ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Cognex VPM-59432-20* at our website: ***Click HERE***

Cognex 5700/5900

Hardware and Software Supplement

Revision 8.1.1

590-6204

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The software described in this document is furnished under license, and may be used or copied only in accordance with the terms of such license and with the inclusion of the copyright notice shown on this page. Neither the software, this document, nor any copies thereof may be provided to or otherwise made available to anyone other than the licensee. Title to and ownership of this software remains with Cognex Corporation or its licensor.

Cognex Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by Cognex Corporation. Cognex Corporation makes no warranties, either express or implied, regarding the described software, its merchantability or its fitness for any particular purpose.

The information in this document is subject to change without notice and should not be construed as a commitment by Cognex Corporation. Cognex Corporation is not responsible for any errors that may be present in either this document or the associated software.

Copyright ©1998
Cognex Corporation
All Rights Reserved
Printed in U.S.A.

The hardware described in this document may be covered by one or more of the following patents: U.S. Patent 4,728,195; 4,972,359; 5,367,439; and 5,371,690. Other U.S. and foreign patents pending.

This document may not be copied in whole or in part, nor transferred to any other media or language, without the written permission of Cognex Corporation.

The following are trademarks of Cognex Corporation:

COGNEX logo	COGNEX 3000	VC1	Cognex 8000
COGNEX 1500	COGNEX 4000	VC2	PatMax
COGNEX 1600	COGNEX 5000	VC3	
COGNEX 2000	VisionPro	VC4	

The following are registered trademarks of Cognex Corporation:

DataMan	CHECKPOINT	COGNEX, Vision for Industry
On-Sight	COGNEX	

Other product and company names mentioned herein may be the trademarks of their respective owners

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Contents

Contents 3

About This Manual 13
- Summary of Documentation Changes 14
- Related Publications 15
- Style Conventions Used in This Manual 16
- Cognex Offices 17

Chapter 1: Before You Install 19
- Host PC Requirements 19
- Requirements for Maximum Frame Rate 20
  - PCI 2.1 Compliance 21
- Software Requirements 21
- Disk Space Requirements 21
- Installation Order 21

Chapter 2: Cognex 5700/5900 Hardware Installation 23
- Installation Overview 24
  - System ISA Bus and PCI Local Bus 24
- Installing Cognex 5700/5900 Hardware 25
  - Placement of Cognex 5700/5900 Hardware 25
  - Summary of the Hardware Installation Process 27
  - Installing and Configuring the Cognex 5700/5900 28
  - Installing Parallel I/O Boards 30
    - Installing a Single Board 31
    - Installing the Double-Board Configuration 32
  - Installing the Serial Interface Adapter 33
  - Hardware Verification 35
  - Closing the System 35

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Contents

Chapter 3: PVE Software Installation 37
- Software Installation Instructions 37
  - Before You Install 37
  - Installing PVE 8.1.1 Software 38
  - Device Driver Installation 45
  - Installing Epsilon 46
  - Installing Adobe Acrobat 47
  - Using the Online PVE Manuals 48
- Uninstalling PVE Software 49

Chapter 4: Cognex 5700/5900 Hardware Description 51
- Cognex 5700/5900 Vision Processor Components 52
  - Processors 52
  - Memory 52
  - PCI Bus Interface 52
  - Video Input System 52
  - Parallel I/O 53
  - Serial I/O 53
  - Video Output 53
- Mechanical Specifications 54
  - Environmental Requirements 54
  - Shipping 55
  - Standards 55
- Electrical Specifications 56
  - Cognex 5700/5900 Power Requirements 56
  - Camera Electrical Specifications 56
  - Diagnostic Display 57
  - Serial I/O (RS-232C) Ports 58
  - Parallel I/O Port 59
    - Hardware Video Triggers 64
    - Using Optically Isolated Outputs 64
- Cognex 5700/5900 Parallel I/O Boards 66
  - Pinout Values for TTL Board 68
  - Pinout Values for OPTO/TTL Board 69
- Cognex 5700/5900 Camera Support 70

Chapter 5: Cognex Video Modules and Cameras 71
- Cognex Video Module Overview 72
  - Baseboard or Daughtercard CVMs 72
  - Input Video Capabilities 72
  - CVM Video Output 73
    - Bus-Based Video Output 73

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Port-Based Video Output ... 73
CVM Descriptions ... 74
CVM0 Description ... 74
CVM0 Camera Support ... 74
CVM0 Analog Camera Connector ... 74
CVM0 Camera Numbering ... 75
CVM1 Description ... 76
CVM1 Camera Support ... 76
CVM1 Analog Camera Connector ... 77
CVM1 Camera Numbering ... 78
CVM3 Description ... 79
CVM3 Camera Support ... 79
CVM3 Analog Camera Connector ... 80
CVM3 Video Output ... 80
CVM3 Camera Numbering ... 81
Connecting Analog Cameras ... 83
Attaching the Camera Breakout Box ... 85
Auxiliary Power Supply ... 86
Hirose Connector Pin Assignments ... 87
Cognex Analog Camera Cables ... 88
Connecting a Sony XC-55 Camera ... 88
Connecting a Pulnix TM-9701 Camera (CVM1 only) ... 89
Optional Camera Lenses ... 89
Analog Camera I/O Electrical Specifications ... 90
Analog Video Signal Timing ... 91

Chapter 6: 5700/5900 User's Guide ... 93
Comparison with 5000-ISA Vision Processors ... 94
Hardware Comparison ... 94
Software Comparison ... 95
Using the 5700/5900 ... 96
Building Volume Files ... 96
Options Not Supported ... 96
Baseline Volume File ... 97
Enabling Live Video ... 97
Saving a Live Video Session ... 98
Running the Search and Train Demo ... 99
Search and Train Demo with Cognex Trackball ... 101
Image Acquisition ... 101
Software Display Interface ... 102
PVE 8.1.1 Software Differences ... 102

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Contents

Chapter 7: Cognex Unified Acquisition 105

Overview of Unified Acquisition 106
Channels 106
Video Formats 107
Vision Processor Supports One Video Format at a Time 108
Display Jitter With Half-Resolution Video Formats 108
Live Display Not Supported for Rapid Reset Video Formats 108
Acquisition Channel Properties 109
Exposure Property 109
Timeout Property 109
Camera Port Property 109
Input Lookup Table Property 109
Video Trigger Property 110
Video Strobe Properties 110
Charge Transfer Interval Properties 111
Brightness and Contrast Properties 112
Simultaneous Acquisitions 112
Frame Buffers 112
Allocating and Freeing Frame Buffers 112

Unified Acquisition Software 114
Object Handles 114
Memory Management with Handles 114
Constants for Graphics Colors 115
Enumeration cua_graphics_colors 115
Channel Functions 116
Function cua_acquire_channel_alloc() 116
Function cua_display_channel_alloc() 117
Function cua_default_monitor_get() 117
Function cua_vp_monitor_get() 118
Function cua_host_monitor_get() 118
Function cua_default_board_get() 119
Function cua_prepare_resources() 119
Function cua_are_resources_prepared() 120
Frame Buffer Functions 120
Function cua_displayable_alloc() 121
Function cua_overlayable_alloc() 121
Function cua_acquisition_buffers_get() 122
Function cua_acquisition_buffers_set() 122
Function cua_image_plane_set() 122
Function cua_overlay_plane_set() 123
Function cua_live_set() 124
Function cua_cip_buffer_get() 125
Acquisition Functions 125
Function cua_acquire() 125

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Function cua_cancel() 126
Function cua_image_get() 126
Function cua_state_get() 126
Enumeration cua_state 127
Function cua_completion_get() 127
Enumeration cua_completion 128
Function cua_master_set() 128
Function cua_master_get() 129
Property Functions 129
Function cua_properties_get() 130
Function cua_properties_set() 130
Function cua_exposure_has() 131
Function cua_exposure_get() 132
Function cua_exposure_set() 132
Function cua_timeout_has() 132
Function cua_timeout_get() 133
Function cua_timeout_set() 133
Function cua_camera_port_has() 133
Function cua_camera_port_get() 134
Function cua_camera_port_set() 134
Function cua_input_lut_256_has() 135
Function cua_input_lut_256_get() 135
Function cua_input_lut_256_set() 135
Function cua_brightness_contrast_has() 136
Function cua_brightness_get() 136
Function cua_contrast_get() 137
Function cua_brightness_contrast_set() 137
Function cua_video_trigger_has() 137
Function cua_video_trigger_enable_get() 138
Function cua_video_trigger_enable_set() 138
Function cua_video_strobe_has() 139
Function cua_video_strobe_enable_get() 139
Function cua_video_strobe_enable_set() 140
Function cua_video_strobe_duration_get() 140
Function cua_video_strobe_duration_set() 140
Function cua_video_strobe_polarity_get() 141
Enumeration cua_polarity 141
Function cua_video_strobe_polarity_set() 142
Function cua_video_strobe_trigger_width_get() 142
Function cua_video_strobe_trigger_width_set() 142
Function cua_cti_has() 143
Function cua_cti_start_even_get() 143
Function cua_cti_end_even_get() 144
Function cua_cti_start_odd_get() 144
Function cua_cti_end_odd_get() 144

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Contents

Function cua_cti_set() 145
Callback Functions 145
Callback Function Definition 146
Function cua_complete_callback_has() 147
Function cua_complete_callback_get() 147
Function cua_complete_callback_set() 147
Function cua_complete_callback_context_get() 148
Function cua_overrun_callback_has() 148
Function cua_overrun_callback_get() 149
Function cua_overrun_callback_set() 149
Function cua_overrun_callback_context_get() 150
Function cua_move_part_callback_has() 150
Function cua_move_part_callback_get() 150
Function cua_move_part_callback_set() 151
Function cua_move_part_callback_context_get() 151
Function cua_strobe_callback_has() 152
Function cua_strobe_callback_get() 152
Function cua_strobe_callback_set() 152
Object Functions 153
Function cua_copy() 153
Function cua_free() 153

Chapter 8: Cognex Software Display 155
Overview of Software Display 156
Using the Software Display Library 162
Sample Code 162
Software Display Limitations 162
Relationship Between Software Display and Unified Acquisition 162
Performance Factors 164
Requirements for Software Display Applications 165
Defining an 8-Bit Palette for Your Software Display Application 165
Using Cognex Software Display in a Multithreaded Application 166
Non-Compliant Chip Set List in Registry 166
Cognex Software Display Interface 168
Cognex Software Display Data Types 168
Handle csd_handle 168
Structure csd_rect 168
Enumeration csd_state 168
Structure csd_version 169
Enumeration csd_pixel_format 169
Typedef csd_keycolor 169
Structure csd_window 170
Cognex Software Display Functions 170
Function csd_open() 170

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Function csd_close() 171
Function csd_image_format_set() 171
Function csd_image_format_get() 172
Function csd_graphics_format_set() 172
Function csd_graphics_format_get() 173
Function csd_keycolor_set() 174
Function csd_keycolor_get() 174
Function csd_image_roi_set() 175
Function csd_image_roi_get() 175
Function csd_state_get() 176
Function csd_live_set() 176
Function csd_next_image_get() 177
Function csd_frozen_set() 178
Function csd_frozen_image_get() 179
Function csd_system_memory() 179
Function csd_version_get() 180

Chapter 9: Using Cogtalk 181

Cogtalk Features 182
Cogtalk Requirements 182
Console Cogtalk Features 182
Windows Cogtalk Features 183

Cogtalk Usage Notes 185
Controlling the Vision Processor 185
Include Path and Include Preprocessor 185
Windows Cogtalk's Include Path and Working Directory 186
Console Cogtalk's Include Path 186
Cogtalk as Preprocessor 187
The #include Directive 187
Handling of Nested Includes 188
The #cwd Directive 189
Escape Commands 189
Reset vision processor (~b) 189
Load Vision Processor (~r) 190
Begin System Save Operation (~x) 190
Begin ASCII Data Capture (~s) 190
End Data Capture (~q) 191
Send Ctrl+C (~c) 191
Send Ctrl+D (~d) 191
Send Ctrl+\: (~\) 191
Send File (~send) 192
Change Current Working Directory (~cwd) 192
Send VP Interrupt (~t) 192
Change Escape Character (~escape) 192
Toggle Quiet Mode (~quiet) 193

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Contents

Display Cogtalk Version (~v) 193
Exit Cogtalk / Exit Port Window (~. or ~close) 194
Display Command List (~? or ~help) 194
Cogtalk as File Server 194
Running the Block Mode Server 194
Stopping the Block Mode Server 195
Using the Host PC's File System 195
Specifying Files and Directories 195
Supported Functions 196
File System Utility Functions 197
Function cfs_shutdown() 197
Unsupported System Save and Load Functions 197
Example Using the File Server Functions 197
Using Console Cogtalk 199
Starting Console Cogtalk 199
Command Line Switches 199
Command History and Editing 201
Using Windows Cogtalk 203
Starting Windows Cogtalk 203
Retained Settings Between Invocations 204
Communication in Port Windows 204
Port Window Display History 205
Command History and Editing 206
Using Session Logs 207
Windows Cogtalk Menu and Toolbar Reference 209
File Menu 209
File->Open Port 209
File->Load System 210
File->Save System 210
File->Create Session Logfile 212
File->Working Directory 212
File->Include File 213
File->Print 213
File->Print Setup 213
File->Exit 214
Edit Menu 214
Edit->Clear Screen 214
Edit->Clear Line 214
Edit->Copy 214
Edit->Paste 214
Edit->Find 215
Edit->Font 215
Options Menu 215
Options->Compile Region 216

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Options->Include Path 216
Options->Block Server 217
Options->Reset VP 217
Options->Send ^\ 218
Options->Send ^C 218
Options->Send ^D 218
Tools Menu 218
View Menu 219
View->Board Status 219
View->Toolbar 219
View->Status Bar 219
Window Menu 219
Window->Cascade 219
Window->Tile Horizontally 220
Window->Tile Vertically 220
Help Menu 220
Help->Cogtalk Help 220
Help->Cognex API Reference 220
Help->About Cogtalk 220
Toolbar Buttons 221
Windows Cogtalk Keyboard Shortcuts 223

Chapter 10: Using Cogview 225
Cogview Usage Notes 226
Cogview Requirements 226
Starting Cogview 226
Retained Settings Between Invocations 228
Window and Image Sizes 229
Using Half Resolution Video Formats 230
Specifying a Region of Interest 230
Live and Frozen Images 231
Using Zoom Mode 232
Using Autosize Mode 233
Using the PC's Mouse with Cogview 233
Palette Tables and Palette Table Files 234
Cogview Menu and Toolbar Reference 236
File Menu 236
Configuration Menu 236
Configuration->Frame Rate 236
Configuration->Mouse Port 237
Configuration->Video Format 238
Configuration->Palette Table 239
View Menu 239

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Contents

View->Autosize Mode 239
View->Freeze Mode 240
View->Capture Image 240
View->Zoom 240
View->Toolbar 240
View->Status Bar 240
Help Menu 240
Help->Contents 240
Help->Using Help 241
Help->About Cogview 241
Toolbar Buttons 241
Cogview Keyboard Shortcuts 242

Index 243

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

About This Manual

This manual provides hardware and software information about the Cognex 5700 and Cognex 5900 vision processors that supplements information published in the PVE 7.6 documentation set. Throughout this manual, the term *Cognex 5700/5900* refers to both Cognex 5700 and 5900 vision processors.

This preface summarizes the content of each chapter and describes the relationship between this manual and the PVE 7.6 documentation set. It also lists related publications and Cognex office locations and phone numbers.

Chapter 1, Before You Install, describes the hardware and software requirements for the PC that will serve as your development host.

Chapter 2, PVE Software Installation, describes the software installation procedure.

Chapter 3, Cognex 5700/5900 Hardware Installation, describes installing a Cognex 5700/5900 vision processor and its associated options into a PC.

Chapter 4, Cognex 5700/5900 Hardware Description, describes the mechanical and electrical interface to the Cognex 5700/5900, and provides information for connection and control of external hardware for use with the Cognex 5700/5900.

Chapter 5, Cognex 5700/5900 User's Guide, describes the hardware and software differences between the 5700/5900 and previous models in the Cognex 5000 series. This chapter also describes additions and changes to the PVE software interface, including features in PVE 7.6 that are not supported in PVE 8.1.1, and provides an introduction to the 5700/5900 for those familiar with previous 5000 series models.

Chapter 6, Cognex Unified Acquisition, describes the software interface used to acquire images using a Cognex 5700/5900 vision processor.

Chapter 7, Cognex Software Display, describes the software interface used to display images over the PCI bus to a window on the 5700/5900 vision processor's host PC.

Chapter 8, Using Cogtalk, describes Cogtalk, a utility that runs on the host PC and can be used to communicate with a vision processor over the PCI bus.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

About This Manual

Chapter 9, Using Cogview, describes Cogview, a utility that runs on the host PC and displays video images and graphics from the vision processor in a PC window. Cogview also accepts mouse input and relays that input to the vision processor.

Summary of Documentation Changes

This manual supplements the information in the PVE 7.6 documentation set. A number of chapters in the PVE 7.6 documentation are superseded one for one by chapters in this manual. Information in other chapters in the 7.6 documentation is not applicable to users of the PVE 8.1.1 software and the 5700/5900 hardware, and is superseded with information in this manual. Table 1 describes the relationships between the contents of this 5700/5900 supplement and the PVE 7.6 documentation set.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Supplement Chapters	Relationship to PVE 7.6 Documentation
Before You Install, PVE Software Installation, Cognex 5700/5900 Hardware Installation	Supersedes Chapters 2 through 4 of the *5000 Getting Started* manual.
Cognex 5700/5900 Hardware Description	Adds the PCI bus vision processors to the set of ISA bus processors described in the *5000 Hardware* manual. Also supplements information in *Chapter 4, Cognex 5000 Camera Support*, and *Chapter 8, Cognex Parallel I/O Board*, of the *5000 Hardware* manual.
5700/5900 User's Guide	Provides an overview of using and developing vision processing applications for Cognex's 5700/5900 vision processors.
Cognex Unified Acquisition and *Cognex Software Display*	For 5700/5900 users, these two chapters supersede *Chapter 1, 5000 Acquisition and Display* in the *5000 Image Acquisition and I/O* manual. (The Cognex 5700/5900 does not support the **caq_** interface for acquisition and display.)
Using Cogtalk, Using Cogview	For 5700/5900 users, the Cogtalk utility supersedes the PCVtalk and Wintalk utilities, and the Cogview utility supersedes the Vmix and PCI Vmix utilities. Thus, none of *Chapter 1, Cognex 5000 Communications Utilities* in the *Development Environment 2* manual applies to 5700/5900 users. Also note that the 5700/5900 displays video images over the PCI bus to the PC without using a video mixing board. Thus, no PVE 7.6 documentation relating to video mixing hardware and software applies to 5700/5900 users.

Table 1. Relationship between the 5700/5900 Supplement and the PVE 7.6 documentation

Related Publications

IEEE 996: Standard for the Industry Standard Architecture Bus of the Personal Computer. This is a formalization of the ISA bus standard, maintained by, and available through, the Institute of Electrical and Electronic Engineers.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PCI Local Bus 2.1 Specification, PCI Special Interest Group. PCI local bus architecture standard developed primarily by Intel Corporation, now controlled by the PCI Special Interest Group.

Style Conventions Used in This Manual

This manual uses the following style conventions:

boldface	Used for C/C++ keywords, function names, class names, structures, enumerations, types, and macros. Also used for user interface elements such as button names, dialog box names, and menu choices.
italic	Used for names of variables, data members, arguments, enumerations, constants, program names, file names. Used for names of books, chapters, and sections. Occasionally used for emphasis.
`courier`	Used for C/C++ code examples and for examples of program output.
`bold courier`	Used in illustrations of command sessions to show the commands that you would type.
<italic>	When enclosed in angle brackets, used to indicate keyboard keys such as *<Tab>* or *<Enter>*.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Cognex Offices

Cognex Corporation serves its customers from the following locations:

Corporate Headquarters	Cognex Corporation Corporate Headquarters One Vision Drive Natick, MA 01760-2059 (508) 650-3000
Web Site	www.cognex.com
U.S. Regional Technology Centers	California (650) 969-4812 Illinois (630) 505-9990 Pennsylvania (610) 975-9592 Texas (512) 370-5245
International Offices	France 33-1-4777-1550 Germany 49-721-96187-0 Italy 39-2-7739-9214 Japan (Osaka) 81-6-329-6671 Japan (Tokyo) 81-3-5977-5400 Korea 82-2-539-9047/8 Netherlands 31-(0)40-2230912 Scotland 44-1292-619670 Singapore 65-773-1261 Sweden 46-21-145588 Taiwan 886-3-562-2660 United Kingdom 44-1908-206000

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

About This Manual

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter

1

Before You Install

This chapter discusses the installation prerequisites for the Cognex 5700/5900 vision processor and the Programmable Vision Engines (PVE) software, revision 8.1.1.

Host PC Requirements

The Cognex 5700/5900 vision processor is a PCI adapter card for IBM-compatible PCs. To develop a vision processing application, you must supply a development PC host. The host PC must meet the following requirements:

- Pentium CPU running at a minimum of 90 MHz. Cognex recommends using Pentium CPUs of 166 MHz or higher.
- One available full-length PCI slot for the 5700/5900.
- One or two available short-length ISA slots for the optional parallel I/O adapters.
- One available back panel slot for the optional serial I/O adapter.
- A PCI video display adapter.
- A CD-ROM drive, or access to one over a network, to install the PVE software.
- Windows NT, version 4.0 or later.

To use the Cogview application, or your own application based on the Cognex Software Display library, the PC must meet these additional requirements:

- The PCI display adapter must have at least 4 MBytes of video memory. Cognex has tested and recommends the following products: ATI Xpert98, ATI Xpert@Work, ATI 3D Xpression + PC2TV, and ATI Graphics Pro Turbo. If your host PC has more than one PCI bus, the display adapter should be on the same bus as the 5700/5900 for maximum performance
- The NT desktop must be in 256-color mode.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- To display full resolution images (512x480 or larger), the NT desktop must be at least 800x600 pixels.
- A mouse or other pointing device is recommended, and is required to use certain features.

Requirements for Maximum Frame Rate

If you are going to use the Cogview application, or your own application based on the Cognex Software Display library, consider the following.

The Cognex 5700/5900 displays video output under control of the Windows NT operating system. The Cognex 5700/5900 vision processor generates high resolution bitmaps and passes them through the PCI bus for display on a high resolution SVGA display under Windows NT control.

To allow the 5700/5900 to communicate with a PCI display adapter at the maximum frame rate of 30 frames per second, the following conditions must be met:

- The PCI chipset on the host PC's motherboard must be fully compliant with the PCI 2.1 specification. This is discussed further in *PCI 2.1 Compliance* below.
- The PC's video display adapter and its device driver must be fully compliant with Microsoft's DirectX specifications, version 2.0 or later. The adapter must support blitting and source color keying in hardware.
- The video display adapter must supply sufficient offscreen video memory to allow hardware blitting of the display image. An adapter with at least 4 Mbytes of video memory meets this requirement.
- The PCI bus must not be saturated with requests from competing PCI adapters, such as network or SCSI cards, or by the use of Cogtalk's block mode file server to transfer data from the vision processor to the PC.

If these conditions are not met, the Cognex software display mechanism uses system memory to transfer images from the 5700/5900 to the video display adapter. This reduces the maximum frame rate for video transfers, and increases host CPU usage. The display mechanisms used by the Cognex Software Display library on compliant and non-compliant PCs are discussed further in *Performance Factors* on page 164.

Cognex has tested and recommends the ATI Xpert98, ATI Xpert@Work, and ATI 3D Xpression + PC2TV graphics adapters with at least 4 Mbytes of video memory. A comparably equipped ATI Graphics Pro Turbo video display adapter may be substituted. Use of other display adapters has not been tested and may result in a reduced frame rate.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PCI 2.1 Compliance

Many early PCI motherboards do not meet the PCI 2.1 specification and cannot support the maximum video transfer frame rate using Cognex 5700/5900 systems. Non-compliant motherboards will work with 5700/5900 systems, but only at a reduced video transfer rate and higher host CPU usage.

Pentium motherboards made with the Intel 430LX, 430NX, or 430FX chip sets do not meet the PCI 2.1 specification. When used with one of these motherboards, the Cognex software display mechanism is known to reduce the frame rate performance with Cognex 5700/5900 vision processors. The 430LX chip set was used with 60 and 66 MHz Pentium systems. The 430NX chip set was used with Pentium CPUs with clock speeds of 75, 90, 100, and 120 MHz. The 430FX chip set was used with Pentium CPUs ranging from 75 to 200 MHz.

There are no known problems using the Cognex 5700/5900 vision processor in host PCs with PCI chip sets meeting the PCI 2.1 specification. These include Intel Pentium support chip sets 430HX, 430VX, and 430TX, Pentium Pro support chip sets 440FX, 450KX, and 450GX, and Pentium II support chip sets 440LX, 440BX, and 440GX. Intel chip sets not listed here (such as those under development at this writing) should be PCI 2.1 compliant.

Cognex has not tested or qualified any motherboards made with non-Intel PCI chip sets.

Software Requirements

The PVE software revision that supports the 5700/5900 vision processor is PVE revision 8.1 or 8.1.1, which are supported under Windows NT 4.0 or later. To perform the required device configuration under Windows NT, you must have Administrator privileges for your Windows NT system.

The PC host demonstration source code was built with Microsoft Visual C++ 4.2 with patch level B applied. Other versions of Visual C++ have not been tested and are not supported.

Disk Space Requirements

PVE revision 8.1.1 requires about 30 Mbytes of disk space. Reserve another 5 to 25 Mbytes of disk space for volume and saved system files that you generate, and for your vision processing application's source code.

Installation Order

Install your 5700/5900 vision processor before installing the PVE software. Doing so allows the PVE installation program to configure and start the vision processor device driver. If your vision processor is not installed first, the PVE installation will install the device driver, but cannot start it until the hardware is installed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1 Before You Install

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter

2

Cognex 5700/5900 Hardware Installation

This chapter describes how to install and configure a Cognex 5700/5900 vision processor in its development host PC.

This chapter contains the following sections:

- *Installation Overview* on page 24 provides an overview of how to install a Cognex 5700/5900 vision processor in your PC.
- *Installing Cognex 5700/5900 Hardware* on page 25 provides considerations, checklists and procedures for installing the Cognex 5700/5900 system components.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Installation Overview

A Cognex 5700/5900 installation includes a PCI vision processor board and, optionally, one or two ISA bus parallel I/O boards and one serial I/O adapter. The development host PC must have a usable full-length PCI bus slot for the vision processor board and a short ISA bus slot for each parallel I/O board to be installed. The serial I/O adapter does not require a bus slot, but requires a free back panel slot for mounting.

You must supply a video display adapter that meets the requirements described in *Requirements for Maximum Frame Rate* on page 20.

System ISA Bus and PCI Local Bus

The Cognex 5700/5900 vision processor is a 32-bit, 5 V PCI bus card that requires an available full size 5 V PCI bus slot. The 5700/5900 vision processor requires +5 V, +12 V, and –12 V DC voltages, which is supplied by the PCI bus.

The two parallel I/O boards available for the Cognex 5700/5900 install into ISA bus slots to provide ESD and chassis ground connection when attaching external devices.

To obtain the maximum image transfer rate when using software display, the development host PC's motherboard must meet the standards defined in the publication *PCI Local Bus Specification Revision 2.1*. See *PCI 2.1 Compliance* on page 21.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Installing Cognex 5700/5900 Hardware

This section describes how the hardware components of the Cognex 5700/5900 are connected and how to install them.

Placement of Cognex 5700/5900 Hardware

The Cognex 5700/5900 vision processor is a full-size PCI card as defined in the PCI specification, revision 2.1. The card has an attached ISA retainer bracket that allows the board to be stabilized in a standard PC board mount. You can install the Cognex 5700/5900 system in any full-size PCI slot, but you must ensure that the placement does not cause heat buildup or airflow restrictions that would result in overheating.

Figure 1 shows a Cognex 5900 vision processor installed with typical components in a Cognex 5700/5900 vision processor system.

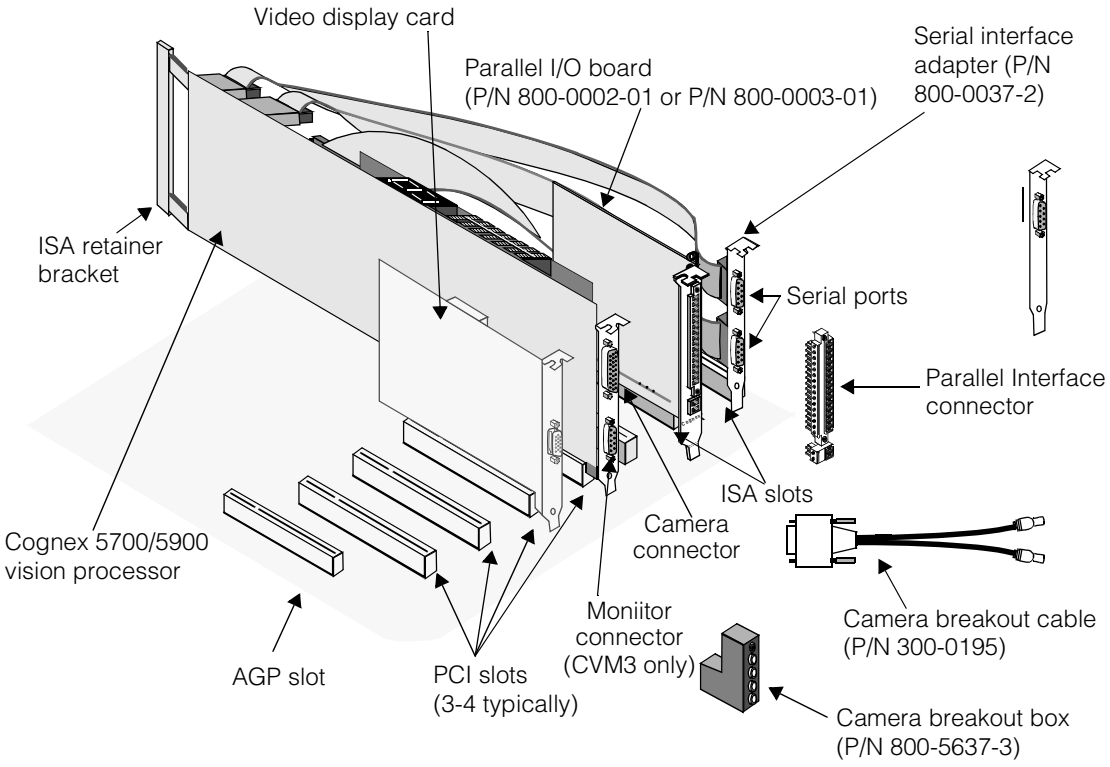


Figure 1. Placement of components in a Cognex 5700/5900 system

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2 Cognex 5700/5900 Hardware Installation

The hardware for a Cognex 5700/5900 system occupies the following positions in your PC:

- The Cognex 5700/5900 vision processor board occupies a 5 V 32-bit PCI bus full-length expansion slot.
- The serial interface adapter can occupy the faceplate slot of any available PCI or ISA bus slot. It does not plug into the bus; it connects directly to the vision processor. All signals, power, and ground connections pass through the ribbon cables from the serial port adapter to the Cognex 5700/5900.
- Two parallel I/O boards (a *TTL* board and an *OPTO/TTL* board) are available for the Cognex 5700/5900.

  Each board carries 16 of the 32 available parallel I/O lines to the PC's back panel and provides two digital grounds. Both boards require an 8-bit ISA bus slot or the 8-bit portion of a 16-bit ISA slot. The ISA slot is used only for grounding purposes. No ISA transfers are made to this board.

  Both boards are equipped with a screw-clamp connector that provides a secure method for attaching wires.
- The DB-26 camera connector on the 5700/5900 supports up to four analog cameras.

  A DB-26-to-Hirose converter cable that supports one or two monochrome cameras is supplied with all Cognex 5700 vision processors, and with Cognex 5900 vision processors that use CVM0. A camera breakout box that supports up to four monochrome cameras is supplied with Cognex 5900 vision processors that use CVM1 or CVM3.
- For a 5700/5900 equiped with CVM3, there is a DB-25 connector to support an external monitor.
- The host PC's video display adapter requires a 32-bit PCI bus expansion slot.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Summary of the Hardware Installation Process

Table 2 provides a summary of the steps to install the Cognex 5700/5900 internal components:

✓	Component	Installation Location
	Cognex 5700/5900 vision processor	Insert in full-length PCI slot on the PC motherboard as described in *Installing and Configuring the Cognex 5700/5900* on page 28.
	Parallel I/O board(s)	When using one board, insert the *TTL* or *OPTO/TTL* board in an ISA slot on the PC motherboard and connect the ribbon cable to the 5700/5900. When using both boards, insert the *TTL* board closest to the 5700/5900 and the *OPTO/TTL* board in the adjacent ISA slot. See the section *Installing Parallel I/O Boards* on page 30.
	Serial interface adapter	Attach the adapter panel to the back panel of the PC and connect the ribbon cable. See *Installing the Serial Interface Adapter* on page 33.

Table 2. Installing Cognex 5700/5900 internal components

If you are going to use the Cogview application, or your own application based on the Cognex Software Display library, a PCI video display adapter with at least 4 Mbytes of video memory must also be installed and configured according to the directions supplied by its manufacturer. On systems having more than one PCI bus, the video display adapter and the Cognex 5700/5900 board should share the same bus for maximum performance. For any questions relating to the installation and configuration of the video adapter, contact the adapter's manufacturer.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2 Cognex 5700/5900 Hardware Installation

Table 3 provides a summary of the steps you follow to install the Cognex 5700/5900 external components.

✓	Component	Installation Location
	Monochrome cameras: Sony XC-75, XC-75CE, XC-75 RR, XC-75CE RR, Pulnix 9701 or custom	Attach the camera breakout box or converter cable supplied with your vision processor to the DB-26 connector on back of Cognex 5700/5900. Connect the camera cable to the camera breakout box or converter cable. See *Connecting Analog Cameras* on page 83.
	Auxiliary camera power supply	When all attached cameras together require more than 750 mA of +12 V power, attach an external power supply to the installed camera breakout box. See *Auxiliary Power Supply* on page 86.
	Parallel interface devices	Attach wiring and secure with clamping screws. See *Cognex 5700/5900 Parallel I/O Boards* on page 66.

Table 3. Installing Cognex 5700/5900 external components

Installing and Configuring the Cognex 5700/5900

Figure 2 shows components and connectors on a Cognex 5900 vision processor board configured with CVM3. A 5900 with CVM1 is the same but with no monitor connector.

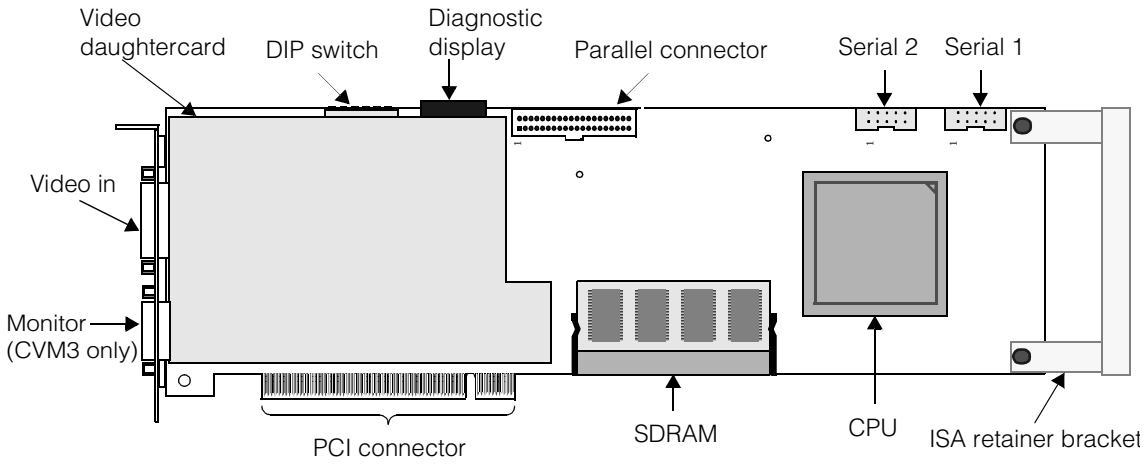


Figure 2. Cognex 5900 vision processor with CVM1 or CVM3

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The presence of a factory-installed video daughtercard distinguishes a Cognex 5900 with two or four video channels (those using CVM1 or CVM3) from a Cognex 5700 or a 5900 with a single video channel (using CVM0). Single-channel 5700/5900 boards have no video daughtercard. Figure 3 shows the components and connectors on a Cognex 5700/5900 configured with a CVM0 single video channel.

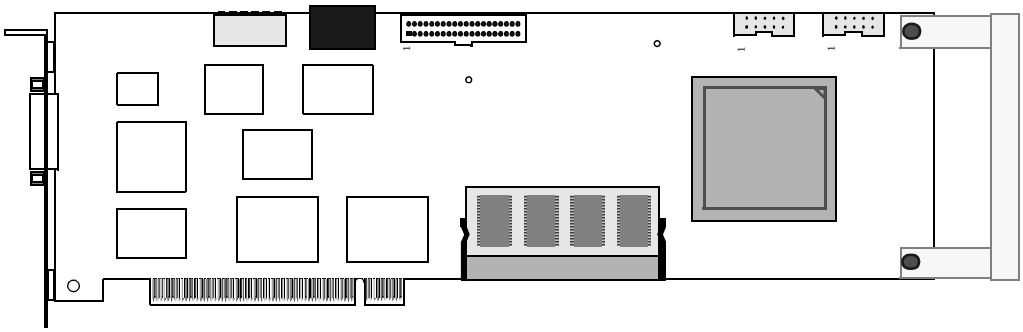


Figure 3. Cognex 5700/5900 board with single video channel

The presence or absence of a video daughtercard has no effect on the installation process for a Cognex 5700/5900.

To install the Cognex 5700/5900 vision processor in your PC:

1. Wear a wrist strap for ESD protection.
2. Set the positions of the DIP switches on the Cognex 5700/5900 board. Figure 4 shows the default switch positions, and Table 4 on page 30 summarizes the functions of the DIP switches.

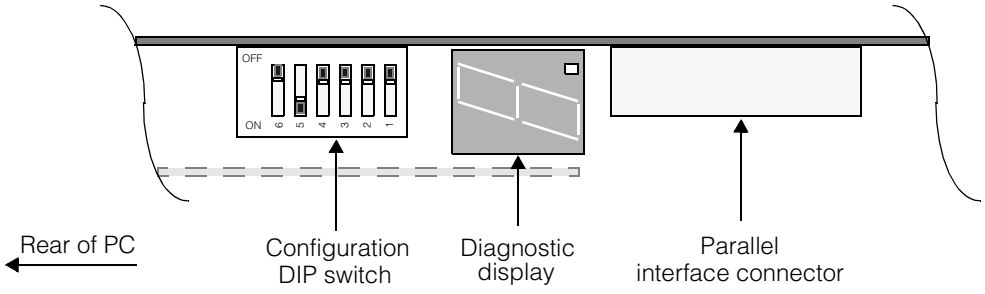


Figure 4. Default DIP switch settings for the Cognex 5700/5900 vision processor

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Switch	Controlled Function	Default	Effect
1 - 4	Reserved for Cognex use	off	*Do not change*
5	Controls how the 5700/5900 responds to break signals on serial port 1. If this switch is on, a break of 32 ms or more causes a restart. If the switch is off, breaks are ignored.	on	Boot on break is enabled.
6	Controls which port is used for communication with the PC host. If the switch is off, the 5700/5900 uses the PCI bus as its console. If switch 6 is on, the 5700/5900 uses serial line 1 as its console.	off	PCI bus is used to communicate with PC host.

Table 4. Cognex 5700/5900 DIP switch functions

3. Turn off the PC and remove its cover.
4. Insert the Cognex 5700/5900 vision processor in a PCI expansion slot in your PC as shown in Figure 1 on page 25. Take care not to damage the fine pitch components near the PCI connector.
5. Screw the metal back panel bracket attached to the 5700/5900 card to the mounting bracket at the back of the PC.

Installing Parallel I/O Boards

Cognex 5700/5900 parallel I/O boards provide external access to the vision processor's parallel I/O lines (and digital ground) using external screw terminal connectors. Two boards are available, *TTL* (P/N 800-0003-01) and *OPTO/TTL* (P/N 800-0002-01). Each carries 16 of the 32 parallel I/O lines on the Cognex 5700/5900 to the PC's back panel.

Specifications for the parallel I/O boards can be found in *Cognex 5700/5900 Parallel I/O Boards* on page 66.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Installing a Single Board

The single-board configuration uses the 1-foot ribbon cable (shown in Figure 5) supplied with the Cognex 5700/5900 parallel I/O boards. It has a 40-pin connector on each end. The connectors are keyed to ensure that they are inserted correctly.

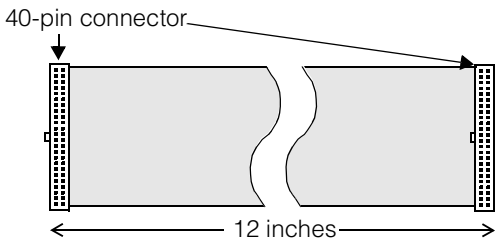


Figure 5. Cognex 5700/5900 parallel I/O board 1-foot ribbon cable (P/N 300-0132)

Using Figure 6 for reference, follow these steps to install a Cognex 5700/5900 parallel I/O board in a single-board configuration:

1. Wear a wrist strap for ESD protection.
2. Turn off the PC and remove its cover.
3. Install the parallel I/O board in an ISA bus slot. Install the mounting screw in the board to secure it to the PC's back panel.
4. Attach one end of the ribbon cable to the Cognex 5700/5900 vision processor's parallel I/O port. (The location of the parallel I/O port on the Cognex 5700/5900 board is shown in Figure 2 on page 28.)
5. Attach the other end of the ribbon cable to the parallel I/O board. For a *TTL* board, connect the cable to either of the board's two connectors.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 6 shows a *TTL* parallel I/O board installed with a Cognex 5900.

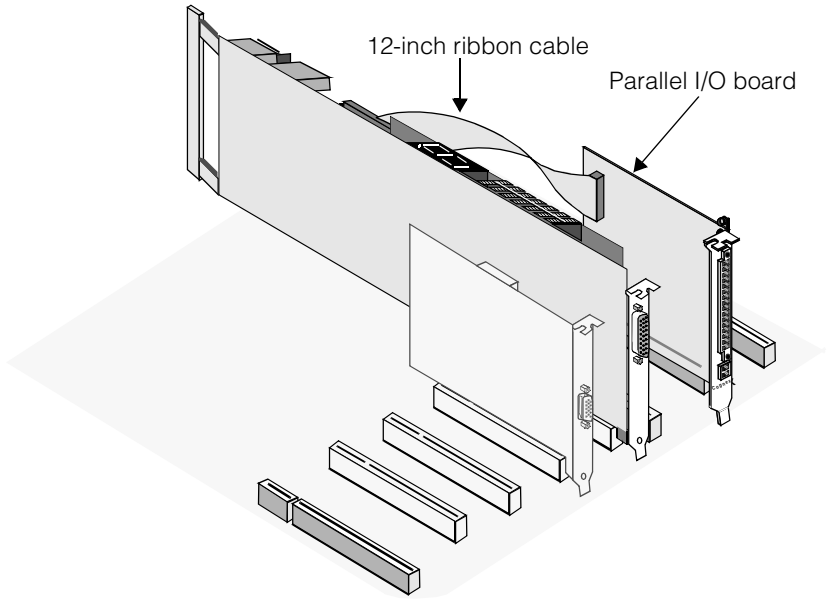


Figure 6. SIngle parallel I/O board installed with Cognex 5700/5900

Installing the Double-Board Configuration

When you install two boards, install the *TTL* board closest to the Cognex 5700/5900 and install the *OPTO/TTL* board in the adjacent ISA slot. Cognex supplies a 4-inch ribbon cable (shown in Figure 7) to connect the two parallel I/O boards. The connectors are keyed to ensure that they are inserted correctly.

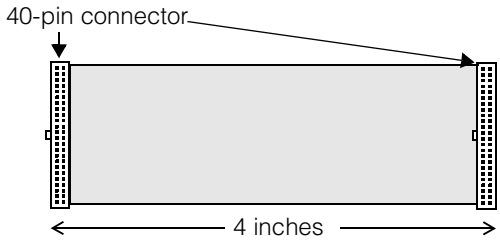


*Figure 7. Cognex 5700/5900 parallel I/O board 4-inch ribbon cable (*P/N 300-0133)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using Figure 8 as a reference, follow these steps to install two Cognex 5700/5900 parallel I/O boards:

1. Wear a wrist strap for ESD protection.
2. Install the *TTL* parallel I/O board in an ISA bus slot near the Cognex 5700/5900. Install the mounting screw in the board to secure it to the PC's back panel.
3. Install the *OPTO/TTL* parallel I/O board in the ISA bus slot adjacent to the one you used for the *TTL* board.
4. Attach the 12-inch ribbon cable to the Cognex 5700/5900 vision processor's parallel I/O port and to either of the two connectors on the *TTL* parallel I/O board.
5. Attach the 4-inch ribbon cable to the remaining connector on the *TTL* board and to the single connector on the *OPTO/TTL* board.

