled video signals [35](#)
- DestroyFrame** method [21](#), [71](#), [92](#), [95](#), [99](#)
- destroying
 - frame buffers [71](#)
 - overlay surface DCs [78](#)
 - overlay surfaces [79](#)
- DestroySurface** method [23](#), [79](#), [99](#)
- device context, getting handle [78](#)
- DeviceCount** property [15](#), [29](#)
- devices, *see* frame grabber boards
- DeviceType** property [15](#)
- DigInCaps** property [26](#), [85](#)
- DigIOConfig** property [26](#), [85](#)
- DigIOPattern** property [26](#), [85](#), [92](#)
- digital camera [66](#)
 - exposure pulse [67](#)
 - properties and methods [21](#)
 - resetting [67](#)
 - type [66](#)
- digital I/O [85](#)
 - configuring [85](#)
 - properties [26](#)
- DigitalCameraType** property [21](#), [66](#), [107](#)
- DigOutCaps** property [26](#), [85](#)
- DisableOverlays** [99](#)
- DisableOverlays** method [23](#), [79](#)

displaying
 acquired image 82
 RGB planes (DT3154) 25
 distributing program 9
DrawAcquiredFrame method 25, 82,
 108
DrawRGBPlane method 25, 82, 108
 DT-Active Open Layers control 2
 adding to project 7
 example program 10
 registering 9
 DT-Open Layers 2

E

edge, *see* transition
 e-mail support 121
EnableExposePulse property 21, 67,
 107
EnableOverlays method 23, 79, 114
EraseSurface method 23, 78, 113
 erasing overlay surfaces 78
 error/status messages 31, 88
 events
 acquisition 24, 81
 defined 3
 passthru 73
 example program 10
ExposePulseDuration property 21, 67,
 107
ExposePulsePolarity property 21, 67,
 107
 exposure pulse 67
ExtendPassthruPalette method 23, 76,
 111
ExtractRGBPlanesEx method 25, 82,
 108

F

false coloring 76
 fax support 121
 field transition, *see* vertical sync
 transition
 field type 68
 files, *see* camera files, bitmap files
 filter, *see* input filter
 first active line 62
 first active pixel 59
FirstActiveLine property 19, 62, 102
FirstActivePixel property 19, 61, 102
 flags
 trigger modes 65
 trigger type 64
 flowcharts
 acquiring frames 89
 passthru with overlays 96
 passthru without overlays 93
 frame buffers 80, 83
 allocating 63, 68, 124
 current 69
 determining number allocated 21
 properties and methods 21
 releasing 71
 transferring data from 82
 frame grabber boards
 closing 29, 31
 determining number installed 29
 determining the type 15
 initializing 29
 returning alias 15
 supported 3
 frame index 68, 71
FrameBaseAddress property 21
FrameBytesPerPixel property 22
FrameCount property 21

FrameHeight property 22
FrameHorizontalScale property 22
FrameLeft property 21
frames 59, 61
 first line (top) 62, 68
 first pixel (left) 62, 68
 height 62, 68
 scaling percentage (horizontal) 69
 scaling percentage (vertical) 69
 storage mode 69
 type 68
 width 62, 68
FrameStorageMode property 22
FrameTop property 21
FrameType property 22
FrameVerticalScale property 22
FrameWidth property 22
frequency
 horizontal and vertical sync signals 52
 internal pixel clock 53, 55
front porch 59

G

gain 37
Gain property 17, 37, 100
GDI functions 78
general imaging properties and methods 15
gen-locking 51
GetDeviceName method 15
GetSurfaceDC method 23, 78, 113, 115
Graphical Device Interface functions 78
GreenOffset property 17, 44, 106
GreenReference property 17, 44, 106

H

hardware requirements 5
help
 e-mail support 121
 fax support 121
 online 11
 technical support 119
 telephone support 119
 web site 121
horizontal
 sync frequency 52
 sync insert position 48, 51
 sync search position 48, 51
 sync transition 48
 sync width 52
 video signal 59
HorizontalFrequency property 18, 52, 105
HorizontalPulseWidth property 18, 52, 105
HorizontalTransition property 18, 48, 100
HSI color model 43
HSyncInsertPos property 18, 51, 101
HSyncSearchPos property 18, 51, 101
hue level 44
Hue property 17, 44, 106
Hue/Saturation/Intensity color model, *see* HSI color model