Figure 8 shows how the *TTL* board connects to the Cognex 5700/5900 and to the *OPTO/TTL* board. Note that the two parallel I/O boards function in exactly the same way when the connections on the *TTL* board are reversed.

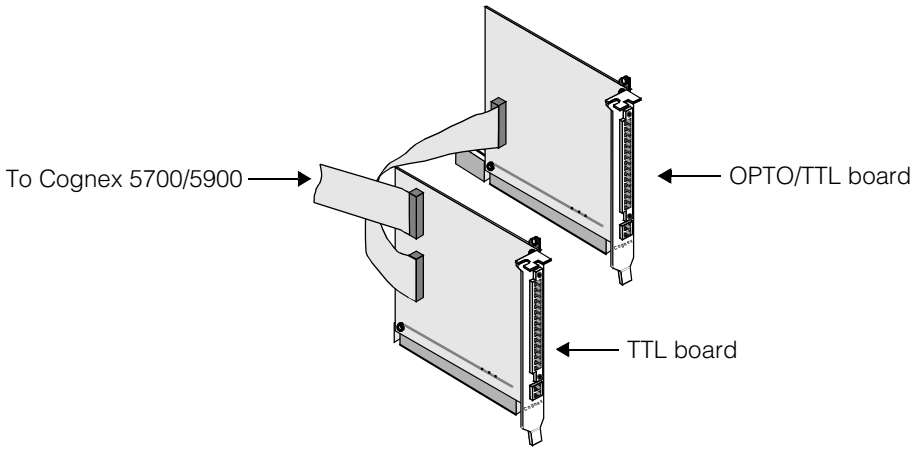


Figure 8. Two-board parallel I/O configuration

Installing the Serial Interface Adapter

The serial interface adapter (P/N 800-0037-2) provides access to the vision processor's serial I/O ports by bringing both of the serial I/O lines from the 10-pin IDC connectors on the Cognex 5700/5900 vision processor to DB-9F connectors on the PC's back panel.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To install the serial interface adapter:

1. Wear a wrist strap for ESD protection.
2. Shut down and power off your PC.
3. Remove the PC's cover.
4. Attach one ribbon cable (P/N 300-0107) to the serial 1 port on the installed Cognex 5700/5900 vision processor. See Figure 2 on page 28 to identify the location of the serial 1 port.

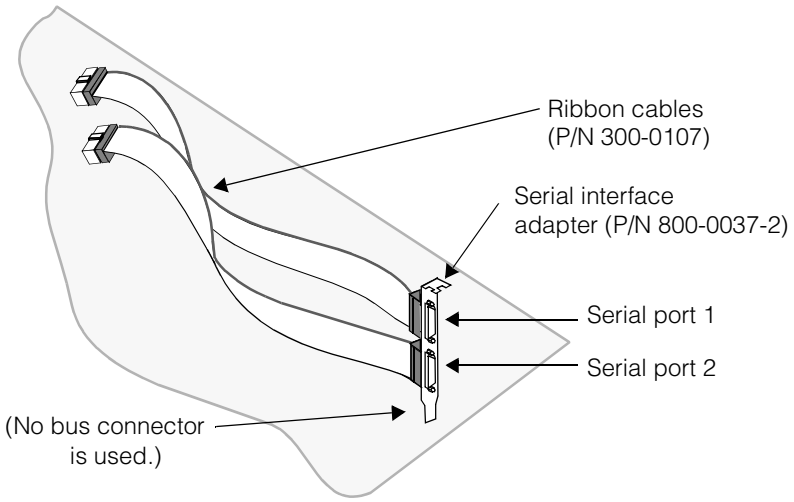


Figure 9. Serial interface adapter assembly

5. Attach the second ribbon cable to serial 2 port on the Cognex 5700/5900 vision processor. See Figure 2 on page 28 to identify the location of the serial 2 port.
6. Remove an available back panel slot cover from the PC and install the serial interface adapter in its place. Note that the serial connector assembly does not use the ISA or PCI bus. You can use any available slot on the PC's back panel.
7. Attach the serial adapter slot cover to the PC's back panel mounting bracket with a retaining screw.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Hardware Verification

Test your Cognex 5700/5900 installation before closing the system. To do so, connect the PC's mouse, keyboard, and display to your PC and follow these steps:

1. Turn on the display and system power. The Cognex 5700/5900 runs its diagnostics at power-on.
2. Allow Windows NT to complete booting and wait until the Cognex 5700/5900 completes its boot diagnostics and is ready. If the diagnostics succeed, the 5700/5900's LED display shows "**P**".

   See Table 8 on page 57 for a list of the other possible LED values.

Closing the System

To close the system, reinstall the enclosure cover. Make sure that internal cables are properly secured and will not be dislodged or interfere with proper air circulation for cooling when the enclosure cover is installed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2 Cognex 5700/5900 Hardware Installation

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 3

PVE Software Installation

This chapter describes the procedure for installing the PVE revision 8.1.1 software that provides support for the Cognex 5700/5900 hardware.

Software Installation Instructions

PVE revision 8.1.1 uses a Windows-based installer. Because PVE 8.1.1 is released for Windows NT only, it is presumed that your development host PC is running Windows NT 4.0 and has a CD-ROM drive, or access to a CD-ROM drive over a network.

Before You Install

If you have an existing *\cognex* directory, Cognex strongly recommends that you rename this directory and install the new PVE software in a new *\cognex* directory. You can rename directories in Windows NT 4.0 as follows: select the directory's name in an Explorer window, press the <*F2*> key, then type the directory's new name.

PVE revision 8.1.1 installs a subset of the demo code installed by previous revisions, because much of the demo code in previous releases presumes the presence of the **caq_** acquisition system. The Cognex 5700/5900 hardware supported by PVE revision 8.1.1 uses the **cua_** acquisition system instead of the **caq_** acquisition system. Thus, if you install PVE revision 8.1.1 in an existing *\cognex* directory, old and new demo code and header files will be mixed, with undefined results.

To perform the required device configuration under Windows NT, you must have Administrator privileges for your Windows NT system.

Install your 5700/5900 vision processor before installing the PVE software. Doing so allows the PVE installation program to configure and start the vision processor device driver. If your vision processor is not installed first, the PVE installation will install the device driver, but cannot start it until the hardware is installed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Installing PVE 8.1.1 Software

To install the PVE revision 8.1.1 software, perform the following steps.

1. Insert the PVE revision 8.1.1 CD-ROM in the CD-ROM drive.
2. The PVE 8.1.1 installation program starts up automatically on insertion of the CD-ROM.

   Note: If you eject and re-insert the CD-ROM while the installation program is running, another instance of the installation program starts automatically. If you must eject and re-insert the CD-ROM during the installation, hold down either *<Shift>* key while re-inserting the CD-ROM and for several seconds thereafter. This suppresses the automatic startup of the installation program.
3. The Setup program displays the **Welcome** dialog box:

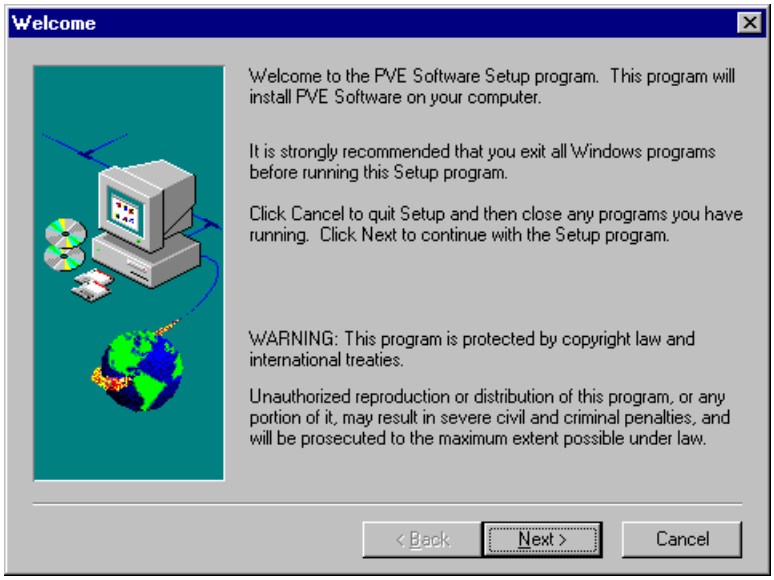


Figure 10. The Welcome dialog box

Click **Next >**.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4. The Setup program displays the **Choose Destination Location** dialog box:

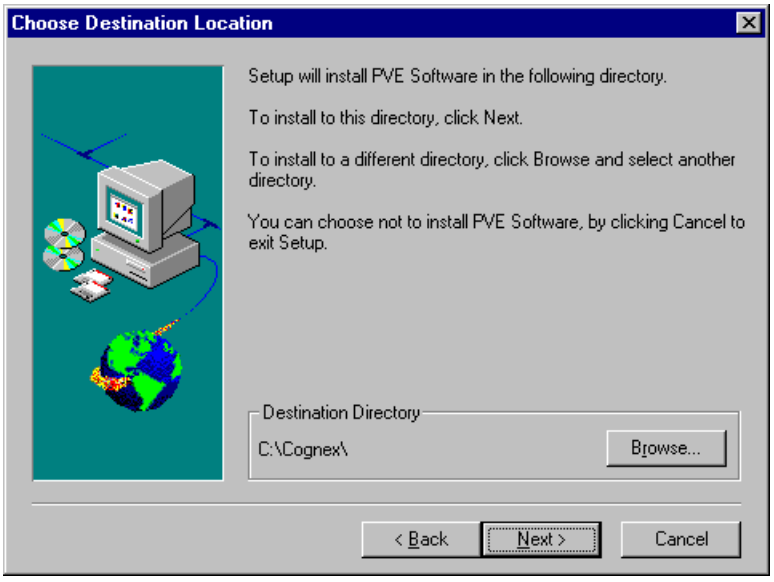


Figure 11. The Choose Destination Location dialog box

Choose the destination for the installation.

To install the PVE software in the default directory, click **Next >**. By default, the Setup program installs the software in a directory named *\cognex* on your system root drive. Cognex recommends installing into a directory named *\cognex.*

To install the PVE software in a different directory, click **Browse** to choose a new directory.

Click **Next >**.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5. The Setup program displays the **Registration** dialog box. Type your PVE Product Key in the text box:

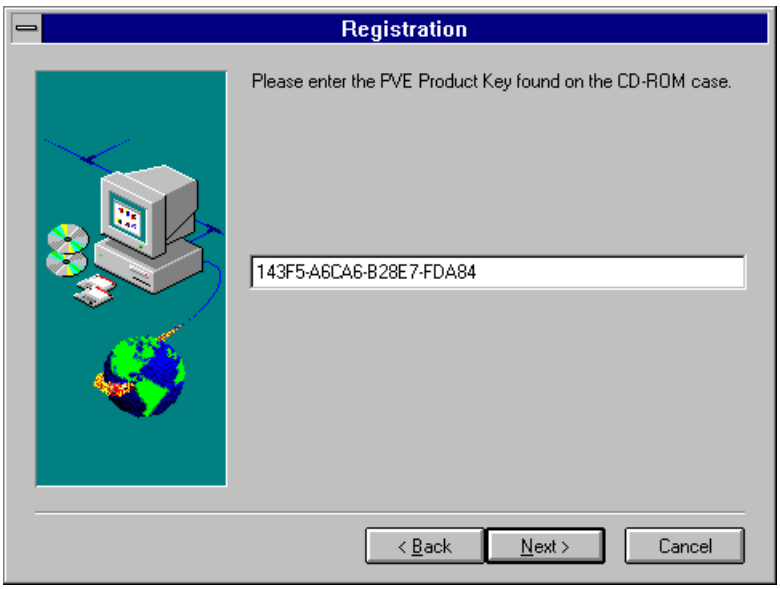


Figure 12. The Registration dialog box

You can find your PVE Product Key on the CD-ROM case. The product key defines the set of PVE components your license entitles you to install. You must type a product key to continue the installation.

Note: Type the product key *exactly* as it appears on the label of your CD-ROM case, including upper- and lower-case letters, numbers, and hyphens.

The installation program writes your product key to a file called *license.txt* in your specified installation directory. You can retrieve the product key from this file if you need to re-install all or part of PVE revision 8.1.1.

Click **Next >**.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6. The Setup program displays the **Options Selection** dialog box, which lists the components you can install for the vision processors you selected.

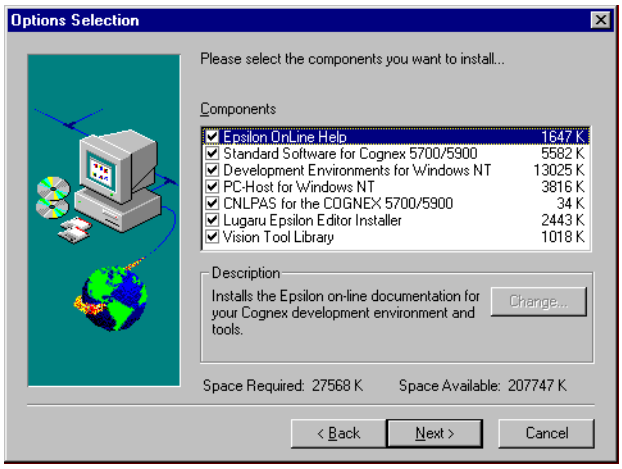


Figure 13. The Options Selection dialog box

7. Select the options you want to install.

   A check mark to the left of a component name means the Setup program will install that component. Click to the left of the component name to enable the check mark.

   To install the complete Vision Tool Library, make sure that there is a check mark to the left of the **Complete Vision Tool Library** component. Then click **Next >** and go on to step 8.

   To install specific vision tools, highlight the **Complete Vision Tool Library** entry, then click **Change**.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The Setup program displays the **Select Sub-components** dialog box:

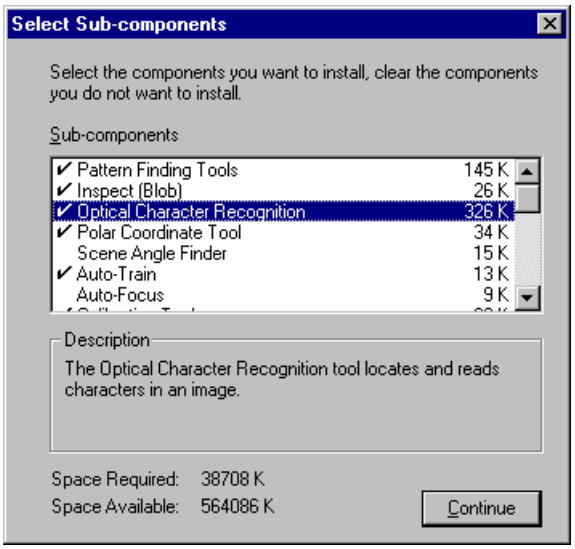
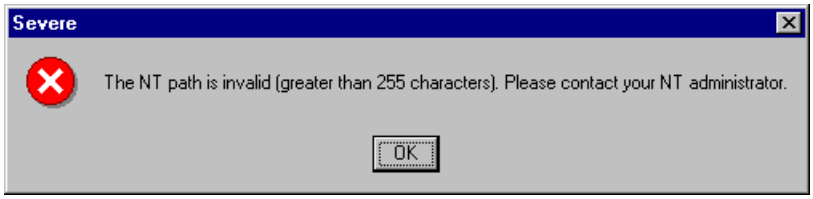


Figure 14. The Select Sub-components dialog box

Click the check marks to the left of the tool names to indicate the tools you want to install.

Click **Continue**.

8. At this point, the Setup program adds any required entries to your system's search path. If it is unable to do so because your search path is too long to accommodate the new entries, the following message is displayed and the installation terminates. You will need to reduce the length of the path before running the Setup program again.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

If the Setup program is able to add to the search path, or if no additions are required because of an earlier installation, the Setup program begins to install the PVE software. During the installation you will see the windows shown below except when the Setup program needs to communicate with you.

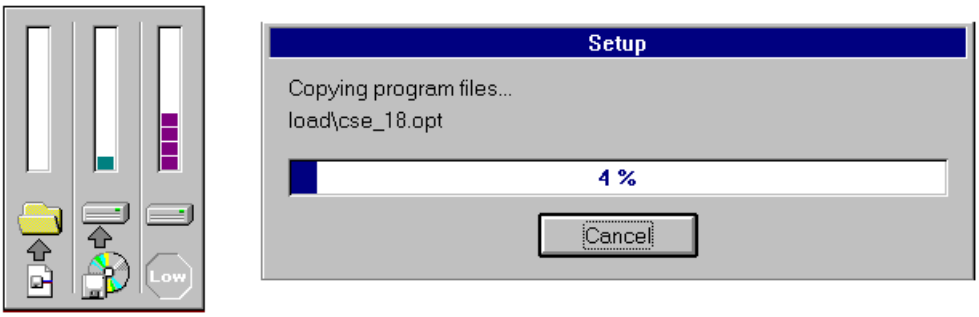


Figure 15. Setup program progress report

9. When all files are copied, Setup invokes Vision Processor Setup to install and configure the 5700/5900's device driver.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

If your 5700/5900 is already installed, you can install and start the device driver. If your 5700/5900 is not yet installed, you can install the device driver now, but will not be able to start it until the hardware is installed.

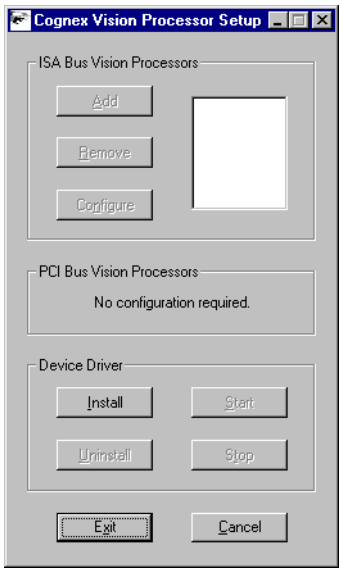
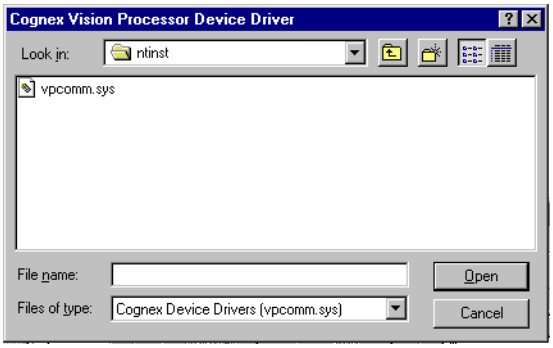


Figure 16. Cognex Vision Processor Setup application window

On a development host with no previous vision processor device driver installed, click the Install button. The utility prompts for the path to the Cognex device driver file, *vpcomm.sys*. Specify the file *\cognex\pchost\ntinst\vpcomm.sys*.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

After installing the device driver file, the utility prompts for permission to start the device driver. If your 5700/5900 is installed, click **Yes**. If your hardware is not yet installed, click **No**; in this case, the driver will start automatically when you restart your computer with the hardware in place.

On a development host with a vision processor device driver from an earlier PVE release, the device setup utility detects the previous driver and prompts for permission to uninstall it before proceeding. For further instructions on this case, see *Device Driver Installation* below.

10. After configuring the device driver, you must log out of Windows NT and log back in for the search path changes to take effect.

Device Driver Installation

The PVE 8.1.1 installation program invokes Vision Processor Setup as part of the installation procedure. If you don't complete the device setup during installation, you can run Vision Processor Setup any time after installation by double-clicking the **Vision Processor Setup** icon in the **Cognex Vision Processor** group in the **Start->Programs** menu.

You can configure your host PC to recognize your vision processor any time after installing the PVE software, but you must do so before you use your vision processor.

Vision Processor Setup searches your system for a previous version of the Cognex vision processor device driver and detects whether that driver is currently loaded and running. Vision Processor Setup responds according to which of the five possible conditions it detects.

1. Case One: No previous vision processor device driver is installed and running. Vision Processor Setup shows only the Install button active. Click this button to install and optionally start the PVE 8.1.1 device driver.
2. Case Two: A pre-8.1.1 device driver is installed and running. Vision processor device drivers shipped in PVE releases prior to 8.1.1 cannot be dynamically unloaded from memory while running. Vision Processor Setup detects this case and shows a message instructing you to stop the device driver manually, reboot, then rerun the configuration utility to remove the old driver. Follow this procedure:
    a. Start the Windows NT Control Panel (**Start**->**Settings**->**Control Panel**).
    b. Double-click the **Devices** icon.
    c. Select the **Vision Processor** device. (Do *not* click the Stop button!)
    d. Click the **Startup** button.
    e. Select **Disabled** and click **OK**.
    f. Close the **Devices** dialog box.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

g. Restart Windows NT.

h. After logging back in to Windows NT, re-run Vision Processor Setup. This time it will detect Case Three.

3. Case Three: A pre-8.1.1 device driver is installed but not running. Vision Processor Setup detects this case and shows a message instructing you uninstall the old device driver before proceeding. Press **OK** to clear this message, then click the **Uninstall** button.

   After uninstalling the old driver, Vision Processor Setup re-displays with only the **Install** button active. Press the **Install** button to install and optionally start the PVE 8.1.1 device driver.

4. Case Four: A PVE 8.1.1 or later device driver is installed and running. In this case, all four buttons in the Device Driver section are active. Click the **Reinstall** button to install a newer version of the device driver. Click the **Uninstall** button to stop the current device driver and remove the device driver file from your disk. Click the **Stop** button to unload the device driver from memory. Click the **Restart** button to stop, then start the current device driver.

5. Case Five: A PVE 8.1.1 or later device driver is installed but is not currently running. In this case, the Device Driver section displays three active buttons. Click the **Start** button to load and start the currently installed device driver. Use the **Reinstall** and **Uninstall** buttons as described for Case Four.

Installing Epsilon

If you chose to install the Lugaru Epsilon Editor Installer files in step 7 on page 41, the directory *\cognex\epsilon* does not contain the Epsilon editor, but contains the installation programs that will place either Epsilon 6.0 or Epsilon 8.0 on your hard disk.

Epsilon 8.0 is intended for use with Cogtalk under Windows NT, and includes both console and Windows versions of Epsilon.

Epsilon 6.0 is intended for use in a console session under Windows NT. Epsilon 6.0 supports a command buffer in which you can run the Console Cogtalk or Vtalk communications utilities.

- To install Epsilon 8.0, run the program *setup.exe* from Windows NT Explorer. When the Epsilon installation program asks for a serial number, enter “69787”.
- To install Epsilon 6.0, run the program *install.exe* from a command prompt.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Note: The installer for Epsilon 6.0 prompts you to install Epsilon 8.0 instead if you intend to run any version of Windows. However, the command prompt version of Epsilon 8.0 does not include the process-buffer feature required to use Epsilon in Vtalk mode. For use with the PVE software, Cognex recommends that you install Epsilon 6.0 for use in command prompt sessions, in contradiction to the message displayed by the Epsilon 6.0 installer.

Cognex recommends that you install Epsilon in its suggested directory or in another directory of your choosing. However, do *not* install Epsilon in *\cognex\epsilon*, because this is the directory containing the Epsilon installation files. After installing Epsilon, you can delete the files in *\cognex\epsilon* if you need to reclaim the disk space.

See *Chapter 3, Software Installation*, in the *5000 Getting Started* manual for instructions on setting up Epsilon to work with your PVE software. Follow those instructions as written, except substitute *cogtalk.e* whenever the instructions refer to *wintalk.e*.

Note: On-line documentation is included with the Epsilon software. If you would like to order a printed Epsilon manual, contact your local Cognex sales representative.

Installing Adobe Acrobat

To view the online versions of the PVE manuals, you must use the Adobe Acrobat reader. If Acrobat is not already on your system, the PVE CD-ROM includes Acrobat installation programs. There are separate versions of Adobe Acrobat for 16-bit and 32-bit Windows.

You can install the Adobe Acrobat Reader under Windows NT on your PVE 8.1.1 development host, or under Windows 3.1, Windows 95, or Windows NT on another PC. Follow these steps:

1. Insert the Cognex CD-ROM in your CD-ROM drive. Under Windows 95 and Windows NT, hold down either *<Shift>* key while inserting the CD-ROM to suppress the automatic startup of the PVE installation program.
2. Using the File Manager or Explorer, run the appropriate installation application located in the *\acroinst* directory on the CD-ROM.
   a. To install the 16-bit version of Acrobat, run *ar16e301.exe.*
   b. To install the 32-bit version of Acrobat, run *ar32e301.exe.*
3. Follow the instructions presented by the Acrobat installation program.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using the Online PVE Manuals

The online versions of the PVE software and hardware manuals are located in the *\pdfs* directory on the CD-ROM. You can copy the entire *\pdfs* directory to your hard disk, or you can open the files from the CD-ROM. If you copy the *\pdfs* directory to your hard disk, Acrobat will work much more quickly. Copying all the files in the *\pdfs* directory requires approximately 25 Mbytes of disk storage.

The online PVE manuals are current as of PVE revision 7.6. Table 5 shows which Adobe Acrobat files contain which online manuals.

PDF file	Contains
5kacq.pdf	*5000 Image Acquisition and I/O*
5khw.pdf	*5000 Hardware*
de1.pdf	*3000/4000/5000 Development Environment 1*
de2.pdf	*5000 Development Environment 2*
gs5k.pdf	*5000 Getting Started*
image.pdf	*3000/4000/5000 Image Processing*
ss.pdf	*3000/4000/5000 System Software*
vision.pdf	*3000/4000/5000 Vision Tools*

Table 5. Cognex PVE documentation in Adobe Acrobat files

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Uninstalling PVE Software

To uninstall PVE 8.1.1 software, do not use the **Add/Remove Software** icon in the NT control panel. Instead, use the **Uninstall PVE Software** icon in the **Cognex Vision Processor** program group.

The **Uninstall PVE Software** icon performs three steps:

1. It invokes Vision Processor Setup, which allows you to stop and uninstall the vision processor device driver. Click the **Uninstall**, then **Exit** buttons to perform this step.
2. It invokes *badchipset.exe* with its *–r* option to remove the non-compliant chip set database from the NT registry.
3. It invokes the PVE uninstallation program.

If you use the Control Panel's uninstallation method, it only performs step 3.

Running the PVE uninstallation program does not completely remove all aspects of the PVE software. The device driver you installed for your Cognex 5700/5900 board remains in place if you didn't uninstall it first. Search path changes remain in place, as do **Start** menu items and any files added to your *\cognex* directory structure since initial installation. All files in *\cognex\working* remain in place, which allows you to uninstall and reinstall the PVE software without affecting any of your application development files.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3 PVE Software Installation

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter

4

Cognex 5700/5900 Hardware Description

This chapter describes the Cognex 5700/5900 hardware in the following sections:

- *Cognex 5700/5900 Vision Processor Components* on page 52 describes the components that make up the Cognex 5700/5900.
- *Mechanical Specifications* on page 54 provides a physical description of the Cognex 5700/5900, including information about shipping, environmental requirements, and safety standards.
- *Electrical Specifications* on page 56 describes the electrical interface to the Cognex 5700/5900, including connector pinouts and circuit descriptions.
- *Cognex 5700/5900 Parallel I/O Boards* on page 66 describes the two parallel I/O boards available to carry the Cognex 5700/5900 parallel I/O signals to the PC's back panel.
- *Cognex 5700/5900 Camera Support* on page 70 describes the Cognex 5700/5900 support for monochrome cameras.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Cognex 5700/5900 Vision Processor Components

This section describes the Cognex 5700/5900 vision processor and components.

Processors

The Cognex 5700 vision processor uses a Motorola MC68040 microprocessor running at 25 MHz. The Cognex 5900 uses a Motorola MC68060 microprocessor running at 50 MHz.

A floating point coprocessor in the MC68882 class is included in the MC68040 and MC68060 CPUs.

The 5700 and 5900 both include Cognex VC3 and VC4 dedicated vision processing chips.

Memory

The Cognex 5700/5900 system supports 16 or 32 MBytes of synchronous DRAM (SDRAM) depending on the system purchased.

Unlike the Cognex 5000-ISA models, the Cognex 5700/5900 does not contain dedicated video memory. The memory for image and graphics buffers is allocated from the vision processor's SDRAM main memory.

PCI Bus Interface

The Cognex 5700/5900 PCI bus interface is a 5 V, 32-bit interface that conforms to the PCI 2.1 standard.

Video Input System

Input video on the 5700/5900 is provided by a Cognex Video Module (CVM). Different models of the 5700/5900 have different CVMs, to meet different customer requirements. CVMs are *not* customer-upgradeable, and are factory-installed purchase options only.

The input video capabilities of available CVMs is discussed in *Cognex Video Modules and Cameras* on page 71.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Parallel I/O

The Cognex 5700/5900 communicates with devices such as strobes, sensors, and programmable controllers over parallel signal interface lines. An internal connector on the vision processor provides the following signals:

- Eight transistor-to-transistor logic (TTL) inputs
- Eight TTL outputs
- Eight bidirectional TTL lines
- Four pairs of opto-isolated outputs

Two parallel I/O boards are available to carry the signals from the internal connector to the PC's back panel.

Serial I/O

All Cognex 5700/5900 vision processors have two RS-232C serial ports, which can be used to connect the Cognex trackball or other serial device, and are identical to the serial ports on Cognex 5000-ISA vision processors. For a description of the serial port functions and pinouts, refer to *Serial I/O (RS-232C) Ports* on page 58.

Video Output

For 5700/5900 vision processors, image output from the CVM can occur:

- Through the vision processor's PCI bus to the hosts's SVGA adapter, for all CVMs
- Through a video output port, for CVM3

See *CVM Video Output* on page 73 for more information on the 5700/5900 video output capability.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Mechanical Specifications

The Cognex 5700/5900's board size is 12.28" x 4.2" (312 mm x 106.68 mm), and is a full length PCI card. Figure 17 shows the dimensions of the Cognex 5700/5900 board.

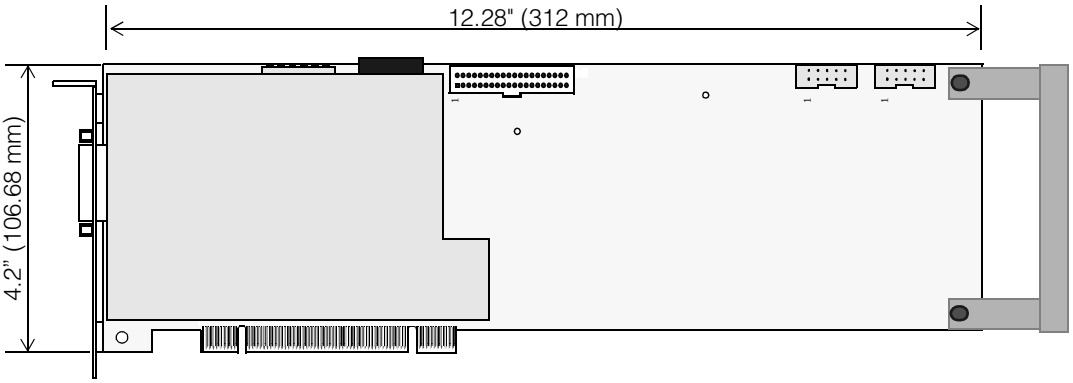


Figure 17. Cognex 5700/5900 board size

The Cognex 5700/5900 vision processor occupies one full-length 32-bit PCI bus slot. A card guide mounting bracket allows the board to use an available front mounting slot for extra stability. The Cognex 5700/5900 must be mounted firmly in place such that it cannot bend or flex under the normal vibrations that occur during the operation of mechanical equipment.

Environmental Requirements

Table 6 lists the environmental requirements for the Cognex 5700/5900 vision processor. These specifications are for the environment inside the PC where the Cognex 5700/5900 is installed.

	Operating Conditions	**Storage Conditions**
Cognex 5700/5900 Temperature	10° to 50° C	–40° to 65° C
Cognex 5700/5900 Humidity (non-condensing)	10% to 90%	10% to 90%

Table 6. Environmental requirements for the Cognex 5700/5900

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Shipping

All Cognex 5700/5900 vision processors are shipped in protective packaging and antistatic bags. Save all packing materials in case you need to ship the Cognex 5700/5900.

Standards

The Cognex 5700/5900 is designed and manufactured to meet the following worldwide standards for safety, electromagnetic compatibility, and electrostatic sensitivity when installed in a compliant PC enclosure:

European Community

- Safety:

  EN 60 950

  EN 60 204-1
- Electromagnetic emissions: EN 55 022 Class A
- Electromagnetic compatibility: EN 50 082-1

United States

- Underwriters Laboratories

  Safety: UL 1950

  Flammability: UL 94V-0
- Code of Federal Regulations Part 15: Federal Communications Commission Class A radio frequency emissions standard

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Electrical Specifications

This section describes the electrical interface to the Cognex 5700/5900, and provides information for connection and control of external hardware.

Cognex 5700/5900 Power Requirements

Input power to the Cognex 5700/5900 vision processor is provided through the PCI bus connector. Table 7 lists the input voltage and tolerance, and maximum PC power consumption required by an installed system.

Voltage	Maximum Cognex 5700/5900 draw	Allowable draw by cameras and serial devices	Total current
+5 V ±3%	3.4 A		3.4 A
+12 V ±5%	50 mA	750 mA	800 mA
-12 V ±5%	10 mA		10 mA

Table 7. Cognex 5700/5900 vision processor power requirements

If you use cameras that draw power from the video acquisition port, the +12 V power use increases accordingly. The combined total draw of cameras and serial devices may not exceed 750 mA, limited by PolySwitch circuit protection on the Cognex 5700/5900. A typical monochrome camera uses from 150 to 250 mA but some cameras draw much more. The standard Cognex trackball draws about 200 mA from the +12 V supply.

An external power adapter can be used to provide supplemental +12 V power to cameras attached to Cognex 5900 systems equipped with a camera breakout box. This avoids the use of system power for that purpose. See *Auxiliary Power Supply* on page 86 for more information.

Camera Electrical Specifications

For descriptions of the camera connectors, see the Camera Connector section for your CVM, in *CVM Descriptions* on page 74.

For a description of the camera input electrical specifications, see *Analog Camera I/O Electrical Specifications* on page 90.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Diagnostic Display

The 7-segment LED display on the Cognex 5700/5900 indicates the status of the vision processor. When the 5700/5900 initializes, it cycles through several numeric codes, which are normal initialization and diagnostic codes. When the board successfully completes its tests, it polls for a boot image over the PCI bus. Table 8 shows the possible display values and their meanings.

Display values	Meaning
1	Testing first megabyte of memory
2	Initializing data
3	Finding CPU clock speed
4	Running diagnostic routines
J	Debugging mode for Cognex internal use only
P	Polling for boot image
(spinning)	Loading a volume file
U	Running a volume file
(one or two horizontal lines)	Loading a saved system file
☰	Running a saved system file
F	Diagnostic, system, or hardware failure
.	Board was reset but is not running

Table 8. Possible values of Cognex 5700/5900 status display

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Serial I/O (RS-232C) Ports

All Cognex 5700/5900 vision processors have two RS-232C serial ports. Hardware handshake is not provided. Figure 18 shows the circuit equivalent for the RS-232C input and output (receive and transmit) for each serial port.

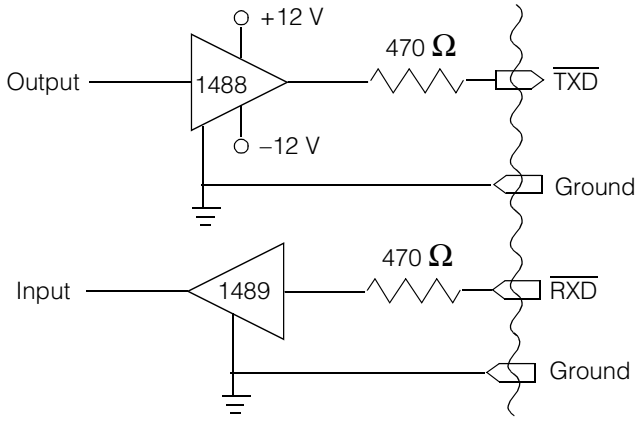


Figure 18. Cognex 5000 RS-232C input and output circuit

The serial connectors on the 5700/5900 are 10-pin 3M/Berg-type headers that mate with 10-pin IDC connectors. Refer to Figure 19 for the pin numbers of the serial I/O connectors.

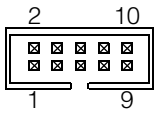


Figure 19. Serial I/O connector pin numbering diagram

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 9 lists the pinouts for serial lines 1 and 2.

Pin	Signal	DB-9F Pin
1	Chassis ground	1
2	No connection	6
3	$\overline{\text{TXD}}$	2
4	No connection	7
5	$\overline{\text{RXD}}$	3
6	No connection	8
7	+12 VDC (fused)	4
8	No connection	9
9	GND	5
10	No connection	N/A

Table 9. RS-232C serial I/O connector pinouts

The RS-232C pinout is designed so that a direct ribbon cable connection between the serial connectors and an industry-standard DB-9 female connector yields the pinout in the DB-9 as shown in Table 9.

Parallel I/O Port

The Cognex 5700/5900 has eight TTL inputs, eight TTL outputs, four opto-isolated outputs, and eight bidirectional general I/O TTL lines.

Cognex provides two Cognex 5700/5900 parallel I/O boards, each of which brings 16 of the 5700/5900's parallel I/O signals, along with two digital grounds, to the back of the PC through a ribbon connector. See *Pinout Values for TTL Board* on page 68 and *Pinout Values for OPTO/TTL Board* on page 69 for pinout information. See *Installing Parallel I/O Boards* on page 30 for installation instructions.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 20 shows the Cognex 5700/5900 TTL parallel input circuit.

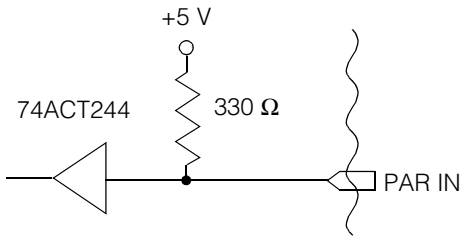


Figure 20. Cognex 5700/5900 parallel input circuit

To guarantee a valid low input, a logic circuit driving a TTL input directly to the vision processor must be capable of sinking a minimum of 14 mA at 0.5 V. Since each TTL input is pulled up by a 330 Ω resistor, you can use an open collector circuit to drive the inputs.

Figure 21 shows the Cognex 5700/5900 TTL parallel output circuit.

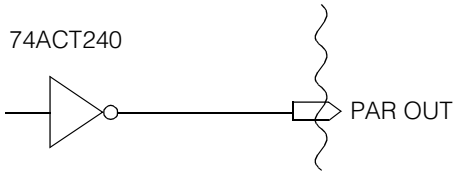


Figure 21. Cognex 5700/5900 parallel output circuit

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 22 shows the circuit for the Cognex 5700/5900 opto-isolated outputs.

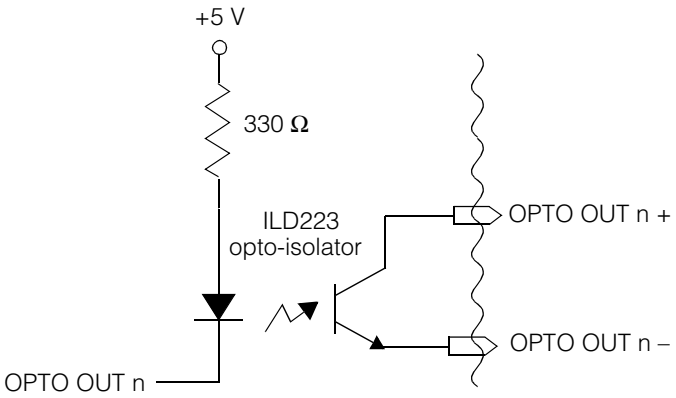


Figure 22. Cognex 5700/5900 opto-isolated output circuit

Figure 23 shows the Cognex 5700/5900 general I/O parallel circuit.

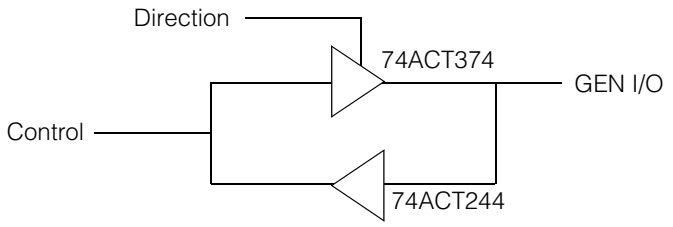


Figure 23. Cognex 5700/5900 general I/O parallel circuit

The Cognex 5700/5900's eight bidirectional general I/O lines are electronically equivalent to Figure 20 when being used for input and to Figure 21 when being used for output. The lines function as outputs by default. For information about the **cio_gio_direction()** function that sets the direction of the general I/O lines, refer to *Chapter 2, Basic I/O* in the *5000 Image Acquisition and I/O* manual.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The Cognex 5700/5900 parallel connector is a 40-pin shrouded double-row, high-density male header. Refer to Figure 24 for a description of the parallel I/O connector pin numbering.

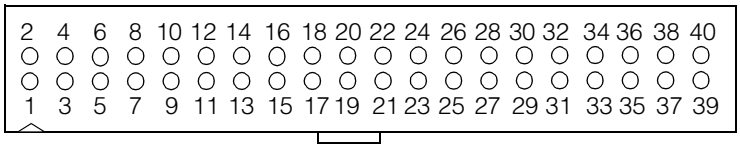


Figure 24. Parallel I/O connector pin numbering

A breakout panel is required for individual I/O connections.

Table 10 shows the pinout for the Cognex 5700/5900 parallel I/O port.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Pin	Signal	Pin	Signal
1	$\overline{\text{PAR IN 1}}$	2	$\overline{\text{PAR IN 2}}$
3	$\overline{\text{PAR IN 3}}$	4	$\overline{\text{PAR IN 4}}$
5	$\overline{\text{PAR IN 5}}$ (Trigger 1)	6	$\overline{\text{PAR IN 6}}$ (Trigger 2)
7	$\overline{\text{PAR IN 7}}$ (Trigger 3)	8	$\overline{\text{PAR IN 8}}$ (Trigger 4)
9	Ground	10	No connection
11	Ground	12	PAR OUT 1
13	PAR OUT 2	14	PAR OUT 3
15	PAR OUT 4	16	PAR OUT 5
17	PAR OUT 6	18	PAR OUT 7
19	PAR OUT 8	20	Ground
21	GEN I/O 8	22	GEN I/O 7
23	GEN I/O 6	24	GEN I/O 5
25	GEN I/O 4	26	GEN I/O 3
27	GEN I/O 2	28	GEN I/O 1
29	Ground	30	OPTO OUT 1 + (Strobe 1)
31	OPTO OUT 1 – (Strobe 1)	32	OPTO OUT 2 + (Strobe 2)
33	OPTO OUT 2 – (Strobe 2)	34	Ground
35	Ground	36	OPTO OUT 3 + (Strobe 3)
37	OPTO OUT 3 – (Strobe 3)	38	OPTO OUT 4 + (Strobe 4)
39	OPTO OUT 4 – (Strobe 4)	40	Ground

Table 10. Cognex 5700/5900 parallel I/O port pinout

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Hardware Video Triggers

Depending on the vision processor's video configuration, some or all of lines PAR IN 5 through PAR IN 8 are available for use as video triggers. A *video trigger* is a very fast, hardware-only acquisition trigger. A signal of 1 µs or longer will trigger an image acquisition if you are using a CCIR camera; a 1.2 µs signal is required for an RS-170 camera.

The number of video triggers that a vision processor supports is equal to the number of video channels on the vision processor. A single-channel Cognex 5700/5900 has one line (PAR IN 5) available for use as a video trigger. A four-channel Cognex 5900 with CVM1 has four available triggers (PAR IN 5 through PAR IN 8). A 5900 using CVM3 has two available triggers (PAR IN 5 and PAR IN 6). The correspondence between trigger lines and camera ports is shown in the *CVM Camera Numbering* section of each CVM's description in *CVM Descriptions* on page 74.

The OPTO OUT 1 through OPTO OUT 4 signals (*Strobe 1* through *Strobe 4* for CVM1, *Strobe 1 and Strobe 2* for CVM3) can be used to fire a strobe during acquisitions initiated by video triggers. The strobe lines and video trigger lines are linked in such a way that Strobe 1 can only be used with Trigger 1, Strobe 2 with Trigger 2, and so on.

For more information on the software interface that configures video triggers to perform strobed or unstrobed acquisitions, refer to *Video Trigger Property* on page 110.

Note that even when video trigger input lines are enabled for use by the video trigger mechanism, they are still available for use with the parallel I/O and trigger (**ctg_***) functions described in *Chapter 2, Basic I/O*, of the *5000 Image Acquisition and I/O* manual. Without video triggers, the **ctg_*** trigger functions are normally used to point to an action routine that performs image acquisition and other functions. When used with a video trigger, an action routine must not perform an image acquisition, but it can still be used to perform other functions, such as logging or record-keeping. In this case, a pulse on the video trigger line performs two actions: (1) an image is acquired, and (2) the vision processor is notified to initiate the action routine registered with **ctg_init()**.

Using Optically Isolated Outputs

You can wire the optically isolated outputs in one of two ways: *voltage source output wiring* or *contact closure output wiring*.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 25 shows an example of the voltage source output wiring configuration. You connect an external power supply (for example, 24 VDC) to Opto Out +, to Opto Out –, and to external equipment. The external equipment should present enough impedance so that no more than 125 mA flows through the circuit.

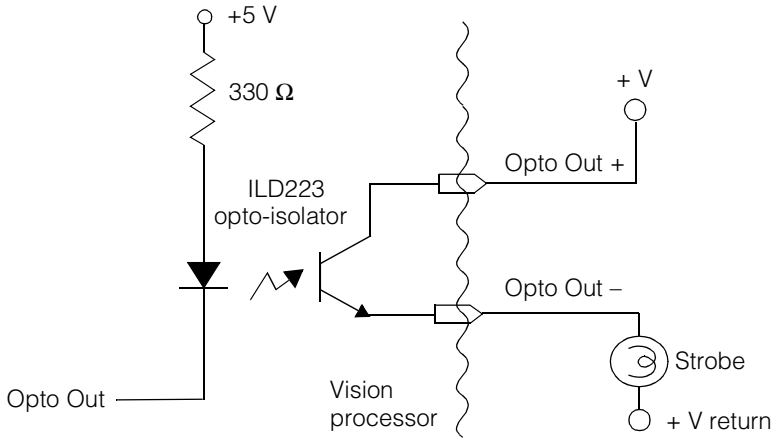


Figure 25. Optically isolated output: voltage source wiring

Figure 26 shows an example using the contact closure output wiring configuration. Output is taken directly from Opto Out + and Opto Out –. This output functions as a contact closure switch. Ensure that current (125 mA) and voltage (30 V) limits are not exceeded.

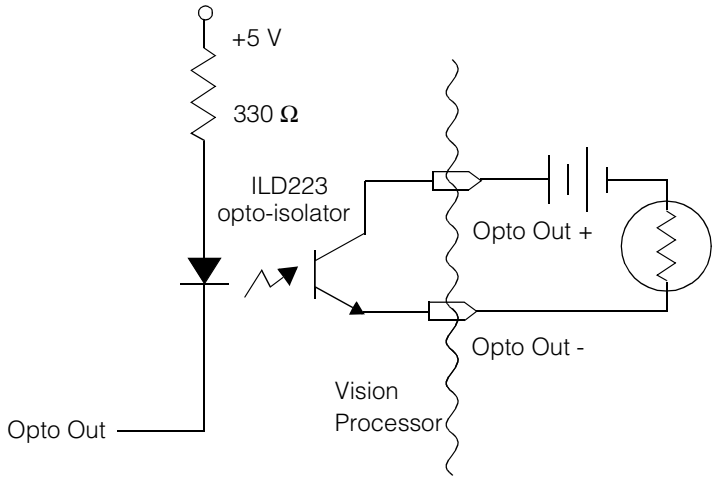


Figure 26. Optically isolated output: contact closure wiring

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Cognex 5700/5900 Parallel I/O Boards

There are two parallel I/O boards available for the Cognex 5700/5900: a *TTL* board (P/N 800-0003-01) and an *OPTO/TTL* board (P/N 800-0002-01). The board(s) you select determine the set of I/O lines available for your application, as shown in Table 11. The two boards can be used together for access to all of the Cognex 5700/5900 parallel I/O signals.

Board	TTL Inputs	TTL Outputs	General TTL I/O	Opto-isolated Outputs (in pairs)	Ground Pins
TTL	4	8	4	-	2
OPTO/TTL	4	-	4	4	2

Table 11. Sets of I/O lines available using Cognex 5700/5900 parallel I/O boards

The video trigger feature of the 5700/5900 makes use of one or more TTL input lines and, optionally, one or more opto-isolated output lines, as described in *Hardware Video Triggers* on page 64. To use the video trigger feature, you must install the OPTO/TTL board or a parallel I/O board of your own construction that makes these signals available.

Both parallel I/O boards plug into the the 8-bit portion of an ISA bus slot. The boards use the slot to connect to power and ground for protection against electrostatic discharge.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 27 shows the layout of the *TTL* parallel I/O board, which is the same as the layout of the *OPTO/TTL* board except where noted in the figure.

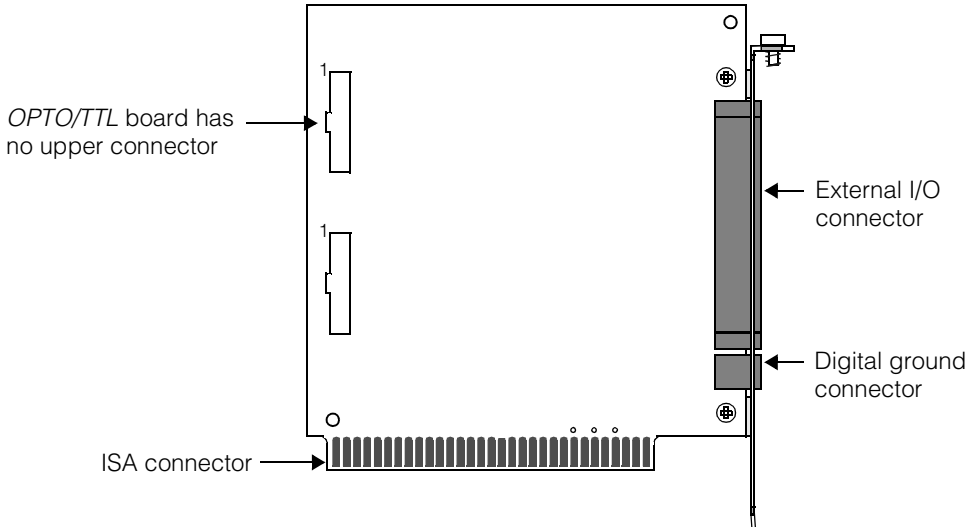


Figure 27. Layout of Cognex 5700/5900 parallel I/O boards

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Pinout Values for TTL Board

The TTL board's external I/O connector pins are labeled on the back panel faceplate. The pins are numbered from the top down, but are labeled on the faceplate as shown in Table 12. The connector plug (provided with each parallel I/O board), to which you attach external connection wires, is not labeled.

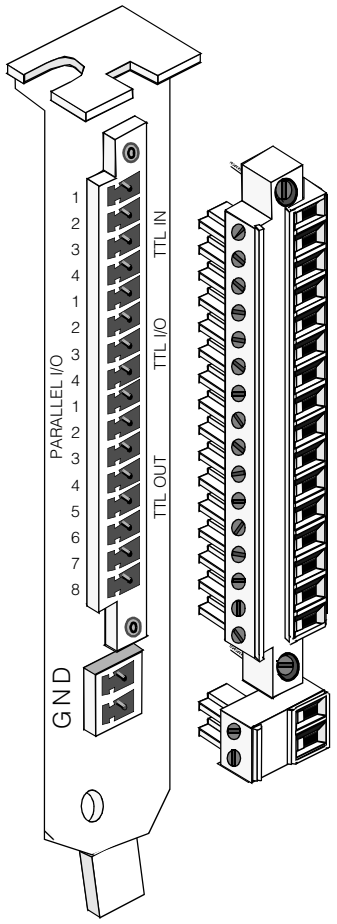
P/N 801-0003-01 (as labeled)	Pin	Interface Signal	Use
	1	$\overline{\text{PAR IN 1}}$	TTL Input 1
	2	$\overline{\text{PAR IN 2}}$	TTL Input 2
	3	$\overline{\text{PAR IN 3}}$	TTL Input 3
	4	$\overline{\text{PAR IN 4}}$	TTL Input 4
	1	5	GEN I/O 1	TTL I/O 1
	2	6	GEN I/O 2	TTL I/O 2
	3	7	GEN I/O 3	TTL I/O 3
	4	8	GEN I/O 4	TTL I/O 4
	1	9	PAR OUT 1	TTL Output 1
	2	10	PAR OUT 2	TTL Output 2
	3	11	PAR OUT 3	TTL Output 3
	4	12	PAR OUT 4	TTL Output 4
	5	13	PAR OUT 5	TTL Output 5
	6	14	PAR OUT 6	TTL Output 6
	7	15	PAR OUT 7	TTL Output 7
	8	16	PAR OUT 8	TTL Output 8
	GND		Ground	Ground
			Ground	Ground

Table 12. Pinout and faceplate connector for TTL parallel board

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Pinout Values for OPTO/TTL Board

The OPTO/TTL board's external I/O connector pins are labeled on the back panel faceplate. The pins are numbered from the top down, but are labeled on the faceplate as shown in Table 13. The connector plug, to which you attach external connection wires, is not labeled.

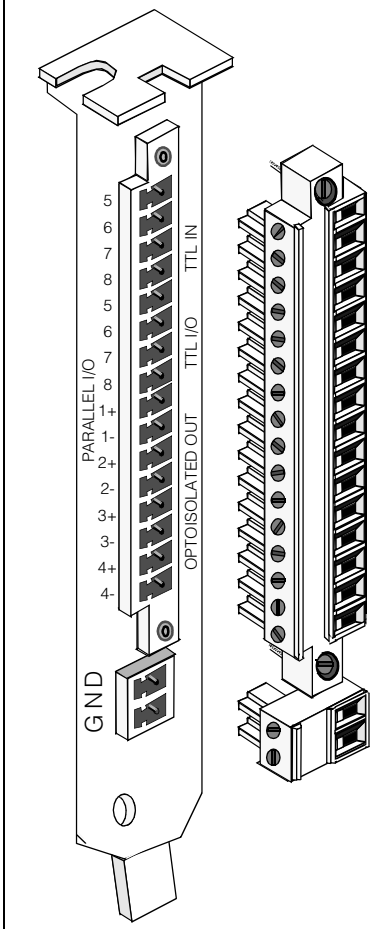
P/N 801-0002-01 (as labeled)		Pin	Interface Signal	Use
	5	1	$\overline{\text{PAR IN 5}}$	Trigger 1
	6	2	$\overline{\text{PAR IN 6}}$	Trigger 2
	7	3	$\overline{\text{PAR IN 7}}$	Trigger 3
	8	4	$\overline{\text{PAR IN 8}}$	Trigger 4
	5	5	GEN I/O 5	TTL I/O 5
	6	6	GEN I/O 6	TTL I/O 6
	7	7	GEN I/O 7	TTL I/O 7
	8	8	GEN I/O 8	TTL I/O 8
	1+	9	OPTO OUT 1+	Strobe 1+
	1-	10	OPTO OUT 1–	Strobe 1–
	2+	11	OPTO OUT 2+	Strobe 2+
	2-	12	OPTO OUT 2–	Strobe 2–
	3+	13	OPTO OUT 3+	Strobe 3+
	3-	14	OPTO OUT 3–	Strobe 3–
	4+	15	OPTO OUT 4+	Strobe 4+
	4-	16	OPTO OUT 4–	Strobe 4–
	GND		Ground	Ground
			Ground	Ground

Table 13. Pinout and faceplate connector for OPTO/TTL parallel board

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Cognex 5700/5900 Camera Support

Cognex 5700/5900 camera support is provided by one of three Cognex Video Modules (CVMs). A CVM is a factory-installed purchase option that provides video input and camera control for your vision processor. See *Cognex Video Modules and Cameras* on page 71 for more information about CVMs.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter

5

Cognex Video Modules and Cameras

This chapter describes one or more Cognex Video Modules (CVMs). A CVM is a factory-installed purchase option that provides video input and camera control for your vision processor. This chapter contains the following sections:

- *Cognex Video Module Overview* on page 72 describes the physical and input characteristics of various CVMs.
- *CVM Descriptions* on page 74 describes individual CVMs.
- *Connecting Analog Cameras* on page 83 describes the support for analog cameras.
- *Analog Camera I/O Electrical Specifications* on page 90 describes the electrical characteristics of the analog camera connector ports on various CVMs.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Cognex Video Module Overview

This section describes Cognex Video Modules (CVMs).

Baseboard or Daughtercard CVMs

A Cognex Video Module is a collection of circuitry that provides support for connecting analog or digital video cameras to a Cognex vision processor or frame grabber. Certain CVMs also provide a video output port.

A CVM can be built into the baseboard of a vision processor, or can be placed on a daughtercard that connects to the vision processor or frame grabber through the proprietary Cognex Video Module bus. Note that CVMs are only available as factory-installed purchase options, and are not user-replaceable or user-upgradeable in any way.

The Cognex 5700 uses CVM0. For the 5900, the following CVMs are supported:

- CVM0
- CVM1
- CVM3

Input Video Capabilities

Different CVMs are distinguished by their input video capabilities. Table 14 shows the input video capabilities of current CVM models.

CVM Model	Number of cameras supported	Acquisition channels	Cameras per acquisition channel	Simultaneous acquisitions	Video output ports
CVM 0	4 analog	1	4	1	0
CVM 1	4 analog	4	1	4	0
CVM 3	4 analog	2	2	2	1

Table 14. Video characteristics of current CVM models

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CVM Video Output

For 5700/5900 vision processors, image output from the CVM can occur:

- Through the vision processor's PCI bus to the hosts's SVGA adapter, for all CVMs
- Through a video output port, for CVM3

Bus-Based Video Output

All 5700/5900 models can pass image output through the PCI bus to a PCI-based video display adapter on the same bus, under the control of the host PC's operating system.

For bus-based image output, the video display adapter and PCI bus must meet certain minimum criteria to allow the 5700/5900 to transfer video images at its maximum frame rate for live video. When the minimum conditions are met, the Cognex 5700/5900 can send images at a real-time acquisition rate of 30 frames/second through an unsaturated PCI bus operating at 33 MHz. See *Requirements for Maximum Frame Rate* on page 20 for a further discussion of the minimum criteria to achieve the 5700/5900's maximum video transfer frame rate.

Port-Based Video Output

Vision processors that use the CVM3 video module can *also* display image output through the CVM3's output video port. The CVM3's HD-15M video output port can be attached to an NTSC/RS-170 monitor, or to a multisynchronous PC monitor that accepts BNC input.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CVM Descriptions

This section includes separate descriptions of each Cognex Video Module.

CVM0 Description

CVM0's functionality is built into the baseboard of the 5700 and of the single-channel 5900. CVM0 has one independent acquisition channel, but allows the connection of up to four analog cameras. One image acquisition can occur on one of the four camera channels at a time.

CVM0 Camera Support

The cameras supported by Cognex PVE software on CVM0 are the same as those supported by CVM1, as shown in Table 17 on page 76. The exception is that CVM0 cannot support two Pulnix 9700 or 9701 cameras simultaneously, because of power consumption restrictions.

CVM0 Analog Camera Connector

Analog cameras are attached to CVM0 through its DB-26F connector. To connect camera cables, you must first connect either the single camera cable (P/N 300-0202) or the double camera cable (P/N 300-0195). Although CVM0 supports up to four connected cameras, you can connect up to two cameras with Cognex-supplied cables. To connect three or four cameras to CVM0, you must construct your own cable or breakout box, using the pinout in Table 15 on page 75 as a guide. For more information, see *Connecting Analog Cameras* on page 83.

Figure 28 shows how the pins are numbered on CVM0's DB-26F connector.

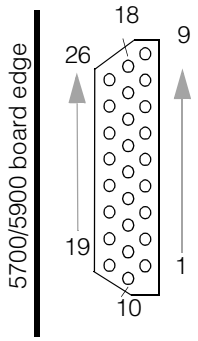


Figure 28. CVM0's DB-26F analog camera connector

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 15 shows the pinout for the DB-26F on the CVM0.

Pin	Signal Name	Pin	Signal Name	Pin	Signal Name
1	VIDEO-Cam2	10	VDrive-Cam1	19	CTRL1-Cam1
2	GND	11	HDrive-Cam1	20	CTRL2-Cam1
3	CTRL2-Cam2	12	CTRL1-Cam2	21	+12V
4	HDrive-Cam2	13	VDrive-Cam2	22	VIDEO-Cam1
5	VIDEO-Cam3	14	GND	23	+12V
6	GND	15	CTRL1-Cam3	24	CTRL2-Cam4
7	VDrive-Cam3	16	CTRL2-Cam3	25	HDrive-Cam4
8	GND	17	HDrive-Cam3	26	VDrive-Cam4
9	VIDEO-Cam4	18	CTRL1-Cam4		

Table 15. CVM0's video input connector pinout

CVM0 Camera Numbering

For purposes of hooking up trigger signals to the OPTO/TTL parallel I/O board, CVM0 cameras are numbered 1, 2, 3, and 4. CVM0 supports one trigger line and one strobe pair of lines at a time. All four cameras are tied to the same trigger and strobe, but only one camera is active at a time. The active camera is selected under PVE software control.

Thus, a pulse on the trigger line triggers an image acquisition on the currently selected camera port, and simultaneously fires the strobe.

Connect your trigger input signal to pin 1 (PAR IN 5, Trigger 1) of the OPTO/TTL board. Connect a strobe to strobe pair 1, which consists of signals OPTO OUT 1+ and OPTO OUT 1–. (Refer to Table 13 on page 69 for the OPTO/TTL board's pin and signal numbering layout.)

When using Cognex PVE software, address the cameras as 1, 2, 3, and 4. Camera 1 is connected to breakout cable port 1 and camera 2 is connected to port 2, and so on.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CVM0's camera numbering scheme is summarized in Table 16.

Camera port on breakout box	Associated trigger line on OPTO/TTL board	Associated strobe pair on OPTO/TTL board	PVE software address
1	1 (PAR IN 5)	1 (OPTO OUT 1+ and 1–)	1
2	same	same	2
3	same	same	3
4	same	same	4

Table 16. CVM0's camera numbering scheme

CVM1 Description

The CVM1 daughtercard has four independent acquisition channels, and allows the connection of up to four analog cameras. Image acquisitions can occur on all four camera channels simultaneously.

CVM1 Camera Support

The cameras supported by PVE software on CVM1 (and CVM0) are shown in Table 17. Note that your software package may not support all the cameras in this list; consult your software's release notes for the latest supported configurations.

Caution *Cameras must be connected using the Cognex cables shown in this table. Using non-Cognex camera cables could damage your vision processor, your camera, or both.*

The term *rapid reset* means the camera supports electronic shutter control.

Camera	EIA/ CCIR	Notes	Cognex Camera Cable Part Number
Sony XC-75	EIA	Interlaced	300-0099
Sony XC-75CE	CCIR	Interlaced	300-0099
Sony XC-77	EIA	Interlaced	300-0099

Table 17. CVM1 supported cameras and cables using PVE software

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Camera	EIA/ CCIR	Notes	Cognex Camera Cable Part Number
Sony XC-77CE	CCIR	Interlaced	300-0099
Sony XC-55	EIA	Rapid reset, progressive scan; must be used in E-DONPISHA II mode (set via switches)	300-0136
Sony XC-55BB	EIA	Remote camera head version of XC-55	300-0136
Pulnix TM-9700/01	EIA	Rapid reset, progressive scan	300-0127

Table 17. CVM1 supported cameras and cables using PVE software

CVM1 Analog Camera Connector

Analog cameras are attached to CVM1 through its DB-26F connector. To connect camera cables, you must first connect a breakout box (P/N 800-5637-3 or double camera cable P/N 300-0197). For more information, see *Connecting Analog Cameras* on page 83.

Figure 29 shows how the pins are numbered on the CVM1's DB-26F connector.

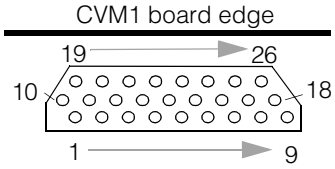


Figure 29. CVM1's DB-26F analog camera connector

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 18 shows the pinout for the analog camera connector on CVM1.

Pin	Signal Name	Pin	Signal Name	Pin	Signal Name
1	VIDEO-Cam3	10	VDrive-Cam4	19	CTRL1-Cam4
2	GND	11	HDrive-Cam4	20	CTRL2-Cam4
3	CTRL2-Cam3	12	CTRL1-Cam3	21	+12V
4	HDrive-Cam3	13	VDrive-Cam3	22	VIDEO-Cam4
5	VIDEO-Cam2	14	GND	23	+12V
6	GND	15	CTRL1-Cam2	24	CTRL2-Cam1
7	VDrive-Cam2	16	CTRL2-Cam2	25	HDrive-Cam1
8	GND	17	HDrive-Cam2	26	VDrive-Cam1
9	VIDEO-Cam1	18	CTRL1-Cam1		

Table 18. CVM1's (and CVM3's) pinout for analog camera connector

CVM1 Camera Numbering

For purposes of hooking up trigger signals to the OPTO/TTL parallel I/O board, CVM1 cameras are numbered 1, 2, 3, and 4. To trigger an acquisition on the camera connected to breakout box port 1, connect your trigger input signal to pin 1 (PAR IN 5, Trigger 1) of the OPTO/TTL board. To trigger an acquisition on the camera connected to port 2, use pin 2 (PAR IN 6, Trigger 2) and so on. (Refer to Table 13 on page 69 for the OPTO/TTL board's pin and signal numbering layout.)

The pairs of strobe signals on the OPTO/TTL board are numbered 1 to 4. That is, strobe pair 1 consists of signals OPTO OUT 1+ and OPTO OUT 1–. For CVM1, strobe pairs 1 to 4 correspond to camera ports 1 to 4. Thus, an acquisition on camera port 1 simultaneously fires a strobe connected to strobe pair 1, an acquisition on camera port 2 fires a strobe on strobe pair 2, and so on.

When using Cognex PVE software, address the cameras as 1, 2, 3, and 4. Camera 1 is connected to breakout box port 1, camera 2 is connected to port 2, and so on.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CVM1's camera numbering scheme is summarized in Table 19.

Camera port on breakout box	Associated trigger line on OPTO/TTL board	Associated strobe pair on OPTO/TTL board	Cognex software address
1	Trigger 1 (PAR IN 5)	Strobe pair 1 (OPTO OUT 1+ and 1–)	1
2	Trigger 2 (PAR IN 6)	Strobe pair 2 (OPTO OUT 2+ and 2–)	2
3	Trigger 3 (PAR IN 7)	Strobe pair 3 (OPTO OUT 3+ and 3–)	3
4	Trigger 4 (PAR IN 8)	Strobe pair 4 (OPTO OUT 4+ and 4–)	4

Table 19. CVM1's camera numbering scheme

CVM3 Description

The CVM3 daughtercard has two independent acquisition channels, and allows the connection of up to four analog cameras. Image acquisitions can occur on two of the four cameras simultaneously, one acquisition on each channel. CVM3 also supports an RS-170 video output port.

CVM3 Camera Support

The cameras supported by PVE software on CVM3 are shown in Table 20. Note that your software package may not support all the cameras in this list; consult your software's release notes for the latest supported configurations.

Caution *Cameras must be connected using the Cognex cables shown in this table. Using non-Cognex camera cables could damage your vision processor, your camera, or both.*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The term *rapid reset* means the camera supports electronic shutter control.

Camera	EIA/ CCIR	Notes	Cognex Camera Cable Part Number
Sony XC-75	EIA	Interlaced	300-0099
Sony XC-75CE	CCIR	Interlaced	300-0099
Sony XC-77	EIA	Interlaced	300-0099
Sony XC-77CE	CCIR	Interlaced	300-0099
Sony XC-55	EIA	Rapid reset, progressive scan; must be used in E-DONPISHA II mode (set via switches)	300-0136
Sony XC-55BB	EIA	Remote camera head version of XC-55	300-0136

Table 20. CVM3 supported cameras

CVM3 Analog Camera Connector

Analog cameras are attached to CVM3 through its DB-26F connector. To connect camera cables, you must first connect a breakout box (P/N 800-5637-3). The CVM3 camera connector is the same as the CVM1 camera connector (see *CVM1 Analog Camera Connector* on page 77).

CVM3 Video Output

CVM3 includes a video output port that allows you to display images from one or more attached cameras onto a monitor. Note that there is no default displayed image; you must initiate display onto the monitor using software commands.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The CVM3 video output port connects to monitors using Cognex cable 300-0209. This cable splits the signals from the CVM3's 15-pin video output port into color-coded red, green, and blue BNC signal cables. You can connect the types of monitors shown in Table 21 to the CVM3's video output port.

Monitor Type	Notes
NTSC/RS-170 with 3 BNC inputs	Match the cable's colors to R, G, and B inputs.
RS-170 monochrome with one BNC input	Connect the green BNC connector on cable 300-0209 to the monitor.
Multisynchronous PC monitor with BNC inputs	Use only R, G, and B inputs, even if the monitor has a 4th or 5th BNC input.

Table 21. Monitor types accepted by CVM3's video output port

As shown in Table 22, the pinout of the CVM3's video output port is compatible with VGA standards.

Pin	Signal Name	Pin	Signal Name	Pin	Signal Name
1	VIDEO-RED	6	GND	11	No Connect
2	VIDEO-GREEN	7	GND	12	No Connect
3	VIDEO-BLUE	8	GND	13	HSYNC
4	No Connect	9	No Connect	14	VSYNC
5	GND	10	GND	15	No Connect

Table 22. Pinout of CVM3's HD-15F video output port

CVM3 Camera Numbering

For purposes of hooking up trigger signals to the OPTO/TTL parallel I/O board, CVM3 cameras are numbered 1, 2, 3, and 4. Cameras 1 and 3 are tied together as one acquisition channel, while cameras 2 and 4 form another acquisition pair. To trigger an acquisition on the camera connected to breakout box ports 1 or 3, connect your trigger input signal to pin 1 (PAR IN 5, Trigger 1) of the OPTO/TTL board. A pulse on PAR IN 5 triggers an image acquisition simultaneously on camera ports 1 and 3.

To trigger an acquisition on the camera connected to ports 2 or 4, use pin 2 (PAR IN 6, Trigger 2). (Refer to Table 13 on page 69 for the OPTO/TTL board's pin and signal numbering layout.)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The pairs of strobe signals on the OPTO/TTL board are numbered 1 to 4. That is, strobe pair 1 consists of signal lines OPTO OUT 1+ and OPTO OUT 1–. For CVM3, strobe pair 1 corresponds to camera ports 1 and 3, while strobe pair 2 corresponds to camera ports 2 and 4. Thus, an acquisition on camera ports 1 or 3 simultaneously fires the strobe connected to strobe pair 1. Likewise, an acquisition on camera ports 2 or 4 fires the strobe connected to strobe pair 2.

When using Cognex PVE software, address the cameras as 1, 2, 3, and 4. Camera 1 is connected to breakout box port 1, camera 2 is connected to port 2, and so on.

CVM3's camera numbering scheme is summarized in Table 23.

Camera port on breakout box	Associated trigger line on OPTO/TTL board	Associated strobe pair on OPTO/TTL board	PVE software address
1	Trigger 1 (PAR IN 5)	Strobe pair 1 (OPTO OUT 1+ and 1–)	1
2	Trigger 2 (PAR IN 6)	Strobe pair 2 (OPTO OUT 2+ and 2–)	2
3	Trigger 1 (PAR IN 5)	Strobe pair 1 (OPTO OUT 1+ and 1–)	3
4	Trigger 2 (PAR IN 6)	Strobe pair 2 (OPTO OUT 2+ and 2–)	4

Table 23. CVM3's camera numbering scheme

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Connecting Analog Cameras

This section discusses how to connect analog cameras to Cognex vision processors and frame grabbers.

You may purchase cameras for use with your Cognex vision processor through Cognex, or you may use your own compatible cameras. Consult with your Cognex representative for cabling and other requirements if you plan to use non-standard cameras. Failing to do so may affect the warranty coverage on your vision processor or frame grabber.

For 5700/5900s that use CVM1 or CVM3, connect the camera breakout box (P/N 800-5637-3 or the double camera cable P/N 300-0197) to the DB-26F connector on the CVM's front panel. The camera breakout box splits the video port signals to provide four Hirose video connectors for image acquisition. Connect monochrome cameras with the appropriate cable on any of the four camera ports on the camera breakout box.

The CVM description section for your CVM lists the Cognex camera cable part numbers for the supported cameras.

For the 5700 or for 5900 models with baseboard CVM0, connect either the single camera cable (P/N 300-0202) or the double camera cable (P/N 300-0195) to the DB-26F connector on the vision processor. Although CVM0 supports up to two connected cameras, you can connect up to two cameras with Cognex-supplied cables. To connect three or four cameras to CVM0, you must construct your own cable or breakout box, using the pinout in Table 15 on page 75 as a guide.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 30 shows a monochrome camera connected to the 5700/5900 through the camera breakout box:

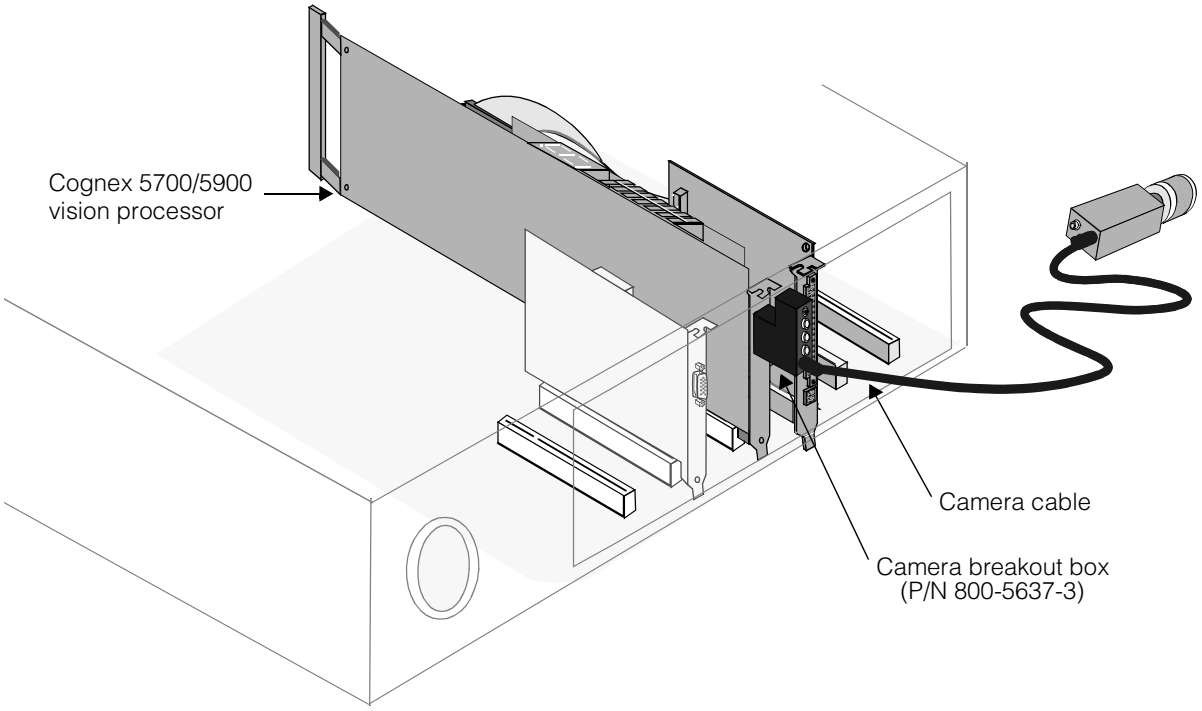


Figure 30. Monochrome camera connected to the Cognex 5700/5900 system

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Attaching the Camera Breakout Box

The Cognex 5900 series camera breakout box attaches to the DB-26F connector on the front panel of daughtercard CVM modules, and provides four Hirose 12P connectors for video cameras. The breakout box attaches securely to the back panel connector using captive screws on the breakout box. Figure 31 shows the camera breakout box.

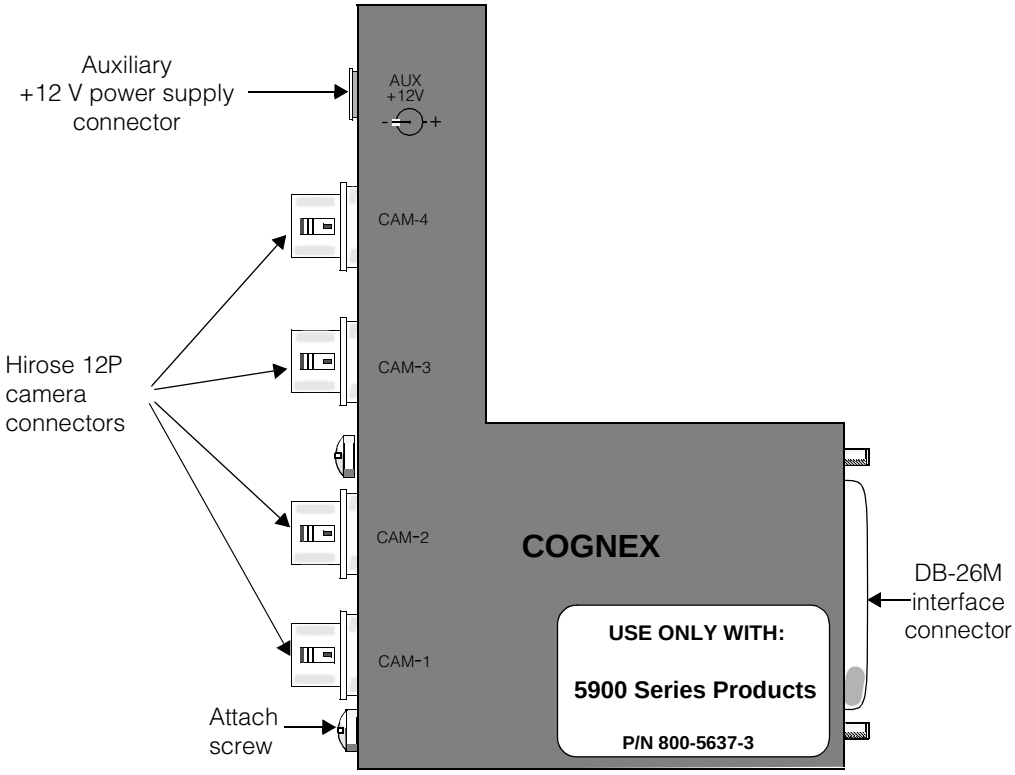


Figure 31. Cognex 5900 series camera breakout box assembly

Caution *Cognex breakout boxes are not interchangeable. The camera breakout box you use must have the 5900 Series Products label, or you may damage your cameras, the vision processor, or both.*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To attach the camera breakout box, follow these steps:

1. With power off to your PC and camera(s), plug the camera breakout box into the DB-26F connector on the back of the CVM. Secure the breakout box to the connector using a slotted screwdriver. The connection should be snug; do not overtighten.
2. For the first camera, use the appropriate Hirose camera cable to connect the DC IN/SYNC connector on the camera to the CAM 1 connector on the camera breakout box. Camera cable part numbers are listed in the CVM description section of this chapter.
3. To connect additional cameras, use the appropriate Hirose camera cable to connect the DC IN/SYNC connector on each camera to one of the open connectors on the camera breakout box.

Auxiliary Power Supply

The total +12 V power draw from cameras attached to CVMs must not exceed 750 mA. If your application requires more power, you must use a +12 V power adapter to supply the additional power. The adapter plugs into the auxiliary power supply socket at the top of the camera breakout box.

For example, if your application requires two Pulnix TM-9701 cameras, use the +12 V power adapter. Each Pulnix camera draws 500 mA for a total of 1 A, which exceeds the 750 mA maximum.

You can use the optional Cognex power adapter (P/N 158-0015) or an adapter that has the characteristics described in this section. Figure 32 shows the optional power adapter.

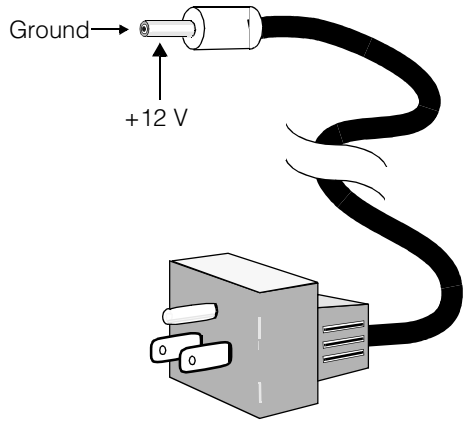


Figure 32. External power supply

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The Cognex external power adaptor (P/N 158-0015) has these characteristics.

- 2.1 mm x 5.5 mm coaxial male connector that is center-negative (ground)
- +12 VDC ±5% regulated output, 2.5 A
- Accepts 90 to 240 VAC input from a grounded wall plug (American or European)

Hirose Connector Pin Assignments

The following table shows the pinout for the Hirose connectors on the back of the Cognex 8000 series camera breakout box and splitter cables.

Pin	Signal
1	Ground
2	+12 VDC (fused)
3	Composite video (ground)
4	Signal (composite video)
5	Horizontal drive (ground)
6	Horizontal drive (signal)
7	Vertical drive (signal)
8	Camera control 2
9	No connection
10	Camera control 1
11	No connection
12	Vertical drive (ground)

Table 24. Hirose 12P connector pinout

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 33 shows the pin numbering sequence for Hirose 12P connectors on the camera breakout box and splitter cables.

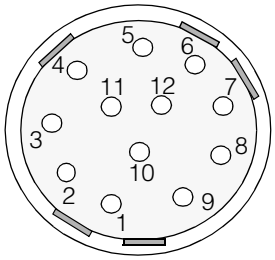


Figure 33. Hirose 12P connector pinout

Cognex Analog Camera Cables

The description for your CVM in this chapter (beginning in *CVM Descriptions* on page 74) lists the cables you must use with cameras supported by Cognex vision processors and frame grabbers. Not all cameras are supported by all vision processors. In all cases, the listed camera cables do not connect directly to the CVM, but connect to the camera breakout box (P/N 800-5637-3), the single camera cable (P/N 300-0202), or the double camera Y-cable (P/N 300-0195 or P/N 300-0197), depending on the CVM in use.

Connecting a Sony XC-55 Camera

The Sony XC-55/55BB camera in its delivered state provides a drop-in replacement for Sony XC-75 cameras. Cognex recommends using the XC-55/55BB in its E-DONPISHA trigger shutter mode for best performance. Two switch settings convert the XC-55/55BB to trigger shutter mode; one switch is external, while the other is internal. Follow this procedure:

1. On the rear panel of the XC-55/55BB, locate the 1N/1L switch. Place this switch in the **1N** position.
2. Remove the cover of the camera control module. Inside the camera control module, locate the SG-257 circuit board.

   On the SG-257 circuit board, locate rotary switch S2. Set this switch to the **E** position.

When used with Cognex vision processors, the Sony XC-55/55BB *must* be connected with Cognex camera cable 300-0136.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Connecting a Pulnix TM-9701 Camera (CVM1 only)

Image acquisition using a Pulnix TM-9701 camera is only supported in non-interlaced, synchronous, double-pulse mode. You must set the TM-9701 switches to **NON** (non-interlaced mode) and **ASY** (asynchronous mode), and you must set the rotary switch to **9** (double-pulse mode) as shown in Figure 34.

To control the TM-9701 shutter speed, you must configure the shutter speed using software controls in the Cognex software. For example, in Object Manager, use the Image object to configure shutter speed. Do not attempt to set the shutter speed using the rotary switch on the TM-9701; the rotary switch must be set to **9**.

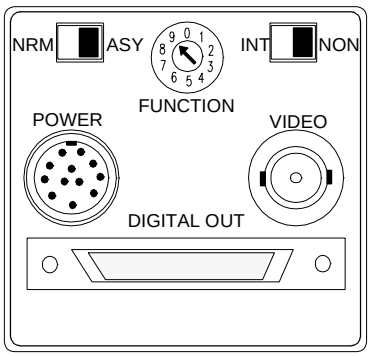


Figure 34. Pulnix TM-9701 back panel switches

Optional Camera Lenses

Cognex offers a lens kit (P/N 800-1000), which includes five lenses (8.5 mm, 12.5 mm, 16 mm, 25 mm, and 50 mm F2.8 macro), a C-mount adapter, and an extension tube set.

Cognex also offers a variety of camera lenses that you can order individually, including all the lenses in the lens kit, plus 50 mm, 75 mm, and 135 mm lenses. You can also separately order the C-mount adapter, a 2X extender lens, or an extension tube set

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Analog Camera I/O Electrical Specifications

The information in this section applies to all CVM modules that support analog video cameras. CVMs support analog cameras in different configurations, as described in Table 14 on page 72.

CAM VIDEO, the video input signal, is approximately 1 V peak to peak and is terminated with a 75 Ω resistor to ground. Each of the analog video signals connects to the CVM as shown in Figure 35.

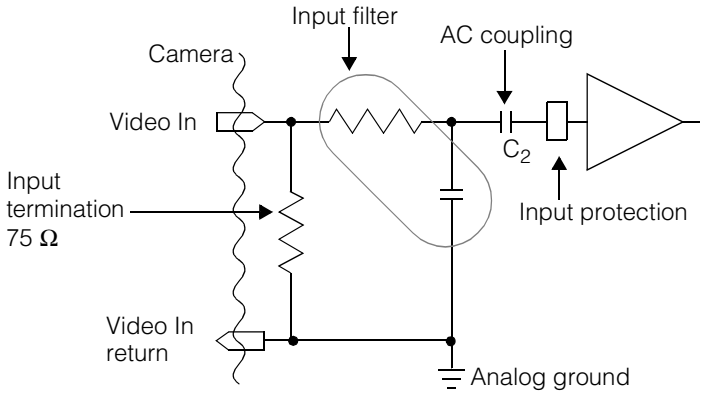


Figure 35. CVM analog video input circuit

The input filter provides noise rejection for switching and random noise above the CAM VIDEO rate. The video signal is terminated with a 75 Ω load and is AC-coupled into the video input section. The video signal is DC-restored internally. Digitization of the input signal is relative to the blanking level and is not affected by any DC offset that the camera may generate.

The input protection device provides over- and under-voltage protection to the A/D converter.

Figure 36 shows the totem-pole or push-pull circuit used to drive the horizontal and vertical drive signals of the CVM:

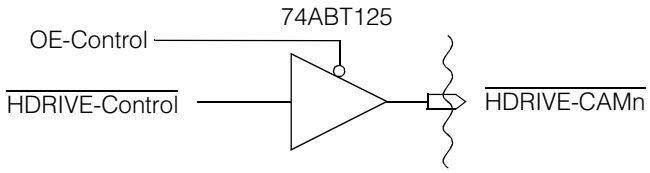


Figure 36. CVM analog camera control circuit

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

High-level and low-level outputs have the following TTL level characteristics:

- The guaranteed high-level output is greater than or equal to 2.7 V if the user load requirement for a TTL high level is less than or equal to 32 mA.
- The guaranteed low-level output is less than or equal to 0.8 V if the user load requirement for a TTL low level is less than or equal to 64 mA.

Analog Video Signal Timing

CVMs support analog video acquisition signal timings for a number of different image resolutions and standards.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 25 shows the video timing for both RS-170 (NTSC) and CCIR (ITU-R or PAL) formats. Notice that some parameters vary slightly, depending on the width and height of the video format you use.

Signal	RS-170 (NTSC)	CCIR (PAL)
Horizontal frequency	15.75 KHz	15.625 KHz
Period	63.5 µs	64 µs
Active scan	41.4 µs	51.5 µs
Blanking	22.6 µs	12.5 µs
Front porch	7 µs	2.2 µs
Sync	4.6 µs	4.9 µs
Back porch	11.0 µs	5.4 µs
Equalizing	2.5 µs	2.8 µs
Horizontal drive	6.4 µs	6.4 µs
Vertical frequency	60 Hz	50 Hz
Period	16.7 ms	20 ms
Active	15.0 ms	16.3 ms
Equalizing	0.18 ms	0.18 ms
Sync	0.18 ms	0.18 ms
Blanking	1.7 ms	3.7 ms
Vertical drive	0.57 ms	0.57 ms
Pixel clock	12.348 MHz	14.75 MHz

Vary slightly depending on width of format

Vary slightly depending on number of rows in format

Table 25. RS-170 and CCIR video timings

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter

6

5700/5900 User's Guide

Cognex's 5000 series of vision processors includes models based on the ISA bus and models based on the PCI bus. Cognex 5000 models based on the ISA bus are referred to as the *5000-ISA* vision processors, while those based on the PCI bus are referred to as the *5000-PCI* vision processors.

This chapter provides an overview of using and developing vision processing applications for Cognex's 5000-PCI vision processors, which includes the Cognex 5700 and 5900. This chapter describes the differences between Cognex's 5000-PCI and 5000-ISA vision processors, and describes the features in PVE 7.6 that are not supported on 5000-PCI vision processors under PVE 8.1.1.

This chapter contains the following sections:

- *Comparison with 5000-ISA Vision Processors* compares the hardware and software similarities and differences between the 5000-PCI and 5000-ISA vision processors.
- *Using the 5700/5900* describes how to use a 5000-PCI vision processor for those familiar with the 5000-ISA vision processors.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Comparison with 5000-ISA Vision Processors

This section compares the 5000-PCI vision processors (Cognex 5700 and 5900) with the 5000-ISA vision processors (Cognex 5200, 5400, and 5600).

Hardware Comparison

The 5000-PCI vision processors have more in common with their predecessors than they have differences. Table 26 compares the two hardware families.

5000-ISA Vision Processors	5000-PCI Vision Processors
CPU from the Motorola 680x0 family (MC68030)	Same (MC68040 or MC68060)
Math coprocessor chip	Built onto CPU chip
On-board RAM for vision processing applications	Same
Cognex proprietary vision coprocessors (VC3 and VC2 or VC4)	Same (VC3 and VC4)
32K shared memory segment on ISA bus for communication with host PC	32K shared memory segment on PCI bus for communication with host PC
Support for one to four analog video input ports for cameras or other video devices	Same
Support for rapid reset cameras	Same
Support for digital cameras (5600)	None
TTL and Opto/TTL parallel I/O lines for communication with devices such as strobes, sensors, and programmable controllers	Same
RS-232 serial I/O ports for communication with the Cognex trackball, or with other processors or devices	Same
On-board SCSI port for direct connection of storage devices	None. All file storage must occur over the PCI bus on the host PC's disks.