I

ILUT, *see* input look-up table
image processing 82
 properties and methods 25
image size 66
ImageFile property 24, 78, 113

imaging properties and methods [15](#)
 initializing a frame grabber board [29](#)
 input channel, *see* video input channel
 input filter [35](#)
 input look-up table [55](#)
 modifying [56](#)
 properties and methods [19](#)
 reading [56](#)
 specifying [56](#)
 input properties and methods [16](#)
 input signal, *see* video input signal
InputFilter property [16](#), [36](#), [100](#)
InputLUT property [19](#), [56](#)
 installing the software [6](#)
 interlaced frame [52](#), [63](#)
IsAsyncAcquireFramesDone method
 [24](#), [81](#), [91](#)

K

keys, query
 external trigger [64](#)

L

languages supported [6](#)
 line transition, *see* horizontal sync
 transition
 lines
 active count [62](#)
 first active [62](#)
 total per field [57](#)
LoadDefaultPassthruLUT method [23](#),
 [76](#), [111](#)
LoadPassthruLUT method [23](#), [76](#), [111](#)
 look-up table, *see* input look-up table,
 passthru look-up table

LUT, *see* look-up table

M

memory, *see* frame buffers, user arrays
 methods
 acquisition [24](#)
 color [17](#)
 defined [2](#)
 digital camera [21](#)
 frame buffers [21](#)
 general imaging [15](#)
 image processing [25](#)
 input look-up table [19](#)
 input source [16](#)
 overlay [23](#)
 passthru [22](#)
 monochrome storage mode [69](#)
 moving
 acquired data [82](#), [83](#)
 user data [83](#)
MultiTriggerMode property [20](#), [65](#),
 [104](#)
MultiTriggerTransition property [20](#),
 [65](#), [104](#)
MultiTriggerType property [20](#), [65](#), [104](#)

N

noninterlaced frame [52](#), [63](#)
 notch filter, *see* input filter

O

offset (color boards) [44](#)
 offset (monochrome boards) [37](#)
Offset property [16](#), [37](#), [100](#)