Table 26. Hardware comparison of 5000-ISA and 5000-PCI vision processors

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5000-ISA Vision Processors	5000-PCI Vision Processors
Dedicated VRAM memory for video and graphics buffers	None. All video and graphics buffers are allocated from the vision processor's main memory.
Monitor port for display of vision processor images	Only on systems equiped with CVM3. All display of vision processor images occurs over the PCI bus by means of the Cognex Software Display library. In addition, systems equiped with CVM3 can display on an attached monitor.
Digital video port for Cognex video mixer	None. Hardware-based video mixing is not required when using the Cognex Software Display library.
Red and green status LEDs	Seven-segment LED diagnostics display
Not available	Support in hardware for video triggers

Table 26. Hardware comparison of 5000-ISA and 5000-PCI vision processors

Software Comparison

Table 27 compares the software differences between the 5000-ISA and 5000-PCI vision processors.

5000-ISA Vision Processors	5000-PCI Vision Processors
Uses PVE revision 7.7.1 or earlier	Requires PVE revision 8.1 or 8.1.1
Supported under DOS, Windows 3.1, Windows 95, and Windows NT 4.0	Supported under Windows NT 4.0
Includes PCVtalk and Wintalk communication applications	Includes Console Cogtalk and Windows Cogtalk communication applications
Includes Vmix and PCI Vmix applications for display of vision processor images via Cognex video mixer boards	Includes Cogview application for software display of vision processor images
Image acquisition uses **caq_*** functions	Image acquisition uses Cognex Unified Acquisition (**cua_***) functions. No support for **caq_*** functions.

Table 27. Software comparison of 5000-ISA and 5000-PCI vision processors

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using the 5700/5900

This section describes the basics of using and developing applications on a 5700/5900 for those familiar with the 5000-ISA models of the Cognex 5000 series.

Building Volume Files

Like previous releases, PVE revision 8.1.1 includes the Sysgen utility for building bootable volume files. Sysgen is described in *Chapter 6, Using Sysgen*, of the *Development Environment 1* manual.

When used with the *-i* option, Sysgen prompts for the vision processor type. Enter *vm21* for the 5700/5900. PVE revision 8.1.1 is shipped with the support files required to generate only *vm21* type volume files. Although you can enter any valid vision processor type in response to Sysgen's prompt, the PVE 8.1.1 Sysgen only builds volume files for the Cognex 5700/5900. If you want to build volume files for other Cognex models, use the PVE release and Sysgen utility that shipped with those models.

Options Not Supported

The following command lists the optional components you can build into your 5700/5900 volume files.

```
sysgen -o -v vm21
```

The following software components are not available for the 5700/5900 in PVE revision 8.1.1:

- Color Acquisition
  Currently supported only on the Cognex 4400 and 5600.
- Command Interpreter
  Obsolete. This option provides an emulation of functionality available on the Cognex 1500 system.
- Image Database
  The Image Database functions currently require a directly attached SCSI disk, which is not available on the 5700/5900.
- RPC Dispatcher
  The PC Development Environment is not currently supported on the 5700/5900.
- Unified Acquisition
  Unified Acquisition is built into the baseline volume file for the 5700/5900, and therefore does not need to be available as an option.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Baseline Volume File

Previous PVE revisions supplied a demo volume file, *demo_xx.vol*, where *xx* is the vision processor family code. (See *Vision Processor Family Codes* in *Chapter 1, Cognex API Introduction*, of the *Development Environment 1* manual.) The demo volume file contained the baseline system software, plus the options and demonstration files necessary to run the Search and Train demo, as discussed in the *5000 Getting Started* manual.

For the 5700/5900, no demo volume file is available, but a baseline volume file is provided. The baseline volume file, installed as *\cognex\build\sss_21.vol,* is a volume file built with no options specified.

Enabling Live Video

On the 5000-ISA vision processors, the connection between camera port and monitor port is based in hardware. When a 5000-ISA vision processor is booted with any volume or saved system file, a live video image is immediately passed from an attached camera to an attached monitor, or to the Vmix or PCI Vmix applications by means of Cognex video mixer hardware. Thus, the 5000-ISA models only need to be loaded and running to display live video images.

If you are going to use the Cogview application, or your own application based on the Cognex Software Display library, display of live images from the vision processor is based in software and must be enabled. Thus, when you load a volume or saved system file on a 5000-PCI vision processor, live video does not display automatically from an attached camera to the Cogview application.

The following example V code represents the minimum set of commands to enable live video on the 5700/5900. This example enables the video format that is Cogview's default display format, which is 640x480 images for a Sony XC-75 RS-170 camera. To enable live video for a CCIR camera or for a different image size, edit these commands in the place marked, and specify a different format constant. The available constants are listed in Table 28 on page 107 in the *Cognex Unified Acquisition* chapter.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
/* cuadisp.v - enable live video on a 5700/5900 */

const cua_board   board   = cua_default_board_get();

/* Use the following for software display on any CVM */
const cua_monitor monitor = cua_default_monitor_get();

/* Use the following for CVM 3's physical monitor */
/* const cua_monitor monitor = cua_vp_monitor_get(board, 1); */

/*
   Change the constant in the following line to match
   the camera and video format desired.
*/
const cua_video_format fmt = cua_xc75_640x480;

cua_channel ac = cua_acquire_channel_alloc(fmt, board);
cua_channel dc = cua_display_channel_alloc(fmt, monitor);
cua_camera_port_set(ac, 1);

cua_live_set(dc, ac);
```

The functions called in this example are discussed in *Cognex Unified Acquisition* on page 105.

Saving a Live Video Session

The 5700/5900 disables live video during system save operations. Thus, when a saved system image is reloaded onto the vision processor, live video is disabled at startup. You can work around this limitation by creating a system saving function that enables live video right after the call to **css_save_system()**.

For example, let's say the currently running system has enabled live video with the commands shown in the *cuadisp.v* example above. In this case, a call to **cua_live_set(dc, ac)** re-enables live video. Create a system saving function like the following:

```
V> int savelive()
>  {
>       css_save_system (css_pci, 0);
>       cua_live_set (dc, ac);
>  }
Function defined using 68 bytes.
V>
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

You can save the current system using either Console Cogtalk's **~x** escape command (page 190) or Windows Cogtalk's **File->Save System** menu option (page 210). When you save the current system, enter the following as your system saving function:

```
savelive();
```

A system loaded with the resulting saved system file boots up with live video enabled.

Running the Search and Train Demo

For the 5700/5900, the instructions in this section replace *Chapter 6, Search and Train Demo*, of the *5000 Getting Started* manual.

Follow these steps to run the Search and Train Demo on a 5700/5900. The demo assumes you're using a Sony XC-75 RS-170 camera. If you're using a CCIR camera, the demo will run as is, or you can edit *cua_cif.v* to designate a different video format on line 65. Mouse button designations are reversed if you've specified a left-handed mouse in the Windows NT control panel.

1. Set up your 5700/5900 according to the instructions in *Chapter 3, Cognex 5700/5900 Hardware Installation*. Connect a video camera and aim it at a well-lit scene of interest.
2. Use Sysgen to build a volume file that contains the Searching option.
3. Start Windows Cogtalk and open a port window on pseudo-serial port 0. Load your volume file into the 5700/5900.
4. If you are using the Cognex Software Display Library, start Cogview and set it to display a video format of 512 x 480 pixels (or a different video format if you've edited *cua_cif.v*). Resize Cogview to display the entire image. Do not enable Cogview's autosize mode.
5. At the V> prompt in the Cogtalk port 0 window, enter the following commands:

   ```
   #include <cse_21.opt>
   #include <cua_srch.v>
   cdemo_init_acquire (cdemo_fmt);
   cdemo_init_display (cdemo_fmt);
   cdemo_init_pd();
   RT
   ```

6. A live video image should now display in Cogview. At the V> prompt, type:

   ```
   T
   ```

   The system responds:

   ```
   Position training window.
   ```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7. A box outline is drawn in black and overlaid on the live image in the Cogview window. If your live image scene is dark, the black box may be difficult to distinguish. In this case, click both PC mouse buttons at the same time to cycle the color of the box between black, white, and cyan colors.

8. Use the PC's mouse to resize and position the box over a target region in your live image. To move the box, hold down the left mouse button while moving the mouse within the Cogview display area. To resize the box, click the right mouse button once to enable resizing mode. In this mode, holding down the left mouse button while moving the mouse moves the bottom and right sides of the box. Click the right mouse button again to return to box-moving mode.

   When you've selected your target area, double-click the left mouse button. The system responds at the V> prompt with a message like the following:

   ```
   window 105 wide by 52 tall at (248, 389); area = 5460 pels
   Trained model size: 10180 bytes.
   ```

   The actual numbers returned will vary according to the size and location of the target region you select.

9. The image in the Cogview window is now the trained model image expanded to fill the display area. Type the following command at the V> prompt to restore the live video image:

   ```
   RT
   ```

10. Draw a search area box over part of the live image, making sure you include your target model in the search area. To draw a search area, type the following command at the V> prompt:

    ```
    BOX
    ```

    A black box outline is drawn and overlaid on the live image. If desired, change the color of the drawing box as described in step 7. Move and resize the search area box as described in step 8.

    When your search area box is ready, double-click the left mouse button. A confirmation message displays at the V> prompt.

11. To search for your trained model within your defined search area, type the following command at the V> prompt:

    ```
    S
    ```

    The system searches the image area and draws a box with cross-hair marks over the area with the best match. A set of search statistics displays at the V> prompt, much like the following:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
Best score: 994 Total time: 15 ms.
Shape: 994 contrast: 1.00 posn: (212,50,208,99)
```

The actual search time and position coordinates will vary according to the complexity of your search target and the size of the search region.

Search and Train Demo with Cognex Trackball

To use the Cognex trackball instead of the PC's mouse, follow these supplementary instructions:

1. Connect the Cognex trackball to the 5700/5900's serial port 2, using the Serial Interface Adapter as described in *Installing the Serial Interface Adapter* on page 33.
2. After step 5 in the previous section, type the following commands at the V> prompt to enable the trackball in place of the PC's mouse:

```
V> fclose (cif_tb_file);
V> cif_tb_file = cif_open_tb ("/dev/ser2");
```

3. Use the following trackball conventions instead of the mouse conventions described in steps 7 and 8 above:
   - Move the trackball button by itself to move the box image.
   - Click the right trackball button to cycle the box outline's color.
   - Click the middle trackball button to toggle between box-moving and box-resizing modes.
   - Click the left trackball button when your box outline is in the desired shape and size.

Image Acquisition

The hardware design of the 5700/5900 dictates the use of a new image acquisition system. The **caq_** family of functions do *not* work on the 5700/5900. You *must* use the Cognex Unified Acquisition (CUA) functions to acquire images on a 5700/5900. The Unified Acquisition system is described in *Cognex Unified Acquisition* on page 105.

The CUA system supports the standard Cognex *cip_buffer* structure, which enables you to run existing Cognex image processing functions on your acquired images. However, because the **caq_** and CUA systems are fundamentally different, a one-to-one mapping of functions is not possible. If you have existing vision processing code you want to port to the 5700/5900, the best approach is to study the *Cognex Unified Acquisition* chapter thoroughly. Then search your code for existing **caq_** system assumptions and replace them with comparable CUA constructs.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The **caq_** acquisition system has a long history on Cognex vision processors, and is the presumed default acquisition method in existing Cognex documentation and sample code. When reading existing documentation and sample code, you must make a mental translation between the presumed **caq_** acquisition system and the CUA system that you will use on the 5700/5900.

Software Display Interface

The 5700/5900 with CVM1 or CVM0 provides a means for directly displaying its output on the host PC's monitor. 5900s with CVM3 provide the same capability, and are also equiped with a hardware monitor port. With CVM1 or CVM0 (and CVM3 if you choose not to use the dedicated monitor port), you must use the Cogview application, or a replacement application of your own, to display images from the vision processor. Cognex supplies the Cognex Software Display (CSD) library of Windows NT functions that allows you to construct a display application.

The CSD library is described in *Cognex Software Display* on page 155. Cogview is described in *Using Cogview* on page 225.

PVE 8.1.1 Software Differences

The revision of the Cognex Programmable Vision Engines (PVE) software that first supported the 5700/5900 is revision 8.1. PVE revision 8.1.1 adds support for 5900s with CVM3. This section summarizes the major differences between PVE 8.1.1 and previous revisions.

- Cognex Unified Acquisition replaces the **caq_** family of image acquisition functions.
- Cognex Unified Acquisition is built in to the baseline, no-options volume file generated with the Sysgen utility.
- Support is included in Cognex Unified Acquisition for the 5700/5900's hardware video trigger feature.
- Most demonstration V files shipped with previous releases were dependent on the **caq_** acquisition system and are no longer instructive demonstrations. Those demonstration files have been removed from PVE 8.1.1.
- The PC Development Environment is not supported on the 5700/5900 and is not shipped with PVE 8.1.1.
- All PC Host components are new, including the device driver, all of the provided applications, and all demonstration code. Console Cogtalk replaces PCVtalk, Windows Cogtalk replaces Wintalk, and Cogview substitutes for Vmix and PCI Vmix.
- In previous releases, Wintalk and PCVtalk supported both serial and block mode file servers. In PVE 8.1.1, Cogtalk supports only the block mode file server, as described in *Cogtalk as File Server* on page 194.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- The behavior of the V compiler was changed from statement-oriented to line-oriented. V now responds to carriage returns without a semicolon, and displays an angle bracket prompt (>) for continuation lines.
- The PVE 8.1.1 installation program automatically calls Vision Processor Setup. Rebooting the host PC is no longer necessary after installation, except when a previous revision's device driver is already loaded and running.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6 5700/5900 User's Guide

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 7

Cognex Unified Acquisition

Cognex Unified Acquisition provides a single application interface for image acquisition and display that works with the Cognex 5700/5900 vision processors. The Unified Acquisition software offers the following features:

- Automatic allocation of frame buffers to store acquired images
- Built-in compatibility with several cameras
- Simple and reliable simultaneous acquisitions
- Support for *video triggers*, hardware-only triggers for acquisition

The Cognex Unified Acquisition functions are not compatible with the older Cognex acquisition software. The **caq_*** interface is not supported in this release.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Overview of Unified Acquisition

The Cognex Unified Acquisition software is a set of functions and data structures that allow you to acquire, display, and store images. Once you have acquired an image, your vision application uses the Cognex image processing functions and vision tools to inspect and analyze it.

Because Unified Acquisition provides an abstract interface to the vision processor, your program does not have to be written for a particular camera or vision processor. Instead, you specify your application's image acquisition requirements in terms of predefined *video formats* that convey such characteristics as camera type and image resolution. The software takes care of setting up and managing the vision processor's resources based on the requested video format.

Unified Acquisition uses two data structures for most of its work: *channels* and *frame buffers*. Channels are data structures that manage acquisition and display. A frame buffer is a region of memory that holds an image or holds graphics that can be overlaid on an image.

When it is acquiring images, the Unified Acquisition software manages the allocation of the frame buffers in which it places the acquired images. You can also allocate frame buffers manually to display images and to overlay graphics on images.

Your vision application follows these steps to acquire an image into a frame buffer through a channel:

1. Allocate an acquire channel to control the acquisition process.
2. Set additional properties of the channel such as the timeout period, and any callback functions you want called during the acquisition process.
3. Either set up a video trigger to enable hardware-triggered acquisition or call a function to begin the acquisition.
4. When the acquisition completes, use the image processing routines or Cognex vision tools on the contents of the frame buffer.
5. Free the frame buffer when you have finished using it.

Channels

The Unified Acquisition software uses data structures called channels to acquire images into a frame buffer and to display images stored in frame buffers. Channels contain the information necessary to configure the vision processor hardware to allow images to be acquired with different video cameras and video formats.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Unified Acquisition uses two kinds of channels: acquisition channels and display channels. To acquire an image, you allocate an acquisition channel and either start the channel's acquisition process or configure the channel to acquire images in response to a video trigger. To display an image, you use a display channel. You cannot use an acquisition channel to display an image or a display channel to acquire an image.

You can allocate as many channels as your application needs (as long as there is sufficient free memory).

Video Formats

Whenever you allocate an acquisition or a display channel, you must specify a video format. For acquisition channels, the video format provides information about the camera and image resolution so that the acquisition software can allocate an appropriate frame buffer. For display channels, the video format gives the display software the information it needs to display the image stored in a frame buffer.

Table 28 lists the video formats.

Camera	CUA video formats	TIming
Sony XC-55	*cua_xc55rr_512x480* *cua_xc55rr_640x480*	RS-170
Sony XC-75	*cua_xc75_640x480* *cua_xc75_512x480* *cua_xc75_320x240* *cua_xc75_256x240*	RS-170
Sony XC-75CE	*cua_xc75ce_760x574* *cua_xc75ce_512x512* *cua_xc75ce_380x287* *cua_xc75ce_256x256*	CCIR
Sony XC-75 Rapid Reset	*cua_xc75rr_320x240* *cua_xc75rr_256x240*	RS-170
Sony XC-75CE Rapid Reset	*cua_xc75cerr_380x287* *cua_xc75cerr_256x256*	CCIR
Pulnix TM-9701	*cua_tm9701_640x480* *cua_tm9701_512x480*	RS-170

Table 28. Video formats

Note: While all cameras should work in this release, only the Sony XC-55 has been tested with CVM3.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Because different cameras behave differently, the video format defines the fundamental behavior of the channel. You cannot change the video format of an existing channel.

Vision Processor Supports One Video Format at a Time

At any given time, the vision processor's hardware resources support just one video format. You cannot use one channel to acquire images with one type of camera at one resolution while using a second channel for a different camera type and/or resolution.

Each time you allocate an acquisition or a display channel, the video format you request for that channel becomes the current video format. Calling **cua_prepare_resources()** for an acquisition or display channel also has the effect of making that channel's video format the current video format.

Note: Whenever you change the current video format by allocating a channel or calling **cua_prepare_resources()**, all existing handles to frame buffers are invalidated and you lose any image and graphics data currently stored in the buffers.

Display Jitter With Half-Resolution Video Formats

Images acquired using the *cua_xc75_320x240* and *cua_xc75_256x240* video formats (XC-75 camera) and the *cua_xc75ce_380x287* and *cua_xc75ce_256x256* formats (XC-75CE camera) may jitter when displayed live in a PC application window that uses Cognex Software Display. (Cogview is an example of such an application.) A one-pixel jitter in the vertical dimension may be detected in the PC window as the odd and even fields of the acquired image are displayed in succession.

Live Display Not Supported for Rapid Reset Video Formats

The video formats for rapid reset cameras (*cua_xc75rr_320x240* and *cua_xc75rr_256x240* for the Sony XC-75 Rapid Reset and *cua_xc75cerr_380x287* and *cua_xc75cerr_256x256* for the Sony XC-75CE Rapid Reset) do not support live display. Calling **cua_live_set()** with these formats results in a blank display. The function does not throw an error.

To simulate live display from a rapid reset camera, execute a loop containing the following steps at an interval that produces the desired frame rate:

1. Acquire an image using **cua_acquire()**.
2. Get a handle to the frame buffer that contains the acquired image using **cua_image_get()**.
3. Display the contents of that frame buffer using **cua_image_plane_set()**.
4. Free the frame buffer using **cua_free()**.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Acquisition Channel Properties

Properties are the configurable aspects of an acquisition such as how long to wait before an acquisition times out or the brightness and contrast settings for acquisitions on a channel.

You can assign different values to an acquisition channel's properties to customize its behavior. Some properties depend on the video format. For example, the exposure property applies only to acquisition channels that control shuttered cameras.

The following sections describe the properties you can change with the Unified Acquisition software.

Exposure Property

The exposure property is available only for acquisition channels whose video format controls a shuttered camera. The exposure property is only relevant with rapid reset cameras. This property specifies the duration of the exposure in microseconds.

Timeout Property

The timeout property specifies the maximum amount of time in milliseconds that an acquisition channel should wait between the time an acquisition request starts and the beginning of the acquisition itself. One reason an acquisition could time out is that the hardware resources needed to acquire the image are in use by another acquisition process.

The default value for this property is 0, which means the acquisition times out if it cannot be started immediately.

Camera Port Property

The camera port property specifies which camera port an acquisition channel should use to acquire an image. The number of camera ports depends on the model of your vision processor. The default value for this property is 1; valid values are 1, 2, 3, and 4.

Input Lookup Table Property

The input lookup table (LUT) property lets you associate a lookup table with an acquisition channel. The values in this 256-element table are applied to the pixel values generated by the analog-to-digital conversion of the camera signal.

On the Cognex 5700/5900, images are displayed in windows on the host PC. Since these are Windows NT windows, the default input lookup table is optimized for the Windows environment. It produces pixel values in the range 10 - 240 to avoid interfering with the static Windows colors.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Video Trigger Property

Acquisitions on specific camera ports can be initiated by a hardware trigger. The hardware trigger goes directly to video hardware to initiate an acquisition, providing a shorter latency than the software-trigger approach where a trigger function that executes in response to a signal on a TTL input line calls **cua_acquire()**.

On vision processors that support video triggers, you have the option of initializing acquisitions by input signals on the trigger line or by the usual method of calling **cua_acquire()**. When the video trigger is used, there is no need to call **cua_acquire()**.

The video trigger property is specified as a Boolean condition, *enabled* or *disabled*.

Four-channel Cognex 5900 vision processors (those that use CVM1) support separate video triggers on each of their four camera ports. In this case, there is a hardware association between each camera port and each video trigger (and video strobe) connection. Camera port 1 can be triggered by video trigger 1, which in turn can fire video strobe 1. Camera port 2 is associated with video trigger 2 and video strobe 2, and so on. Thus, a signal on an enabled video trigger line initiates an image acquisition on the associated camera port.

Single-channel Cognex 5700 and 5900 vision processors (those that use CVM0) support up to four cameras, but support only one acquisition at a time. The hardware supports a single video trigger line (and video strobe line), but it is not associated with a particular camera port. On these models, a signal on an enabled video trigger line initiates an image acquisition on the camera port currently associated with the current acquisition channel. Thus, on these models, a single video trigger can initiate successive image acquisitions on one of up to four cameras.

Two-channel 5900s (those that use CVM3) support up to four cameras, but two acquisitions at a time. A pulse on trigger line 1 causes an image acquistion on camera ports 1 and 3, while a pulse on trigger line 2 affects cameras 2 and 4.

For information about setting up the parallel I/O lines used as video triggers and strobe lines on the Cognex 5700/5900, refer to *Hardware Video Triggers* on page 64. For the correspondence between camera port numbers and trigger/strobe lines, see the *CVM Camera Numbering* section of each CVM's description in *CVM Descriptions* beginning on page 74.

Video Strobe Properties

Channels that can use video triggers also can be configured to fire a strobe to illuminate the video-triggered acquisitions. The firing of the strobe is handled by the video hardware; no software is involved.

The video strobe property is specified as a Boolean condition, *enabled* or *disabled*.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

On four-channel Cognex 5900 vision processors with CVM1, the video strobe lines are associated with specific video trigger lines and camera ports. Acquisitions on camera port 1 use video trigger 1 and the strobe 1 lines. Camera port 2 uses video trigger 2 and the strobe 2 lines, and so on.

Single-channel Cognex 5700 and 5900 vision processors using CVM0 have a single set of video strobe lines, which are associated with the video trigger line, but are *not* associated with a particular camera port. On these models, a signal on an enabled video trigger line fires the strobe connected to the strobe lines, and an image is acquired on the camera port currently associated with the current acquisition channel.

Two channel 5900s using CVM3 have two video strobe lines, each associated with two camera ports. Strobe line 1 is associated with camera ports 1 and 3 while strobe line 2 is associated with camera ports 2 and 4.

Video strobes have several configurable properties, including the width of the pulse required to fire the strobe and the polarity and duration of the strobe. The default width of the pulse to the strobe is 15 µs. This value is used internally to calculate the timing of the pulse to the strobe. The default signal polarity is high-to-low and the default duration of the strobe is 10 µs. The duration value is used internally. The duration value is a property of the strobe itself that can be found in the strobe's operating manual.

Charge Transfer Interval Properties

The Unified Acquisition video formats include appropriate charge transfer interval settings for the supported cameras. The charge transfer interval is the period of time during which the charge representing the intensity values in the scene is transferred from the camera's individual sensor elements to an output register in the camera. Firing the strobe while the charge transfer occurs produces unpredictable results.

Separate charge transfer intervals are defined for the odd and even fields of the image. The channel's start-even and end-even properties specify the starting and ending line numbers of the charge transfer interval for the even field. The start-odd and end-odd properties provide the line numbers for the odd field. (Only the start-odd and end-odd properties are used with noninterlaced cameras.)

If you find that your strobe is not firing at the proper time, you will need to change the default settings to enlarge the charge transfer interval or to shift it forward or back. Note that the vision processor's settings for the charge transfer interval are not the same as the ones provided in a camera's specifications. You should use the property-get functions to determine the vision processor's settings, then use the property-set functions to adjust the interval. The charge transfer interval properties are implemented as doubles so that you have the option of specifying fractional line numbers.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Brightness and Contrast Properties

The brightness and contrast properties let you set the brightness and contrast of the image that is being acquired or displayed by the channel. The properties alter image brightness and contrast by influencing the analog-to-digital conversion of the signal from the camera to the vision processor. The values for both properties range from 0.0 to 1.0. The brightness property ranges from dark (0.0) to light (1.0). The contrast property ranges from no additional contrast (0.0) to maximum contrast (1.0).

The default value for brightness is 1.0. The default value for contrast is 0.0.

Simultaneous Acquisitions

Acquisition channels can be chained together in a master/slave relationship. Initiating an acquisition on a master channel automatically initiates an acquisition on its slave channel. An acquisition can be master to one channel and slave to another.

Simultaneous acquisition is only supported on a Cognex 5900 with four video channels (CVM1) or two video channels (CVM3). Note that a *video channel* is not the same as an *acquisition channel*. The term video channel refers to video hardware on a vision processor (video-timing generator, A/D converter, and so on). The number of acquisitions that a vision processor can perform simultaneously is equal to the number of its video channels. Single-channel vision processors can perform only one acquisition at a time.

Frame Buffers

A frame buffer is a region of memory that holds an image or graphics that can be overlaid on an image. You can acquire images into a frame buffer, display an image stored in a frame buffer, or superimpose an image in a frame buffer over a live or frozen image.

Allocating and Freeing Frame Buffers

The Unified Acquisition software automatically allocates the frame buffers in which it stores acquired images. The default limit on the number of acquisition buffers depends on the vision processor's video configuration. The default limit for vision processors with a single video channel is four. The default limit for vision processors with multiple video channels is ten. You can specify a different number using **cua_acquisition_buffers_set()** if your application requires more.

In addition to the automatically allocated acquisition buffers, your vision application can allocate one displayable and one overlayable frame buffer.

Your application is responsible for returning all frame buffers that are no longer in use to the memory pool.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- For the overlayable and displayable buffers that your application allocates explicitly, you pass the buffer handle returned by the allocation function to **cua_free()**.
- For a frame buffer that holds an image acquired using **cua_acquire()**, you obtain a handle to the automatically allocated buffer using **cua_image_get()**, and pass that handle to **cua_free()**.
- For the image buffer that is automatically allocated to perform a live display, calling **cua_live_set()** with the *cua_no_object* argument both terminates the live display and frees the frame buffer being used.

Note: You should free all frame buffers before you change the video format by allocating a new channel or calling **cua_prepare_resources()**.

The Unified Acquisition software allocates all frame buffers in a contiguous block of memory that it allocates from main memory on the 5700/5900. The size of this video memory block is determined by a combination of the currently specified video format and the currently specified limit on the number of acquisition buffers. At startup, the memory block is allocated assuming a 512x480 video format. With each subsequent change to the video format (caused by the allocation of a new acquisition or display channel or the use of **cua_prepare_resources()** for a channel with a video format different from the current format), the existing video memory is freed and a new video memory block is allocated.

A situation can occur where the Unified Acquisition software is unable to allocate a block of memory large enough to accommodate a change to the video format or the number of acquisition buffers. When this occurs, the channel-allocation or **cua_prepare_resources()** function throws a *CUA_ERR_VIDMEM_ALLOC* error.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Unified Acquisition Software

This section describes the Cognex Unified Acquisition functions and data types.

Object Handles

Except for the functions that return scalar values (integers, Booleans, or floating point numbers), all of the **cua_*** functions return *object handles*. Object handles are references to objects in memory that only the Unified Acquisition software can manipulate. They are very much like pointers, but the only way for you to work with their contents is to use the Unified Acquisition functions described in this section.

For example, the function **cua_properties_get()** returns an object handle to a data structure that describes the properties of a channel. The only way to get the value of a specific property is to use a function that operates on properties such as **cua_timeout_get()**.

The Unified Acquisition interface includes the following handle data types:

- **cua_channel**, a handle to an acquisition or display channel
- **cua_monitor**, a handle to the display device to which a display channel sends images
- **cua_board**, a handle to the vision processor
- **cua_frame_buffer**, a handle to a frame buffer
- **cua_properties**, a handle to the data structure that stores an acquisition channel's properties

The special value *cua_no_object* denotes a null object handle. Many of the functions that allocate data, such as frame buffers, return *cua_no_object* when the allocation fails. Other functions take *cua_no_object* as an argument to sever a relationship between two objects.

Memory Management with Handles

While some of Unified Acquisition functions return handles to existing data structures, other functions return handles to newly allocated instances of data structures. In those cases, it is up to your application to call **cua_free()** to reclaim the memory used by the data structure when it is no longer in use.

For example, **cua_properties_get()** returns a handle to the actual data structure that belongs to a channel, so you should never pass that object handle to **cua_free()** to release the memory it uses. **cua_overlayable_alloc()**, on the other hand, returns a newly allocated frame buffer. It is your application's responsibility to call **cua_free()** to reclaim the memory it uses.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Constants for Graphics Colors

Unified Acquisition defines a set of color constants that are used to specify the color of graphics created in overlay buffers. The functions that you use to create graphics and text in an overlay buffer's **cip_buffer** are described in the *Image Processing* manual.

Enumeration cua_graphics_colors

cua_graphics_colors defines the color constants used to specify the color of graphics created in a **cip_buffer**.

```
#if C_BOARD==C_VM21
#include <pixel.h>

enum
{
    cua_pass    = 240,
    cua_black   = 0,
    cua_white   = 255,
    cua_blink   = 241,
    cua_blue    = 252,
    cua_brown   = 242,
    cua_cyan    = 254,
    cua_green   = 250,
    cua_grey    = 247,
    cua_ovl_black = 248,
    cua_magenta = 253,
    cua_maroon  = 1,
    cua_orange  = 245,
    cua_pink    = 243,
    cua_purple  = 244,
    cua_red     = 249,
    cua_yellow  = 251
} cua_graphics_colors;
```

The values assigned to these color constants are specific to the Cognex 5700/5900. Because images and graphics from the Cognex 5700/5900 are usually displayed in Windows NT windows, the values of the color constants correspond whenever possible to the static Windows colors. The Cogview display application relies on these values to display the colors in overlay graphics. If you are creating your own PC display application, refer to *Defining an 8-Bit Palette for Your Software Display Application* on page 165 for additional information.

Use *cua_ovl_black* when defining the color black in an overlay graphics palette for use on an external monitor with a 5900 and CVM3.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Note: The value of these color constants may change in future versions of the software. Use the color constants in your vision application. Do not use their defined values.

Channel Functions

A channel is a data structure that contains all the information needed to interact with a vision processor to acquire or display images.

Function cua_acquire_channel_alloc()

cua_acquire_channel_alloc() allocates and returns a handle to an acquisition channel. If **cua_acquire_channel_alloc()** cannot allocate an acquisition channel, it throws a standard memory allocation error.

Note: Always allocate an acquire channel with **cua_acquire_channel_alloc()** before you allocate a display channel with **cua_display_channel_alloc()**.

Note: Whenever you change the current video format by allocating a channel or calling **cua_prepare_resources()**, you lose any image and graphics data currently stored in frame buffers.

```
#include <cua/channel.h>

cua_channel cua_acquire_channel_alloc(cua_video_format format,
    cua_board board);
```

format is the video format that the channel uses. The video format specifies the kind of camera you are using and the image resolution. It also establishes default values for the configurable properties of the acquisition channel. When you start an acquisition, the channel uses the format information to allocate an appropriate frame buffer. Table 28 on page 107 lists the valid values for *format*.

board must be the value returned by **cua_default_board_get()**.

Allocating an acquisition channel does not allocate a frame buffer. Frame buffers for acquisition channels are allocated automatically when an acquisition begins (see **cua_acquire()**).

cua_acquire_channel_alloc() throws the following errors:

- *CGEN_ERR_BADARG* if *board* is not **cua_default_board_get()**
- *CUA_ERR_TYPE_MISMATCH* if *format* is not a valid video format
- *CUA_ERR_VIDMEM_ALLOC* if *format* is different from the current video format and it is not possible to allocate a block of video memory large enough for the new format

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Function cua_display_channel_alloc()

cua_display_channel_alloc() allocates and returns a handle to a display channel. If **cua_display_channel_alloc()** cannot allocate a display channel, it throws a standard memory allocation error.

Note: Always allocate an acquire channel with **cua_acquire_channel_alloc()** before you allocate a display channel with **cua_display_channel_alloc()**.

Note: Whenever you change the current video format by allocating a channel or calling **cua_prepare_resources()**, you lose any image and graphics data currently stored in frame buffers.

```
#include <cua/channel.h>

cua_channel cua_display_channel_alloc(cua_video_format format,
    cua_monitor display);
```

format is the video format to use to display the image. See Table 28 on page 107 for a list of valid video formats.

display must be a value returned by **cua_default_monitor_get()**, **cua_vp_monitor_get()**, or **cua_host_monitor_get()**.

Allocating a display channel does not allocate a frame buffer. To allocate a frame buffer whose contents you will display through a display channel, use the functions **cua_image_plane_set()** or **cua_overlay_plane_set()**. To display a live image through a display channel, use **cua_live_set()**.

cua_display_channel_alloc() throws the following errors:

- *CGEN_ERR_BADARG* if *display* is not a valid display
- *CUA_ERR_TYPE_MISMATCH* if *format* is not a valid video format
- *CUA_ERR_VIDMEM_ALLOC* if *format* is different from the current video format and it is not possible to allocate a block of video memory large enough for the new format

Function cua_default_monitor_get()

cua_default_monitor_get() returns a handle to the default monitor of the default vision processor.

```
#include <cua/channel.h>

cua_monitor cua_default_monitor_get(void);
```

Use this function to obtain a **cua_monitor** handle and pass it to any functions that require a **cua_monitor** argument. Only one monitor at a time can be used.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This function transparently allocates a monitor handle independent of the vision processor in use. On a 5900 with CVM3, this function is equivalent to calling:

```
cua_vp_monitor_get(cua_default_board_get(),1).
```

When using a 5700/5900 with CVM0 or a 5900 with CVM1, this function is equivalent to calling:

```
cua_host_monitor_get(cua_default_board_get(),1)
```

cua_default_monitor_get() throws the following error:

- *CUA_ERR_NO_MONITOR* if the default vision processor has no monitors

Function cua_vp_monitor_get()

```
#include <cua/channel.h>
```

cua_vp_monitor_get() returns a handle to the vision processor's physical monitor when using a 5900 with CVM3.

Only one physical monitor is supported. Use this function to obtain a **cua_monitor** handle for CVM3's external monitor and pass it to any functions that require a **cua_monitor** argument.

```
#include <cua/channel.h>

cua_monitor cua_vp_monitor_get(cua_board board,
   c_Int32 monitor_number);
```

board: a handle to the display device to which a display channel sends images.

monitor_number: the monitor number. Set this value to 1.

cua_vp_monitor_get() throws the following error:

- *CUA_ERR_NO_MONITOR* if the default vision processor has no monitors

Function cua_host_monitor_get()

```
#include <cua/channel.h>
```

cua_host_monitor_get() returns a handle to one of the vision processor's host monitors. A host monitor is typically a window of an application running in the host environment.

Use this function to obtain a **cua_monitor** handle for a software display monitor, and pass it to any functions that require a **cua_monitor** argument. This function can be used with any CVM, but is typically used with CVM0 or CVM1.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Function cua_default_board_get()

cua_default_board_get() returns a handle to the vision processor where the Unified Acquisition software is executing.

Use this function to obtain a **cua_board** handle and pass it to any functions that take a **cua_board** argument.

```
#include <cua/channel.h>

cua_board cua_default_board_get(void);
```

Function cua_prepare_resources()

cua_prepare_resources() prepares the hardware resources of the vision processor, such as the video format and synchronization, according to a channel's properties. Channel operations, such as acquisition or display, cannot begin or continue when the hardware resources are not prepared.

Note: Whenever you change the current video format by allocating a channel or calling **cua_prepare_resources()**, you lose any image and graphics data currently stored in frame buffers.

```
#include <cua/channel.h>

void cua_prepare_resources(cua_channel channel);
```

channel is the acquisition channel whose properties are used to prepare the hardware resources.

You do not need to call this function when you allocate an acquisition or display channel because those functions call **cua_prepare_resources()** for you.

You may need to call this function when you modify channel properties or when you switch between two channels that require the hardware to be set up differently. For example, if you are acquiring images using two different video formats, you would create a channel for each format. Since each channel requires the vision processor hardware to be set up differently, you must call **cua_prepare_resources()** when you switch between channels.

If the hardware resources are already prepared to support this channel, this function has no effect.

Note that if you use **cua_prepare_resources** to switch from an RS-170 video format to a CCIR format, or vice versa, it may take as long as a second after the function returns for hardware resources to stabilize. Images acquired during this period may contain noise.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

cua_prepare_resources() throws the following errors:

- *CUA_ERR_TYPE_MISMATCH* if *channel* is not a channel
- *CUA_ERR_VIDMEM_ALLOC* if the video format of *channel* is different from the current video format and it is not possible to allocate a block of video memory large enough for the new format

Function cua_are_resources_prepared()

cua_are_resources_prepared() returns *TRUE* if a vision processor's hardware resources are prepared to support a given acquisition channel.

```
#include <cua/channel.h>

c_Bool cua_are_resources_prepared(cua_channel channel);
```

channel is an acquisition channel whose properties are being checked against the current preparation of the vision processor's hardware resources.

You can use this function to see if the resource preparation you requested with **cua_prepare_resources()** has completed.

cua_are_resources_prepared() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *channel* is not a channel

Frame Buffer Functions

A frame buffer is an area of memory that holds an image or holds graphics that can be overlaid on an image when the image is displayed.

When you acquire an image, the Unified Acquisition software automatically allocates a frame buffer from its free frame buffer pool. To display an image, you can use a frame buffer that holds an acquired image, or you can allocate displayable and overlayable frame buffers yourself.

Since the number of available frame buffers is limited, it is important that your application release the memory used by the frame buffer. To release a frame buffer, use **cua_free()**.

The Cognex 5700/5900 allocates frame buffers in main memory on the 5700/5900; the number of frame buffers available for acquired images is configurable.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Function cua_displayable_alloc()

cua_displayable_alloc() allocates and returns a handle to a frame buffer suitable for displaying an image on the specified display channel. If this function cannot allocate a frame buffer, it returns *cua_no_object*. A vision application can allocate one displayable frame buffer.

```
#include <cua/channel.h>

cua_frame_buffer cua_displayable_alloc(cua_channel display);
```

display is the display channel that will be used to display the frame buffer.

You are responsible for releasing the frame buffer when you no longer need it. To deallocate the memory, use **cua_free()**. If you free a frame buffer that is being displayed, the Unified Acquisition software defers releasing the frame buffer until it is no longer displayed.

cua_displayable_alloc() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *display* is not a display channel

Function cua_overlayable_alloc()

cua_overlayable_alloc() allocates and returns a handle to a frame buffer suitable for displaying overlay graphics on the specified display channel. If this function cannot allocate a frame buffer, it returns *cua_no_object*. A vision application can allocate one overlayable frame buffer.

```
#include <cua/channel.h>

cua_frame_buffer cua_overlayable_alloc(cua_channel display);
```

display is the display channel that will be used to display the overlay graphics in the frame buffer.

You are responsible for releasing the frame buffer when you no longer need it. To deallocate the memory, use **cua_free()**. If you free a frame buffer that is being displayed, the Unified Acquisition software defers releasing the frame buffer until it is no longer displayed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

cua_overlayable_alloc() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *display* is not a display channel

Function cua_acquisition_buffers_get()

cua_acquisition_buffers_get() returns the maximum number of frame buffers that the Unified Acquisition software will allocate to store acquired images.

```
#include <cua/vidmem.h>

c_Int32 cua_acquisition_buffers_get();
```

The number returned is either the default value of ten or the last value specified using **cua_acquisition_buffers_set()**.

Function cua_acquisition_buffers_set()

cua_acquisition_buffers_set() specifies the number of frame buffers to be used to store acquired images. This function is only available on systems like the Cognex 5700/5900 that allocate frame buffers in main memory.

Note: **cua_acquisition_buffers_set()** invalidates the handles to your currently allocated frame buffers. You should not call this function if there is data in buffers that you wish to retain.

```
#include <cua/vidmem.h>

void cua_acquisition_buffers_set(c_Int32 numBuffers);
```

numBuffers is the new number of frame buffers to be used for acquired images.

The change in the number of available buffers is made the next time **cua_prepare_resources()** is called. (Note that **cua_acquire_channel_alloc()** and **cua_display_channel_alloc()** call **cua_prepare_resources()** internally.)

cua_acquisition_buffers_set() throws the following error:

- *CUA_ERR_VIDMEM_ALLOC* if it is not possible to allocate a block of video memory large enough for the requested number of buffers, given the current video format

Function cua_image_plane_set()

cua_image_plane_set() displays a frame buffer through a display channel.

Note: It is recommended that you call **cua_prepare_resources()** before displaying a frame buffer after a change in video format.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
#include <cua/channel.h>

void cua_image_plane_set(cua_channel display,
       cua_frame_buffer buffer);
```

display is a display channel.

buffer is a displayable frame buffer that you wish to display through the *display* channel. If *buffer* is *cua_no_object*, no image (all black) is displayed.

This function has the side effect of calling

```
cua_live_set(display, cua_no_object);
```

to disable any current live display.

If a channel is displaying the contents of a frame buffer, and you try to deallocate the frame buffer with the function **cua_free()**, this function waits until the frame buffer is no longer displayed to return the frame buffer to free memory. For example:

```
cua_frame_buffer fb;

fb = cua_image_get(acqChan); // get an acquired image
cua_image_plane_set(dispChan, fb); // display it

cua_free(fb); // the frame buffer is not released...
cua_image_plane_set(dispChan, cua_no_object); //...until now
```

cua_image_plane_set() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *display* is not a display channel or if *buffer* is not a frame buffer or *cua_no_object*

Function cua_overlay_plane_set()

cua_overlay_plane_set() overlays the contents of a frame buffer on the frozen or live image being displayed on a display channel.

Note: Overlaying a frame buffer on a display channel that is performing a live display will freeze the live display. Call **cua_overlay_plane_set()** before calling **cua_live_set()** to avoid this problem. Also, it is recommended that you call **cua_prepare_resources()** before overlaying a frame buffer after a change in video format.

```
#include <cua/channel.h>

void cua_overlay_plane_set(cua_channel display,
       cua_frame_buffer buffer);
```

display is a display channel.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

buffer is an overlayable frame buffer that you wish to overlay on the image displayed through the *display* channel. If buffer is *cua_no_object*, any existing overlay image is removed.

If a channel is displaying the contents of a frame buffer, and you try to deallocate the frame buffer with the function **cua_free()**, this function waits until the frame buffer is no longer displayed to return the frame buffer to free memory. See the example on page 123.

cua_overlay_plane_set() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *display* is not a display channel or if *buffer* is not a frame buffer or *cua_no_object*

Function cua_live_set()

cua_live_set() activates a live display on a display channel from an acquisition channel.

Note: It is recommended that you call **cua_prepare_resources()** before activating a live display for the first time after a change in video format.

```
#include <cua/channel.h>

void cua_live_set(cua_channel display,
       cua_channel acquire);
```

display is a display channel.

acquire is an acquisition channel. If *acquire* is *cua_no_object*, the current live display (if any) is terminated and the frame buffer used for the live display is freed.

If you were previously using the display channel to display a live image from a different acquisition channel, **cua_live_set()** switches to the new acquisition channel.

If you were previously displaying a frozen image (the contents of a displayable frame buffer) through the display channel, **cua_live_set()** switches to the live image. Remember that it is your responsibility to deallocate a frame buffer that is no longer in use.

cua_live_set() throws the following errors:

- *CUA_ERR_TYPE_MISMATCH* if *display* is not a display channel or if *acquire* is not an acquisition channel or *cua_no_object*
- *CUA_ERR_UNPREPARED* if the channel's resources are not prepared

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Function cua_cip_buffer_get()

cua_cip_buffer_get() returns a pointer to a **cip_buffer**. You can then use the Cognex image processing and vision tools to manipulate and to analyze the image or to create graphics in an overlay buffer. The *cip_buffer* that this function returns belongs to the frame buffer and is valid only until the frame buffer is freed.

```
#include <cua/channel.h>

cip_buffer *cua_cip_buffer_get(cua_frame_buffer buffer);
```

buffer is a frame buffer previously allocated.

Use the function **cua_image_get()** to get a frame buffer associated with an acquisition channel.

cua_cip_buffer_get() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *buffer* is not a frame buffer

Acquisition Functions

To acquire an image, you start an acquisition through one of the channels you allocated. Each channel maintains a queue of outstanding acquisition requests. That means that you can issue as many acquisition requests as you like at the same time. The Unified Acquisition software takes each request in order.

In some vision applications, the acquisition request is part of the program's normal flow. In most typical applications, however, the acquisition request is triggered asynchronously by an external event, such as an object entering the inspection station.

Function cua_acquire()

cua_acquire() begins an acquisition through the specified channel.

Note: It is recommended that you call **cua_prepare_resources()** before acquiring an image for the first time after a change in video format.

```
#include <cua/channel.h>

void cua_acquire(cua_channel channel);
```

channel is the acquisition channel you use to acquire an image.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

cua_acquire() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *channel* is not an acquisition channel

Function cua_cancel()

cua_cancel() stops and discards the oldest acquisition on the specified channel. This function has no effect if the channel is idle. (See **cua_state_get()**.)

```
#include <cua/channel.h>

void cua_cancel(cua_channel channel);
```

channel is the channel on which you are canceling the acquisition.

cua_cancel() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *channel* is not an acquisition channel

Function cua_image_get()

cua_image_get() waits for the oldest acquisition request to complete on the specified channel and returns a frame buffer that contains the acquired image. The acquisition request is removed from the channel's acquisition queue.

```
#include <cua/channel.h>

cua_frame_buffer cua_image_get(cua_channel channel);
```

channel is the channel you are using to acquire an image.

To use image-processing functions to analyze the image, use **cua_cip_buffer_get()** (see page 125) to obtain the **cip_buffer** that contains the image.

Use **cua_image_plane_set()** to display the contents of the frame buffer (see page 122). When the frame buffer is no longer in use, you must call **cua_free()** to return the frame buffer to the frame buffer pool.

cua_image_get() throws the following errors:

- *CUA_ERR_TYPE_MISMATCH* if *channel* is not an acquisition channel
- *CUA_ERR_FAILED_ACQUISITION* if the acquisition failed

Function cua_state_get()

cua_state_get() returns the state of a given acquisition channel.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
#include <cua/channel.h>

cua_state cua_state_get(cua_channel channel);
```

channel is an acquisition channel.

Enumeration cua_state

cua_state_get() returns one of the following:

```
typedef enum cua_state
{
    cua_idle,
    cua_waiting,
    cua_running,
    cua_completed
} cua_state;
```

cua_idle means that the channel has no outstanding acquisition requests.

cua_waiting means that the channel is currently unable to allocate the resources it needs for the oldest acquisition in its queue. Another acquisition request may be using the resources, or the resources may not be in the proper state. (See **cua_prepare_resources()**). The acquisition request will leave the waiting state and enter the running state when the resources become available and are prepared. If the waiting time exceeds the timeout limit specified in the timeout property, the acquisition request enters the completed state with the *cua_timed_out* completion state. (See **cua_completion_get()**).

cua_running means that the channel is currently acquiring an image.

cua_completed means that the acquisition has finished. Use **cua_completion_get()** to check whether it was successful or whether the acquisition request timed out.

cua_state_get() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *channel* is not an acquisition channel

Function cua_completion_get()

cua_completion_get() returns the completion status of the oldest acquisition on the specified channel. If there are no acquisition requests in the queue, this function waits for an acquisition to start.

```
#include <cua/channel.h>

cua_completion cua_completion_get(cua_channel channel);
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

channel is the channel you are using to acquire an image.

Enumeration cua_completion

cua_completion_get() returns one of the following values:

```
typedef enum cua_completion
{
    cua_succeeded,
    cua_timed_out
} cua_completion;
```

cua_succeeded means that the acquisition completed successfully.

cua_timed_out means that the acquisition did not complete successfully because the amount of time specified in the timeout limit elapsed before the acquisition could begin.

For vision applications that demand the highest possible throughput, you may wish to use the callback functions to determine when an acquisition request has completed. See the section *Callback Functions* on page 145.

cua_complete() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *channel* is not an acquisition channel

Function cua_master_set()

cua_master_set() establishes a master/slave relationship between two acquisition channels. Starting an acquisition on the master channel triggers a simultaneous acquisition on the slave channel.

```
#include <cua/channel.h>

void cua_master_set(cua_channel slave, cua_channel master);
```

slave is the secondary channel. Starting an acquisition with the *master* channel triggers a simultaneous acquisition on this channel.

master is the primary channel. Starting an acquisition with this channel triggers a simultaneous acquisition on the *slave* channel. If *master* is *cua_no_object*, any existing relationship between *slave* and another channel is broken.

Each channel can have only one master and one slave. A channel can be the master of one channel and the slave of another channel. For example, on a 5900 with the four-channel video configuration, four acquisition channels *A*, *B*, *C*, and *D* (one for each camera port) could be chained together for simultaneous acquisition in the following way

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
cua_master_set(B, A);
cua_master_set(C, B);
cua_master_set(D, C);
```

This code produces that following relationships.

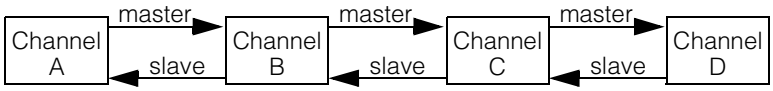


Figure 37. Acquisition channels chained for simultaneous acquisition

To acquire images on all four channels simultaneously, you would call:

```
cua_acquire(A);
```

cua_master_set() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *slave* is not an acquisition channel, or if *master* is neither an acquisition channel nor *cua_no_object*

Function cua_master_get()

cua_master_get() returns the specified channel's master channel. If the channel has no master, this function returns *cua_no_object*.

```
#include <cua/channel.h>

cua_channel cua_master_get(cua_channel channel);
```

channel is the acquisition channel whose master channel you want to get.

cua_master_get() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *channel* is not acquisition channel

Property Functions

Every property has at least three functions that operate on the property. At minimum, there are functions that let you test for the existence of the property, get the value of the property, and set the value of a property.

This section describes functions that operate on the following properties:

- Exposure
- Timeout

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- Camera port
- Input LUT
- Brightness and contrast
- Video trigger
- Video strobe
- Charge transfer interval

For descriptions of how these properties are used to control acquisitions, refer to *Acquisition Channel Properties* on page 109.

All of the functions that take a properties data structure (**cua_properties**) as an argument can accept a channel (**cua_channel**) argument as well. When you pass these functions a channel as an argument, they implicitly call **cua_properties_get()** for you.

Function cua_properties_get()

cua_properties_get() returns a handle to the properties associated with a specified channel.

```
#include <cua/prop.h>

cua_properties cua_properties_get(cua_channel channel);
```

channel is the channel whose properties this function returns.

This function returns a handle to the properties owned by the channel. If *channel* is an acquisition channel, changes that you make to the properties take effect with the next acquisition request. If *channel* is a display channel, changes take effect immediately if the display is active.

For instance, to change the exposure time for a channel, use code like this:

```
myProps = cua_properties_get(myChannel);
cua_timeout_set(myProps, 100);
```

Do not pass the properties returned by this function to **cua_free()**. If you need to make a copy of a set of properties, you can pass the value returned by this function to **cua_copy()**.

cua_properties_get() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *channel* is not a channel

Function cua_properties_set()

cua_properties_set() sets all of a specified channel's properties at once.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
#include <cua/props.h>

void cua_properties_set(cua_channel channel, cua_properties prop);
```

channel is the channel whose properties you want to set.

prop is the collection of properties to set for the channel.

The properties in *prop* are copied to the properties structure that belongs to the channel. Existing references to the channel's properties remain valid, but the values of the individual properties may be changed. For example:

```
p1 = cua_properties_get(chan1);
p2 = p1;

cua_timeout_set(p1, 100);
p3 = cua_copy(p1);
cua_timeout_set(p3, 200);

   // value in p3 copied to chan1's properties
cua_properties_set(chan1, p3);

cua_free(p3); // p3 no longer valid

   // p1 and p2 still refer to chan1's properties
val = cua_timeout_get(p1); // value is 200
val = cua_timeout_get(p2); // value is 200
```

The properties in *prop* must have come from a channel whose video format is the same as *channel*.

cua_properties_set() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_exposure_has()

cua_exposure_has() returns *TRUE* if a channel or a set of properties has an exposure property. The exposure property is only relevant with rapid reset cameras.

```
#include <cua/prop.h>

c_Bool cua_exposure_has(cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel whose exposure property you are interested in.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

cua_exposure_has() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel

Function cua_exposure_get()

cua_exposure_get() returns the exposure duration in microseconds from a channel or from a set of properties.

```
#include <cua/prop.h>

cua_exposure cua_exposure_get(cua_properties prop_or_channel);
```

prop_or_channel is a set of properties that contains the exposure property.

cua_exposure_get() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel, or if the specified channel does not support the exposure property

Function cua_exposure_set()

cua_exposure_set() sets the channel's exposure in microseconds.

```
#include <cua/prop.h>

void cua_exposure_set(cua_properties prop_or_channel, c_Int32
exposure);
```

prop_or_channel is the set of properties or the channel whose exposure value you want to change.

exposure is the new value of the exposure in microseconds.

cua_exposure_set() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_timeout_has()

cua_timeout_has() returns *TRUE* if a channel or a set of properties has a timeout property.

```
#include <cua/prop.h>

c_Bool cua_exposure_has(cua_properties prop_or_channel);
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

prop_or_channel is the set of properties or the channel that you want to test for the presence of the timeout property.

cua_timeout_has() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel

Function cua_timeout_get()

cua_timeout_get() returns acquisition timeout period in milliseconds.

```
#include <cua/cua.h>

c_Int32 cua_timeout_get(cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel whose timeout period you want to get.

cua_timeout_get() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_timeout_set()

cua_timeout_set() lets you set a new value for the timeout period.

```
#include <cua/cua.h>

void cua_timeout_set(cua_properties prop_or_channel, c_Int32
timeout);
```

prop_or_channel is the set of properties or the channel whose timeout value you want to change.

timeout is the new timeout period in milliseconds. A value of 0 means that the acquisition times out if it cannot begin immediately. A negative value means that the acquisition will never time out.

cua_timeout_set() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_camera_port_has()

cua_camera_port_has() returns *TRUE* if a channel or a set of properties has a camera port property.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
#include <cua/prop.h>

c_Bool cua_camera_port_has(cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel that you want to test for the presence of the camera port property.

cua_camera_port_has() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel

Function cua_camera_port_get()

cua_camera_port_get() returns the camera port associated with a set of properties or a channel.

```
#include <cua/prop.h>

c_Int32 cua_camera_port_get(cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel whose camera port setting you want to get.

cua_camera_port_get() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_camera_port_set()

cua_camera_port_set() lets you select a new camera port.

```
#include <cua/prop.h>

void cua_camera_port_set(cua_properties prop_or_channel,
    c_Int32 port);
```

prop_or_channel is the set of properties or the channel whose camera port value you want to change.

port is the new camera port to associate with the properties or the channel. This will be a value between 1 and the number of cameras ports available on your vision processor (typically, 2 or 4).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

cua_camera_port_set() throws the following errors:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property
- *CGEN_ERR_BADARG* if the port number is less than 1 or greater than the number of camera ports your vision processor supports

Function cua_input_lut_256_has()

cua_input_lut_256_has() returns *TRUE* if the channel supports a 256-element input lookup table.

```
#include <cua/prop.h>

c_Bool cua_input_lut_256_has(cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel that you want to test for the presence of the input lookup table property.

cua_input_lut_256_has() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel

Function cua_input_lut_256_get()

cua_input_lut_256_get() returns the digitizer's 256-element lookup table.

```
#include <cua/prop.h>

const unsigned char *cua_input_lut_256_get(
    cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel whose input lookup table you want to get.

The value that this function returns is a 256-element array of bytes. The typical values for the input lookup table range from 0 to 255.

cua_input_lut_256_get() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_input_lut_256_set()

cua_input_lut_256_set() lets you set a new input lookup table.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
#include <cua/prop.h>

void cua_input_lut_256_set(cua_properties prop_or_channel,
    const unsigned char *lut);
```

prop_or_channel is the set of properties or the channel whose input LUT you want to change.

lut is a 256-element array of bytes.

cua_input_lut_256_set() throws the following errors:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property
- *CGEN_ERR_BADARG* if the port number is less than 1 or greater than the number of camera ports your vision processor supports

Function cua_brightness_contrast_has()

cua_brightness_contrast_has() returns *TRUE* if a channel or a set of properties has brightness and contrast properties.

```
#include <cua/prop.h>

c_Bool cua_brightness_contrast_has(cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel that you want to test for the presence of the brightness and contrast properties.

cua_brightness_contrast_has() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel

Function cua_brightness_get()

cua_brightness_get() returns the brightness level in the range 0.0 to 1.0.

```
#include <cua/prop.h>

double cua_brightness_get(cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel whose brightness level you want to get.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

cua_brightness_get() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_contrast_get()

cua_contrast_get() returns the contrast level in the range 0.0 to 1.0.

```
#include <cua/prop.h>

double cua_contrast_get(cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel whose contrast level you want to get.

cua_contrast_get() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_brightness_contrast_set()

cua_brightness_contrast_set() lets you set new brightness and contrast levels.

```
#include <cua/prop.h>

void cua_brightness_contrast_set(cua_properties prop_or_channel,
    double brightness, double contrast);
```

prop_or_channel is the set of properties or the channel whose brightness and contrast levels you want to change.

brightness is the brightness level from 0.0 to 1.0.

contrast is the contrast level from 0.0. to 1.0.

cua_brightness_contrast_set() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_video_trigger_has()

cua_video_trigger_has() returns *TRUE* if a channel or a set of properties has a video trigger property.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Video triggers are supported on certain vision processors and, depending on the vision processor's video configuration, on specific camera ports. Refer to *Video Trigger Property* on page 110 for more information.

```
#include <cua/prop.h>

c_Bool cua_video_trigger_has(cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel that you want to test for the presence of the video trigger property.

cua_video_trigger_mode_has() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel

Function cua_video_trigger_enable_get()

cua_video_trigger_enable_get() returns a Boolean indicating whether the video trigger is currently enabled or disabled. *TRUE* indicates that the trigger is enabled; *FALSE* indicates disabled.

```
#include <cua/prop.h>

c_Bool cua_video_trigger_enable_get(
    cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel whose video trigger you are checking.

cua_video_trigger_enable_get() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_video_trigger_enable_set()

cua_video_trigger_enable_set() lets you enable or disable video trigger support.

```
#include <cua/prop.h>

void cua_video_trigger_enable_set(cua_properties prop_or_channel,
    c_Bool enabled);
```

prop_or_channel is the set of properties or the channel whose video trigger you want to enable or disable.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Note: The **cua_state_get()** function does not accurately describe the state of an acquisition channel that uses a video trigger. When using a video trigger, do not use **cua_state_get()** to detect when an acquisition completes. Set up a completion-callback function instead. See *Callback Function Definition* on page 146 for more information.

enabled is a Boolean that enables or disables the video trigger. *TRUE* enables the video trigger; *FALSE* disables the trigger.

cua_video_trigger_enable_set() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_video_strobe_has()

cua_video_strobe_has() returns *TRUE* if a channel or a set of properties has a video strobe property.

Video strobes are only supported when video triggers are supported.

```
#include <cua/prop.h>

c_Bool cua_video_strobe_has(cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel that you want to test for the presence of the video strobe property.

cua_video_strobe_has() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel

Function cua_video_strobe_enable_get()

cua_video_strobe_enable_get() returns a Boolean indicating whether the video strobe is currently enabled or disabled. *TRUE* indicates that the trigger is enabled; *FALSE* indicates disabled.

```
#include <cua/prop.h>

c_Bool cua_video_strobe_enable_get(
    cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel whose video strobe you are checking.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

cua_video_strobe_enable_get() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_video_strobe_enable_set()

cua_video_strobe_enable_set() lets you enable or disable video strobe support.

```
#include <cua/prop.h>

void cua_video_strobe_enable_set(cua_properties prop_or_channel,
    c_Bool enabled);
```

prop_or_channel is the set of properties or the channel whose video strobe you want to enable or disable.

enabled is a Boolean that enables or disables the video strobe. *TRUE* enables the video strobe; *FALSE* disables the strobe.

cua_video_strobe_enable_set() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_video_strobe_duration_get()

cua_video_strobe_duration_get() returns the strobe duration in microseconds. The strobe duration is a characteristic of the strobe itself. Unified Acquisition uses this information internally.

```
#include <cua/prop.h>

c_Int32 cua_video_strobe_duration_get(
    cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel whose strobe duration property you are getting.

cua_video_strobe_duration_get() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_video_strobe_duration_set()

cua_video_trigger_duration_set() lets you set the strobe duration in microseconds. The strobe duration is a characteristic of the strobe itself.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Setting the duration does not change the strobes behavior. Unified Acquisition uses this information internally to time the firing of the strobe to avoid the charge transfer interval. If you do not specify a duration, 10 µs is used by default.

```
#include <cua/prop.h>

void cua_video_strobe_duration_set(cua_properties prop_or_channel,
    c_Int32 duration);
```

prop_or_channel is the set of properties or the channel whose strobe duration you want to change.

duration is the new value for the strobe duration.

cua_video_strobe_duration_set() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_video_strobe_polarity_get()

cua_video_strobe_polarity_get() returns the polarity of the signal that will be sent to the strobe when an acquisition is triggered using the video trigger. (This assumes that the video strobe has been enabled using **cua_video_strobe_enable_set()**.)

```
#include <cua/prop.h>

cua_polarity cua_video_strobe_polarity_get(
    cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel whose video strobe polarity you are checking.

cua_video_strobe_polarity_get() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Enumeration cua_polarity

cua_video_strobe_polarity_get() returns one of the following:

```
typedef enum cua_polarity
{
    cua_high_to_low,
    cua_low_to_high
} cua_polarity;
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

cua_high_to_low specifies a negative transition.

cua_low_to_high specifies a positive transition.

Function cua_video_strobe_polarity_set()

cua_video_strobe_polarity_set() lets you set the polarity of the signal sent to the strobe when an acquisition is triggered using the video trigger.

```
#include <cua/prop.h>

void cua_video_strobe_polarity_set(cua_properties prop_or_channel,
    cua_polarity polarity);
```

prop_or_channel is the set of properties or the channel whose strobe polarity you want to change.

polarity is either *cua_high_to_low* or *cua_low_to_high*. The default is *cua_high_to_low*.

cua_video_strobe_polarity_set() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_video_strobe_trigger_width_get()

cua_video_strobe_trigger_width_get() returns the strobe trigger width in microseconds. This is the width of the output pulse needed to fire the strobe.

```
#include <cua/prop.h>

c_Int32 cua_video_strobe_trigger_width_get(
    cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel whose strobe trigger width you are checking.

cua_video_strobe_trigger_width_get() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_video_strobe_trigger_width_set()

cua_video_strobe_trigger_width_set() lets you set the strobe trigger width. This is the width of the output pulse needed to fire the strobe. The Unified Acquisition software uses this information to determine when it should fire the strobe to correctly illuminate the acquisition.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The default strobe trigger width is 15 µs.

```
#include <cua/prop.h>

void cua_video_strobe_trigger_width_set(cua_properties
    prop_or_channel, c_Int32 strobe_trigger_width);
```

prop_or_channel is the set of properties or the channel whose strobe trigger width you want to change.

strobe_trigger_width is the new strobe trigger width, in microseconds.

cua_video_strobe_trigger_width_set() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_cti_has()

cua_cti_has() returns a Boolean indicating whether the channel has charge transfer interval properties.

```
#include <cua/prop.h>

c_Bool cua_cti_has(cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel you are checking for charge transfer interval properties.

cua_cti_has() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_cti_start_even_get()

cua_cti_start_even_get() returns the charge transfer interval's start-even value.

```
#include <cua/prop.h>

double cua_cti_start_even_get(
    cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel whose change-transfer-interval value you want to get.

cua_cti_start_even_get() throws the following error:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_cti_end_even_get()

cua_cti_end_even_get() returns the charge transfer interval's end-even value.

```
#include <cua/prop.h>

double cua_cti_end_even_get(
    cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel whose charge-transfer-interval value you want to get.

cua_cti_end_even_get() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_cti_start_odd_get()

cua_cti_start_odd_get() returns the charge transfer interval's start-odd value.

```
#include <cua/prop.h>

double cua_cti_start_odd_get(cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel whose start-odd value you want to get.

cua_cti_start_odd_get() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_cti_end_odd_get()

cua_cti_end_odd_get() returns the charge transfer interval's end-odd value.

```
#include <cua/prop.h>

double cua_cti_end_odd_get(cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel whose end-odd value you want to get.

cua_cti_end_odd_get() throws the following error:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_cti_set()

cua_cti_set(() lets you set new charge transfer interval values.

```
#include <cua/prop.h>

void cua_cti_set(cua_properties prop_or_channel,
    double start_odd, double end_odd,
    double start_even, double end_even);
```

prop_or_channel is the set of properties or the channel whose charge transfer interval you want to change.

start_odd specifies the new start odd-value.

end_odd specifies the new end odd-value.

start_even specifies the new start-even value. (Not used for noninterlaced cameras.)

end_even specifies the new end-even value. (Not used for noninterlaced cameras.)

cua_video_trigger_mode_set() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Callback Functions

The callback functions provide a way for the image acquisition software to call your own functions at specific times during the acquisition process. Pointers to your functions, the callback functions, are stored and treated the same way as other channel properties.

You can specify up to four kinds of callback functions:

- Completion function called when an acquisition has completed successfully.
- Overrun function called when an image acquisition could not begin within the timeout period you specified.
- Move part function called when it is safe for your vision application to move the item being inspected.
- Strobe function called when a strobe should be fired for an acquisition on a given channel.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

When you set up a callback function, you can also specify a pointer to your own data. Whenever the image acquisition software calls your callback function, it will pass this pointer as an argument to your function.

The completion callback function is called after an acquisition has completed successfully. You can use a completion callback function to signal that the acquisition is complete. Using a callback function is faster than using the polling functions **cua_state_get()** and **cua_completion_get()**.

The overrun callback function is called when the acquisition could not begin within the period specified in the timeout property. A typical use for overrun callback functions is to signal that acquisition requests are coming faster than they can be processed. Note that if you specified a negative value for the timeout property, you will never get timeout errors and the overrun function will never be called.

The move part callback function is called when it is safe to move the target object because the image has been integrated (stored in the camera, but not yet transferred to the frame buffer) or because the acquisition has timed out. One way to use this function is to have it signal a conveyor to prepare the next item for inspection and begin the next acquisition.

The strobe callback function is called when the image acquisition software is integrating an image on the specified acquisition channel. (The timing of the call to the strobe callback function is determined by the acquisition software.) The callback function should use the parallel I/O functions to send a pulse of the correct width and polarity to the output line that is connected to the strobe. Note that, for vision processors that support them, video triggers provide a simpler interface for firing a strobe during acquisition. Also note that if a strobe callback function is defined for an acquisition channel for which a video strobe has also been enabled, the callback function will never be called.

Callback Function Definition

The four kinds of callback functions are defined the same way:

```
void UserCallbackFunction(void *context);
```

context is a pointer to your private data used by your function. It can have any value or point to any data you like.

To set up a completion callback function named **MyCompletionFn()**, you might use the following:

```
my_data_structure myData = MyDataAlloc();

cua_complete_callback_set(myChan, MyCompletionFn, myData);
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Function cua_complete_callback_has()

cua_complete_callback_has() returns *TRUE* if a channel or a set of properties is capable of calling a completion callback function.

```
#include <cua/prop.h>

c_Bool cua_complete_callback_has(cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel that you want to test for completion callback function support.

cua_completion_callback_has() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel

Function cua_complete_callback_get()

cua_complete_callback_get() returns a pointer to the completion callback function. If there is no completion function, this function returns *NULL*.

```
#include <cua/prop.h>

cua_callback cua_complete_callback_get(
       cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel whose completion callback function you want to get.

cua_complete_callback_get() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_complete_callback_set()

cua_complete_callback_set() sets the callback function and user context pointer for a set of properties or a channel. The callback function executes after an image has been successfully acquired on the channel to which the properties belong.

```
#include <cua/prop.h>

void cua_complete_callback_set(cua_properties prop_or_channel,
   cua_callback callback_fn, void *context);
```

prop_or_channel is the set of properties or the channel whose completion callback function you want to set.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

callback_fn is a pointer to a function that executes after a successful acquisition. Use *NULL* to remove the currently defined callback function.

context is a pointer to any data you like. The Unified Acquisition software invokes your callback function with this pointer after a successful acquisition.

cua_complete_callback_set() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_complete_callback_context_get()

cua_complete_callback_context_get() gets the user context pointer for a completion callback function associated with a channel or with a set of properties.

```
#include <cua/prop.h>

void *cua_complete_callback_context_get(
       cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel whose user context pointer you want to get.

cua_complete_callback_context_get() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_overrun_callback_has()

cua_overrun_callback_has() returns *TRUE* if a channel or a set of properties is capable of calling an overrun callback function.

```
#include <cua/prop.h>

c_Bool cua_overrun_callback_has(cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel that you want to test for overrun callback function support.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

cua_overrun_callback_has() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel

Function cua_overrun_callback_get()

cua_overrun_callback_get() returns a pointer to the overrun callback function. If there is no overrun function, this function returns *NULL*.

```
#include <cua/prop.h>

cua_callback cua_overrun_callback_get(
       cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel whose overrun callback function you want to get.

cua_overrun_callback_get() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_overrun_callback_set()

cua_overrun_callback_set() sets the callback function and user context pointer for a set of properties or a channel. The callback function executes after an overrun occurs during an acquisition on the channel that the properties set belongs to.

```
#include <cua/prop.h>

void cua_overrun_callback_set(cua_properties prop_or_channel,
    cua_callback callback_fn, void *context);
```

prop_or_channel is the set of properties or the channel whose overrun callback function you want to set.

callback_fn is a pointer to a function that executes after an acquisition error. Use *NULL* to remove the currently defined callback function.

context is a pointer to any data you like. The Unified Acquisition software invokes your callback function with this pointer after an acquisition error.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

cua_overrun_callback_set() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_overrun_callback_context_get()

cua_overrun_callback_context_get() gets the user context pointer of an overrun callback function associated with a channel or with a set of properties.

```
#include <cua/prop.h>

void *cua_overrun_callback_context_get(
       cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel whose user context pointer you want to get.

cua_overrun_callback_context_get() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_move_part_callback_has()

cua_move_part_callback_has() returns *TRUE* if a channel or a set of properties is capable of calling an move part callback function.

```
#include <cua/prop.h>

c_Bool cua_move_part_callback_has(cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel that you want to test for move part callback function support.

cua_move_part_callback_has() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel

Function cua_move_part_callback_get()

cua_move_part_callback_get() returns a pointer to the move part callback function. If there is no move part function, this function returns *NULL*.

```
#include <cua/prop.h.h>

cua_callback cua_move_part_callback_get(
       cua_properties prop_or_channel);
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

prop_or_channel is the set of properties or the channel whose move part callback function you want to get.

cua_move_part_callback_get() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_move_part_callback_set()

cua_move_part_callback_set() sets the move part callback function and user context pointer for a set of properties or a channel. The move part function executes after the image has been integrated (stored in the camera) or when the acquisition has timed out. In either case, it is safe to move the item being inspected without blurring the acquired image.

```
#include <cua/prop.h>

void cua_move_part_callback_set(cua_properties prop_or_channel,
    cua_callback callback_fn, void *context);
```

prop_or_channel is the set of properties or the channel whose move_part callback function you want to set.

callback_fn is a pointer to a function that executes when it is safe to move the item being inspected. Use *NULL* to remove the currently defined callback function.

context is a pointer to any data you like. The Unified Acquisition software invokes your callback function with this pointer when it is safe to move the item being inspected.

cua_move_part_callback_set() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_move_part_callback_context_get()

cua_move_part_callback_context_get() gets the user context pointer of a move part callback function associated with a channel or with a set of properties.

```
#include <cua/prop.h>

void *cua_move_part_callback_context_get(
       cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel whose user context pointer you want to get.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

cua_move_part_callback_context_get() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_strobe_callback_has()

cua_strobe_callback_has() returns *TRUE* if a channel or a set of properties is capable of calling a strobe callback function.

```
#include <cua/prop.h>

c_Bool cua_strobe_callback_has(cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel that you want to test for strobe callback function support.

cua_strobe_callback_has() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel

Function cua_strobe_callback_get()

cua_strobe_callback_get() returns a pointer to the strobe callback function. If there is no strobe function, this function returns *NULL*.

```
#include <cua/prop.h>

cua_callback cua_strobe_callback_get(
    cua_properties prop_or_channel);
```

prop_or_channel is the set of properties or the channel whose strobe callback function you want to get.

cua_strobe_callback_get() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Function cua_strobe_callback_set()

cua_strobe_callback_set() sets the strobe callback function and user context pointer for a set of properties or a channel.

```
#include <cua/prop.h>

void cua_strobe_callback_set(cua_properties prop_or_channel,
    cua_callback callback_fn, void *context);
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

prop_or_channel is the set of properties or the channel whose strobe callback function you want to set.

callback_fn is a pointer to a function that executes at the correct time for firing a strobe. The acquisition software calculates the timing and avoids firing during the CTI. Use *NULL* to remove the currently defined callback function.

context is a pointer to any data you like. The Unified Acquisition software invokes your callback function with this pointer.

cua_strobe_callback_set() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *prop_or_channel* is not a set of properties or a channel or if the channel does not support this property

Object Functions

Use the object functions to make copies of an object and to release the memory they occupy. In general, you must be careful not to release objects that are in use by other objects.

Function cua_copy()

cua_copy() returns a copy of a properties data structure.

```
#include <cua/object.h>

cua_object cua_copy(cua_object object);
```

object is the set of properties you want to copy.

The newly copied object is not associated with any channel. Changing properties in the newly copied object does not affect any channels or the vision processor hardware.

Use **cua_free()** to deallocate the memory allocated by this function.

cua_copy() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *object* cannot be copied

Function cua_free()

cua_free() deallocates the memory used by a unified acquisition object.

```
#include <cua/object.h>

void cua_free(cua_object object);
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

object is the object whose memory you want to deallocate.

This function has no effect if the value of *object* is *cua_no_object*.

Note: When working with acquisition channel properties, call this function only with the copy of the properties data structures that you obtain using **cua_copy()**. Do *not* use it with the handle returned by **cua_properties_get()**.

cua_free() throws the following error:

- *CUA_ERR_TYPE_MISMATCH* if *object* is not a valid unified acquisition object

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter

8

Cognex Software Display

The Cognex Software Display system allows you to display images and graphics from a 5700/5900 vision processor in a window on the host PC. Images and overlay graphics on the vision processor are transferred over the PCI bus. Software Display overlays the graphics on the image and displays the resulting image in a Windows NT window.

This chapter includes the following sections:

Overview of Software Display describes the operation of the Software Display system.

Requirements for Software Display Applications describes requirements for applications that use Software Display, including palette requirements and considerations for multithreaded applications.

Performance Factors describes the factors that influence the performance of the Software Display system.

Cognex Software Display Interface describes the data types and functions that make up the Software Display interface.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Overview of Software Display

Image and graphics data on the vision processor are posted for transfer to the PC host using the Unified Acquisition software. It is up to the display application on the PC, using Software Display, to communicate with the vision processor so that the data being posted is transferred to the PC and displayed.

The vision processor posts images and (optionally) overlay graphics in separate memory buffers. The size of these buffers is determined by the Unified Acquisition video format being used to acquire images. The available video formats range from 640x480 pixels to 256x240 pixels for RS-170 cameras and from 760x574 to 256x256 pixels for CCIR cameras.

Figure 38 shows 256x240 image and graphics buffers.

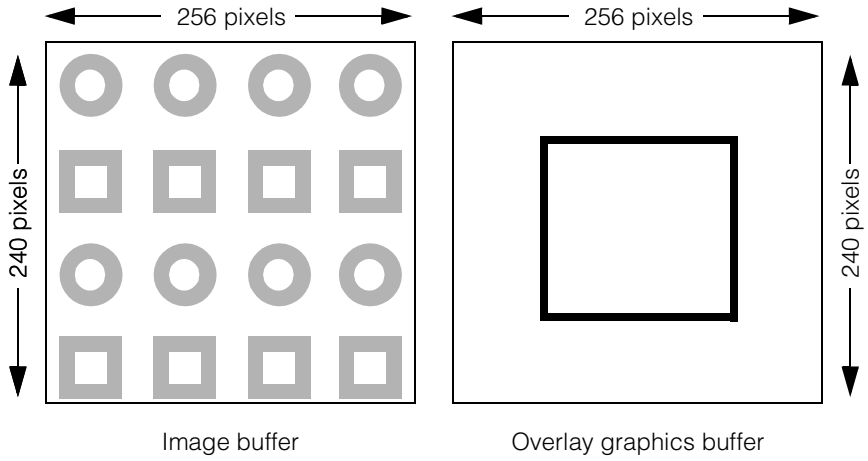


Figure 38. Image and graphics buffers on vision processor

In the graphics buffer, all the pixels except the ones that make up the graphical elements (here, a rectangle) have the value defined on the vision processor by *cua_pass*. In Microsoft's DirectDraw terminology, this value is a *color key*. These pixels are not copied when the graphics buffer is overlaid on the image buffer.

Note: The existence of the graphics buffer on the vision processor is not automatic. It requires that the vision processor application populate a graphics buffer and explicitly assign it to the image currently being displayed. To simplify the following discussion, the examples assume the existence of graphics data on the vision processor.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 39 shows the result of this graphics overlay.

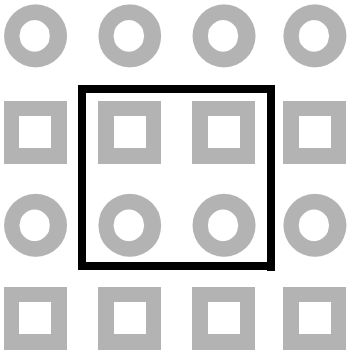


Figure 39. Effect of overlaying graphics buffer on image buffer

In your Windows application, you open display-related communication with the vision processor by opening a software display device with **csd_open()**. You set an image format and a graphics format for the display device. These specify the pixel format of the data on the vision processor and how much of the data in the vision processor's image and graphics buffers you want to display.

The following figure shows the results of setting the image and graphics formats of the display device to a width and height of 200. Setting the formats establishes an initial region of interest for the data on the vision processor. The regions set by the image and graphics formats always have their origins at (0,0), and must be the same size.

Figure 40 shows the effect of setting the 200x200 image and graphics formats. The pixels outside the 200x200 region of interest area are effectively removed from subsequent display on the PC for as long as the image and graphics formats remain in effect. The 200x200 region of pixels also represents an upper limit on the amount of data that Software

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Display will display in a PC window. Regardless of any zoom operation that you implement, Software Display will not display more than 200x200 pixels for as long as the image and graphics formats remain set at 200x200.

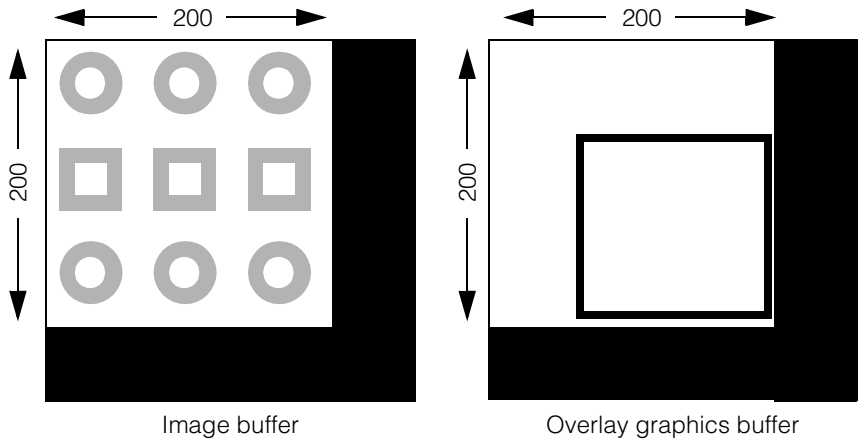


Figure 40. Setting image and graphics formats of the display device establishes initial region of interest

If you set an image format but no graphics format, no graphics buffer will be transferred to the PC. Once you *do* set a graphics format, the graphics buffer is transferred whenever it is posted on the vision processor.

Once you have set the image and (usually) the graphics format of your display device, you can display the data from the vision processor in an application window. You have three options:

- You can request a frozen display to transfer the currently posted image and graphics buffers from the vision processor to the PC, overlay the graphics on the image, and then both display the resulting image and cache it. The cached image can be used to repaint the application window where the image is displayed.
- You can request a live display at a specified frame rate. When you request a live display, images and graphics are transferred, the graphics are overlaid, and the resulting image is displayed. Software Display attempts to transfer and display at the requested rate, subject to available bandwidth on the PCI bus. A live display is terminated by requesting a frozen display.

  Note that while this display is live from the perspective of the PC application in that a new image is displayed at (or near) a requested rate, the vision processor's display output is under the control of the Unified Acquisition software, and could be live, frozen, or blank.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- You can request the next image from the vision processor to transfer the current image and graphics buffers from the vision processor to the PC, overlay the graphics on the image, and display the resulting image. No image is cached. The next-image approach allows an application to transfer and display images only when they are needed.

Note: An application can use either the live display or the next image option; it cannot use both.

Figure 41 shows the 200x200 region established by the image and graphics formats displayed in an application window.

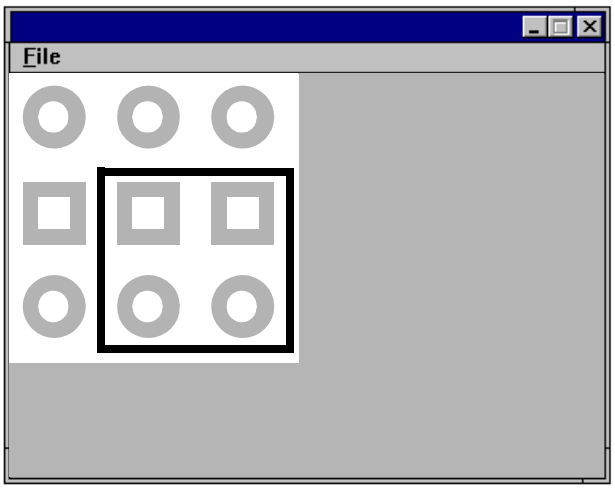


Figure 41. Image and graphics displayed in PC window

Regardless of which approach is used to display an image, there are several things to point out about the way Software Display displays the image in your application window.

- The image always displays at the origin of the window's drawing area.
- When the drawing area of the window is larger than the image (as shown in Figure 41), Software Display does not control the pixels in the drawing area beyond the 200x200 area established by the image and graphics formats.
- When the window's drawing area is smaller than the image, your application is responsible for providing scroll bars.

You can use Software Display to set a region of interest within the area established by the image and graphics formats.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 42 shows the effect of defining a 140x140 region of interest with its origin at (60, 60).

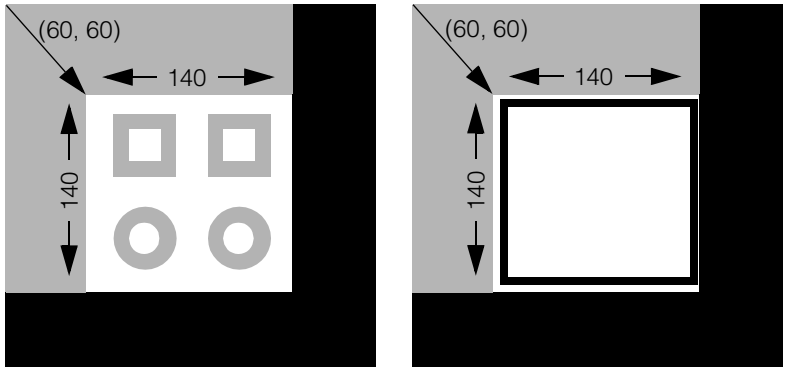


Figure 42. Effect of setting the region of interest within the region established by image and graphics formats

When you set a region of interest, you set the value of a *stretch* flag. When the stretch option is used, the pixels within the region of interest are scaled to fit the window drawing area as the window is resized. The pixels cannot, however, be stretched beyond the dimensions defined by the image/graphics formats (200x200 in this discussion).

Figure 43 shows the region of interest displayed with the *stretch* option off.

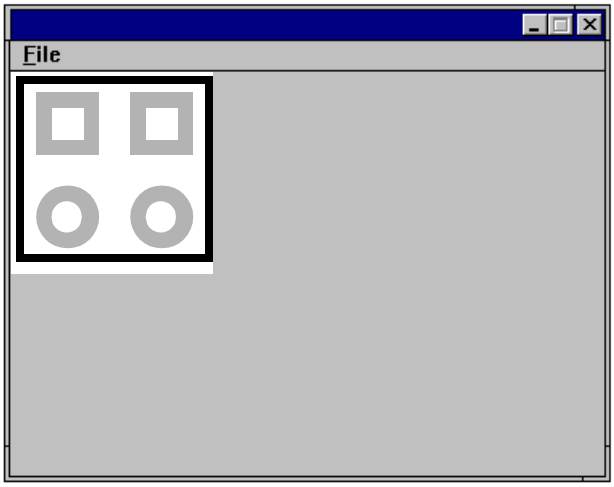


Figure 43. Region of interest displayed without stretch option

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 44 shows the region of interest displayed with the *stretch* option on when the window's drawing area is smaller than the image/graphics format.

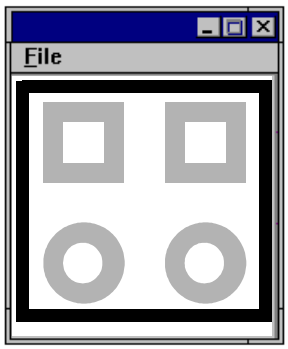


Figure 44. Region of interest displayed with stretch option in small window

Figure 45 shows the region of interest displayed with the *stretch* option on when the window's draw area is larger than the image and graphics formats. The scaling of the region of interest stops, therefore, at 200x200.

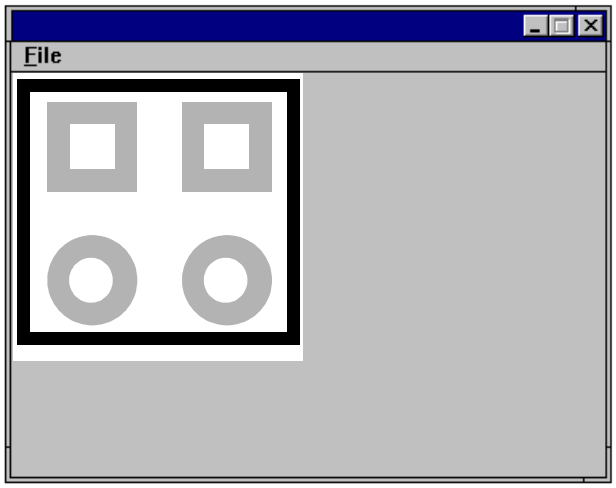


Figure 45. Region of interest displayed with stretch option in large window

There is a performance impact associated with the stretch option, which depends on the capabilities of your video display adapter.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8 Cognex Software Display

Using the Software Display Library

Cognex Software Display is implemented as a DLL, *csd.dll*, which calls a subordinate DLL, *cvideo.dll*. Copies of both DLLs must be delivered to your end-users with your software display application. These DLLs are installed in *\cognex\pchost\ntbin*.

Software Display is supported only under Windows NT 4.0 or later. The supported compiler is Microsoft Visual C++ 4.2, with patch level B applied. Other compiler configurations may work but have not been tested and are not supported by Cognex.

Software Display ships with a header file, *\cognex\pchost\include\csd.h*, and an import library, *\cognex\pchost\ntlib\csd.lib*.

Sample Code

For examples of the operations discussed here, see the Cogview source files, which are installed in *\cognex\pchost\ntsample\cogview*. Note that Cogview does not set a live display; it uses the next-image approach and a timer to display images.

Software Display Limitations

Software Display has the following limitations:

- You can only open one display device in your application.
- The pixel format must be 8-bit greyscale (which includes support for the colors in the graphics buffer).

Relationship Between Software Display and Unified Acquisition

Because Software Display offers **csd_live_set()** and **csd_frozen_set()** functions, you may think that calling these functions in your PC application controls the acquisition of images on the vision processor. This is not the case. The writing of image and graphics data by the vision processor and the reading and displaying of that data on the PC are independent processes.

Requesting a live display in your PC application using **csd_live_set()** does not establish a live display on the vision processor. For example, if the vision processor is currently providing a frozen display, calling **csd_live_set()** in the PC application will repeatedly

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

transfer and display the same frozen image from the vision processor for as long as the vision processor display remains frozen and the PC display remains live. To obtain a live display from the camera connected to the vision processor, two things must happen:

1. The vision processor application must establish a live display using the Unified Acquisition software's **cua_live_set()** function, as described on page 124.
2. Your PC application must call **csd_live_set()**.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Performance Factors

To perform optimally, Software Display requires that the following conditions be met:

- The PCI chip set on the host PC's motherboard must be fully compliant with the PCI 2.1 specification. (Refer to *PCI 2.1 Compliance* on page 21 for more information.)

  Software Display compares the host PC's PCI chip set ID with a list of known non-compliant chip sets stored in the NT registry. If the comparison finds a match, or if Software Display can't find the chip set list in the registry, the test fails. (See *Non-Compliant Chip Set List in Registry* on page 166 for more information.)