OLC_FG_IC_INPUT_FILTER_LIMITS
36

OLC_FG_IC_INPUT_SOURCE_
COUNT 46

OLC_FG_CLOCK_EXTERNAL 54

OLC_FG_CLOCK_INTERNAL 54

OLC_FG_CSYSNCS_CURRENT_SRC 47

OLC_FG_CSYSNCS_EXTERNAL_LINE
47

OLC_FG_CSYSNCS_SPECIFIC_SRC 47

OLC_FG_DOES_PASSTHRU_
SNAPSHOT 77

OLC_FG_FILT_AC_50 36

OLC_FG_FILT_AC_60 36

OLC_FG_FILT_AC_NONE 36

OLC_FG_FILT_DC_NONE 36

OLC_FG_IC_ACTIVE_HEIGHT_
LIMITS 62

OLC_FG_IC_ACTIVE_LINE_LIMITS
62

OLC_FG_IC_ACTIVE_PIXEL_LIMITS
60

OLC_FG_IC_ACTIVE_WIDTH_LIMI
TS 60

OLC_FG_IC_BACK_PORCH_START_
LIMITS 60

OLC_FG_IC_CLAMP_END_LIMITS
60

OLC_FG_IC_CLAMP_START_
LIMITS 60

OLC_FG_IC_CLOCK_FREQ_LIMITS
54

OLC_FG_IC_CLOCK_SOURCE_
LIMITS 54

OLC_FG_IC_CSYSNCS_SOURCE_
LIMITS 46, 47

OLC_FG_IC_DOES_ACTIVE_VIDEO
58, 60, 62

OLC_FG_IC_DOES_FRAME_SELECT
70

OLC_FG_IC_DOES_INPUT_FILTER
36, 56

OLC_FG_IC_DOES_PIXEL_CLOCK
54

OLC_FG_IC_DOES_PROG_A2D 37

OLC_FG_IC_DOES_SYNC_
SENTINEL 50

OLC_FG_IC_DOES_TRIGGER 64

OLC_FG_IC_DOES_VIDEO_SELECT
34

OLC_FG_IC_FRAME_HEIGHT_
LIMITS 70

OLC_FG_IC_FRAME_HINC_LIMITS
70

OLC_FG_IC_FRAME_LEFT_LIMITS
70

OLC_FG_IC_FRAME_TOP_LIMITS
70

OLC_FG_IC_FRAME_WIDTH_
LIMITS 70

OLC_FG_IC_ILUT_COUNT 56

OLC_FG_IC_INPUT_SOURCE_
COUNT 35

OLC_FG_IC_MAX_FRAME_SIZE 70

OLC_FG_IC_MAX_ILUT_INDEX 56

OLC_FG_IC_MAX_ILUT_VALUE 56

OLC_FG_IC_MULT_TRIGGER_
MODE_LIMITS 64

OLC_FG_IC_MULT_TRIGGER_TYPE_
LIMITS 64

OLC_FG_IC_PIXEL_DEPTH 70

OLC_FG_IC_SENTINEL_TYPE_
LIMITS 50

- OLC_FG_IC_TOTAL_LINES_PER_FLD_LIMITS 58
- OLC_FG_IC_TOTAL_PIX_PER_LINE_LIMITS 58
- OLC_FG_IC_TRIGGER_TYPE_LIMITS 64
- OLC_FG_IC_VIDEO_TYPE_LIMITS 34
- OLC_FG_MODE_EACH 65
- OLC_FG_MODE_START 65
- OLC_FG_PASSTHRU_ASYNC_BITMAP 73
- OLC_FG_PASSTHRU_ASYNC_BITMAP_EXTENDED 73
- OLC_FG_PASSTHRU_SYNC_BITMAP 73
- OLC_FG_PC_DOES_PASSTHRU_LUT 76
- OLC_FG_PC_DOES_SOURCE_ORIGIN 75
- OLC_FG_PC_MAX_PALETTE_INDEX 77
- OLC_FG_PC_MAX_PALETTE_VALUE 77
- OLC_FG_PC_MAX_PLUT_INDEX 76
- OLC_FG_PC_MAX_PLUT_VALUE 76
- OLC_FG_PC_PASSTHRU_MODE_LIMITS 73
- OLC_FG_PC_SRC_ORIGIN_X_LIMITS 75
- OLC_FG_PC_SRC_ORIGIN_Y_LIMITS 75
- OLC_FG_SYNC_SENTINEL_FIXED 50
- OLC_FG_SYNC_SENTINEL_VARIABLE 50
- OLC_FG_TRIG_EVENT 64
- OLC_FG_TRIG_EXTERNAL_LINE 64
- OLC_FG_TRIG_ONE_EVENT_DELAY 64
- OLC_FG_VID_COMPOSITE 34
- OLC_FG_VID_VARSCAN 34
- OLC_QUERY_COLOR_INTERFACE_HARDWARE_SCALING 70
- OLC_QUERY_COLOR_INTERFACE_IMAGE_PARAMETER 42
- OLC_QUERY_COLOR_INTERFACE_SIGNAL_TYPE 34
- OLC_STS_ACTIVE 81
- OLC_STS_NORMAL 31
- OLC_STS_PENDING 81
- OLE control, *see* DT-Active Open Layers control
- OnAcquireNotify** event 24, 81, 91
- online help 11
- OnPassthruNotify** event 23, 73
- opaque overlays 79
- OpenDevice** method 15, 29, 89, 93, 96
- opening a frame grabber board 29
- operating systems supported 6
- output pulse, *see* exposure pulse, integration pulse, line-sync pulse
- overlays
 - acquiring with snapshots 77
 - allocating surfaces 78
 - destroying surfaces 79
 - erasing surfaces 78
 - properties and methods 23
 - releasing surface DCs 78
 - setting see-through state 79
- OverlaySurfaceOnFrame** method 24, 77, 112

P

- passthru 72
 - asynchronous 72
 - continuous-acquire 73
 - flowcharts 93, 96
 - properties, methods, and events 22, 73
 - starting 73
 - stopping 73
 - synchronous 72
 - taking a snapshot 77
- passthru look-up table 75
 - adding colors 76
 - modifying 76
 - using monochrome values 76
- passthru operation
 - source origin 74
- PassthruSnapshot** method 23, 77, 112
- phase between sync signals 52
- Phase** property 18, 52, 105
- pixel clock, *see* clock
- pixels 53
 - active count 59
 - first active 59
 - modifying values 55
 - total per line 57
- processing data, *see* image processing
- product support 119
- program, distributing 9
- programming languages supported 6
- project, adding control 7
- properties
 - acquisition 24
 - clock 19
 - color 17
 - defined 2
 - digital camera 21

- digital I/O 26
- frame buffers 21
- general imaging 15
- image processing 25
- input look-up table 19
- input source 16
- overlay 23
- passthru 22
- sync signal 18
- triggers 20
- video area 19

Q

- querying the board
 - color settings 42
 - frame size 69
 - horizontal video signal 60
 - input filter 36
 - input look-up table 56
 - number of channels 35
 - offset, gain, and reference 37
 - passthru 73
 - passthru look-up table 76
 - pixel clock 54
 - snapshots 77
 - source origin 74
 - Sync Sentinel 49
 - sync source 46
 - sync threshold 47
 - total video area 57
 - triggering 64
 - vertical video signal 62
 - video signal type 34
- QueryInputCaps** method 16, 34, 35, 36, 37, 46, 47, 49, 54, 56, 57, 60, 62, 64, 70