- The PC's video display adapter and its device driver must be fully compliant with Microsoft's DirectX specification, version 2.0 or later. The adapter must support blitting and source color keying in hardware, and provide sufficient offscreen video memory to allow hardware blitting of the display image. At least 4 Mbytes of video memory are recommended.

  Cognex has tested and recommends the following video display adapters (with 4 or more Mbytes of memory): ATI Xpert98, ATI Xpert@Work, ATI 3D Expression + PC2TV, and ATI Graphics Pro Turbo.

- The PCI bus must not be saturated with requests from competing PCI adapters, such as network or SCSI cards, or by the use of Cogtalk's block mode server to transfer data from the vision processor to the PC.

When these conditions are met, and depending on CPU speed and availability, Cognex Software Display provides a frame rate of up to 30 frames per second in the following way:

- It uses the PCI bus's DMA mechanism to transfer image and graphics data from the vision processor directly to offscreen memory on the video display adapter.
- It overlays the graphics data on the image in offscreen video memory.
- It uses DirectDraw to display the image and graphics in the PC window.

If any of these conditions are not met, the frame rate is reduced because image and graphics data are first transferred to system memory on the PC, and from there to the video display adapter. The **csd_system_memory()** function is provided to indicate whether or not system memory is being used, and if so, for which of the reasons listed above.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Requirements for Software Display Applications

A Windows NT application that uses Software Display must meet a number of conditions:

- The desktop where the application runs must be set to 256 colors.
- The application must define its 8-bit palette to correctly display both the greyscale image data and the color graphics data from the vision processor.
- If the application is multithreaded, it must guard against suspending a thread while the thread is executing certain Software Display functions. Refer to *Using Cognex Software Display in a Multithreaded Application* on page 166 for more information.
- It must run *badchipset.exe* as part of its installation program.

These requirements are discussed in the following sections.

Defining an 8-Bit Palette for Your Software Display Application

A PC application using Software Display must run on a desktop with 256 colors. To correctly display both the greyscale image data and the colors used in overlay graphics, the PC application also must define an appropriate 8-bit palette.

When the vision processor application uses the default input lookup table during image acquisition, the range of greyscale pixel values in the images transferred from the vision processor is continuous from 10 to 240, inclusive. Entries 10 to 240 in the PC application's palette should produce appropriate grey levels. (Note that if the input lookup table were changed on the vision processor, you would need to adjust your PC application's 8-bit palette to correctly display the image data coming from the vision processor.)

The color values in overlay graphics are specified on the vision processor using Unified Acquisition color constants such as *cua_white* and *cua_yellow* (defined in *\cognex\defs\cua\pixel.h*) Whenever possible, these constants have been defined with values that map to entries in the Windows system palette that store the static Windows colors (entries 0 to 9 and 246 to 255). When there is no match for a Unified Acquisition color constant among the static colors, as with *cua_pink* and *cua_brown*, the color constant has a defined value between 241 and 245. In your application palette, entries 241-245 should be defined with RGB values that produce the intended colors.

To summarize the vtpalette requirements:

- Entries 0-9 and 246-255 should be mapped to the corresponding entries in the Windows system palette.
- Entries 10-240 should produce appropriate grey levels.
- Entries 241-245 should be defined appropriately for *cua_blink*, *cua_brown*, *cua_pink*, *cua_purple*, and *cua_orange*.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

For an example of a working palette, see the Cogview palette generated in *\cognex\pchost\ntsample\cogview\src\palette.cpp*.

Using Cognex Software Display in a Multithreaded Application

Software Display maintains global data (for example, the image format established by **csd_image_format_set()**). Software Display uses the NT operating system's critical section mechanism to synchronize access to its global data in multithreaded applications.

If you are writing a multithreaded application, you must write your program so that it does not suspend a thread that is executing a critical section of a Software Display function. If a Software Display function is suspended in the middle of a critical section, a situation is created in which all threads are blocked from executing critical sections in Software Display functions. The only way to end the deadlock is to resume the suspended thread.

The following Software Display functions contain critical sections:

- **csd_image_format_get()**
- **csd_graphics_format_get()**
- **csd_keycolor_set()**
- **csd_keycolor_get()**
- **csd_image_roi_set()**
- **csd_image_roi_get()**
- **csd_state_get()**
- **csd_next_image_get()**
- **csd_live_set()**
- **csd_frozen_set()**
- **csd_frozen_image_get()**

Non-Compliant Chip Set List in Registry

The console application *badchipset.exe* places in the system registry a *bad chipset* key that stores numeric identifiers of PCI chip sets that are known to be non-compliant with the PCI 2.1 specification. When one of these chip sets is present on the motherboard of a PC where your Software Display application is running, performance degrades because Software Display cannot use the DMA mechanism to write image and graphics data directly to memory on the video display adapter.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The **csd_system_memory()** function (page 179) gets the identifier of the PCI chip set of the machine where your Software Display application is running and compares it with the list created by *badchipset.exe*. If it finds a match, or if it can't find the chip set list in the registry, it passes back a value that indicates a non-compliant PCI chip set.

Cognex runs the *badshipset.exe* utility on your development machine during PVE installation. It is your responsibility to run this utility on the PCs on which you plan to deploy your Software Display application. If the program is not run, **csd_system_memory()** cannot detect the presence of a non-compliant chip set. This condition forces your Software Display application to use system memory for all image transfers from the vision processor.

The *badchipset.exe* application can be run with its *-r* option to remove the *bad chipset* key from the registry.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Cognex Software Display Interface

This section describes the data types and functions that make up the Cognex Software Display interface. The documentation of the Software Display functions includes the possible return values for each function, which indicate whether a function has completed successfully or with an error condition.

Cognex Software Display Data Types

This section describes the Cognex Software Display data types.

Handle csd_handle

A **csd_handle** is a 32-bit value returned by the **csd_open()** function that specifies a display device.

Structure csd_rect

A **csd_rect** is used by the region of interest functions: **csd_image_roi_get()** and **csd_image_roi_set()**. The origin is in the upper left corner; x and y values increase to the right and down.

```
#include <csd.h>

typedef struct csd_rect {
    c_Int32   x, y;
    c_Int32   width, height;
} csd_rect;
```

x and *y* are the (x, y) location of the upper left corner of the rectangle, measured from the origin of the image and graphics buffers on the vision processor.

width and *height* are the width and height of the rectangle in pixels.

Enumeration csd_state

csd_state describes the state of a display device. Use the function **csd_state_get()** to get the state of a display device.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
#include <csd.h>

typedef enum csd_state {
   csd_state_frozen = 0,
   csd_state_live
} csd_state;
```

csd_state_frozen indicates that the display device is displaying a frozen image.

csd_state_live indicates that the display device is displaying a live image.

Structure csd_version

csd_version stores version information for the Software Display library in use. Use the function **csd_version_get()** to retrieve the version information.

```
#include <csd.h>

typedef struct csd_version {
   c_Int32   major;
   c_Int32   minor;
} csd_version;
```

major is the major revision level; *minor* is the minor revision level.

Enumeration csd_pixel_format

csd_pixel_format describes the pixel format of a display device. Software Display uses this value to determine how to interpret pixel information.

```
#include <csd.h>

typedef enum csd_pixel_format{
   csd_format_grey256 = 8
} csd_pixel_format;
```

csd_format_grey256 means that the pixels are interpreted as 8-bit greyscale values.

Typedef csd_keycolor

csd_keycolor describes the pass-through color in the graphics plane. When you use this color in a graphics plane, the underlying image shows through.

```
#include <csd.h>

typedef c_UInt32 csd_keycolor;
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Structure csd_window

csd_window is an abstract data type that Software Display uses to identify the window in which images are displayed.

```
#include <csd.h>

typedef struct csd_window {
    HWND      hWnd;
} csd_window;
```

hWnd: since Software Display currently runs only under Windows NT, a **csd_window** variable is always an HWND, which is a handle to a Windows NT window.

Cognex Software Display Functions

This section describes the Cognex Software Display functions.

Function csd_open()

csd_open() creates a handle to a display device. Only one display device can be open in an application.

```
#include <csd.h>

c_Int32 csd_open(csd_window window, c_Int32 board_num,
       c_Int32 display_num, csd_handle* handle);
```

window can be any window from which Software Display can obtain information it uses to initialize the Microsoft Windows DirectDraw software. It could be the window in which you will display images, the main window of your application, or the desktop window.

board_num must be 0.

display_num must be 0.

handle is a pointer to a variable that will hold the handle to the vision processor display device. The handle is invalid if **csd_open()** returns an error.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

csd_open() returns the following values:

- *CSD_OK* if the function completed successfully
- *CSDERR_NOTIMP* if a parameter requests a feature is not implemented
- *CSDERR_DEVICE_OPEN* if a display device is already open
- *CSDERR_ADAPTER_FAIL* if there is a hardware problem in the vision processor
- *CSDERR_RESOURCE_ALLOC* if this function was unable to allocate the necessary resources
- *CSDERR_INVALID_PARAM* if an invalid parameter was passed
- *CSDERR_DESKTOPNOT256* if the current desktop display is not 256 colors

Function csd_close()

csd_close() closes a display device.

```
#include <csd.h>

c_Int32 csd_close(csd_handle handle);
```

handle is the handle to the display device that you want to close; the handle is not valid after the function returns.

csd_close() returns the following values:

- *CSD_OK* if the function completed successfully
- *CSDERR_NOTINITIALIZED* if the display has not been initialized
- *CSDERR_BAD_HANDLE* if *handle* is not a handle to a display device

Function csd_image_format_set()

csd_image_format_set() sets the image format parameters for a display device.

```
#include <csd.h>

c_Int32 csd_image_format_set(csd_handle handle, c_Int32 width,
       c_Int32 height, csd_pixel_format format);
```

handle is the handle to the display device whose parameters you want to change.

width and *height* are the new width and height of the display device. The same width and height must be specified using **csd_graphics_format_set()**. If different values are specified, Software Display uses the maximum width and height.

format must be *csd_format_grey256*.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

csd_image_format_set() returns the following values:

- *CSD_OK* if the function completed successfully
- *CSDERR_NOTINITIALIZED* if the display has not been initialized
- *CSDERR_BAD_HANDLE* if *handle* is not a handle to a display device
- *CSDERR_NOTIMP* if a parameter requests a feature that is not implemented
- *CSDERR_RESOURCE_ALLOC* if this function was unable to allocate the necessary resources
- *CSDERR_INVALID_PARAM* if an invalid parameter was passed
- *CSDERR_DISPLAY_INPROGRESS* if a live display is currently being provided

Function csd_image_format_get()

csd_image_format_get() gets the current format parameters of a display device.

```
#include <csd.h>

c_Int32 csd_image_format_get(csd_handle handle, c_Int32* width,
       c_Int32* height, csd_pixel_format* format);
```

handle is the handle to the display device whose parameters you want to get.

width and *height* are pointers to variables that hold the width and height of the display device.

format is a pointer to a variable to hold the pixel format. This value is always *csd_format_grey256*.

csd_image_format_get returns the following values:

- *CSD_OK* if the function completed successfully
- *CSDERR_NOTINITIALIZED* if the display has not been initialized
- *CSDERR_BAD_HANDLE* if *handle* is not a handle to a display device
- *CSDERR_INVALID_PARAM* if an invalid parameter was passed

Function csd_graphics_format_set()

csd_graphics_format_set() lets you set the graphics format parameters of a display device.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
#include <csd.h>

c_Int32 csd_graphics_format_set(csd_handle handle, c_Int32 width,
       c_Int32 height, csd_pixel_format format);
```

handle is the handle that refers to the display device whose graphic parameters you want to change.

width and *height* are the new width and height of the display device. The same width and height must be specified using **csd_image_format_set()**. If different values are specified, Software Display uses the maximum width and height.

format must be *csd_format_grey256*.

csd_graphics_format_set() returns the following values:

- *CSD_OK* if the function completed successfully
- *CSDERR_NOTINITIALIZED* if the display has not been initialized
- *CSDERR_BAD_HANDLE* if *handle* is not a handle that refers to a display device
- *CSDERR_NOTIMP* if a parameter requests a feature that is not implemented
- *CSDERR_RESOURCE_ALLOC* if this function was unable to allocate the necessary resources
- *CSDERR_INVALID_PARAM* if an invalid parameter was passed
- *CSDERR_DISPLAY_INPROGRESS* if a live display is currently being provided

Function csd_graphics_format_get()

csd_graphics_format_get() gets the current values of the graphic parameters.

```
#include <csd.h>

c_Int32 csd_graphics_format_get(csd_handle handle, c_Int32* width,
       c_Int32* height, csd_pixel_format* format);
```

handle is the handle that refers to the display device whose parameters you want to get.

width and *height* are pointers to variables that hold the width and height of the display device.

format is a pointer to a variable to hold the pixel format. This value is always *csd_format_grey256*.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

csd_graphics_format_get returns the following values:

- *CSD_OK* if the function completed successfully
- *CSDERR_NOTINITIALIZED* if the display has not been initialized
- *CSDERR_BAD_HANDLE* if *handle* is not a handle to a display device
- *CSDERR_INVALID_PARAM* if an invalid parameter was passed

Function csd_keycolor_set()

csd_keycolor_set() sets the key color that will be treated as transparent when the graphics plane is overlaid on the image plane.

This value should be equal to the value of the *cua_pass* constant that is used on the vision processor to identify the pass-through pixels in the graphics plane. The default value of *cua_pass* is 240.

```
#include <csd.h>

c_Int32 csd_keycolor_set(csd_handle handle, csd_keycolor keycolor);
```

handle is the handle to the display device whose key color you want to change.

keycolor is the new value between 0 and 255.

csd_keycolor_set returns the following values:

- *CSD_OK* if the function completed successfully
- *CSDERR_NOTINITIALIZED* if the display has not been initialized
- *CSDERR_BAD_HANDLE* if *handle* is not a handle to a display device

Function csd_keycolor_get()

csd_keycolor_get() gets the current value of the key color. The key color defines the pixel value in the graphics plane that will not be copied when the graphics plane is overlaid on the image plane. To match up with the *cua_pass* constant used on the vision processor, the value of *keycolor* should be 240.

```
#include <csd.h>

c_Int32 csd_keycolor_get(csd_handle handle,
    csd_keycolor *keycolor);
```

handle is the handle to the display device whose key color you want to get.

keycolor is a pointer to a variable that will hold the key color, which will be between 0 and 255.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

csd_keycolor_get returns the following values:

- *CSD_OK* if the function completed successfully
- *CSDERR_NOTINITIALIZED* if the display has not been initialized
- *CSDERR_BAD_HANDLE* if *handle* is not a handle to a display device
- *CSDERR_INVALID_PARAM* if an invalid parameter was passed

Function csd_image_roi_set()

csd_image_roi_set() sets the region of interest.

```
#include <csd.h>

c_Int32 csd_image_roi_set(csd_handle handle, csd_rect rect,
       c_Bool stretch);
```

handle is the handle to the display device whose region of interest you want to set.

rect is a rectangle that describes the new region of interest.

stretch is a Boolean. If *stretch* is true, the image is enlarged to fit the viewing area of the window up to the size of the current image and graphics formats. Otherwise, the image is constrained to the size of *rect*. By setting this value to *true*, you can implement a zooming mechanism.

csd_image_roi_set returns the following values:

- *CSD_OK* if the function completed successfully
- *CSDERR_NOTINITIALIZED* if the display has not been initialized
- *CSDERR_BAD_HANDLE* if *handle* is not a handle to a display device
- *CSDERR_INVALID_PARAM* if an invalid parameter was passed

Function csd_image_roi_get()

csd_image_roi_get() gets the current region of interest.

```
#include <csd.h>

c_Int32 csd_image_roi_get(csd_handle handle, csd_rect* rect,
       c_Bool *stretch);
```

handle is the handle to the display device whose region of interest you want to get.

rect is a pointer to a rectangle that will hold the current region of interest.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

stretch is a pointer to a Boolean that will hold the current value of the region of interest's *stretch* flag.

csd_image_roi_get returns the following values:

- *CSDERR_NOTINITIALIZED* if the display has not been initialized
- *CSDERR_BAD_HANDLE* if *handle* is not a handle to a display device
- *CSDERR_INVALID_PARAM* if an invalid parameter was passed

Function csd_state_get()

csd_state_get() gets the current state of the display device.

```
#include <csd.h>

c_Int32 csd_state_get(csd_handle handle, csd_state* state);
```

handle is the handle to the display device whose state you want to set.

state is a pointer to a variable that will hold the current state. The possible values for state are *csd_frozen* and *csd_live*.

csd_state_get returns the following values:

- *CSD_OK* if the function completed successfully
- *CSDERR_NOTINITIALIZED* if the display has not been initialized
- *CSDERR_BAD_HANDLE* if *handle* is not a handle to a display device
- *CSDERR_INVALID_PARAM* if an invalid parameter was passed

Function csd_live_set()

csd_live_set() creates a live display where the selected region of data and graphics on the vision processor is transferred to the PC and displayed in a window at a specified frame rate. An application can use one or the other of **csd_live_set()** and **csd_next_image_get()**. It cannot use both.

Note: Using **csd_live_set()** has no effect on the vision processor's display output, which is under the control of the vision processor application using the Unified Acquisition software and could be live, frozen, or blank.

```
#include <csd.h>

c_Int32 csd_live_set(csd_handle handle, csd_window window,
    c_Int32 fps, csd_callback user_fp, void* user_param);
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

handle is the handle to the display device. The region of interest selected for the display device determines which region of the vision processor's image and graphics buffers will be transferred and displayed. If no region of interest is selected using **csd_image_roi_set()**, the region established by the current image and graphics formats is used.

window is the window in which to draw the image. If a frame contains both an image plane and a graphics plane, the window will display the results of merging the image and graphics planes.

fps is the number of times per second that you want the function to get an image from the vision processor and update the display.

user_fp is a pointer to a user function that is called after this function has mixed the image and graphics planes but before it copies the image into the device context.

user_param is a parameter that will be passed to the user function.

The user function takes one parameter, *user_param*. The prototype for the user function is as follows:

```
void liveset_callback(void *user_param);
```

csd_live_set returns the following values:

- *CSD_OK* if the function completed successfully
- *CSDERR_NOTINITIALIZED* if the display has not been initialized
- *CSDERR_BAD_HANDLE* if *handle* is not a handle to a display device
- *CSDERR_DISPLAY_INPROGRESS* if a live display is currently being provided

Function csd_next_image_get()

csd_next_image_get() transfers the currently selected region of image and graphics data from the vision processor and displays it.

Note: An application can use one or the other of **csd_next_image_get()** and **csd_live_set()**. It cannot use both.

```
#include <csd.h>

c_Int32 csd_next_image_get(csd_handle handle,
    csd_window window);
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

handle is the handle to the display device. The region of interest selected for the display device determines which region of the vision processor's image and graphics buffers will be transferred and displayed. If no region of interest is selected using **csd_image_roi_set()**, the region established by the current image and graphics formats is used.

window is the window in which to draw the image.

csd_next_image_get returns the following values:

- *CSD_OK* if the function completed successfully
- *CSDERR_NOTINITIALIZED* if the display has not been initialized
- *CSDERR_BAD_HANDLE* if *handle* is not a handle to a display device
- *CSDERR_DISPLAY_INPROGRESS* if a live display is currently being provided
- *CSDERR_DISPLAYHW_FAIL* if there is a Windows display communication failure

Function csd_frozen_set()

csd_frozen_set() transfers the currently selected region of image and graphics data from the vision processor and displays it.

Note: Using **csd_frozen_set()** has no effect on the vision processor's display output, which is under the control of the vision processor application using the Unified Acquisition software and could be live, frozen, or blank.

```
#include <csd.h>

c_Int32 csd_frozen_set(csd_handle handle, csd_window window);
```

handle is the handle to the display device. The region of interest selected for the display device determines which region of the vision processor's image and graphics buffers will be transferred and displayed. If no region of interest is selected using **csd_image_roi_set()**, the region established by the current image and graphics formats is used.

window is the window in which to draw the image. To refresh the frozen image when this window is subsequently repainted, use **csd_frozen_image_get()**.

csd_frozen_set() returns the following values:

- *CSD_OK* if the function completed successfully
- *CSDERR_NOTINITIALIZED* if the display has not been initialized
- *CSDERR_BAD_HANDLE* if *handle* is not a handle to a display device
- *CSDERR_ADAPTER_FAIL* if unable to communicate with the vision processor

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Function csd_frozen_image_get()

csd_frozen_image_get() displays the image displayed by a previous call to **csd_frozen_set()**.

```
#include <csd.h>

c_Int32 csd_frozen_image_get(csd_handle handle, csd_window window);
```

handle is the handle to the display device.

window is the window in which to display the frozen image. The call to **csd_frozen_image_get()** should reside in the code that executes when *window* receives a paint message.

csd_frozen_image_get() returns the following values:

- *CSD_OK* if the function completed successfully
- *CSDERR_NOTINITIALIZED* if the display has not been initialized
- *CSDERR_DISPLAY_INPROGRESS* if a live display is currently being provided
- *CSDERR_BAD_HANDLE* if *handle* is not a handle to a display device
- *CSDERR_DISPLAYHW_FAIL* if there is a Windows display communication failure

Function csd_system_memory()

csd_system_memory() indicates whether Software Display is using the PC's system memory and, if it is, the reason why. (Using system memory in conjunction with video memory on the video display adapter results in a slower frame rate than the exclusive use of video display memory. Refer to *Performance Factors* on page 164 for additional information.)

```
#include <csd.h>

c_Int32 csd_system_memory(csd_handle handle, c_Int32* status);
```

handle is the handle to the display device.

status is a pointer to a variable to hold the value that indicates whether system memory is being used, and why. Possible values are:

- 0 if system memory is not being used
- 1 if system memory is being used because the video display adapter or its software driver does not support DirectDraw
- 2 if DirectDraw is supported, but system memory is being used because the PC's PCI chip set is not compliant with the PCI 2.1 specification

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- 3 if the function cannot find the bad-chipset key in the system registry (See *Non-Compliant Chip Set List in Registry* on page 166 for more information.)

csd_system_memory() returns the following values:

- *CSD_OK* if function returned successfully
- *CSDERR_NOTINITIALIZED* if the display has not been initialized
- *CSDERR_BAD_HANDLE* if *handle* is not a handle to a display device

Function csd_version_get()

csd_version_get() gets the version of the Cognex Software Display library (*csd.dll*).

```
#include <csd.h>

c_Int32 csd_version_get(csd_version *version);
```

version is a pointer to a structure that holds the version information. See *Structure csd_version* on page 169.

csd_version_get() returns the following values:

- *CSD_OK* if the function completed successfully
- *CSDERR_INVALID_PARAM* if an invalid parameter was passed

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 9

Using Cogtalk

PVE revision 8.1 introduced Cogtalk, an application used to communicate with the vision processor. Cogtalk replaces the Wintalk and PCVtalk applications shipped with previous PVE revisions.

Cogtalk is delivered in two versions:

- The version for Windows NT console sessions is called Console Cogtalk.
- The version of Cogtalk with a Windows interface is called Windows Cogtalk.

When the word Cogtalk is used by itself throughout this manual, it refers to both versions, unless the context clearly refers to one version.

This chapter describes Cogtalk in the following sections:

- *Cogtalk Features* provides an introductory overview of Cogtalk.
- *Cogtalk Usage Notes* describes features common to both versions of Cogtalk.
- *Using Console Cogtalk* describes features exclusive to the console version of Cogtalk.
- *Using Windows Cogtalk* describes features unique to the Windows version.
- *Windows Cogtalk Menu and Toolbar Reference* explains the use of each menu item and icon in the Windows Cogtalk user interface.
- *Windows Cogtalk Keyboard Shortcuts* lists the shortcuts available when running Windows Cogtalk.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Cogtalk Features

This section introduces Cogtalk and compares it with Wintalk and PCVtalk.

Cogtalk Requirements

To run either version of Cogtalk, the host PC must meet these requirements:

- Must be running Windows NT, version 4.0 or later.
- Must have a Cognex 5700/5900 and its device driver installed and running.
- The Cognex-supplied DLLs *mid_mnt.dll* and *bserver.dll* must be in NT's DLL search path.
- A mouse or other pointing device is recommended for Windows Cogtalk.

Only one application at a time can have open any one pseudo-serial port on the vision processor. You can run Windows Cogtalk at the same time as Console Cogtalk, or your own application built with the Cognex mid-level library, but each application must use a separate pseudo-serial port. Note that Cogview uses pseudo-serial port 2 for communicating the PC's mouse data to the vision processor, when Cogview's mouse feature is enabled.

Console Cogtalk Features

Console Cogtalk offers the features shown in Table 29, compared to its predecessor, PCVtalk:

Console Cogtalk	PCVtalk
Win32 console application; supported under Windows NT	DOS application; not supported under Windows NT
Command history recall and editing	Primitive command editing
Command history survives exit in a single console window	Primitive command history
Presumes default include directory path	Must specify include directive at each invocation
Arguments to escape commands are specified on the same command line	Certain escape commands prompt for arguments.

Table 29. Console Cogtalk compared to PCVtalk

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Console Cogtalk	PCVtalk
On vision processor load, does not reset the vision processor if it's already in Boot Polling mode	Always resets the vision processor during system load, even if not required
Offers a verbose mode to show the processing of included files as it proceeds	Not available
Can communicate with separate vision processor pseudo-serial ports in separate console sessions	Same
Can boot and/or load the vision processor, then exit (for use in batch files)	Same
Supports a block mode file server to allow transfer of images and files between the vision processor and the host	Same
Can generate a session history log file	Same

Table 29. Console Cogtalk compared to PCVtalk

Windows Cogtalk Features

Windows Cogtalk offers the features shown in Table 30, compared to its predecessor, Wintalk.

Windows Cogtalk	Wintalk
Communicates with vision processor through the Cognex mid-level library and device driver	Same
Supports multiple port windows, each communicating over a separate pseudo-serial port	Supports single communication window; requires separate Wintalk sessions to communicate with separate ports
Each port window keeps track of the volume or saved system file currently loaded in the vision processor	Not available

Table 30. Windows Cogtalk compared to Wintalk

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Windows Cogtalk	Wintalk
Command history recall and editing	Not available
Can select font for port windows; fixed-pitch font is the default	Supports only the system font
Can print the contents of a port window's display history	Not available
Can reliably copy and paste from display history or from another window. Cutting from display history is disallowed.	Incorrectly allows cutting from display history. Cut and paste from display history can confuse Wintalk's current line count.
Optional menu-based method of including files	Not available
On vision processor load, does not reset the vision processor if it's already in Boot Polling mode	Always resets the vision processor during system load even if not required
Supports vision processor status dialog box	Not available
Can invoke Cogview, Sysgen, and Epsilon from Cogtalk	Not available
Supports searching for text in a port window's display history	Not available
Fully supports Windows path separator (backslash) everywhere path names are specified	Required Unix path separator (slash) in include path specifications
Recognizes the same escape commands as Console Cogtalk	Not available

Table 30. Windows Cogtalk compared to Wintalk

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Cogtalk Usage Notes

This section provides overviews of features common to both Console Cogtalk and Windows Cogtalk.

Controlling the Vision Processor

Both versions of Cogtalk can be used to control the operation of the vision processor, including support for the following operations:

- Initial boot or reset of the vision processor
- Loading a volume file or saved system file from the PC to the vision processor
- Saving the currently running vision processor system to a saved system file on the PC

Only one application can communicate with a given pseudo-serial port at a time. This means that either Windows Cogtalk or Console Cogtalk, but not both, can have open a V session at a given moment, because V uses pseudo-serial port 0 by default. However, Windows Cogtalk can communicate over one pseudo-serial port while Console Cogtalk communicates over a different pseudo-serial port at the same time.

The most common use for Cogtalk is to communicate with the V compiler running on the vision processor. Cogtalk can also be used to communicate with an alternate command processor you have written. If you have generated a V-less system (as described in *Chapter 10, Components of a Vision Processor Program*, of the *Development Environment 1* manual), and replaced V with a command interpreter of your own, then your command prompt is displayed instead of V>.

Status and information messages from Cogtalk are prefixed with the prompt *[Cogtalk]* to distinguish them from V's responses.

Include Path and Include Preprocessor

Cogtalk has special features for communicating with the V compiler running on the vision processor.

You can send code fragments or entire files of V code to the V compiler by specifying the file's name as an argument to the **#include** directive at the *V>* prompt. Cogtalk preprocesses all **#include** directives to prevent sending redundant, duplicate, or undesired code to V.

Cogtalk does not maintain a local symbol table, as do Wintalk and PCVtalk. Instead, it queries the vision processor to determine if a file should be included. This avoids the issue of keeping the symbol table in sync with the one on the vision processor.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

You can place the name of the file to be included in angle brackets or quotes. If the file's path name is in angle brackets, Console Cogtalk searches for the file in every directory in Cogtalk's include path. If the file name is in quotes, Console Cogtalk first searches the current directory for the file before searching the include path.

Windows Cogtalk always searches its current working directory first in its include file search. Thus, Windows Cogtalk behaves the same whether the file path to be included is entered with quotes or angle brackets.

Cogtalk's include preprocessor is discussed in more detail in *Cogtalk as Preprocessor*.

Windows Cogtalk's Include Path and Working Directory

Specify Windows Cogtalk's include path in a dialog box invoked with the **Options->Include Path** menu item. Specify a list of absolute or relative directory path names, separated by semicolons. Do not enter spaces between path names. The current directory is always searched first, as part of Windows Cogtalk's internal path searching mechanism, whether or not it is explicitly part of the include path you specify. A recommended include path for Cognex software installed in the default directory structure is the following:

```
.;c:\cognex\working;c:\cognex\defs;c:\cognex\load;
```

Windows Cogtalk appends a closing semicolon to the include path.

If you specify a relative path name (such as *..\defs*), the path is relative to the current working directory. You specify Windows Cogtalk's initial working directory in the setup of the shortcut that invokes Windows Cogtalk. A newly installed Windows Cogtalk shortcut specifies *<cogroot>\working,* where *<cogroot>* is the top-level Cognex directory specified during installation. For example, the default initial working directory for Windows Cogtalk is *c:\cognex\working*.

You can set a new working directory for Windows Cogtalk by means of the **File->Working Directory** menu item or with the **#cwd** directive or **~cwd** command. If you change the current working directory, the include path becomes relative to Cogtalk's new working directory.

Console Cogtalk's Include Path

Specify Console Cogtalk's include path by means of the *-i* command line switch. If no *-i* switch is specified, Console Cogtalk presumes the following default include path:

```
..\load;..\defs
```

Relative directory specifications in Console Cogtalk's include path are relative to Cogtalk's initial working directory. You can change Cogtalk's working directory at run time by means of the **#cwd** directive or **~cwd** command.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Cogtalk as Preprocessor

Cogtalk traps and interprets two preprocessor directives, **#include** and **#cwd**, before sending them on to the vision processor. In the case of **#include**, preprocessing prevents sending redundant information to the vision processor. The preprocessing of **#cwd** allows directory change requests to occur on the host PC.

The older Cognex utilities Wintalk, PCVtalk, and Vtalk preprocessed a larger set of directives, and maintained a local symbol table, which could get out of synchronization with the V compiler's symbol table under certain circumstances. Cogtalk does not maintain a local symbol table, but instead queries the vision processor to determine whether it should send a given file.

The #include Directive

The **#include** directive transmits files from the host PC to the V compiler. When Cogtalk encounters an **#include** directive, it first queries the vision processor to determine if the file should be included. If so, it opens the indicated file and sends its contents to the vision processor. After Cogtalk has processed an **#include** directive, it does not send the **#include** line itself to the V compiler. Cogtalk interprets any **#include** directive within a transmitted file the same way it would as when typed at the V> prompt.

Before sending a file, Cogtalk sends a **#line** directive to the V compiler. This tells V to number any error messages relative to the first line of the file being processed. At the end of the transmission of a simple file (containing no imbedded **#include** directives), Cogtalk sends to V another **#line** directive with a null file name, which tells V to re-enter interactive mode. (V's interactive and noninteractive modes are discussed in *Chapter 4, Using V*, of the *Development Environment 1* manual.)

A *line* is defined as all of the input between one carriage return and the next. Cogtalk displays an error message if the **#include** syntax is faulty or if it cannot open the specified file. The latter problem is usually due to an incomplete include path.

The following example describes what steps Cogtalk takes internally when you enter an **#include** directive that specifies the simple file *file.v*:

1. Cogtalk queries the vision processor to see if the file should be included.
2. If so, Cogtalk opens *file.v*.
3. Cogtalk takes V out of interactive mode by sending the following line to the vision processor:

   ```
   #line 1 "file.v"
   ```

4. Cogtalk sends the contents of *file.v* to V.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5. Cogtalk puts V back in interactive mode by sending a **#line** directive with a null file name:

   ```
   #line n ""
   ```

   where *n* is V's internal line count of interactive lines typed so far at the V> prompt.

Handling of Nested Includes

If a nested **#include** directive occurs in a file that Cogtalk is transmitting to V, Cogtalk resets V's current file name and line count to the first line of the new file, transmits the contents of the new file, then restores the file name and line count to the next line in the original file.

The following example analyzes Cogtalk's internal behavior when nested include files are sent to V. This example assumes that *1st_file.v* contains ten lines, and that the fourth line of *1st_file.v* is the following:

```
#include <2nd_file.v>
```

Under these circumstances, Cogtalk goes through the following steps when *1st_file.v* is transmitted to V:

1. Cogtalk queries the vision processor to see if *1st_file.v* should be included.
2. If so, Cogtalk opens *1st_file.v*.
3. Cogtalk sends the following line to the vision processor:

   ```
   #line 1 "1st_file.v"
   ```

4. Cogtalk sends the first three lines of *1st_file.v* to the vision processor.
5. Cogtalk queries the vision processor to see if *2nd_file.v* should be included. If not, it skips to step 10.
6. If so, Cogtalk opens *2nd_file.v*.
7. Cogtalk records the current line number in *1st_file.v* (which is line 4) and sends the following directive to V:

   ```
   #line 1 "2nd_file.v"
   ```

8. Cogtalk sends the contents of *2nd_file.v* to V.
9. Cogtalk restores the next line number for *1st_file.v,* which is the line following the nested **#include** directive (line 5). Cogtalk then sends the following line to V:

   ```
   #line 5 "1st_file.v"
   ```

10. Cogtalk sends the remaining six lines of *1st_file.v* to V.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

11. Cogtalk puts V back in interactive mode by sending a **#line** directive with a null file name:

    ```
    #line n ""
    ```

    where *n* is V's internal line count of interactive lines typed so far at the V> prompt.

The #cwd Directive

You use the **#cwd** directive to change the current working directory used by Cogtalk. Any relative path names in Cogtalk's include path are relative to the current working directory. Cogtalk also recognizes the **~cwd** escape command as an alias for the **#cwd** directive.

The **#cwd** directive takes a single parameter, the path name including drive letter that you want to become the current working directory. Note that the path name you specify with the **#cwd** directive or the **~cwd** escape command is *not* enclosed in quotation marks, and can use either slash or backslash as path separators.

Escape Commands

Cogtalk recognizes a set of commands, prefixed (by default) with the tilde character (~), that are interpreted by Cogtalk instead of being passed to the V compiler. These commands are recognized identically by both Console Cogtalk and Windows Cogtalk, although Windows Cogtalk offers dialog boxes and menu items to accomplish most of the same tasks.

The escape commands must be typed as the first characters of each input line. That is, the escape character, tilde by default, must be the first character typed at the left margin of a port window, or the first character typed after the vision processor's prompt.

Cogtalk's escape commands are similar to those used in PCVtalk, but do not have identical functionality. In particular, PCVtalk's escape commands prompted for arguments. By contrast, when Cogtalk's escape commands take arguments, they must be supplied on the command line.

The escape commands are listed in the following sections in the order shown by the **~?** command.

Reset vision processor (~b)

Resets the vision processor. In Windows Cogtalk, **~b** must be typed in the port 0 window, which must be the only port window open.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Load Vision Processor (~r)

This command takes one argument, the path name of a loadable system file. Cogtalk first checks the validity of the specified file path, then checks the status of the vision processor. If the vision processor is in Boot Polling mode, Cogtalk immediately loads the specified system file. If the vision processor is running another system file, Cogtalk resets the vision processor before loading the new system file. In Windows Cogtalk, **~r** must be typed in the port 0 window, which must be the only port window open.

Begin System Save Operation (~x)

Use this command with the **css_save_system()** function, or with your own function that calls **css_save_system()**, to save the current state of the vision processor to a saved system file. For Cognex 5700/5900s, be sure to invoke **css_save_system()** with the *css_pci* parameter, so that the output stream is directed over the PCI bus. See *Chapter 5, Save System*, in the *System Software* manual for a full description of **css_save_system()**.

The **~x** command takes two arguments:

1. The path name of the PC file to receive the saved system image.
2. The V statement that will initiate the save system operation. Be sure to close the V statement with a semicolon, just as you would at the V> prompt.

In Windows Cogtalk, **~x** must be typed in the port 0 window.

The following is an example system-saving session. In this example, the saved system is stored in the file *sysimage.sav.* The *css_pci* argument to the **css_save_system()** function specifies the destination of the save as pseudo-serial port 0 on the PCI bus.

```
V>~x sysimage.sav css_save_system(css_pci, 0);
[Cogtalk] Saving system...
.....................................................
[Cogtalk] Save complete.
0, 00, 0x0
V>
```

Begin ASCII Data Capture (~s)

Use this command to capture ASCII data into a file on your host PC. This data might be the output of a data-generating function such as **cip_print()** or **cip_nprint()**. If you supply a data-generating function, it should append the Ctrl+Z end-of-file character to the output stream, or you must manually cancel the data capture with the **~q** command.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The **~s** command takes two arguments:

1. The path name of the PC file to receive the ASCII data.
2. The complete set of V statements that will initiate the data transfer. Be sure to close the V statements with semicolons, just as you would at the V> prompt.

The following is a sample data capture session. The data-generating function is **cip_nprint()** and the data is stored in *image.dat:*

```
V>~s image.dat cip_nprint(fb_image, "imgbuff");
[Cogtalk] Save completed.
V>
```

If errors occur in the function or macro called to save data, the error messages from the V compiler are written to your output file. The save function may not complete execution, any end-of-file character may not be sent, and thus Cogtalk may wait indefinitely for the end-of-file character. In this case, use **~q** to end the data capture, then look at the output file for the cause of the transfer failure.

End Data Capture (~q)

Abort a data capture session begun with **~s**. Use this command to terminate an ASCII data capture session.

Send Ctrl+C (~c)

Aborts the current operation and sends the Ctrl+C character (ASCII 3) to the vision processor, which sends the signal *CIC_ERR_INT* to V. This signal suspends execution of the current instance of V. Typically, this causes a stack trace, and starts a new instance of V.

Send Ctrl+D (~d)

Sends the End-of-File character, Ctrl+D (ASCII 4), to the vision processor, which causes the V compiler to exit its current instance. If the current instance of V is recursive (if the prompt is, for example, *V2>*), that instance exits and leaves the calling instance open (the prompt returns to *V1>*).

Send Ctrl+\: (~\)

Aborts the current operation and sends the Quit character, Ctrl+\ (ASCII 28), to the vision processor. This sends the signal *CIC_ERR_QUIT* to V, which causes V to clean up and pass the signal to its invoker. Use this command to halt a vision processing program stuck in an infinite loop or stuck waiting for input.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Send File (~send)

Use this command to send a file or files of commands or other ASCII data to V, or to your own command interpreter on the vision processor. This is not the same as using **#include** to send the file, because **~send** does not route the commands through Cogtalk's include preprocessor. Therefore, the files you send to V with **~send** cannot contain any **#include** preprocessor directives.

The **~send** command takes one or more arguments, the path name of the file or files to send to the vision processor. Multiple file names can be listed in a single command. For example:

```
V>~send commands.txt command2.txt
```

Change Current Working Directory (~cwd)

This command changes Cogtalk's current working directory. The **~cwd** command takes one argument, the path name, including drive letter, of the directory to become the new working directory. The **~cwd** command serves the same function as the **#cwd** preprocessor directive.

Send VP Interrupt (~t)

This command generates either a bus trigger interrupt or a pseudo expansion interrupt on the vision processor. You can use this command to debug your own interrupt service routines running on the vision processor. Note that pseudo expansion interrupts are *not* the same as ISA or PCI bus interrupts. Pseudo expansion interrupts occur only on the vision processor.

The **~t** command takes one argument, which is either of two literal strings *trigger* or *interrupt*. Specifying *trigger* is the same as invoking **cvp_trigger()** in a PC host application. Specifying *interrupt* is the same as invoking **cvp_exp_int()**. These functions are described in *Chapter 2, Cognex 5000 Communications API*, of the *Development Environment 2* manual.

Change Escape Character (~escape)

Use this command to change the escape character for the escape commands from tilde (~) to the character or character string of your choice. For example, use the following command to change the escape character to a dollar sign:

```
V>~escape $
```

Thereafter, you must use the dollar sign as a prefix for the escape commands, until the current session of Console Cogtalk ends, or until the active port window closes in Windows Cogtalk.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Toggle Quiet Mode (~quiet)

Use the **~quiet** command to toggle the display of file-processing progress messages. Quiet mode, in which no processing messages are displayed, is Cogtalk's default mode. Type **~quiet** to toggle quiet mode on or off. Toggling quiet mode affects only the amount of information returned by Cogtalk; it does not affect V in any way. Turning quiet mode off is especially useful when using the **File->Include File** menu item, which otherwise includes files silently.

File-processing progress messages show the steps that Cogtalk takes when processing an **#include** directive or **~send** command. For example, when using **~send** to send a small set of V commands to V, Cogtalk displays the following when quiet mode is off:

```
V>~quiet
[Cogtalk] Quiet: off
V>~send commands.txt
[Cogtalk] Sending ‘commands.txt’...
[Cogtalk] Sending ‘commands.txt’...done
V>
```

When processing an **#include** directive with quiet mode off, Cogtalk shows each included file as it is processed, with indentation showing the nesting levels of the included files. The following example shows the first few lines of nested include files processed when a demonstration V file is processed:

```
V>~quiet
[Cogtalk] Quiet: off
#include <demo.v>
[Cogtalk] Sending `c:\cognex\working\demo.v'...
[Cogtalk] --Sending `c:\cognex\load\cse_21.opt'...
[Cogtalk] ----Sending `../defs/search/ccal_def.h'...
[Cogtalk] ------Sending `../defs/search/cse_parm.h'...
[Cogtalk] ------Sending `../defs/search/cse_parm.h'...done
[Cogtalk] ------Sending `../defs/search/cse_rslt.h'...
[Cogtalk] ------Sending `../defs/search/cse_rslt.h'...done
[Cogtalk] ----Sending `../defs/search/ccal_def.h'...done
[Cogtalk] ----Sending `../defs/search/crsw_def.h'...
[Cogtalk] ----Sending `../defs/search/crsw_def.h'...done
...
```

Display Cogtalk Version (~v)

Use this command to display Cogtalk's version information.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Exit Cogtalk / Exit Port Window (~. or ~close)

In Console Cogtalk, tilde-period (**~.**) or **~close** closes the current session and exits back to the NT command prompt. You can also type *<Ctrl+Z>* and press *<Enter>*.

In Windows Cogtalk, **~.** or **~close** closes and exits the current port window.

Display Command List (~? or ~help)

The **~?** or **~help** command displays usage information on the Cogtalk escape commands.

Cogtalk as File Server

Unlike the Cognex 5000-ISA vision processors, the Cognex 5700/5900 does not support directly-attached SCSI disks. All file system access must occur over the PCI bus to the host PC's file system. Cogtalk supports a block mode file server that allows the vision processor to communicate with the PC's file system.

Cogtalk's block server support is the same as in the latest revisions of Wintalk and PCVtalk. Cogtalk does not support the earlier serial mode file server.

In order to use the PC's file system from a vision processing application, the block mode server must be running on the PC host. The block mode server is incorporated into both Console Cogtalk and Windows Cogtalk. Either version of Cogtalk must be running on the host PC with the block server enabled before your vision processing application can use the PC's file system.

Running the Block Mode Server

For Console Cogtalk, start the block mode file server by specifying the **-s** or **--server** command line switches when you start Cogtalk.

For Windows Cogtalk, select **Options->Block Server**, or click the  button on the toolbar. In the **Block Server** dialog box, pseudo-serial port 3 is pre-selected. Click the **Enable** button to start the block server. The menu item changes to **Server Running** when the block server is enabled.

You can use the block server toolbar button even when no port windows are open. This allows Windows Cogtalk to serve as a standalone file server, with your own application communicating independently with the vision processor.

Note: You can start only one instance of the block server at a time, either from Windows Cogtalk or Console Cogtalk.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

For PVE 8.1, the block mode server requires the use of pseudo-serial port 3 for communicating between the host PC and the vision processor. If your application uses pseudo-serial port 3 for another purpose, you must change your application to use a different port if you want to use the block server at the same time.

Stopping the Block Mode Server

To stop the block mode server in Windows Cogtalk, select **Options->Block Server** (or click the  button on the toolbar), then click the **Disable** button.

To stop Console Cogtalk's block mode server, exit Cogtalk back to the NT command prompt, then restart Cogtalk without the **-s** or **--server** switch.

Using the Host PC's File System

Use the standard C run-time library I/O and Cognex file system (**cfs_***) functions to manipulate files on the host PC's file system. You must observe the following rules to manipulate files on the PC's file system:

- Your vision processing application *must* call **cfs_init()** before it attempts to use files on the PC's file system.
- To work with a file or directory on the PC's file system, you must precede the name of the file or directory with the string "*/dev/host/*".
- You must specify the full path name for all PC files and directories, including the drive letter.
- Only a specified subset of the standard I/O and Cognex file system functions are supported. The functions to set and get the current working directory are *not* supported. See *Supported Functions* on page 196.
- Your vision processing application must call **cfs_shutdown()** after accessing the PC's file system, to clean up and shut down the Cognex file system software.

These restrictions are discussed in the following sections.

Specifying Files and Directories

The Cognex file system (**cfs_***) functions were originally written to handle file I/O on SCSI disks directly attached to Cognex vision processors. To use these functions on the 5700/5900 to specify a file or directory on the host PC's file system, you must precede the file or directory name with the string "*/dev/host/*".

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The functions that set and get the current drive or working directory only work for files and directories on a directly-attached SCSI drive. You cannot specify or retrieve the current drive or working directory on the PC's file system. Whenever you specify a file or directory on the PC's file system, you must specify a full, absolute path that includes the "*/dev/host/*" string and a drive letter.

You can use either slash or backslash characters to separate directory names in the path name, but backslashes must be escaped with a preceding backslash as per C language syntax. The **cfs_*** functions support only DOS-like 8.3 format file and directory names, even when accessing an NTFS or extended FAT file system under Windows NT.

The following examples show valid ways to address files with the **cfs_*** functions:

- "*/dev/host/c:/cognex/working/test.txt*"
- "*/dev/host/c:\\temp\\image.dat*"
- "*/dev/host/c:\\progra~1\\cognex\\working\\testvol.vol*"

Supported Functions

The following standard C run-time library I/O functions are supported for accessing files on the PC's file system.

clearerr()
fclose()
feof()
ferror()
fflush()
fgetpos()
fopen()
fread()
fseek()
fsetpos()
ftell()
fwrite()
remove()
rename()
rewind()

The following Cognex file system functions are supported for accessing files on the PC's file system.

cfs_diskopen()
cfs_free_space()
cfs_gdone()
cfs_gfirst()
cfs_gnext()
cfs_init()
cfs_mkdir()
cfs_rmdir()
cfs_shutdown()
cfs_util_copy()

In all cases where the function takes a file or directory name as an argument, you must precede the file name with the string "*/dev/host/*" and you must specify a complete path name, including drive letter.

Functions that are not listed in this section are not supported for use with the block mode file server.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

File System Utility Functions

You must use the functions **cfs_init()** and **cfs_shutdown()** to initialize and terminate the file system software. See *Chapter 2, File System*, in the *System Software* manual for information on **cfs_init()**. The function **cfs_shutdown()** is described below.

Function cfs_shutdown()

cfs_shutdown() closes all open files on the host PC's file system that were opened with the block mode server. You must call this function before your application terminates. **cfs_shutdown()** returns 0 if the shutdown succeeded, or a nonzero value if it failed.