QueryPassthruCaps method 22, 73, 74, 76, 77

R

ReadFrameEx method 25, 82, 109

reading digital I/O data 85

ReadInputLUT method 19, 56

RedOffset property 17, 44, 106

RedReference property 17, 44, 106

reference (color boards) 44

reference (monochrome boards) 37

Reference property 17, 37, 100

region of interest, *see* video area

registering control 9

related documents ix

ReleaseSurfaceDC method 23, 78, 113, 115

releasing

frame buffers 71

frame grabber boards 29, 31

overlay surface DCs 78

removing the software 12

requirements 5

ResetCamera method 21, 67, 107

resetting a digital camera 67

RGB

storage modes 69

video type 33, 44

S

S_OK 31

saturation level, *see* U-saturation level,

V-saturation level

SaveAcquiredFrame event 25

SaveAcquiredFrame method 82, 108

scaling a frame 69

separating RGB images (DT3154) 25

service department 119

signals

composite 33

exposure output pulse 67

noisy 48

RGB 33, 44

sync 45

variable-scan 33

video input 33, 37, 45

Y/C 33

size of image 66

snapshots 77

acquiring 77

software

installing 6

removing 12

requirements 5

source of input, *see* video input channel

source of sync, *see* sync source

source origin 74

starting

acquisition (of frames) 80

passthru operation 73

StartPassthru method 22, 73, 110

status codes 31, 88

StatusMessage property 15, 31

StopAsyncAcquireFrames method 24, 81, 91

StopPassthru method 22, 73, 95, 99

stopping

asynchronous acquisition (of frames) 81

passthru operation 73

storage mode 69

support 119

SurfaceCount property 24, 78

surfaces

- allocating 78
- destroying 79

S-video signal, *see* Y/C video type

sync

- frequency of horizontal sync 52
- frequency of vertical sync 52
- horizontal insert position 48, 51
- horizontal search position 48, 51
- period 46
- properties 18
- signal 45
- source 46
- threshold 46, 47
- transition 48
- vertical insert position 49, 51
- vertical search position 49, 51
- width of horizontal sync 52
- width of vertical sync 52

Sync Master mode 51

Sync Sentinel 48

synchronous

- acquisition 80
- passthru 72

SyncMasterEnabled property 18, 52, 105

SyncSentinel property 18, 50, 101

SyncSource property 18, 47, 100

SyncThresh property 18, 47, 100

system requirements 5

T

technical support 119

telephone support 119

threshold, *see* sync threshold

timeout period 30

Timeout property 15, 30, 89, 93, 96

timing, *see* sync signal

total lines per field 57

total pixels per line 57

total video area, *see* video area

TotalLinesPerField property 19, 58, 102

TotalPixelsPerLine property 19, 58

transferring

- acquired data 82, 83
- user data 83

transition

- external pixel clock 54, 55
- external trigger 65
- horizontal and vertical sync 48

translucent overlays 79

Translucent property 24, 79, 114

triggers 64

- mode for starting multiple-frame acquisition 65
- properties 20
- transition 65

TriggerTransition property 20, 65, 104

TriggerType property 20, 65, 104

troubleshooting checklist 118

U

U-saturation level 44

USaturation property 17, 44, 106

V

variable-scan

- specifying the sync transition 48
- video signal type 33

vertical
 sync frequency 52
 sync insert position 49, 51
 sync search position 49, 51
 sync transition 48
 sync width 52
 video signal 61
VerticalFrequency property 18, 52, 105
VerticalPulseWidth property 18, 52, 105
VerticalTransition property 18, 48, 100
 video area
 active 57
 frame 62
 properties 19
 total 57
 video input
 channel 33, 35
 signal 37, 45
 type 33
 video signal, *see* video input signal
VideoInputSource property 16, 35, 89, 93, 96
VideoSignalType property 16, 34, 100, 106
 Visual Basic, adding control 8
 Visual C++, adding control 8
 V-saturation level 43
VSaturation property 17, 43, 106
VSyncInsertPos property 18, 51, 101
VSyncSearchPos property 18, 51, 101

W

web site 121

white level 38
see also offset (monochrome boards),
 gain, reference (monochrome boards)
 world-wide web site 121
WriteFrame method 25, 83, 109
WriteInputLUT method 19, 56, 90, 94, 97
 writing digital I/O data 85

X

x-coordinate, *see* source origin
XPassthruOrigin property 22, 75, 94, 97

Y

Y/C video type 33
 y-coordinate, *see* source origin
YPassthruOrigin property 23, 75, 94, 97
 YUV storage mode 69