```
#include <cfs/cfs_defs.h>

c_Int32 cfs_shutdown(void);
```

Unsupported System Save and Load Functions

The following file system functions are not supported when using the block mode server with Cogtalk:

cfs_host_init_save()
cfs_host_reload()
cfs_host_wait_for_reboot()

Example Using the File Server Functions

This section contains an example of using the host PC's file system from a Cognex 5700/5900 application.

This example opens and reads the first hundred bytes of a file on the host's file system, and displays it at the V> prompt. The target file must exist and must contain text. For the example code to work, Cogtalk must be running with the block mode server enabled, as described in *Running the Block Mode Server* on page 194.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
#include <stdio.h>
#include <cfs\cfs_defs.h>

int open_and_read_file()
{
   char *buff = NULL;
   FILE *infile;
   int status, numbytes;

   status = cfs_init();
   printf ("\ncfs_init returned status %d\n\n", status);
   if (status)
      return (status);

   infile = fopen ("/dev/host/c:\\temp\\test.txt", "r");
   if (!infile)
   {
      printf ("Error opening file.\n");
      return (-1);
   }

   buff = cvc_alloc(100);
   numbytes = fread(buff, 1, 100, infile);
   printf ("%d bytes read from file.\n\n", numbytes);
   printf ("Here's what we read from the file:\n%s\n", buff);
   free (buff);

   status = fclose (infile);
   printf ("fclose returned status %d\n", status);
   status = cfs_shutdown();
   printf ("cfs_shutdown returned status %d\n", status);
   return 0;
}
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using Console Cogtalk

This section discusses issues specific to the console version of Cogtalk.

Starting Console Cogtalk

Console Cogtalk is intended to be used in Windows NT command sessions, and thus no icon is provided. Start Console Cogtalk by typing *cogtalk* at the prompt in a command session.

```
C:> cogtalk
[Cogtalk] Status: Running
V>
```

You can run Console Cogtalk from any directory, but Cogtalk's default include path presumes that it is run from your working subdirectory in the Cognex directory structure, usually *\cognex\working*. Console Cogtalk's default include path (*..\load;..\defs*) specifies relative path names, and Cogtalk expects to be run from a directory one level below the Cognex top-level directory. Thus, if you start Cogtalk from any other directory, be sure to specify a full include path with absolute path names, and use the *--nostdinc* switch to suppress inclusion of the standard directories. Specify an include path with the *-i* or *--include* command line switches, as discussed in the next section.

Command Line Switches

Console Cogtalk recognizes the following command line switches, which are listed in the same order shown by the *--help* switch. Single letter switches are prefixed by a single hyphen, but full word switches are prefixed with two hyphens.

-i or --include
: Append the specified directories to Cogtalk's include search path. This switch takes one argument, a colon- or semicolon-separated list of directory path names. If no *-i* or *--include* switch is specified, the following standard include path is presumed: *..\load;..\defs*

--stdinc (--nostdinc)
: Add (do not add) the standard directories to the include search path. By default, after any *-i* or *--include* switch has been processed, the directories *..\load* and *..\defs* are appended to the include search path. Use the *--nostdinc* switch if you run Cogtalk from a directory that's not one level below the Cognex top-level directory.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

--localinc (--nolocalinc)
For **#include** directives that specify a file name surrounded by quotes, search (do not search) the local directory before searching the include path. The meaning of *local directory* depends on the origin of the **#include** directive. If the **#include** directive is a line in a file being read by V (for example, #include "/red/green.h"), then the local directory is /red/. If the **#include** directive is typed at the V> prompt, then the local directory is the current working directory. Cogtalk's default condition is *--localinc*.

-e or --escape
This switch takes one argument, the character or character string to use as the prefix for the escape commands. The default is the tilde character (~). This switch sets the initial condition at Cogtalk's startup. You can change the escape character during a Cogtalk session with the *~escape* command, as discussed in *Change Escape Character (~escape)* on page 192.

-p or --port
Identifies the pseudo-serial port through which this Cogtalk session will communicate with the vision processor. Takes an integer argument between 0 and 3, inclusive. Port 0 is the default.

-q or --quiet (--noquiet)
Suppress (don't suppress) file-processing progress messages. The default condition is *--quiet*. This switch sets the initial condition at Cogtalk's startup. You can toggle the state of quiet mode during a Cogtalk session with the *~quiet* command, as discussed in *Toggle Quiet Mode (~quiet)* on page 193.

-h or --help
Display usage help for the command line switches.

-b or --boot (--noboot)
Reset (don't reset) the vision processor and return to the command prompt. The default condition is *--noboot*. You can use *cogtalk -b* in batch files to reset the vision processor. Use *-b* with *--noexit* to remain in Cogtalk after resetting the vision processor.

-r or --load
This switch takes one argument, the path name of a bootable volume or saved system file to load into the vision processor. If a system is currently running on the vision processor, Cogtalk first resets the vision processor, then loads the specified system file. If the vision processor is in Boot Polling mode, Cogtalk immediately loads the vision processor without resetting it. By default, the *-r* switch returns to the command prompt after loading the vision processor. Use *-r* with *--noexit* to reverse the default behavior.

-s or --server (--noserver)
Enable (disable) the block-mode file server. The default condition is *--noserver*.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

-l or --logfile
This switch takes one argument, the path name of a log file to open. The specified file is overwritten if it already exists. With this option specified, Cogtalk logs the commands you type and their responses during the current session in the specified file.

--exit (--noexit)
Use *--noexit* to keep Cogtalk running after processing a *-b* and/or *-r* option. By default, Cogtalk exits back to the command prompt after processing these options.

Command History and Editing

Console Cogtalk passes through to the vision processor the command history and command editing features of the underlying NT console session. This means that most of the command editing features available at an NT console prompt are available within Console Cogtalk. This includes the following features:

- Use the up and down arrow keys to recall commands typed in a Console Cogtalk session.
- Use *<Page Up>* and *<Page Down>* to jump to the first and last commands, respectively, in the command history.
- Use the left and right arrow keys, *<Ctrl+left>*, *<Ctrl+right>*, *<Home>*, *<End>* to move the cursor, and use the function keys *<F1>*, *<F2>*, *<F3>*, *<F4>*, and *<F5>* to edit the current command line.
- Use *<Del>* to remove the character at the cursor position. Use *<Backspace>* to remove the character to the left of the cursor position.
- Use *<Ctrl+left>* and *<Ctrl+right>* to move one word at a time in the current line. Word boundaries are defined for Console Cogtalk as character strings delineated by white space. Note that this is different behavior than in Windows Cogtalk, where words are delineated by any non-alphanumeric character.
- Use *<Esc>* to clear the current command line.
- Use *<F7>* to display the command history list.
- Use *<Alt+F7>* to clear the command history list.
- Type the first few characters of a recent command, then press *<F8>* successively to recall each recent command beginning with those characters.
- The Console Cogtalk command line is in insert mode by default. Press the *<Insert>* key to toggle between insert and overstrike modes.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

For a complete list of NT's console prompt editing features, consult the online help for NT, as follows:

1. Invoke **Start->Help**. This displays Windows NT Help.
2. In the **Contents** tab, double-click the book icon labeled **Windows NT Commands**.
3. With the book contents open, double-click the question-mark icon labeled **Windows NT Commands**.
4. In the next window, click the **D** button.
5. Click the button next to **doskey**.
6. Scroll to the bottom of the next window, and click the **Doskey—Notes** button.

The only feature of NT command editing that is *not* available within Console Cogtalk is the Doskey macro feature. You cannot run a Doskey macro within Cogtalk.

Note the following additional features of Console Cogtalk's command history feature:

- In any one NT command session, the history of commands typed within Console Cogtalk survive repeated invocations of Cogtalk. That is, let's say you enter Cogtalk, type several V commands, then exit. At the NT console prompt, you type several NT commands, then re-enter Console Cogtalk. The history of the V commands you typed in the first Cogtalk session is still available.
- When you exit the current NT console session, the Cogtalk command history is lost. There is no way to save your Cogtalk command history to a file, but you can log the entire session with the *--logfile* switch.
- NT console sessions keep track of separate command histories for display with the *<F7>* key. While in Cogtalk, *<F7>* shows you the commands you typed in the current Cogtalk session. When you exit back to the NT console prompt, *<F7>* shows the history of NT commands you typed at the console prompt. You can go back and forth into Cogtalk, and the history lists remain separate.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using Windows Cogtalk

This section discusses issues specific to the Windows version of Cogtalk.

Starting Windows Cogtalk

Windows Cogtalk is installed as an entry in the Cognex Vision Processor program group in the Windows NT start menu. To start Cogtalk, invoke **Start->Programs->Cognex Vision Processor->Cogtalk**.

As installed, the shortcut that starts Windows Cogtalk specifies *<cogroot>\working* as Cogtalk's initial working directory, where *<cogroot>* is the top-level Cognex directory specified during installation. Edit the shortcut's properties to specify another initial working directory, as described in *File->Working Directory* on page 212.

Figure 46 shows the components of a typical Cogtalk window.

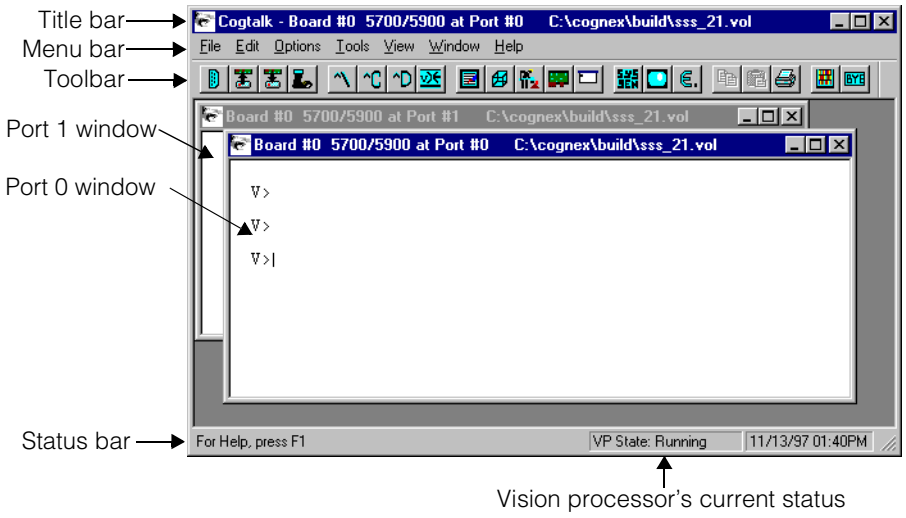


Figure 46. Components of the Cogtalk window

Each port window's title bar shows the current board, pseudo-serial port, and loaded system file. Cogtalk's title bar reflects the active port window's title bar.

When you right-click in any port window, Cogtalk displays a duplicate of the **File** menu as a pop-up menu.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Retained Settings Between Invocations

Windows Cogtalk retains the following settings between invocations, and uses them at its next startup:

- Window size
- Window position
- Include path
- The path names of the programs you can start from the **Tools** menu
- Any port windows open at shutdown are reopened at the next startup, if the vision processor is still running. If the vision processor is reset between invocations of Windows Cogtalk, Cogtalk starts up without port windows.
- Information about the current session, such as the vision processor's board type and the volume or saved system file currently loaded.

Windows Cogtalk maintains a settings file, *cogtalkw.ini*, in the NT system directory (usually *c:\winnt*), which records the information kept between invocations.

Communication in Port Windows

Windows Cogtalk allows you to open multiple windows for communication with the vision processor. Each window must communicate over a separate pseudo-serial port, which must not be already in use by Cogtalk or any other application.

A window on port 0 normally is used to interact with V, or with your alternate command processor. V communicates by default over pseudo-serial port 0, so a port window opened on that port normally shows the V> prompt. Until you load a bootable volume image file to the vision processor, the Windows Cogtalk communication window is blank. If you start Windows Cogtalk when the vision processor is already booted and running, it opens a port 0 window automatically.

When resetting, loading, or saving the vision processor, the port 0 window must be active. When resetting or loading the vision processor, *only* the port 0 window can be active; windows on other ports must be closed first.

Your vision processing application can make use of other port windows as required. For example, you can use a window on pseudo-serial port 1 to accept messages from your debugging routines. To do so, open pseudo-serial port 1 from V and write to it as a file. The following V code fragment shows how to write a simple message to a port 1 window.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
FILE *port1;
port1 = fopen ("/dev/bus1", "w");
setvbuf (port1, NULL, _IOLBF, 0);
fprintf (port1, "Debugging message 1.\n");
fprintf (port1, "Debugging message 2.\n");
fclose (port1);
```

Windows Cogtalk's preprocessor runs equally on all port windows. This allows you to redirect V to an alternate port window, and still include files with full **#include** preprocessing. Note that **#include** preprocessing occurs even if you're running your own alternate command processor to replace V. In this case, use **~send** to send files to your command processor instead of using the **#include** directive.

Open a port window with **File->Open Port**, with the  toolbar button, or with the *<Ctrl+O>* keyboard shortcut. These actions invoke the **Vision Processor Selection** dialog box, as shown in Figure 47 on page 209. When selecting a pseudo-serial port for a Cogtalk port window, remember that Cogview uses pseudo-serial port 2 by default for sending the PC's mouse data to the vision processor, and remember that Cogtalk uses pseudo-serial port 3 for the block mode file server. If you are using these features, plan your port usage accordingly.

Port Window Display History

Each port window maintains an effectively unlimited display history. The following commands and keystrokes show you different parts of the active window's display history. Note that the display history feature is separate from the command editing and recall feature discussed in the next section.

- *<Page Up>* and *<Page Down>* move up and down one screen at a time in the display history.
- *<Ctrl+Home>* places the cursor at the top of the window's display history.
- *<Ctrl+End>* places the cursor at the end of the current command line.
- *<Ctrl+Page Up>* places the cursor at the top left corner of the active port window.
- *<Ctrl+Page Down>* places the cursor at the bottom left corner of the active port window.

You can clear a port window's display history with **Edit->Clear Screen**, or by clicking the  toolbar button. Note that clearing a window's display history does not affect its command history, which remains in place.

Very large display history buffers (over 200,000 characters) can slow down response time on the current command line. If you run a V command that produces a great deal of output, and thus fills the display history, clear the display history to restore response time.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Command History and Editing

Port windows in Windows Cogtalk maintain a history of commands typed, and allow you to recall, edit, and re-run those commands. A window's command history is maintained until that window is closed, and is not affected by resetting or reloading the vision processor. Each port window maintains a separate command history.

Command history and editing work well with V, but do not presume that V is the command processor. For each new command line, the cursor is placed after any text returned from the vision processor. In the case of V, the text returned is the V> prompt. In the case of your alternate command processor, the text returned is your command processor's prompt.

Cogtalk's command editing mechanism borrows from the Windows rich text editing standards. Use the same editing keystrokes you would use in Wordpad. The following points summarize Windows Cogtalk's command recall and command editing features.

- To recall previous commands, use the up and down arrow keys. Unlike Console Cogtalk, there are no other mechanisms for recalling previous commands.
- The command history is maintained as a simple list, with no wrapping between top and bottom. If you go back to the first command typed in the current session, the up arrow does not proceed farther. Likewise, the down arrow does not proceed farther down the command history list than the most recent command typed.
- For any recalled command, you can edit the command, then press *<Enter>* to send the edited command to the vision processor. The cursor can be anywhere in the current line, or, in fact, anywhere in the active port window, when you press *<Enter>*.
- Type *<Ctrl+Break>* to clear the current line back to the prompt. You can also press *<Keypad 5>* to clear the current line, if your keyboard's Num Lock feature is toggled off.
- If the cursor is not in the current command line, it is automatically placed at the end of the current command line in two cases:
  - When you type any printable character
  - When you paste from the Windows clipboard

  This feature allows you to scroll back in the port window's display history, copy text to the clipboard, then immediately paste without having to scroll back down first. The pasted text is always placed at the end of the current command line.
- Use the left and right arrow keys to move the cursor within the current command line.
- Use *<Home>* to move to the beginning of the current command line; use *<End>* to move to the end of the current line.
- Use *<Del>* to remove the character at the cursor position. Use *<Backspace>* to remove the character to the left of the cursor position.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- Insert mode is on by default for the current command line. Press the *<Insert>* key to toggle between overstrike and insert modes.
- Use *<Ctrl+left>* and *<Ctrl+right>* to move one word at a time in the current line. Word boundaries are defined for Windows Cogtalk as character strings delineated by any non-alphanumeric character. Note that this is different behavior than in Console Cogtalk, where words are delineated by white space.
- Highlight text for copying to the clipboard by holding down either *<Shift>* key while moving the cursor. For example, *<Shift+right>* highlights one character at a time, while *<Shift+Ctrl+right>* highlights one word at a time.
- You cannot *cut* text from the display history, but you can *copy* text from the display history, or from any other application.
- When pasting to the current command line, the pasted text is placed according to the cursor's current position. If the cursor is in the current command line, the text is pasted at the cursor's position. If the cursor is anywhere else in the port window, the text is pasted at the end of the current command line.
- White space and carriage returns are stripped off the end of text pasted from the clipboard. This means that pasted text will never be inadvertently sent to the vision processor. You must deliberately press *<Enter>* to send your edited command line.

Using Session Logs

Windows Cogtalk supports the logging of all commands typed in a port window and their responses from the vision processor. Enable session logging for the active port window by using **File->Create Session Logfile**, or by clicking the  toolbar button. When the session log is enabled, this menu item changes to **File->Close Session Logfile** and a checkmark displays to the left of the item. Select this item, or re-click the  toolbar button to close the log file.

The session log file is opened in Cogtalk's current working directory, and is always named *Cogtalk0.log*. If this file exists, new log entries are appended to the existing file. (Note that this behavior is different than in Console Cogtalk, where specifying an existing log file name always overwrites the previous file of that name.) If you change Windows Cogtalk's current working directory while the session log is open, the log file continues to be written to the original file in the original working directory, and a log entry is entered showing the directory change.

The session log is enabled separately for each port window, but log entries for all ports are intermixed in the same file. For this reason, it is recommended that you enable logging for one port window at a time.

When a session log is opened, a timestamp entry is written to the log, much like the following example.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
Cogtalk Session - Fri Nov 12 11:08:50 1999
VP Board - 5700/5900
System File Loaded - c:\cognex\working\sss_21.vol
```

When logging an interactive session with V, a log entry is generated for the following events:

- Every V command you type
- The response to each V command from the vision processor
- Every escape command you type, and the response from Cogtalk and/or from the vision processor
- Every *<Ctrl+C>, <Ctrl+D>, or <Ctrl+\>* you generate, whether from the toolbar buttons, the menu items on the **Options** menu, or typed at the prompt as an escape command
- Application-level events such as changing the working directory

You can also log port windows other than the one communicating with V. For example, let's say your application uses a window on pseudo-serial port 1 to display debugging and error messages, as described in *Communication in Port Windows* on page 204. If you enable session logging for port 1 only, you effectively generate a log of error messages from your application.

If session logging is enabled when you enable the block mode file server, a second log file named *bserver0.log* is opened that records the start and end times of the block server, and any error messages from the block server.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Windows Cogtalk Menu and Toolbar Reference

This section describes, in order, each menu item and toolbar icon in the Cogtalk interface.

File Menu

Use items on the **File** menu to control the vision processor, enable session logging, change working directories, send files to the vision processor, or print the contents of the active port window. When you right-click in any port window, Cogtalk displays a duplicate of the **File** menu as a pop-up menu.

File->Open Port

This menu item opens the **Vision Processor Selection** dialog box, as shown in Figure 47.

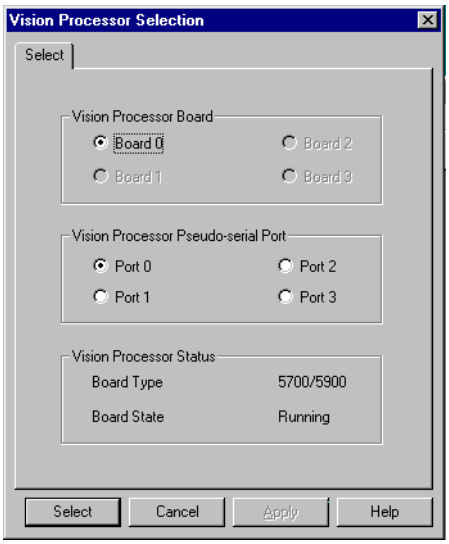


Figure 47. Vision Processor Selection dialog box

This dialog always opens with Board 0 and Port 0 preselected. Click the **Select** button, or press the *<Enter>* key to open a port window on pseudo-serial port 0.

PVE revision 8.1 supports one vision processor per host PC, so board selections other than the first are greyed out. Select a pseudo-serial port and click **Select** or press *<Enter>* to open a window on that port. Before opening the window, Cogtalk queries the vision processor device driver to determine if that port is in use by another application. If so, Cogtalk displays an error message.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The status section shows the status of the currently selected vision processor board. If the vision processor is not running a system file, you can only select pseudo-serial port 0.

File->Load System

The port 0 window must be active and must be the only window open when loading a system file. This menu item first checks whether the active port window is port 0, and displays a reminder message if not. It then checks whether other port windows are open; if so, it displays a reminder to close all other port windows. It then opens the **Load Executable Image** dialog box, as shown in Figure 48.

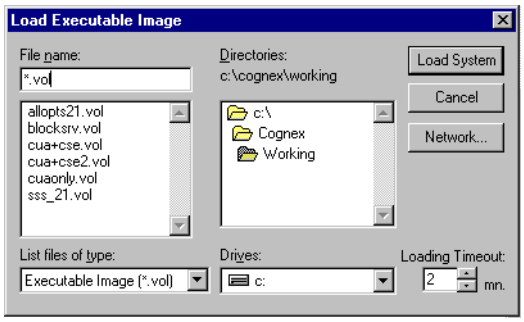


Figure 48. Load Executable Image dialog box

Use this dialog to select a volume image file (traditionally with a *.vol* extension) or saved system file (traditionally with a *.sav* extension). The directory shown when this dialog box opens is Cogtalk's current working directory.

When you've selected a vision processor image file, click the **Load System** button (or press the *<Enter>* key) to begin the system load operation.

Use the **Loading Timeout** field to specify a time in minutes that Cogtalk will wait before determining that the system load operation has failed. The default is 2 minutes.

File->Save System

Use this item to save an image of the running system on the vision processor to a bootable saved system file. This operation proceeds in two steps:

1. Specify the command that starts the operation on the vision processor.
2. Specify the name of the PC file to contain the saved system image.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The port 0 window must be active when saving a system file, but other port windows can be open. This menu item first checks whether the active port window is port 0, and displays a reminder message if not. If so, it opens the **Save Options** dialog box, as shown in Figure 49.

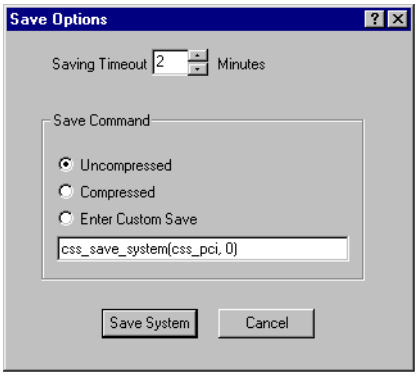


Figure 49. Save Options dialog box

Use this dialog to specify the command on the vision processor that begins the save system operation. When you select the **Uncompressed** or **Compressed** buttons, you tell Cogtalk to use one of two forms of the **css_save_system()** function. Select the **Enter Custom Save** button to specify the name of your own system-saving function, which must contain **css_save_system()**.

Use the **Saving Timeout** field to specify a time in minutes that Cogtalk will wait before determining that the system save operation has failed. The default is 2 minutes.

When you click the **Save System** button, Cogtalk displays the **Save As** dialog box, as shown in Figure 50.

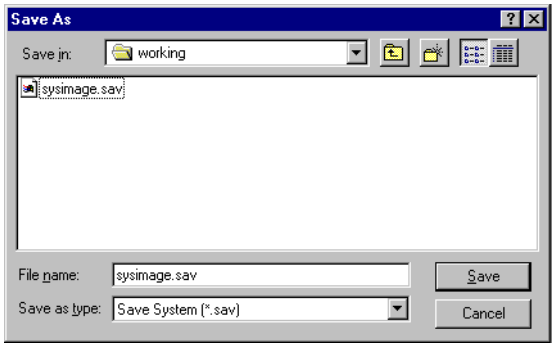


Figure 50. Save As dialog box

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Use this dialog to specify the path and name of the PC file to contain the saved system image.

File->Create Session Logfile

This menu item enables the logging to a file of all commands typed, and their responses, for the active port window. When the session log is enabled, this menu item changes to **Close Session Logfile**, and a checkmark displays to the left of the item. The log file is written to Cogtalk's current working directory, and is always named *Cogtalk0.log*. If this file exists, new log entries are appended. If the block mode server is enabled while session logging is enabled, a separate log file named *bserver0.log* is generated. See *Using Session Logs* on page 207 for more information.

File->Working Directory

This item opens the **Current Working Directory** dialog box, as shown in Figure 51.

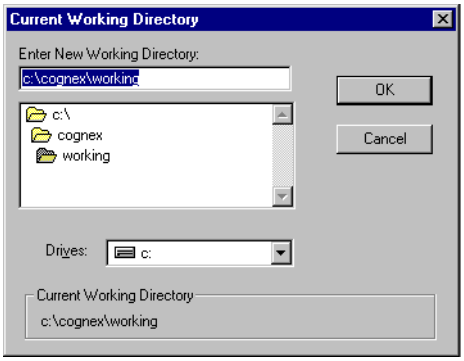


Figure 51. Current Working Directory dialog box

Use this dialog to specify a new working directory for Cogtalk. The current working directory can also be specified by typing **#cwd** or **~cwd** in a port window.

Specify Cogtalk's *initial* working directory by editing the properties of the shortcut that starts Cogtalk. To do so, right-click the **Start** menu and select the **Explore All Users** menu item (this item only displays if you are logged in with NT administrator rights). This opens an Explorer window with **All Users->Start Menu->Programs** visible. Open the **Programs** folder and locate the **Cognex Vision Processor** folder. In that folder, right-click the Cogtalk icon, then select **Properties** from the pop-up menu. Select the **Shortcut** tab, then type an initial working directory in the **Start in** field.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

File->Include File

This item opens the **Choose File to #Include** dialog box, as shown in Figure 52.

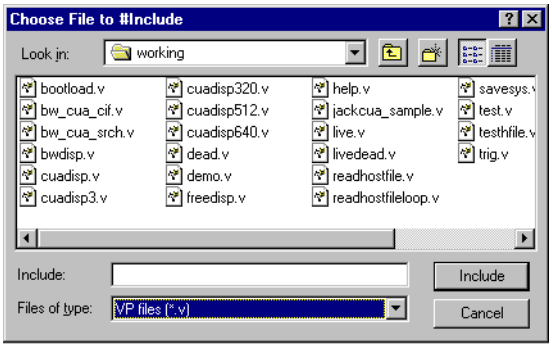


Figure 52. Choose File to #Include dialog box

Use this menu item as an alternative to typing an **#include** directive at the V> prompt. This dialog box initially shows all header files with the *.h* extension in Cogtalk's current working directory. Use the **Files of type** field to show all *.v* files, or all files of any type.

Select a file from the ones showing, or browse the PC's file system for a file. When selected, click the **Include** button or press the *<Enter>* key. The file is passed through Cogtalk's **#include** preprocessor, and then on to the vision processor through the pseudo-serial port associated with the active port window.

Files included by this menu item are normally included silently. Set quiet mode off with the **~quiet** command to watch the include command proceed.

File->Print

When you select **File->Print**, Cogtalk immediately sends the contents of the active port window to the Windows default printer.

When you use the *<Ctrl+P>* keyboard shortcut for this item, Cogtalk opens the standard Windows **Print** dialog box, which allows you to select a different printer or different print properties.

File->Print Setup

This item opens the standard Windows **Print Setup** dialog box, which allows you to change the default printer and print properties for subsequent printing from this session of Cogtalk.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

File->Exit

Use **File->Exit** to quit the current Cogtalk session.

Edit Menu

Use items on the **Edit** menu to clear the current line or port window, to copy and paste text, to locate text in the active port window, and to change the font in the active port window.

Edit->Clear Screen

Use this menu item to clear the display history for the active port window. This item does not affect the window's command history.

Edit->Clear Line

Use this menu item to clear the current line in the active port window back to the prompt. Use *<Ctrl+Break>* to perform the same action. You can also use the *<5>* key on the numeric keypad if the keyboard's Num Lock feature is toggled off.

Edit->Copy

Use this menu item to copy the highlighted text in the active port window to the Windows clipboard.

Edit->Paste

Use this menu item to paste text from the Windows clipboard to the active port window. The text is pasted according to the position of the cursor. If the cursor is in the current command line, the text is pasted at the cursor position. If the text is anywhere else in the active port window, the text is pasted at the end of the current command line.

You can scroll back in the display history, select text, and immediately paste it to the current command line without having to scroll back down in the window.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Edit->Find

This menu item opens the standard Windows **Find** dialog box, as shown in Figure 53.

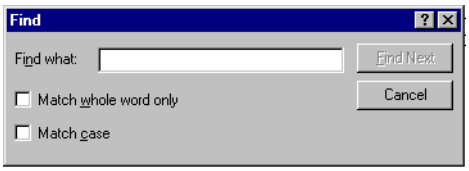


Figure 53. Find dialog box

Use this feature to locate text in the display history of the active port window.

Edit->Font

This menu item opens the standard Windows **Font** dialog box, as shown in Figure 54.

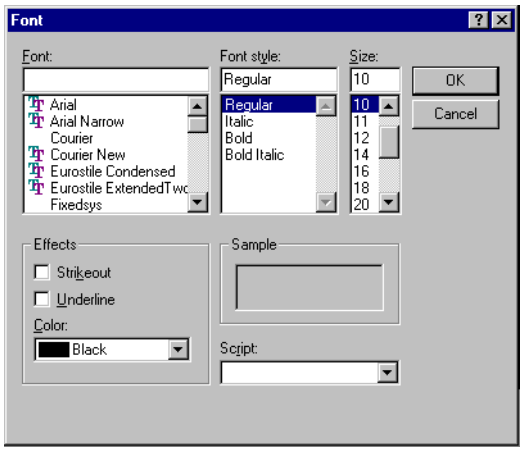


Figure 54. Font dialog box

Use this feature to change the font used in the active port window.

Options Menu

Use items in the **Options** menu to enable certain Cogtalk features.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Options->Compile Region

This menu item toggles Cogtalk's support for the regional compile feature of Cognex-extended Epsilon. The Epsilon text editor is included with the PVE software. You can install the Cognex Epsilon extensions, by following the instructions in the *5000 Getting Started* manual, and *Chapter 7, The Cognex Epsilon Enhancements*, of the *Development Environment 1* manual.

Cognex-extended Epsilon supports highlighting a region of text in an Epsilon buffer (presumably a region containing a compilable unit of V code), and sending that region to the vision processor with the help of Vtalk, Wintalk, or PCVtalk. Cogtalk also supports this feature.

When you select the **Compile Region** menu item, the item changes to **Compile Region Enabled** with a checkmark to the left. While this menu item is enabled, text selected and tagged for regional compilation with the Epsilon *compile-region* command is sent to the vision processor by Cogtalk as if you had used **#include** to send that code region to the vision processor.

Options->Include Path

This menu item opens the **Cogtalk Include Path** dialog box, as shown in Figure 55.

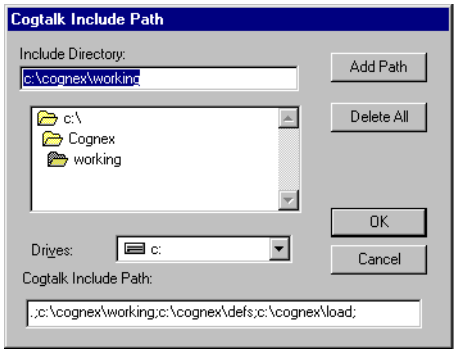


Figure 55. Cogtalk Include Path dialog box

Use this dialog to specify Cogtalk's include path, which is the list of directories searched for files specified to be included by **#include** at the V> prompt. You can type a full include path in the **Cogtalk Include Path** field, or edit the path already in this field. As an alternative, you can browse for individual directories, then click the **Add Path** button to add the currently selected directory to the end of the include path.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The **Delete All** button clears the current include path, except for the initial period-semicolon. This is left as a reminder that Windows Cogtalk always searches its current working directory first in the include path. You can backspace over and remove the period-semicolon, but Cogtalk still keeps the current directory first in its internal include path.

Options->Block Server

This menu item opens the **Block Server** dialog box, as shown in Figure 56. The menu item changes to **Server Running** when the block server is enabled.

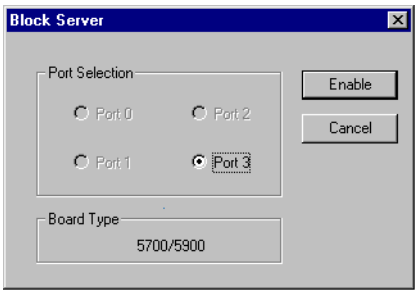


Figure 56. Block Server dialog box

In the current implementation of Cogtalk, the block mode file server always uses pseudo-serial port 3 for part of its communication protocol with the vision processor. Thus, the other pseudo-serial port selections are greyed out.

To enable the block mode file server, click the **Enable** button. The text in this button changes to **Disable** when the file server is running. Click the **Disable** button to stop the file server.

The block mode file server is discussed in more detail in *Cogtalk as File Server* on page 194.

Options->Reset VP

The port 0 window must be active and must be the only window open when resetting the vision processor. This menu item first checks whether the active port window is port 0, and displays a reminder message if not. It then checks whether other port windows are open; if so, it displays a reminder to close all other port windows.

If invoked on an active port 0 window with no other port windows open, this menu item prompts for confirmation, then resets the vision processor.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Options->Send ^\

Use this menu item to abort the current operation and send the Quit character, Ctrl+\ (ASCII 28), to the vision processor. This sends the signal *CIC_ERR_QUIT* to V, which causes V to clean up and pass the signal to its invoker. Use this command to halt a vision processing program stuck in an infinite loop or stuck waiting for input.

Options->Send ^C

Use this menu item to abort the current operation and send the Ctrl+C character (ASCII 3) to the vision processor, which sends the signal *CIC_ERR_INT* to V. This signal suspends execution of the current instance of V. Typically, this causes a stack trace, and starts a new instance of V.

Options->Send ^D

Use this menu item to send the End-of-File character, Ctrl+D (ASCII 4), to the vision processor, which causes the V compiler to exit its current instance. If the current instance of V is recursive (if the prompt is, for example, *V2>*), that instance exits and leaves the calling instance open (the prompt returns to *V1>*).

Tools Menu

Use items on the **Tools** menu to start applications related to Cogtalk.

- **Tools->Cogview** starts the Cogview utility, as described in *Using Cogview* on page 225.
- **Tools->Sysgen** opens an NT command session, and invokes *sysgen -i* in Cogtalk's current working directory. The Sysgen utility is used to build volume image files for the vision processor, and is described in *Chapter 6, Using Sysgen* of the *Development Environment 1* manual.
- **Tools->Epsilon** opens the Epsilon text editor, as published by Lugaru Software. A copy of Epsilon is included with the PVE software. For more on using the Cognex Epsilon extensions, see the *5000 Getting Started* manual, and *Chapter 7, The Cognex Epsilon Enhancements*, of the *Development Environment 1* manual.

The items on this menu attempt to start these programs as installed in their default locations. If an item can't find its program in its default location, Cogtalk prompts you to specify the new location. If necessary, you can adjust the path names of the programs started by the **Tools** menu by editing the *cogtalkw.ini* file located in the NT system directory (usually *c:\winnt*).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

View Menu

Select items in the **View** menu to change the way Cogtalk looks or to display the vision processor's current status.

View->Board Status

This menu item displays the **Board Status** message window, as shown in Figure 57.



Figure 57. Board Status message window

The message window shows information returned by the vision processor device driver about the current hardware and software status of the vision processor, and shows the state of Cogtalk's block mode file server.

View->Toolbar

The **Toolbar** menu item toggles the display of Cogtalk's toolbar.

View->Status Bar

The **Status Bar** menu item toggles the display of Cogtalk's status bar.

Window Menu

The Window menu allows you to change the way port windows appear within Cogtalk's main window.

Window->Cascade

Use this item to stack all port windows in sequence, overlapping by the height of their title bars.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Window->Tile Horizontally

Use this item to fill Cogtalk's main window with all port windows drawn with equal lengths. The active port window is moved to the top.

Window->Tile Vertically

Use this item to fill Cogtalk's main window with all port windows drawn with equal heights. The active port window is moved to the leftmost position.

Help Menu

Select the **Help** menu to display online help for Cogtalk.

Help->Cogtalk Help

The **Cogtalk Help** menu item displays the contents page of the Cogtalk online help file. If Cogtalk cannot find the help file, it prompts you to specify its location. In this case, search for the file *cogtalkw.hlp*, normally installed in *\cognex\pchost\ntbin*.

Help->Cognex API Reference

This menu item opens the Windows help file containing reference pages for each function and data type in the Cognex vision processing API. This reference is complete as of PVE revision 7.6; APIs added since that release are documented in this manual and in the Release Notes for the current PVE revision.

If Cogtalk cannot find the API Reference help file, it prompts you to specify the location of the file. Search for *prog_ref.hlp*, normally installed in *\cognex\ntbin*.

Help->About Cogtalk

The **About Cogtalk** menu item shows the product information box, containing copyright and version information for Cogtalk.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Toolbar Buttons

Use the buttons in Cogtalk's toolbar as single-click shortcuts for certain functions.

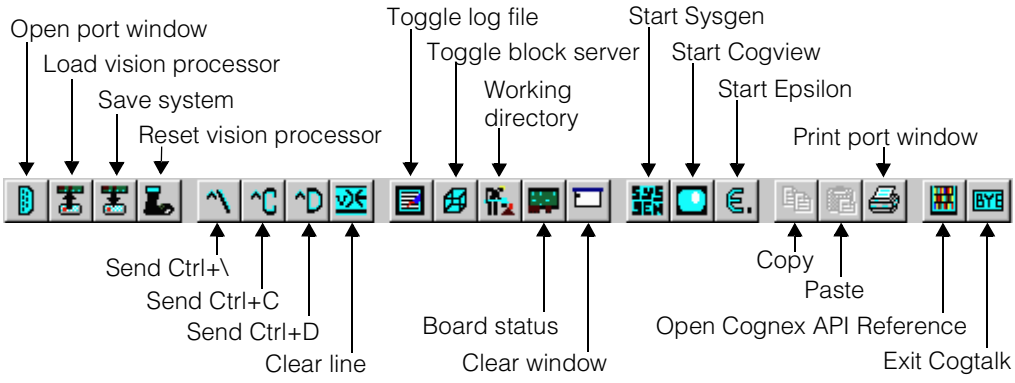


Figure 58. Toolbar in Windows Cogtalk

The functions invoked by Cogtalk's toolbar buttons are shown in Table 31.

Button	Button Name	Function
	Open port window	Opens the Vision Processor Selection dialog box, with which you open a port window for communication with the vision processor over the specified pseudo-serial port
	Load vision processor	Opens the Load Executable Image dialog box which you use to load a volume or saved system file to the vision processor. The active port window must be the port 0 window and no other port windows can be open.
	Save system	Opens the Save Options dialog box which lets you specify a system-saving command to send to the vision processor. The active port window must be the port 0 window.
	Reset vision processor	Resets the vision processor. The active port window must be the port 0 window and no other port windows can be open.

Table 31. Cogtalk's toolbar buttons

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Button	Button Name	Function
	Send Ctrl+\	Aborts the current operation and sends *<Ctrl+\>* to the vision processor
	Send Ctrl+C	Aborts the current operation and sends *<Ctrl+C>* to the vision processor
	Send Ctrl+D	Sends *<Ctrl+D>* to the vision processor
	Clear line	Clears the current command line back to the prompt
	Toggle log file	Toggles on and off the logging of the active window's entries to *Cogtalk0.log* in the current working directory
	Toggle block server	Toggles on and off the block mode file server
	Working directory	Opens the Current Working Directory dialog box in which you change Cogtalk's current directory
	Board status	Displays the Board Status message window
	Clear window	Clears the display history for the active port window
	Start Sysgen	Starts an NT command session running *sysgen -i* in Cogtalk's current working directory
	Start Cogview	Starts the Cogview application
	Start Epsilon	Starts the Epsilon editor
	Copy	Copies the currently selected text to the Windows clipboard
	Paste	Pastes text from the Windows clipboard
	Print	Prints the contents of the active window's display history
	Open Cognex API Reference	Opens the Cognex API Reference help file
	Exit Cogtalk	Exits the current session of Cogtalk

Table 31. Cogtalk's toolbar buttons

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Windows Cogtalk Keyboard Shortcuts

Table 32 lists the keyboard shortcuts available when running Windows Cogtalk.

Keyboard shortcut	Action performed
Ctrl + Break	Clears the current line in the active port window back to the prompt. *<Keypad 5>* (with Num Lock off) is an alias for this operation.
Ctrl + C	Copies the current text selection to the Windows clipboard.
Ctrl + F	Invokes the Find dialog box, to locate specified text within the display history of the active port window.
Ctrl + L	Invokes the Load Executable Image dialog box to load a volume or saved system file to the vision processor. The active port window must be the port 0 window and no other port windows can be open.
Ctrl + O	Invokes the Open Port dialog box, which opens a port window for communication with the vision processor over a specified pseudo-serial port.
Ctrl + P	Invokes the Print dialog box, allowing you to print the contents of the active port window.
Ctrl + S	Invokes the Save Options dialog box, which you use to save an image of the running vision processor system to a file on the PC. The port 0 window must be active.
Ctrl + V	Pastes text from the Windows clipboard at the current cursor position in the command line, or at the end of the command line if the cursor is elsewhere in the window.

Table 32. Cogtalk keyboard shortcuts

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9 Using Cogtalk

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter

10

Using Cogview

The Cognex 5000-ISA vision processors, including models 5200, 5400, and 5600, are equipped with a hardware monitor port to which you can connect an RGB monitor to display vision processor images. The 5000-ISA boards can also be used with one of the Cognex Video Mixer boards and a Cognex utility named Vmix to display vision processor images in a window on the host PC's monitor.

By contrast, the Cognex 5700/5900 vision processors do not include a physical monitor port and cannot be used with a Cognex Video Mixer board. Instead, Cognex supplies the Software Display library, which allows the construction of applications that can display images acquired by the 5700/5900 on the host PC's monitor.

Cogview is a Windows NT utility, built with the Software Display library, that serves two purposes:

- Cogview lets you display the video output from the Cognex 5700/5900 in a window on your host PC's monitor.
- Cogview's display area can be toggled to serve as a mouse drawing space when using the host PC's mouse in place of the Cognex trackball.

This chapter is divided into the following sections:

- *Cogview Usage Notes* presents an overview of using the Cogview utility.
- *Cogview Menu and Toolbar Reference* explains the use of each menu item and icon in the Cogview user interface.
- *Cogview Keyboard Shortcuts* lists the shortcuts available when running Cogview.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Cogview Usage Notes

This section provides overview information on using the Cogview utility.

Cogview Requirements

To run Cogview, the host PC must meet these requirements:

- The PC must be running Windows NT, version 4.0 or later.
- The PC must have a Cognex 5700/5900 and its device driver installed.
- The Cognex-supplied DLLs *csd.dll* and *cvideo.dll* must be in NT's DLL search path.
- The NT desktop must be in a 256-color mode.
- To display full resolution images (512x480 or larger), the NT desktop must be at least 800x600 pixels.
- A mouse or other pointing device is recommended, and is required for certain features.

To run Cogview at its maximum frame rate, the host PC must meet the following additional requirements:

- The PC's PCI chipset must be compliant with the PCI 2.1 specification.
- The PC's display driver must support Microsoft's DirectX specifications, version 2.0 or later.

These additional requirements are discussed in *Requirements for Maximum Frame Rate* on page 20.

Starting Cogview

The Cogview utility is installed as an entry in the Cognex Vision Processor program group in the Windows NT start menu. To start Cogview, invoke **Start->Programs->Cognex Vision Processor->Cogview.**

On startup, if Cogview detects a display mode with a color depth other than 256 colors, it displays a message, then exits.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

On startup, Cogview's software display subsystem queries the host PC's PCI chipset for compliance with the PCI 2.1 specification, and queries the PC's display driver for compliance with the DirectX 2.0+ specification. Under the following conditions, Cogview displays a reminder that image display performance will be slower than running Cogview on a compliant PC:

- If Cogview detects a non-compliant chipset.
- If Cogview detects a non-compliant display driver.
- If Cogview can't find its chipset database in the NT registry.

Click the **Yes** button in this message box to continue starting Cogview.

Note: On compliant host PCs, Cogview requests image transfer directly from the vision processor to the PC's video card, using the PCI bus's DMA transfer mechanism. On non-compliant host PCs, images are transferred first to system memory, and then to the video card's display memory. See *Performance Factors* on page 164 and *Non-Compliant Chip Set List in Registry* on page 166 for further information.

When running Cogview with system memory transfers, the NT Task Manager will show a higher host CPU usage than when running Cogview on a PC that supports video memory transfers. The higher host CPU usage is due to the overhead of writing image data from system memory to the display adapter.

When Cogview is started, the image shown in its display area depends on the state of the vision processor. If live display mode has been requested on the vision processor (for example, by invoking **cua_live_set()** from the V> prompt in Cogtalk), Cogview displays the live video image from the camera attached to the vision processor. Otherwise, Cogview starts up with a blank display area.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 59 shows the components of the Cogview window.

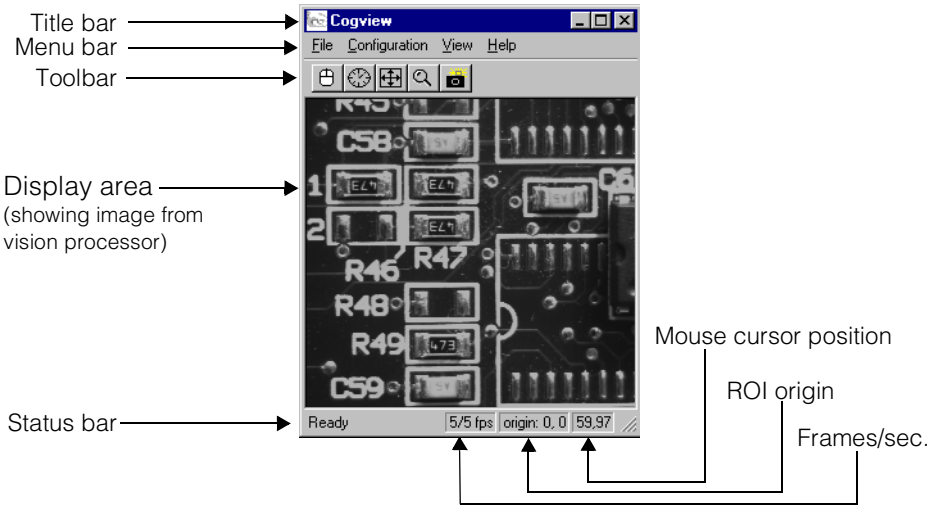


Figure 59. Components of the Cogview window

Cogview's status bar shows the following information:

- Frames per second shows the actual and requested rate in frames per second at which Cogview retrieves images from the vision processor. The actual frame rate may be lower than the requested rate on a non-compliant host PC, as discussed in *Cogview Requirements* on page 226.
- Region of interest origin shows the coordinates of the upper left corner of the current region of interest, relative to the origin of the current image. You specify the region of interest by means of the **Video Format** tab of the **Cogview Configuration** dialog box (as shown in Figure 65 on page 238).
- Mouse cursor position shows the coordinates of the mouse cursor within the Cogview window, or the last position of the mouse as it exited the Cogview window. Mouse position is reported whether or not mouse support is enabled.

Retained Settings Between Invocations

On exit, Cogview retains the following settings and uses them at its next startup:

- Window size
- Window position
- Current frame rate

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- Current state of PC mouse support (enabled or disabled)
- Current video format selection
- Current region of interest origin and dimensions
- Current palette table file

On the other hand, Cogview always starts up in the following modes, independent of the last session's settings on exit.

- Autosize mode is off.
- Freeze mode is off.
- Zoom mode is off.

Window and Image Sizes

Cogview supports several video format selections, specified with the **Video Format** tab of the **Cogview Configuration** dialog box (as shown in Figure 65 on page 238). For best results, select the video format size that matches the current video mode specified on your vision processor.

For example, let's say you specify an acquisition video format at the V> prompt, or in your vision processing application, using a line like the following:

```
const cua_video_format fmt = cua_xc75_640x480;
```

In this case, select 640x480 as the video format in Cogview.

When you specify a video format in Cogview, you are actually specifying a maximum image size for Cogview's display area. The format for the acquired image is specified on the vision processor with functions from the Unified Acquisition software. Table 28 on page 107 lists the available video formats. (See *Video Formats* on page 107 for further information.)

Best results are obtained when you match the video format (image size) in Cogview to the format specified on the vision processor, but you are free to select any Cogview video format. If you specify a larger image size in Cogview than on the vision processor, then images from the vision processor are displayed in the upper left corner of Cogview's display area, with the remaining display area blacked out. If you specify a smaller image size than vision processor video format, Cogview displays the cropped upper left corner of the vision processor's images.

When you select a video format (image size) smaller than the current one, the Cogview window resizes itself to match. However, when you select a larger image size than is currently displayed, Cogview draws scroll bars, but retains its current window size. You can then resize the window manually. If you resize a window to its maximum size for the current image size, the scroll bars are eliminated.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using Half Resolution Video Formats

The following video formats from Table 28 on page 107 are half resolution formats:

- *cua_xc75_320x240*
- *cua_xc75_256x240*
- *cua_xc75ce_380x287*
- *cua_xc75ce_256x256*

Full resolution formats display the odd and even scan fields from the video camera interlaced together to form a single image. By contrast, the display for half resolution formats is composed by alternating, for each line, between the odd and even scan fields. When Cogview displays live video images at half-resolution, it passes the image unaltered from the vision processor. The rapid alternation for each line can appear as a flickering or a periodic vertical “crawl” of the image.

The video formats for the rapid reset cameras (the Sony XC-75 RR and XC-75CE RR) are considered half-resolution, but live video mode is not supported for rapid reset cameras.

Specifying a Region of Interest

Cogview allows you to specify a region of interest within the displayed image, by specifying the *x* and *y* positions of the region’s origin and its width and height. Once the region is specified, its origin is fixed in the upper left corner of Cogview’s display area, and the Cogview window resizes, if necessary, to accommodate the new display area.

You specify a region of interest by means of the **Video Format** tab of the **Cogview Configuration** dialog box, as shown in Figure 65 on page 238. You can specify a region’s origin and dimensions independently, or together. When you specify a region’s origin, the width and height fields are automatically adjusted so that the region of interest does not extend beyond the current video format size.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

As examples, Table 33 shows the region settings for the four quadrants of a 640x480 image, and for a 1/16th-sized center image.

Portion of 640x480 image	X Origin	Y Origin	Width	Height
Upper left quadrant	0	0	320	240
Upper right quadrant	320	0	320	240
Lower left quadrant	0	240	320	240
Lower right quadrant	320	240	320	240
Center 1/16th image	240	180	160	120

Table 33. Example region of interest settings for portions of a 640x480 image

Live and Frozen Images

Strictly speaking, Cogview does not pass live video streams from the vision processor's camera to the Cogview window. Cogview's job is to request and display images from the vision processor's image buffer at the requested frame rate. It is up to the vision processing application running on the vision processor to enable or disable live video from the attached camera. It is important to understand that these are independent processes.

Think of the vision processing application as a pen, continually drawing image information on a pane of glass. In this metaphor, the pane of glass is a buffer in memory on the vision processor. Think of Cogview on the other side of this pane of glass, reading the image information as it's written, and passing it on to the system's video display card. The writing of image data on one side and the reading of that data from the other side are independent processes that occur at independent rates.

Participants on either side of the image buffer can issue commands to enable or disable live video, or to freeze the image. The results are different, depending on the participant.

When an application running on the vision processor disables live video, the vision processor stops writing data to its image buffer. However, Cogview, on the other side, may continue to request images from that buffer at a specified frame rate. In this case, Cogview continually retrieves and re-displays the same unchanging image from the vision processor's image buffer. However, Cogview has not done the work of freezing the image, and thus Cogview is not in freeze mode.

When you place Cogview in freeze mode, Cogview stops reading image data from the vision processor, and displays the last frame read. The vision processor may still be in live video mode, and may still be writing a continuous stream of data to its image buffer.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

You place Cogview in freeze mode in one of three ways:

- Select **View->Freeze Mode.**
- Click the  icon in the Cogview toolbar.
- Type *<Ctrl+F>.*

While in freeze mode, you can display an updated image by selecting **View->Capture Image** or by typing *<Ctrl+I>*. In this case, Cogview reads and displays the vision processor's image buffer one time, which effectively captures a new image for display.

Note: The Unified Acquisition software does not support live video for rapid reset cameras (the Sony XC-75 RR and XC-75CE RR). When used with a rapid reset camera, Cogview continually re-displays a static image from the vision processor's image buffer.

Using Zoom Mode

Cogview supports zoomed images, where the current live or frozen image is magnified. When you place Cogview in zoom mode, the cursor changes to one of the zoom cursors as shown in Figure 60. The number in the center of the zoom cursor shows the current Cogview magnification level.



Figure 60. Zoom cursors in Cogview

You place Cogview in zoom mode in one of three ways:

- Select one of the submenus under **View->Zoom**.
- Click the  icon in the Cogview toolbar.
- Type *<Ctrl+Z>.*

In zoom mode, use the PC's mouse to select magnification levels. Click the right mouse button to increase to the next magnification level, and use the left mouse button to decrease the magnification. (Remember that mouse button designations are reversed if you've specified a left-handed mouse in the NT control panel.)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 61 shows a Cogview image along with the three magnifications of zoom mode. Note that the origin of the zoomed images remains in the upper left corner. Use the scroll bars to view other portions of a zoomed image. Contrary to standard Windows scroll bar behavior, you cannot resize the Cogview window while in zoom mode in order to see more of the image.

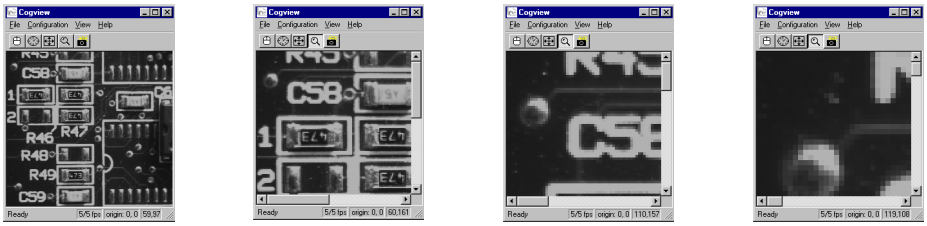


Figure 61. Original and zoomed images

Note: Cogview cannot be in zoom mode and autosize mode at the same time.

Using Autosize Mode

Cogview supports autosizing of the displayed image. When autosize mode is on, the displayed image scales to fit the Cogview window as you resize the window. With autosize mode off, the displayed image is cropped as the Cogview window is resized, with scroll bars available to view other portions of the image.

You place Cogview in autosize mode in one of three ways:

- Select **View->Autosize Mode.**
- Click the  icon in the Cogview toolbar.
- Type *<Ctrl+A>*.

Note: Cogview cannot be in autosize mode and zoom mode at the same time.

Using the PC's Mouse with Cogview

Cogview allows you to use the host PC's mouse as the pointing device for the vision processor, replacing the Cognex trackball. Your vision processing application must specify that trackball data is to be read on a certain pseudo-serial port, using a line like the following:

```
cif_tb_file = cif_open_tb ("/dev/bus2", "r");
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Cognex sample code and applications usually use pseudo-serial port 2 for mouse data, but you can use any port that's not already in use. Note that Cogtalk uses pseudo-serial port 0 by default for its communication window.

Once your vision processing application on the vision processor is set to read trackball information from a pseudo-serial port, set up Cogview to send the mouse's tracking information to the same port. You specify this port using the **Mouse Port** tab of the **Cogview Configuration** dialog box, as shown in Figure 64 on page 237.

While mouse support is enabled, the PC's mouse cursor remains in the PC's default shape when over the Cogview window. When mouse support is disabled, the mouse cursor becomes the mouse-inactive cursor (as shown in Figure 62) when over Cogview's display area.



Figure 62. Mouse-inactive cursor

The Cognex vision processor interface functions were written to interpret information received from the Cognex three-button trackball. To replace the trackball with the PC's mouse, you must emulate the action of the trackball buttons with the mouse buttons. PC mice can have more than two buttons, but only a two-button mouse interface is supported. Table 34 shows the correspondence between the Cognex trackball buttons and the host PC's mouse buttons.

Trackball button	2-button PC mouse	3-button PC mouse
left	left	left
middle	right	right
right	left + right	left + right

Table 34. Trackball buttons and corresponding mouse buttons

To emulate trackball movement, hold down the left mouse button while moving the mouse.

Mouse button designations are reversed if you have specified a left-handed mouse in the Windows control panel.

If no volume or saved system file is loaded, Cogview disables PC mouse support.

Palette Tables and Palette Table Files

On startup, Cogview generates a 256-color palette that combines color definitions in positions 0 through 9 and 240 through 255, with ascending grey-scale values in the interim positions. You can change Cogview's current palette by reading in a palette table from a

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

file, using the **Palette Table** tab of the **Cogview Configuration** dialog box (as shown in Figure 66 on page 239). A sample palette table file is installed as *\cognex\pchost\ntbin\cogview.lut*. The palette defined in this sample is the same as the one generated by Cogview.

To create a new palette table, copy the sample file and edit your copy to specify your desired palette. For guidelines on creating a valid palette table, see *Defining an 8-Bit Palette for Your Software Display Application* on page 165. A palette table file uses C language syntax to define an array of 256 triplets of RGB values. When reading the palette table, Cogview ignores any values in positions 0 through 9 and 246 through 255, because these positions are reserved for the static Windows colors.

For definitions of the CUA color constants in positions 241 through 245, refer to the palette-generating code in *\cognex\pchost\ntsample\cogview\src\palette.cpp.*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Cogview Menu and Toolbar Reference

This section describes, in order, each menu item and toolbar icon in the Cogview interface.

File Menu

The **File** menu contains only the **Exit** item. Use **File->Exit** to quit the current Cogview session.

Configuration Menu

Each item in the **Configuration** menu opens the same tabbed dialog box, with each menu item specifying a different tab.

Configuration->Frame Rate

The **Frame Rate** menu item invokes the **Cogview Configuration** dialog box with the **Frame Rate** tab uppermost, as shown in Figure 63.

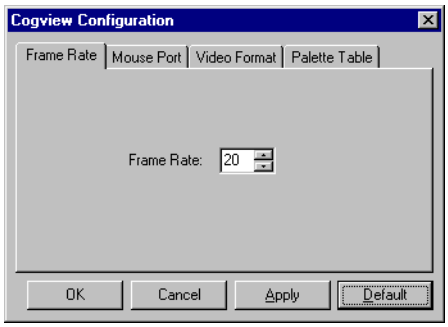


Figure 63. Frame Rate tab of Cogview Configuration dialog box

The **Frame Rate** entry field specifies a requested rate in frames per second at which Cogview attempts to retrieve an image from the vision processor's image buffer. Specify an integer between 1 and 30, inclusive.

Cogview's actual frame rate may not match the requested frame rate, depending on the conditions described in *Performance Factors* on page 164. The actual and requested frame rates are shown in Cogview's status bar.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Configuration->Mouse Port

The **Mouse Port** menu item invokes the **Cogview Configuration** dialog box with the **Mouse Port** tab uppermost, as shown in Figure 64.

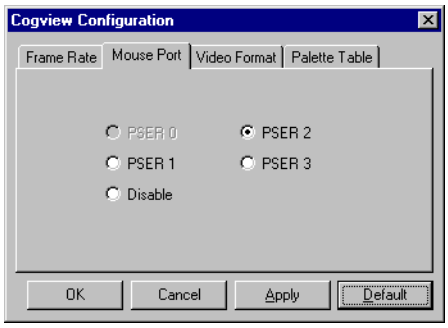


Figure 64. Mouse Port tab of Cogview Configuration dialog box

Select an available pseudo-serial port on which to send the mouse's tracking information to the vision processor. Select the same port that you designated in your vision processing application to receive mouse (trackball) information. Pseudo-serial port 2 is typically used for mouse communication.

Select the Disable button to stop sending the mouse's tracking information to the vision processor. In this state, the PC's mouse cursor changes to the mouse-inactive cursor (see Figure 62 on page 234) when over the Cogview display area.

See *Using the PC's Mouse with Cogview* on page 233 for further information on using the PC's mouse to emulate the Cognex trackball.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Configuration->Video Format

The **Video Format** menu item invokes the **Cogview Configuration** dialog box with the **Video Format** tab uppermost, as shown in Figure 65.

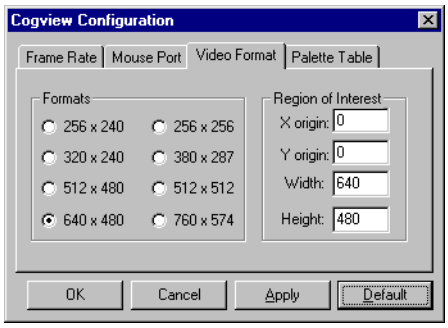


Figure 65. Video Format tab of Cogview Configuration dialog box

Use the **Formats** section of this dialog box to specify the maximum image size for the Cogview display area. For best results, select a image size format that matches the video format you specified in your vision processing application. (Vision processor video formats are shown in Table 28 on page 107).

The formats in the left column correspond to the available formats for RS-170 cameras. The formats in the right column of the **Formats** section correspond to the available formats for CCIR cameras. See *Window and Image Sizes* on page 229 for more information on matching vision processor video formats with Cogview's maximum image size.

Use the **Region of Interest** section to specify a rectangular region within the current image. The region specified cannot be larger than the current maximum image size. Specify the region's origin and/or its dimensions in pixels. When a region of interest is specified, its origin becomes fixed in the upper left corner of Cogview's display area. If the specified region is smaller than the current Cogview window, Cogview resizes itself to correspond to the region's dimensions. If the specified region is larger than the current Cogview window, Cogview draws scroll bars on the current window.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Configuration->Palette Table

The **Palette Table** menu item invokes the **Cogview Configuration** dialog box with the **Palette Table** tab uppermost, as shown in Figure 66.

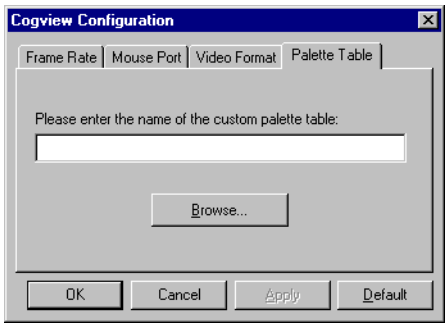


Figure 66. Palette Table tab of Cogview Configuration dialog box

Use the **Palette Table** dialog box to designate the path name, relative to Cogview's current directory, of a customized palette table file, or use the Browse button to locate the file. Once specified, Cogview retains the palette table entry between invocations. If Cogview can't find a specified palette table on startup, it displays an error message. To return to the default palette, delete the entry in the palette table field.

See *Palette Tables and Palette Table Files* on page 234 for more on palette table files and their format.

View Menu

Select items in the **View** menu to change the way Cogview looks or to enable certain operating modes.

View->Autosize Mode

Use the **Autosize Mode** menu item to toggle on or off the automatic resizing of images within the Cogview display area as the Cogview window is resized. A checkmark to the left of the menu item shows that autosize mode is enabled. Note that autosize and zoom modes are mutually exclusive.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

View->Freeze Mode

Use the **Freeze Mode** menu item to toggle on or off the active retrieval of images from the vision processor's image buffer. When this mode is toggled on, Cogview displays the last image retrieved from the vision processor. Use the **Capture Image** menu item to capture another image. A checkmark to the left of the menu item shows that freeze mode is enabled.

View->Capture Image

The **Capture Image** menu item is greyed out unless Cogview is in freeze mode. If so, use the **Capture Image** item to retrieve and display a single image frame from the vision processor's image buffer.

View->Zoom

The **Zoom** menu item has four submenu items (**x1**, **x2**, **x4**, and **x8**), which correspond to the available magnification levels for displayed images. Select the **x2**, **x4**, or **x8** submenu to display the current image in times-two, times-four, or times-eight modes, respectively. These selections place Cogview in zoom mode, in which the cursor becomes a magnifying glass with a centered digit denoting the current magnification level. Select the **x1** menu item to return to the non-magnified original image.

Zoom mode is independent of the current image's live or frozen state. Zoom and autosize modes are mutually exclusive.

View->Toolbar

The **Toolbar** menu item toggles the display of Cogview's toolbar.

View->Status Bar

The **Toolbar** menu item toggles the display of Cogview's status bar.

Help Menu

Select the **Help** menu to display online help for Cogview.

Help->Contents

The **Contents** menu item displays the contents page of the Cogview online help file.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Help->Using Help

The **Using Help** menu item opens a description of how to use Windows online help.

Help->About Cogview

The **About Cogview** menu item shows the product information box, containing copyright and version information for Cogview.

Toolbar Buttons

Use the buttons in Cogview's toolbar as single-click shortcuts for certain functions.

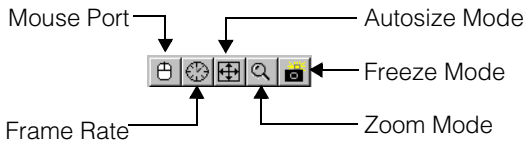


Figure 67. The Cogview toolbar

The functions invoked by Cogview's toolbar buttons are shown in Table 35. The three mode-changing buttons show the state of the current mode. When the button appears pressed, the corresponding mode is enabled. Note that zoom and autosize modes are mutually exclusive: only one can be selected at a time.

Button	Button Name	Function
	Mouse Port	Invokes the **Cogview Configuration** dialog box with the **Mouse Port** tab uppermost, as shown in Figure 64 on page 237
	Frame Rate	Invokes the **Cogview Configuration** dialog box with the **Frame Rate** tab uppermost, as shown in Figure 63 on page 236
	Autosize Mode	Toggles on or off the automatic resizing of images within the Cogview display area as the Cogview window is resized

Table 35. Cogview toolbar buttons

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Button	Button Name	Function
	Zoom Mode	Places Cogview in zoom mode, with times-two magnification selected. In zoom mode, use the right mouse button to increase magnification level and use the left mouse button to decrease magnification
	Freeze Mode	Toggles on or off the active retrieval of images from the vision processor's image buffer

Table 35. Cogview toolbar buttons

Cogview Keyboard Shortcuts

Table 36 lists the keyboard shortcuts available when running Cogview.

Keyboard shortcut	Action performed
Ctrl + A	Toggle autosize mode on or off.
Ctrl + F	Toggle freeze mode on or off.
Ctrl + I	In freeze mode, retrieve and display another image from the vision processor.
Ctrl + Z	Toggle zoom mode on or off.

Table 36. Cogview keyboard shortcuts

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Index

Symbols

~. command 194
~? command 194
~\ command 191
~b command 189
~c command 191
~close command 194
~cwd command 192
~d command 191
~escape command 192
~help command 194
~q command 191
~quiet command 193
~r command 190
~s command 190
~send command 192
~t command 192
~v command 193
~x command 190

Numerics

256-color desktop
    requirement for 165
430xX, 440xX Intel chip sets 21
5000-ISA boards
    hardware comparison with 5000-PCI boards 94
    software comparison with 5000-PCI boards 95

A

acquisition
    callback functions for 146
    channels 106
    hardware-triggered 110
    overview 101
    simultaneous 112
    software-triggered 125

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

acquisition properties
  brightness and contrast 112
  camera port 109
  charge transfer interval 111
  exposure 109
  for simultaneous acquisition 112
  input LUT 109
  overview 109
  timeout 109
  video strobe 110
  video trigger 110
Adobe Acrobat 47
ATI display adapters 20
autosize mode 233, 239

B

baseboard CVMs 72
bidirectional general I/O TTL lines 59
block mode file server
  Cogtalk as 194
  example 197
  in Windows Cogtalk 217
  session logging 208
  starting 194
  stopping 195
  supported functions 196
  unsupported functions 197
  utility functions 197
board size 54
boot on break 30
booting vision processor 189
breakout box 77, 80, 83, 85, 86, 87, 88
  camera 26, 56
  pinout 87
brightness and contrast 112
bus slot
  PCI bus 24
bus trigger interrupt 192
bus-based video output 73

C

C language
  run-time library 196
cable
  camera 26
callback function
  when acquisition completes 147
  when acquisition times out 149
  when part can be moved 150
  when strobe should fire 152
camera
  auxiliary power supply 28
  cable 26
  cables 88
  CCIR 64
  connector 26
  digital 94
  lenses 89
  port 109
  Pulnix TM-9701 77, 86, 89
  rapid reset 94
  RS-170 64
  shuttered 109
  Sony 28
  Sony XC-55 88, 107
  Sony XC-75 88, 107
  Sony XC-75 Rapid Reset 107
  Sony XC-75CE 107
  Sony XC-75CE Rapid Reset 107
  support for 26
  XC-55 77, 80
  XC-55BB 77, 80
  XC-75 76, 80
  XC-75CE 76, 77, 80
camera breakout box
  auxiliary power supply 56
  for 5900 26, 56
  installing 85

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

camera numbering
CVM1 75, 78
CVM3 81
cameras supported
CVM1 76
CVM3 74, 79
capture image
in Cogview 240
capturing data
with Cogtalk 190
CCIR cameras 64
cfs_diskopen() 196
cfs_free_space() 196
cfs_gdone() 196
cfs_gfirst() 196
cfs_gnext() 196
cfs_host_init_save() 197
cfs_host_reload() 197
cfs_host_wait_for_reboot() 197
cfs_init() 196, 197
cfs_mkdir() 196
cfs_rmdir(196
cfs_shutdown() 196, 197
cfs_util_copy(196
character, escape
in Cogtalk 192
charge transfer interval 111
chip sets
Intel 430xX, 440xX 21
clear
command line 214
command line in Console Cogtalk 201
display history 214
clearerr() 196
clipboard 207
closing window
in Cogtalk 194, 214
Cognex 5700/5900
32K shared memory segment 94
board size 54
components 52
hardware comparison with 5000-ISA boards 94
hardware description 51
image output 53
overview of using 93
PCI interface 52
PolySwitch 56
power requirements 56
safety standards 55
serial ports 53
shipping 55
software comparison with 5000-ISA boards 95
video output 53
vision processor 52
Cognex API Reference
starting from Cogtalk 220
Cognex trackball 101
Cognex video module 52, 85

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Index

Cogtalk
- alternate command processor 185
- as preprocessor 187
- block mode file server 194
- board status 219
- booting vision processor 185, 189
- bus trigger interrupt 192
- clear display history 205, 214
- clear line 214
- close window 194, 214
- command line switches 199
- common functionality usage notes 185
- compile region 216
- Console Cogtalk
  - command line switches 199
  - compared to PCVtalk 182
  - edit command line 201
  - starting 199
- Console vs. Windows 181
- Ctrl+Z 194
- current directory 212
- data capture 190
- dialog box, see dialog box
- directory 212
- DLLs required 182
- edit command line 201, 206
- end data capture 191
- end-of-file signal 191, 218
- escape character 192
- escape commands
  - ASCII data capture 190
  - change working directory 192
  - control+\ 191
  - control+C 191
  - end data capture 191
  - end-of-file 191
  - escape character 192
  - exit 194
  - help 194
  - load 190
  - overview 189
  - quiet mode 193
  - reboot 189
  - send file 192
  - send trigger interrupt 192
  - system save 190
  - version information 193
- exiting 194, 214
- expansion interrupt 192
- file server 194
- find 215
- font 215
- help 194
- include file 213
- include path 186
- include preprocessor 185
- includes, nested 188
- interrupt signal 191, 218
- interrupt, bus trigger 192
- keyboard shortcuts 223
- load system menu 210
- loading vision processor 185, 190, 210
- logging 201, 207, 212
- menu reference 209
- nested includes 188
- port window 209
- print 213
- prompt 185
- quiet mode 193
- quit signal 191, 218
- quitting 194, 214
- regional compile 216
- requirements 182
- reset vision processor 217
- right-click menu 203
- saving system 185, 190, 210
- sending
  - control-\ 218
  - control-C 218
  - end-of-file 218
  - files 192
  - interrupt 192
- session logging 201, 207, 212
- shortcuts 223
- starting
  - Console Cogtalk 199
  - Windows Cogtalk 203
- title bar 203
- toolbar reference 221
- tools menu 218
- transfer files 192
- V compiler 185

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

version 193
Windows Cogtalk
clear display history 205
compared to Wintalk 183
display history 205
edit command line 206
find 215
font 215
include file 213
menu reference 209
port windows 204
print 213
retained settings 204
right-click menu 203
session logging 207, 212
starting 203
title bar 203
using 203
window components 203
Windows vs Console 181
working directory 192, 212

Cogview
autosize mode 233, 239
capture image 240
DLLs required 226
exiting 236
frame rate 236
freeze mode 240
half-resolution video formats 230
image capture 240
keyboard shortcuts 242
live and frozen images 231
mouse port 237
mouse, using 233
palette table 234, 239
quitting 236
region of interest 230, 238
requirements 226
retained settings 228
shortcuts 242
starting 226
starting from Cogtalk 218
status bar 228
toolbar reference 241
using 225
video formats 229, 238
window and image sizes 229
window components 228
zoom mode 232, 240

color key 156

color, see palette

command line switches
Console Cogtalk 199

compile region 216

conditions
for maximum frame rate 20

connector
DB-26 camera 26
DC IN/SYNC 86

console
setting serial line as 30

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Index

Console Cogtalk
  command history and editing 201
  command line switches 199
  compared to PCVtalk 182
  feature list 182
  starting 199
  using 199
  vs. Windows Cogtalk 181

contrast 112

critical section mechanism
  use by Software Display 166

csd_close() 171

csd_frozen_image_get() 179

csd_frozen_set() 178

csd_graphics_format_get() 173

csd_graphics_format_set() 172

csd_handle 168

csd_image_format_get() 172

csd_image_format_set() 171

csd_image_roi_get() 175

csd_image_roi_set() 175

csd_keycolor 169

csd_keycolor_get() 174

csd_keycolor_set() 174

csd_live_set() 176

csd_next_image_get() 177

csd_open() 170

csd_rect 168

csd_state 168

csd_state_get() 176

csd_system_memory() 179

csd_version 169

csd_version_get() 180

csd_window 170

css_save_system() 211

ctg_init() 64

Ctrl+Z
  in Console Cogtalk 194

cua_acquire()
  defined 125
  use with video triggers 110

cua_acquire_channel_alloc() 116

cua_acquisition_buffers_get() 122

cua_acquisition_buffers_set() 122

cua_are_resources_prepared() 120

cua_board handle 114

cua_brightness_contrast_has() 136

cua_brightness_contrast_set() 137

cua_brightness_get() 136

cua_camera_port_get() 134

cua_camera_port_has() 133

cua_camera_port_set() 134

cua_cancel() 126

cua_channel handle 114

cua_cip_buffer_get() 125

cua_complete_callback_context_get() 148

cua_complete_callback_get() 147

cua_complete_callback_has() 147

cua_complete_callback_set() 147

cua_completion enumeration 128

cua_completion_get() 127

cua_contrast_get() 137

cua_copy() 153

cua_cti_end_even_get() 144

cua_cti_end_odd_get() 144

cua_cti_has() 143

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

cua_cti_set() 145
cua_cti_start_even_get() 143
cua_cti_start_odd_get() 144
cua_default_board_get() 119
cua_default_monitor_get() 117, 118
cua_display_channel_alloc() 117
cua_displayable_alloc() 121
cua_exposure_get() 132
cua_exposure_has() 131
cua_exposure_set() 132
cua_frame_buffer handle 114
cua_free()
    defined 153
    with object handles 114
cua_graphics_colors 115
cua_host_monitor_get() 118
cua_image_get() 126
cua_image_plane_set() 122
cua_input_lut_256_get() 135
cua_input_lut_256_has() 135
cua_input_lut_256_set() 135
cua_live_set()
    defined 124
    effect of Cogview 227
cua_master_get() 129
cua_master_set() 128
cua_monitor handle 114
cua_move_part_callback_context_get() 151
cua_move_part_callback_get() 150
cua_move_part_callback_has() 150
cua_move_part_callback_set() 151
cua_no_object 114
cua_overlay_plane_set() 123
cua_overlayable_alloc() 121
cua_overrun_callback_context_get() 150
cua_overrun_callback_get() 149
cua_overrun_callback_has() 148
cua_overrun_callback_set() 149
cua_pass 156
cua_polarity enumeration 141
cua_prepare_resources()
    defined 119
    effect on current video format 108
cua_properties handle 114
cua_properties_get() 130
cua_properties_set() 130
cua_state enumeration 127
cua_state_get() 126
cua_strobe_callback_get() 152
cua_strobe_callback_has() 152
cua_strobe_callback_set() 152
cua_timeout_get() 133
cua_timeout_has() 132
cua_timeout_set() 133
cua_video_strobe_duration_get() 140
cua_video_strobe_duration_set() 140
cua_video_strobe_enable_get() 139
cua_video_strobe_enable_set() 140
cua_video_strobe_has() 139
cua_video_strobe_polarity_get() 141
cua_video_strobe_polarity_set() 142
cua_video_strobe_trigger_width_get() 142
cua_video_strobe_trigger_width_set() 142
cua_video_trigger_enable_get() 138
cua_video_trigger_enable_set() 138

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

cua_video_trigger_has() 137

cua_vp_monitor_get() 118

CVM 83, 85
  defined 52, 72

cvp_exp_int() 192

cvp_trigger() 192

#cwd directive 189

D

data capture
  ending 191
  with Cogtalk 190

daughtercard CVMs 72, 85

DB-26 connector 26

DC IN/SYNC connector 86

demo
  seach and train 99

device driver
  installing, starting, stopping 45

dialog box
  Block Server (Cogtalk) 217
  Choose Destination Location (installation) 39
  Choose File to #Include (Cogtalk) 213
  Cognex Vision Processor Setup 44
  Cogtalk Include Path 216
  Cogview Configuration, Frame Rate 236
  Cogview Configuration, Mouse Port 237
  Cogview Configuration, Palette Table 239
  Cogview Configuration, Video Format 238
  Current Working Directory (Cogtalk) 212
  Find (Cogtalk) 215
  Font (Cogtalk) 215
  Load Executable Image (Cogtalk) 210
  Options Selection (installation) 41
  Registration (installation) 40
  Save As (Cogtalk) 211
  Save Options (Cogtalk) 211
  Select Sub-components (installation) 42
  Vision Processor Selection (Cogtalk) 209
  Welcome (installation) 38

digital cameras 94

DIP switch
  default position 29
  settings 30

directory, working
  changing in Cogtalk 192

disk space requirements 21

display channels 106

display history
  clearing in Windows Cogtalk 205
  in Windows Cogtalk 205

DLLs required
  Cogtalk 182
  Cogview 226

documentation
  changes 14
  online versions 48

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

E

edit command history 201, 206

electromagnetic
- compatibility standard 55
- emissions standard 55

end data capture
- with Cogtalk 191

end-of-file
- sending with Cogtalk 191
- signal 218

Epsilon
- starting from Cogtalk 218

escape character
- in Cogtalk 192

exiting
- Cogtalk 194, 214
- Cogview 236

expansion interrupt 192

exposure 109

F

FAT file system 196

fclose() 196

feof() 196

ferror() 196

fflush() 196

fgetpos() 196

file
- 8.3 file names 196
- sending with Cogtalk 192

file server
- see block mode file server

file-processing progress messages 193

fopen() 196

frame buffers
- allocating 112, 121
- automatically allocated 112
- data structure in Unified Acquisition 106
- for acquired images 112
- for overlay graphics 112
- number available 112, 121
- overlayable 121

frame rate
- in Cogview 236
- maximizing 20

fread() 196

freeze mode 240

frozen video
- in Cogview 231
- in Software Display 158

fseek() 196

fsetpos() 196

ftell() 196

fwrite() 196

G

graphics buffer
- pass-through color 156

H

half-resolution video formats 230

handle, see object handles

hardware comparison 94

hardware trigger 110

help
- Cogtalk escape command 194

history buffer
- in Windows Cogtalk 205

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

host PC
requirements 19

I

IEEE 996 15

image acquisition
overview 101

image capture
in Cogview 240

images
function for acquiring 125
triggers for acquiring 110

#include directive
using 187
with quiet mode off 193

include path
Console Cogtalk 186
Windows Cogtalk 186, 216

include preprocessor
Cogtalk as 185, 187
nested includes 188

input lookup table 109

input video
provided by CVM 52

inputs
TTL 53

insert mode 201

installation
5700/5900 29
before you install 37
device driver 45
double parallel I/O board 32
Epsilon 46
hardware before software 21, 37
serial interface adapter 34
single parallel I/O board 31
software installation instructions 37

installing
camera breakout box 85

interrupt
bus trigger, sending with Cogtalk 192
signal 218
signal sent by Cogtalk 191

ISA bus standard 15

K

keyboard shortcuts
Cogview 242
Windows Cogtalk 223

L

lenses and lens kits 89

live display
enabling 97
in Cogview 231
incompatible with next-image display 159

loading vision processor 190, 210

logging Cogtalk sessions 201, 207, 212

M

manuals
online versions 48

maximum frame rate, requirements for 20

memory requirements
effect of video formats 113

menu reference
Windows Cogtalk 209

messages, file-processing progress 193

motherboards
PCI chip set 21

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

mouse
compared to Cognex trackball 234
in Cogview 237
recommended for Cogview 20
recommended for Windows Cogtalk 182
using with Cogview 233

multithreaded applications
considerations in 166

N

nested includes
Cogtalk's handing of 188

next-image display 158
incompatible with live display 159

non-compliant motherboards 21

NT
command editing feature 202

NTFS file system 196

O

object handles
cua_board 114
cua_channel 114
cua_frame_buffer 114
cua_monitor 114
cua_no_object 114
cua_properties 114
null handles 114
overview 114
using cua_free() with 114

optically isolated outputs
example 65

Opto/TTL lines 53, 94

opto-isolated outputs 53, 59

outputs
opto-isolated 53
TTL 53, 59

overlay graphics
allocating frame buffer for 121
color constants for 165

overlayable frame buffer 121

overrun 146

overstrike mode 201

P

palette
defining for software display applications 165
in Cogview 234
palette table files 239
palette tables in Cogview 239

pane of glass 231

parallel I/O board
installation
double board 32
single board 31
uses ISA slot 26

parallel I/O boards 59

part numbers
for 5700/5900 accessories 25

PC requirements 19

PCI
2.1 specification 21, 25
bus interface 52
chip set 21, 164
slot 24
standard 16

PCVtalk
compared to Console Cogtalk 182

performance
Software Display 164

PolySwitch 56

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

pop-up menu
in Windows Cogtalk 203
port windows
in Windows Cogtalk 204
using more than one 204
port-based video output 73
power
+12 V adapter 86
power adapter
auxiliary 28
preprocessor
Cogtalk as 187
progress messages 193
pseudo-serial port 194, 237
Pulnix TM-9701 77, 86, 89
PVE 7.6 documentation
changes to 14
online versions 48
PVE software
8.1 differences 102
installation 37

Q

quiet mode
in Cogtalk 193
quit signal 191, 218
quitting
Cogtalk 194, 214
Cogview 236

R

rapid reset cameras 94
rebooting vision processor 189
recall commands
in Console Cogtalk 201
in Windows Cogtalk 206
recommended
PCI chip set 21, 164
video display adapters 20, 164
region of interest
csd_rect structure 168
in Cogview 230, 238
set by image and graphics formats 157
setting 159
stretch option 160
regional compile
supported by Cogtalk 216
remove() 196
rename() 196
requirements
disk space 21
for host PC 19
for maximum frame rate 20
for software display applications 165
power 56
software 21
to run Cogtalk 182
to run Cogview 19, 226
reset vision processor 217
retained settings
in Cogview 228
in Windows Cogtalk 204
rewind() 196
right-click menu
in Windows Cogtalk 203
RS-170 camera 64
run-time library
standard C 196

S

safety standards 55

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

saving vision processor system 190, 210

search demo 99

send files
  with Cogtalk 192

serial interface adapter
  installing 34
  uses faceplate slot 26

serial lines
  pinout 59

serial port as console 30

session logging 201, 207, 212

settings
  in Cogview 228
  in Windows Cogtalk 204

shared memory segment 94

shipping 55

shortcuts 223
  Cogview 242

shuttered camera 109

signal
  end-of-file, sent by Cogtalk 191, 218
  interrupt, sent by Cogtalk 191, 218
  quit, sent by Cogtalk 191, 218

simultaneous acquisitions 112, 128

software
  comparing 5000-ISA and 5000-PCI boards 95
  differences in PVE 8.1 102
  installation of PVE 37
  requirements for 5700/5900 21

Software Display
  256-color limitation 165
  frozen display 158
  global data 166
  image and graphics formats 157
  limitations 162
  live display 158
  next-image display 158
  overview 102
  PCI chip set for good performance 164
  performance factors 164
  recommended display adapters 164
  region of interest 159
  synchronization of threads 166

Sony cameras 28, 107

Sony XC-55 77, 80, 88

Sony XC-55BB 77, 80

Sony XC-75 76, 80, 88

Sony XC-75CE 76, 77, 80

standard
  electromagnetic compatibility 55
  electromagnetic emissions 55
  ISA bus 15
  PCI bus 16
  Underwriters Laboratories 55

starting
  Cogview 226
  Console Cogtalk 199
  Windows Cogtalk 203

strobes
  hardware-triggered 110
  link with video triggers 64
  software-triggered 146

supported cameras
  CVM1 76
  CVM3 74, 79

switch settings
  for Pulnix TM-9701 89
  for Sony XC-55 88

switches, command line
  for Console Cogtalk 199

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Sysgen
starting from Cogtalk 218

T

timeout 109, 146

timing
video 91

title bar
port windows in Windows Cogtalk 203

TM-9701 77, 89

toolbar
Cogview 241
Windows Cogtalk 221

trackball 101

transfer files
with Cogtalk 192

trigger interrupt 192

TTL
inputs 53, 59
lines 53, 94
outputs 53, 59

U

Underwriters Laboratories 55

Unified Acquisition
channels for acquisition and display 106
frame buffers 106
overview 105, 106
simultaneous acquisitions 112
use of frame buffers 106
video formats 107
video triggers 110

V

VC2, VC3, VC4 vision coprocessors 52, 94

version
Cogtalk 193

video display adapters
recommended 20, 164

video formats
effect on memory requirements 113
half-resolution 230
in Cogview 229, 238
table of supported 107

video memory block 113

video module
CVM overview 72

video output
bus-based 73
port-based 73

video strobes
acquisition channel property 110

video timing 91

video transfer rate, maximizing 20

video triggers
acquisition channel property 110
and strobe lines 64
parallel input lines to use 64
used with software trigger functions 64

W

white space 207

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Windows Cogtalk
board status 219
close window 214
command history and editing 206
compared to Wintalk 183
current directory 212
directory 212
display history 205
exiting 214
feature list 183
find 215
font 215
include file 213
keyboard shortcuts 223
load system menu 210
menu reference 209
port window 209
port windows 204
print 213
quitting 214
reset vision processor 217
retained settings 204
right-click menu 203
saving system 210
shortcuts 223
starting 203
toolbar reference 221
tools menu 218
using 203
vs. Console Cogtalk 181
window componets 203
working directory 212

Windows NT
command editing feature 202

Windows system palette 165

Wintalk
compared to Windows Cogtalk 183

working directory
changing in Cogtalk 192

X

XC-55 77, 80, 88

XC-55BB 77, 80

XC-75 76, 80, 88

XC-75CE 76, 77, 80

Z

zoom mode 232, 240

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Index

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP