

**OMRON**

# Automation Control Environment

---

## User's Guide



---

## **Copyright Notice**

---

The information contained herein is the property of Omron Adept Technologies, Inc., and shall not be reproduced in whole or in part without prior written approval of Omron Adept Technologies, Inc. The information herein is subject to change without notice and should not be construed as a commitment by Omron Adept Technologies, Inc. The documentation is periodically reviewed and revised.

Omron Adept Technologies, Inc., assumes no responsibility for any errors or omissions in the documentation. Critical evaluation of the documentation by the user is welcomed. Your comments assist us in preparation of future documentation. Please submit your comments to: [techpubs@adept.com](mailto:techpubs@adept.com).

Copyright © 2006 - 2016 by Omron Adept Technologies, Inc. All rights reserved.

Created in the United States of America



---

## Table Of Contents

<b>Introduction .....</b>	<b>17</b>
Overview .....	18
Configuration Tools .....	18
Application Setup and Operation .....	18
The ACE Workspace .....	19
Feature Summary .....	20
Comparison to Adept Utilities .....	21
Comparison to Adept AIM Software .....	22
Comparison to Adept DeskTop Software .....	23
How Do I Begin? .....	25
 <b>Getting Started with the ACE Software .....</b>	 <b>27</b>
Installing the Software .....	28
Licensing Requirements .....	28
Beginning the Installation .....	30
Installing the ACE Software .....	30
Configuring Ethernet Communications .....	32
Configuring One PC and Controller With User-Assigned IP Address .....	32
Configuring Serial Communications .....	36
Required Cabling and Hardware .....	36
Configuring the Serial Protocol .....	36
Running ACE for the First Time .....	37
Starting the ACE Software .....	37
Starting a Localized Version of the ACE Software .....	38
Loading or Creating a Workspace .....	39
Controller Connection Errors .....	40
Starting ACE from the Windows Command Prompt .....	40
Creating Shortcuts for Starting ACE .....	42
Understanding the Interface .....	46
The Main User Interface .....	46
Icons in the Workspace .....	46
The 3D Visualization Window .....	47
The Robot Jog Control .....	48
Programming Tools and Editors .....	49
Task Status Control .....	50
Workspace Loading and Unloading .....	52
Loading a Workspace .....	52
Dragging an AWP File into the Workspace .....	54
Saving a Workspace .....	54
User and Startup Configuration .....	55

---

---

Configuring User Access .....	55
Configuring System Startup .....	55
<b>Robot Concepts .....</b>	<b>57</b>
The Coordinate System .....	58
Locations and Precision Points .....	58
Defining a Location .....	59
Workspace Coordinates .....	60
Basic Robot-Motion Parameters .....	61
Speed, Acceleration, and Deceleration .....	61
Approach and Depart .....	61
Arm Configuration .....	62
Location Precision .....	63
Continuous-Path Motion .....	63
Joint-Interpolated Motion vs. Straight-Line Motion .....	64
Performance Considerations .....	64
Understanding Conveyors (Belts) .....	66
Indexing versus Tracking Conveyors .....	66
Conveyor Terminology .....	66
Defining a Pallet Layout .....	68
Why is Gripper Orientation Important? .....	70
<b>User Interface .....</b>	<b>71</b>
Main Menu .....	72
File Menu .....	72
View Menu .....	73
Tools Menu .....	74
Help Menu .....	74
Objects and Object Editors .....	75
Workspace Explorer .....	75
Objects in the Workspace .....	85
Objects and Object Editors .....	92
Controller Editor .....	97
System Options .....	104
Gripper Editor .....	108
Note Editor .....	113
Robot Editor .....	114
System Startup Editor .....	119
User Manager Editor .....	125
Data Mapper Editor .....	129
Variable Editor .....	131
Adept AnyFeeder Object .....	138
FlexiBowl Feeder Object .....	144
IO Feeder Object .....	148
CAD Data Object .....	152

---

---

Toolbar .....	157
Toolbar - General .....	157
Toolbar - Connections .....	158
Toolbar - Task Status Control .....	160
Save Workspace .....	161
Robot Jog Control .....	163
Digital I/O Window .....	166
File Explorer .....	168
3D Visualization .....	176
Function Keys .....	180
Wizards .....	181
Wizard Elements .....	182
Application Sample and Calibration Wizards .....	184
Emulation Mode Wizards .....	185
<b>V+ System Configuration Tools .....</b>	<b>189</b>
Controller Installation .....	190
Using the Controller Installation Wizard .....	190
DeviceNet Scan .....	193
Digital I/O Configuration .....	195
Using the Configure (FireWire) Nodes Dialog .....	195
Using the V+ System Configuration Dialog .....	196
Encoder Latches .....	198
FireWire Node Firmware Upgrade .....	200
Python Module/Smart Servo Kit Configuration .....	202
Robot and Motor FireWire Configuration .....	205
Setting the Robot and Motor Numbers .....	206
Setting the sDIO Block Numbers .....	208
Flashing the Node LED .....	208
Viewing the Node Information .....	209
Setting the Node Clock .....	210
Robot Installation .....	211
Adding a Robot With Auto Configuration .....	213
V+ Backup .....	214
V+ Restore .....	215
DeviceNet Configuration .....	216
LOCAL Statement .....	217
MACID Statement .....	218
MAPPING Statement .....	218
V+ System Configuration Editor .....	221
Adding, Editing, or Removing a Statement .....	222
Adding Multiple Item Values Within a Statement .....	224

---

---

Invalid Inputs .....	225
Saving the Configuration File .....	226
V+ Upgrade .....	228
V+ License Configuration .....	230
<b>Robot and Motor Configuration .....</b>	<b>233</b>
Collecting Data .....	234
Adding and Removing Monitored Items .....	237
Data Collection Parameters .....	239
Configuring Axes, Options, and Kinematic Parameters .....	243
Enabled Axes .....	244
Robot Options .....	244
Kinematics .....	245
Configuring S-Curve Profiles .....	246
Editing Obstacles .....	249
Applying or Saving Your Changes .....	250
Loading and Saving Robot/Motor Specification Files .....	251
Loading a Robot/Motor Specification File .....	251
Saving a Robot/Motor Specification File .....	253
Safety Settings .....	254
Tuning the Motors .....	256
Adding and Removing Monitored Items .....	259
Using a Tool Offset .....	261
Teaching a Tool Offset .....	261
<b>Cobra i600/i800 Robot Configuration .....</b>	<b>265</b>
Configuration Overview .....	266
Working with Configuration Files .....	268
Configuration Manager .....	269
Menu Items .....	270
Toolbar Items .....	271
Editor Parameters .....	271
Tabs .....	271
Buttons .....	272
Controller View .....	273
System Information .....	273
Inputs/Outputs .....	274
Password-Protect Configuration Data .....	275
Change Access Level .....	275
Robot Configuration Pages .....	276

---

---

Tabs .....	276
Buttons .....	277
Hand Control Signals .....	277
Specify Hand Control Signals .....	278
Workcell Obstacles .....	279
Manage Obstacles in the Workcell .....	281
Advanced Robot Configuration .....	283
Select Advanced Configuration Options .....	284
Edit Joint Limits .....	284
Apply Changes to the Controller .....	287
Option Descriptions .....	288
Writing Data to the Controller .....	289
Load Configuration Data from a File .....	290
Load-File Report .....	291
Save Configuration Data to a File .....	292
Restore Configuration .....	294
Option Descriptions .....	294
Buttons .....	295
License Manager .....	296
Menu Items .....	296
Editor Parameters .....	297
Installing and Removing a License .....	298
Switches and Parameters .....	299
Opening the Switches and Parameters Editor .....	299
Toolbar Items .....	300
Value .....	300
State .....	300
<b>Vision .....</b>	<b>301</b>
Vision Overview .....	301
Vision Devices .....	301
Vision Tools .....	302
Sequences .....	302
Related Topics .....	302
Vision Basics .....	303
Vision Basics - Resolution .....	303
Vision Basics - Mode .....	305
Vision Basics - Camera Calibration and Lighting .....	311
Vision Devices .....	315
Adding a Camera .....	316
Standalone Camera Calibration .....	320
Creating a Dot Target .....	326
Adding an Emulation Device .....	327
Adding a Virtual Camera .....	333

---

---

Camera Properties .....	335
<b>Vision Tools .....</b>	<b>345</b>
Vision Tools Summary Table .....	349
Adding Vision Tools .....	352
Arc Caliper .....	355
Arc Edge Locator .....	371
Arc Finder .....	387
Blob Analyzer .....	399
Calculated Arc .....	428
Calculated Frame .....	431
Calculated Line .....	433
Calculated Point .....	437
Calibration Grid Locator .....	440
Caliper .....	442
Color Matching .....	455
Custom Vision Tool .....	465
Edge Locator .....	469
Gripper Clearance Tool .....	481
Image Histogram .....	489
Image Processing .....	500
Image Sampling .....	525
Image Sharpness .....	527
Inspection .....	536
Line Finder .....	547
Locator .....	557
Locator Model .....	574
Pattern Locator .....	580
Point Finder .....	592
Remote Vision Tool .....	600
<b>ACE Sight Special Tools .....</b>	<b>607</b>
ACE Sight Sequence .....	608
Communication Tool .....	610
Overlap Tool .....	618
Gripper Offset .....	621
<b>ACE Sight Calibrations .....</b>	<b>625</b>
ACE Sight Belt Calibration .....	627
ACE Sight Camera Calibration .....	641
ACE Sight Latch Calibration .....	742
<b>Programming ACE .....</b>	<b>745</b>
Programming Overview .....	746
Task Status Control .....	747
C# Language Programming .....	750
Using the C# Program Editor .....	750
Toolbar Items .....	752
Tab Items .....	753

---

---

Code Editor .....	753
<b>Profiler .....</b>	<b>755</b>
Operation .....	756
Choosing Displayed Tasks .....	756
Profiler Menus .....	757
	759
<b>V+ Programs and Variables .....</b>	<b>760</b>
V+ System Modules .....	760
V+ User Modules .....	760
V+ User Variables .....	762
<b>V+ Editor Tool .....</b>	<b>765</b>
Editor Tool .....	765
<b>V+ Debugger Tool .....</b>	<b>769</b>
The Program Variable Shortcut Menu .....	771
<b>V+ Task Manager .....</b>	<b>772</b>
Toolbar .....	774
Task List Area .....	775
Task Status Area .....	776
<b>Monitor Window .....</b>	<b>777</b>
Executing a V+ Monitor Command .....	777
V+ Startup Messages .....	777
Opening a V+ Program .....	778
Creating a New V+ Program .....	778
<b>Error Messages .....</b>	<b>780</b>
<b>Find Dialog .....</b>	<b>782</b>
Finding a String .....	782
Replacing a String .....	783
<b>OPC Data Access and Process Control .....</b>	<b>785</b>
Installing and Enabling OPC .....	785
Exposing Object Data Elements .....	786
Communicating Variable Values .....	787
Controlling Applications .....	790
Controlling a Process .....	793
<b>Watch Variable Tool .....</b>	<b>794</b>
Variable Browser Window .....	795
Add Workspace Variable Window .....	797
<b>Process Control .....</b>	<b>799</b>
Overview .....	800
Process Components .....	800
Line Balancing .....	804
V+ Controller Software .....	805
Process Flow .....	805

---

---

Process Menu and Features .....	807
Belt .....	809
Creating a Belt Object .....	809
Menu Items .....	812
Editor Parameters .....	813
Progressive Processing of Pallets .....	817
Viewing Belt Encoders and Testing Latches .....	818
Defining Multiple Latch Signals for a Belt Encoder .....	822
Custom Allocation Script .....	825
Using the Custom Allocation Script Editor .....	825
Toolbar Items .....	827
Tab Items .....	828
Code Editor .....	828
Example .....	828
Part .....	829
Menu Items .....	829
Configuration, Editor Parameters, and Online Help .....	830
Creating a Part Configuration .....	830
Part Buffer .....	836
Menu Items .....	836
Configuration, Editor Parameters, and Online Help .....	837
Creating a Part Buffer Configuration .....	837
Adding a Part Buffer to a Process .....	838
Setting the Part Buffer Access Order .....	839
Part Target .....	841
Menu Items .....	841
Configuration, Editor Parameters, and Online Help .....	842
Process Manager .....	843
Creating a Process Manager Object .....	843
Menu Items .....	844
Editor Parameters .....	844
Configuration Errors .....	845
Creating an Application .....	845
Creating Recipes .....	847
Belt Calibrations .....	848
Configuration Items .....	856
Control Sources .....	869
Process Strategies .....	878
Processes .....	886
Sensor Calibrations .....	909
Process Manager Control .....	920
Using the Process Manager Control .....	920
Editor Parameters .....	922
Controlling an Application .....	925
Process Pallet .....	927

---

---

Menu Items .....	928
Configuration, Editor Parameters, and Online Help .....	928
Creating a Custom Pallet .....	929
Pallet Visualization .....	931
Status/Error Codes .....	932
Category Key .....	932
Code and Description List .....	932
Custom V+ Programs .....	937
To create or select a custom V+ program .....	937
Vision Refinement Station .....	941
Menu Items .....	941
Configuration, Editor Parameters, and Online Help .....	942
Adding the Vision Refinement Station to a Process .....	942
Editing the Motion Parameters .....	943
Workspace Positioning .....	945
<b>Application Samples .....</b>	<b>947</b>
Overview .....	948
Selecting the Application Sample .....	949
Introduction and Tasks Overview .....	949
To Program the Application: .....	950
Cobra i-Series Pick and Place Samples .....	953
What happens next? .....	953
To program the application: .....	953
Auto starting the application: .....	960
Static Pick and Place Sample .....	961
What happens next? .....	961
To program the application: .....	961
Belt Camera Part Pick Sample .....	966
What happens next? .....	966
To program the application: .....	966
Fixed Camera Part Pick Sample .....	974
What happens next? .....	974
To program the application: .....	974
<b>Recipe Manager .....</b>	<b>983</b>
Overview .....	984
Using the Recipe Manager .....	985
Understanding Editor Menus .....	985
Configuration Sources .....	986
ACE Sight Index .....	987
Configuration Editor .....	987

---

---

Script .....	989
Editing the Recipe .....	991
ACE Task Status Control: Recipe Editing .....	991
Recipe Editor .....	993
Recipe Selection .....	999
<b>User Interface Designer .....</b>	<b>1000</b>
Overview .....	1001
Understanding the Interface .....	1003
Tab Items .....	1004
Menu Items .....	1004
Toolbar Icons .....	1004
Toolbox Items .....	1006
Properties .....	1011
Adding and Editing UI Tools .....	1016
Adding Tools to the Form .....	1016
Editing a Tool on the Form .....	1019
Using the Editing Commands and Z-Order .....	1020
Editing the Item Properties .....	1022
Using the Appearance, Behavior and Layout Controls .....	1025
The Appearance Parameters .....	1025
The Behavior Parameters .....	1025
The Layout Parameters .....	1026
Applying the Appearance and Layout Parameters .....	1026
Using the Condition Editor .....	1030
Conditional Properties .....	1031
Condition Editor .....	1031
Applying a Condition to a Tool .....	1034
Using Property Links .....	1039
Linking the Object to a Data Item .....	1039
Using the Formatting Parameters .....	1040
Using the User Access Controls .....	1043
Applying the User Manager Parameters .....	1043
Creating a Combo Box or List Box .....	1045
Using a List Box .....	1046
Creating a Graph .....	1052
The Appearance Group .....	1052
The Graph Group .....	1053
Using the Graph Tool .....	1054
Creating an Image List .....	1058
Using an Image List .....	1059
Creating a Split Form .....	1066

---

---

Creating a Tabbed Form .....	1070
Using Images Instead of Tab Names .....	1073
Applying Behavior Parameters to the Tab Control .....	1074
Creating an Example User Interface .....	1076
Deploying the Custom User Interface .....	1082
Using the User Interface Deployment Tool .....	1082
Using a Command Line to Deploy the UI .....	1084
Using a Batch File to Deploy the UI .....	1085
Using a Shortcut to Deploy the UI .....	1085
Controlling the User Interface through Scripts .....	1087
<b>Emulation Mode .....</b>	<b>1089</b>
Overview .....	1090
Features and Differences .....	1091
Features .....	1091
Differences .....	1091
V+ versus Emulation Mode .....	1092
Enabling Emulation Mode .....	1093
Emulation Mode Checklist .....	1100
Latch Signals .....	1100
Belt Speed .....	1100
Quattro Robot Platform .....	1101
Camera Trigger Mode .....	1102
Latch Period (Distance) .....	1102
Virtual Camera Emulation Configuration (Behavior) .....	1102
<b>Diagnostic Tools .....</b>	<b>1105</b>
About ACE .....	1106
Event Log .....	1108
FireWire Event Log .....	1110
Viewing the FireWire Event Log .....	1110
Clearing the FireWire Event Log .....	1110
Pending Errors Log .....	1112
Robots and Encoder Diagnostics .....	1113
Robot Diagnostics .....	1113
External Encoder Diagnostics .....	1115
System Monitor .....	1117
Menu Items .....	1120
Adding and Removing Graph Items .....	1123
Setting the Options .....	1124
Resetting the Maximum Recorded Values .....	1125
Monitoring Process Manager Statistics .....	1126

---

---

System Diagnostic Summary .....	1128
Cobra i600/i800 Robot Diagnostics .....	1130
Using the Robot Diagnostics Wizard .....	1130
Robot Diagnostics Wizard Test Summary .....	1132
<b>Extending ACE .....</b>	<b>1135</b>
<b>Appendix 1: Using OPC through DCOM .....</b>	<b>1136</b>
Overview .....	1137
Applicable Products .....	1137
Notes .....	1137
Summary .....	1137
Procedure .....	1139
Step 1: Create a computer name and define the workgroup / domain .....	1139
Step 2: Create matching user accounts and passwords .....	1141
Step 3: Configure the DCOM settings for system-wide security parameters .....	1142
Step 4: Configure the DCOM settings for all computers using OPC .....	1151
Step 5: Configure OpcEnum for all computers using OPC .....	1158
Step 6: Activate the Guest account .....	1160
Step 7: Configure the Windows Firewall to allow exceptions .....	1162
Step 8: Restart all computers and OPC clients / servers .....	1166
Reference Documents .....	1167
<b>Appendix 2: User Access Item List .....</b>	<b>1168</b>
Operator Access Levels .....	1169
Technician Access Levels .....	1170
Engineer Access Levels .....	1172

## Introduction

The topics in this chapter provide introductory material on the ACE software.

Overview .....	18
The ACE Workspace .....	19
Feature Summary .....	20
How Do I Begin? .....	25

## Overview

The Adept Automation Control Environment (ACE) software is a PC-based software package that contains a collection of tools for configuring, programming, controlling, and monitoring Adept equipment in a work-cell. These tools are accessed through the ACE™ graphical user interface (GUI), which provides a user-friendly, point-and-click environment.

## Configuration Tools

When you first receive an Adept system, you may need to configure that system for your particular application. The ACE software contains a set of configuration tools, which provide GUI-driven alternatives to the Adept V+ operating system command-line utilities. Most of the ACE configuration tools can be accessed directly from the Controller Editor main window. For more details on the configuration tools, see the topics [Comparison to Adept Utilities](#) and [Configuration Tools](#).

## Application Setup and Operation

After your Adept system is configured, you will need to program it to do its job. There are two major programming approaches available under ACE:

- ACE Sight
- ACE PackXpert

The ACE Sight module utilizes the PC to handle all vision operations and the customer is responsible for supplying the V+ code that drives the robot motions.

The ACE PackXpert Process Manager provides a point-and-click interface for configuring and programming your workcell specifically for packaging applications. You can program many types of packaging applications without having to write any code — the ACE PackXpert Process Manager handles this for you, along with balancing the work between multiple robots and conveyors in the workcell, and much more. For more details, see [Process Control](#) on page 799.

**NOTE:** Because the ACE software runs on the PC, the PC must remain connected when the application is running. The only exception to this rule is if the application does not rely on any ACE functionality at run-time. In this case, ACE can be used for setup and configuration, then removed for the operation of the system.

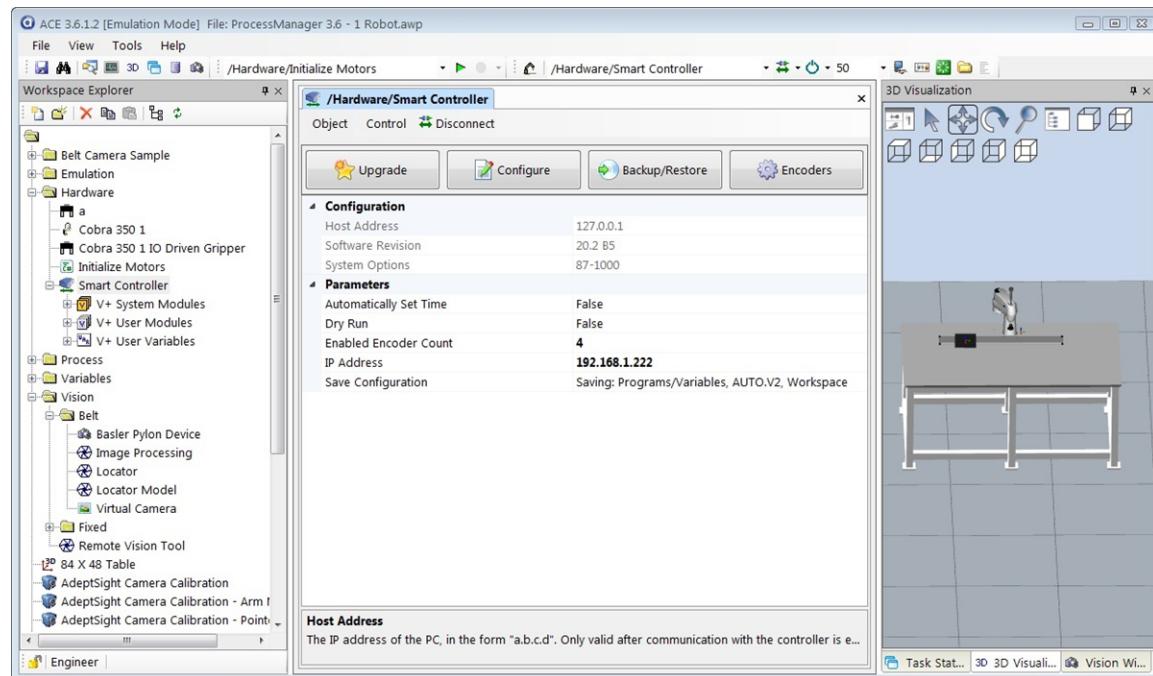
## The ACE Workspace

A fundamental concept in the ACE software is the workspace. The workspace is composed of a hierarchy of objects, both physical (like robots and end-effectors) and non-physical (like programs and variables).

The workspace configuration can be saved to and loaded from an AWP (ACE Workspace Project) file. This file contains all the information describing the application.

**NOTE:** You can also click the Save icon (■) or select **File > Save** or **File > Save As** to save the workspace. For details, see Save Workspace on page 161.

The following figure shows the Workspace Explorer, Editor window, and 3D virtual display. The left part of the window shows the folder view of the workspace; the center part of the window shows an editor used to view and change object properties; the right part shows the 3D virtual display.



ACE Workspace Explorer and Object Editor

For more details, see the section [Workspace Explorer](#).

## Feature Summary

The ACE software includes configuration, programming and application-development features, such as:

- GUI-driven alternatives to the V+ operating system command-line utilities. See Comparison to Adept Utilities on page 21 for more details.
- Point-and-click application configuration and programming. See the Process Control on page 799 for more details.
- Custom workspace layouts that can be created/edited, saved, and opened. See The ACE Workspace on page 19 for more details.
- .NET Framework GUI, which allows greater control of items in the workspace.

**NOTE:** This version of ACE requires the Microsoft .NET Framework, version 2.0 or later, which is supplied on the ACE software installation media. See Installing the Software on page 28 for more details.

- Multi-axis mechanism support
- [Task Status](#) control, which provides the following program controls:
  - Start (execute)
  - Pause
  - Proceed
  - Retry
  - Info
  - Edit
  - Abort
  - monitoring of current tasks and current exceptions
- [File Explorer](#), which provides:
  - Access to files on the connected Adept controller
  - Drag and drop capability
- [Workspace Explorer](#), which displays the workspace objects (controller, robot, gripper, etc.) in a tree structure on one side of the display, and editors for those objects on the other side of the display.
- [Program Editors](#), which provide online programming tools for:
  - V+ Programs
  - C# Programs
- [Digital I/O](#) display of Inputs / Outputs / Soft and internal robots signals.
- Support for Basler 1394 and Gigabit Ethernet cameras.

The following sections describe how the ACE software features compare to those found in other Adept software.

- [Comparison to Adept Utilities](#)
- [Comparison to Adept AIM](#)

## Comparison to Adept Utilities

The ACE configuration tools provide a graphical version of the command-line configuration utilities found in the V+ operating system. Most of these configuration tools can be accessed directly from the Controller Editor main window. For details on the Controller Editor, see the topic [Controller Editor](#). For details on the configuration utilities, see the topic [Configuration Tools](#).

V+ Utility	ACE
CONFIG_C: Configures V+ and the controller	Included as a system configuration editing dialog box, V+ upgrade utility dialog box, and robot append/remove/select dialog box, which are accessed from the <a href="#">Controller Editor</a> .
SPEC: Configures robot specifications and motor gains.	Included as property editors and GUI dialogs, which are accessed from the <a href="#">Robot Editor</a> and the <a href="#">Controller Editor</a> .
DC_SETUP: Configures FireWire nodes and sets up single-axis calibration.	Included as GUI dialogs for configuring FireWire nodes and loading specs, which are accessed from the <a href="#">Controller Editor</a> .
DC_UPDAT: Updates firmware on FireWire nodes.	Included as a GUI dialog box, which is accessed from the <a href="#">Controller Editor</a> .
FAULTLOG: Reads the FireWire node fault logs.	Included as a GUI dialog box, which is accessed from the <a href="#">Controller Editor</a> .
FLASHUPD: Updates SmartController FPGA.	Included as a GUI dialog box, which is accessed from the <a href="#">Controller Editor</a> .
ASL: Advanced Servo Library.	Included as API calls. See the ACE Reference Guide for details.
DISKCOPY: Copies files across directories and networks.	Included as a graphical interface, which is accessed from the <a href="#">File Explorer</a> .
VPROTECT: Encrypts V+ files to provide intellectual property protection.	Included as the "VPROTECT" (VPROTECT.EXE) PC command-line utility, which is located in the ACE installation directory "Tools" subfolder. The file(s) being encrypted must be located on the PC.

V+ Utility	ACE
tection.	
SQUEEZE: Eliminates white space and comments to reduce file size.	Included as the "VPP" (VPP.EXE) PC command-line utility, which is located in the ACE installation directory "Tools" subfolder. The file(s) being squeezed must be located on the PC.

## Comparison to Adept AIM Software

The table below highlights the differences between AIM and ACE software.

Feature	AIM	ACE
Runtime Execution	AIM executes sequences on the Adept controller.	Execution is performed on the PC, which allows access to a greater variety of hardware and software capabilities. However, the PC must remain connected to the controller. V+ execution is still supported for customers who want to run AIM or V+ programs.
Persistence	AIM uses one file per database, with each database containing any number of records with a fixed record size. The file is stored on the Adept controller.	The entire workspace is portable as a single ZIP or AWP (ACE Workspace Project) file composed of one XML file per workspace object. The file is stored on the PC.
Framework	AIM uses libraries of V+ routines. These routines cannot be used outside of Adept AIM. The framework runs in V+.	Libraries of .NET software, accessible through .NET languages such as VB, C#, and C++. The ACE .NET libraries can be used either within a standard Adept GUI or in a customer-written application. The ACE software can also invoke V+ programs on the controller.
Utilities	AIM has very limited utility support. The user must run command-line utilities outside of Adept AIM, using the standard V+ text interface.	Utilities run as plug-ins into framework.
Vision	AIM VisionWare is not accessible outside of Adept AIM. Vision in the V+ environment	ACE Sight can take advantage of fast PC processors and networked cameras, and is accessible from V+ programs and the ACE software.

Feature	AIM	ACE
	onment requires extensive coding.	
Controllers	AIM can control one Adept controller.	Multiple Adept controllers can be controlled from a single ACE application.
Programming	AIM programming is generally done online, because there is no offline interpreter.	Programming is generally done online. C# language scripting can occur offline or online.
Diagnostics	AIM is used as a control application, rather than a diagnostic application, because it contains few diagnostics.	By providing a foundation for utility and test software, many diagnostics have already been developed, and new ones are planned.

## Comparison to Adept DeskTop Software

The section highlights the similarities and differences between Adept DeskTop and ACE software

In general, Adept DeskTop and ACE software provide a PC-based graphical user interface for Omron Adept systems. In the Adept DeskTop software, programs must be written in the V+ or MicroV+ programming languages. In the ACE software, you can build and configure your applications without any knowledge of the V+ programming language. However, Cobra i-series (MicroV+) robot programs must be written in the MicroV+ programming language.

### ***Similarities***

Adept DeskTop and ACE software provide the following similar features:

- a PC-based interface to Adept controllers.
- a file explorer for moving files between the PC and the controller.
- a digital I/O status display for digital signals, soft signals, and robot signals.
- a vision interface.
- works with both MicroV+ and V+ systems.

### ***Differences***

Adept DeskTop and ACE software differ in the following features:

- The Adept DeskTop software has limited V+ configuration capability; the ACE software has complete V+ configuration capability.
- The Adept DeskTop software has limited point-and-click application-development capability; the ACE software has complete point-and-click application-development capability, while still allowing access to the V+ programming language for custom development.

## How Do I Begin?

Before you can use the ACE software, you must complete the following steps:

1. Install the Omron Adept controller and robot in the workcell. See the Omron Adept user guides that were supplied with your equipment for installation instructions.
2. Install ACE on your PC. See [Installing the Software](#) on page 28 for details.
3. Set up a physical connection to the controller. For Adept robots controlled by the Adept SmartController, see [Configuring Ethernet Communications](#) on page 32. For the Cobra i600/i800 robots, see [Configuring Serial Communications](#) on page 36.
4. Start the ACE software. For details, see [Running ACE for the First Time](#) on page 37.



## Getting Started with the ACE Software

The topics in this chapter describe how to install and start the ACE software.

Installing the Software .....	28
Configuring Ethernet Communications .....	32
Configuring Serial Communications .....	36
Running ACE for the First Time .....	37
Understanding the Interface .....	46
Workspace Loading and Unloading .....	52
User and Startup Configuration .....	55

## Installing the Software

This section describes the procedure for installing the ACE software and supporting libraries onto your PC.

The ACE software is distributed on installation media. The media contains installation programs to properly install the software on the PC. During the installation procedure, any prerequisite software (for example, Microsoft .NET Framework and DirectX) and the following application software will be installed:

- ACE (requires the USB hardware key [dongle], installed on the PC, for full functionality)
- ACE Sight (requires the ACE Sight USB hardware key [dongle], installed on the PC, for full functionality)
- Online Documentation (User Guides, ACE Reference Guide, and ReadMe file)

**NOTE:** Please see the ReadMe file for the complete list of changes, system requirements, and usage considerations. To access this file, select ReadMe File from the ACE software Help menu.

## Licensing Requirements

**NOTE:** To enable full functionality the ACE software requires the V+ controller licenses and PC licenses (supplied on the USB hardware key [dongle]) described below. For details on obtaining these items, please contact Omron Adept.

The ACE software requires the following licenses:

### **Omron Adept Controller Licenses**

The following licenses are required on the Adept controller:

License	Description
ACE Motion (license bit 44)	These licenses enable robot motion, access to additional user tasks, belt tracking and other advanced programming controls.  A check is made by each controller-queue object after the Process Manager Control start button is pressed. If a controller fails a license check, a "license not found" error message is displayed for the controller associated with the Process Manager Control.
V+ Extensions (license bit 2)	<b>NOTE:</b> Any OPC variables mapped to the status message/code for the controller will display the same error message.

**NOTE:** The ACE software occupies 2 user tasks plus 1 additional user task for each robot configured on the Adept controller. Therefore, beginning with V+ version 17.2B, for standard V+ software installations (those not using the V+ Extensions license), the number of user tasks was changed from 7 to 10, to compensate for the tasks used by the base ACE software. In applications using the ACE

PackXpert option, the Process Manager control will allocate additional tasks based on the specific process configuration. For more details on the ACE PackXpert option license, see the table in the next section. For more details on the Process Manager, see Process Control on page 799.

### **PC Licenses**

The following licenses are required on the PC (the PC licenses are supplied on the USB hardware key [dongle]). To view the licenses installed on the dongle, from the ACE menu bar, select **Help > Diagnostic Summary**. For more details on the Diagnostic Summary, see System Diagnostic Summary on page 1128.

<b>License</b>	<b>Description</b>
ACE PackXpert	<p>(Optional) Enables full functionality of the ACE PackXpert software. You will need to add the appropriate controller, vision, camera, and OPC server/client licenses required for your application.</p> <p>Without the ACE PackXpert option license OR the ACE Sight vision license, the system is the same as the "Configuration Only" version of ACE. If one of these licenses is not enabled, you can still load and edit a previously-saved workspace containing ACE PackXpert or ACE Sight objects. The licensing restricts <i>execution</i> of those objects in the ACE PackXpert or ACE Sight software.</p> <p><b>NOTE:</b> License checking is enforced only on the PC running the ACE server.</p>
Controller	<p>Provides support for the Adept controllers in the ACE PackXpert and ACE Sight application modules. You must have one controller license for each Adept controller you wish to use with those application modules. For example, 4 Controller licenses would allow you to connect (and communicate with) up to 4 Adept controllers in your application.</p>
OPC Server	<p>(Optional) Enables OPC data communications. For details, see OPC Data Access and Process Control on page 785 .</p>
OPC Client	<p>(Optional) Provides OPC client capability for each controller (client) you wish to obtain information from. You must have one OPC Client license for each OPC client you wish to communicate with in your application. For example 2 OPC Client licenses would allow you to use OPC data communications with 2 controllers in your application.</p>
ACE Sight Vision	<p>(Optional) Enables the ACE Sight vision software functionality. See the <i>ACE Sight User's Guide</i> for details.</p>
Camera	<p>(Optional) Provides support for physical cameras. You must have one Camera license for each physical camera you wish to use in your application. For example, 3 Camera licenses would allow you to use up to 3 physical cameras in your application. See the <i>ACE Sight User's Guide</i> for details.</p>

License	Description
Color Vision	(Optional) Provides support for color vision tools. See the <i>ACE Sight User's Guide</i> for details.

## Beginning the Installation

To begin the installation:

1. Insert the ACE software installation media into your PC.
  - If AutoPlay is enabled, the Welcome to ACE menu is automatically displayed.
  - If AutoPlay is disabled, you will need to manually start the installation media. Use Windows Explorer to browse to the installation media, right-click on the drive, and select AutoPlay from the menu.
2. Click **Read Important Information**. The ReadMe file is displayed.
3. Verify that your system meets the minimum requirements described in the System Requirements section.
4. After reviewing the information, click the close icon () in the upper-right corner of the ReadMe file window to close it.

## Installing the ACE Software

Complete the following steps to install the ACE software:

1. Click **Install ACE**. The ACE installer opens.
2. The installer scans your system for prerequisite files. If any files are needed, you will be prompted to acknowledge the installation of those files.
3. Follow the on-screen instructions to complete the installation. You will be prompted to accept the License Agreement and confirm the destination folder for the software installation. Then the installation will proceed.
4. After the installation has completed, you may be prompted to restart your computer.

The ACE software installer adds an "ACE" shortcut to your desktop.



ACE Icon

You can also launch ACE from the program group added to the **Omron gt; ACE** folder on your Windows Programs menu.

## Configuring Ethernet Communications

**NOTE:** If your system requires a serial communications link, see Configuring Serial Communications on page 36 for connection details.

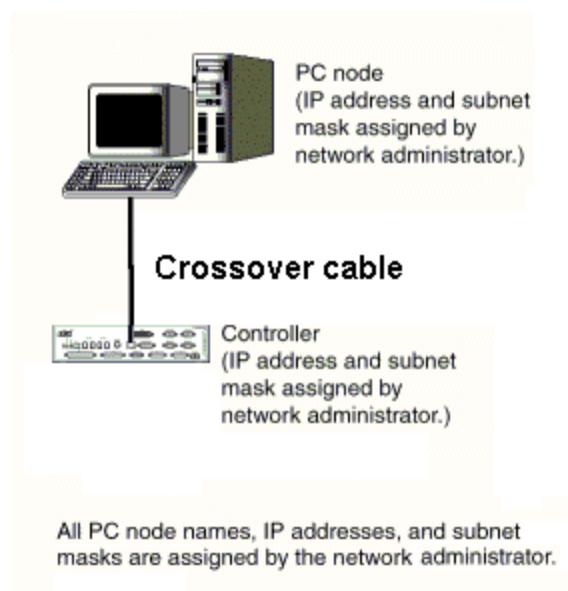
There are many ways to configure the Ethernet communications for your ACE system. You can set up a basic Ethernet installation between one PC and one Omron Adept controller. Or, you can set up more complex systems involving multiple PCs and controllers.

When connecting your PC to your Omron Adept controller, there are times when you may not know the IP address of the Omron Adept controller. The Controller IP Address Configuration tool (see the figure [Controller IP Address Configuration Dialog on page 34](#)) can help solve this problem. Even if the controller IP address is incompatible with the IP address of your PC, it will reveal the controller IP address to you. Then you can set the desired IP and subnet addresses to values that are compatible with those on your PC, which will allow your PC and controller to communicate properly. See the next section for more details.

### Configuring One PC and Controller With User-Assigned IP Address

As shown in the following figure, there is one PC networked to one Omron Adept controller. This configuration assumes that the PC has already been assigned an IP address (for example, for use on a corporate network). Therefore, the controller must have a user-assigned IP address rather than the default IP address assigned by V+.

**NOTE:** This configuration requires the Omron Adept-supplied crossover cable and CompactFlash card that was supplied with the Omron Adept controller.



*PC and SmartController Network*

To create the network shown above, complete the following steps:

1. Disconnect power from the controller. If needed, install the Omron Adept-supplied CompactFlash that was included with the controller. See the *SmartController User's Guide* for details.
2. Verify that the controller DIP switch (located on the front of the SmartController chassis) is set to the following:

**SW1 SW2 SW3 SW4**

OFF OFF OFF OFF

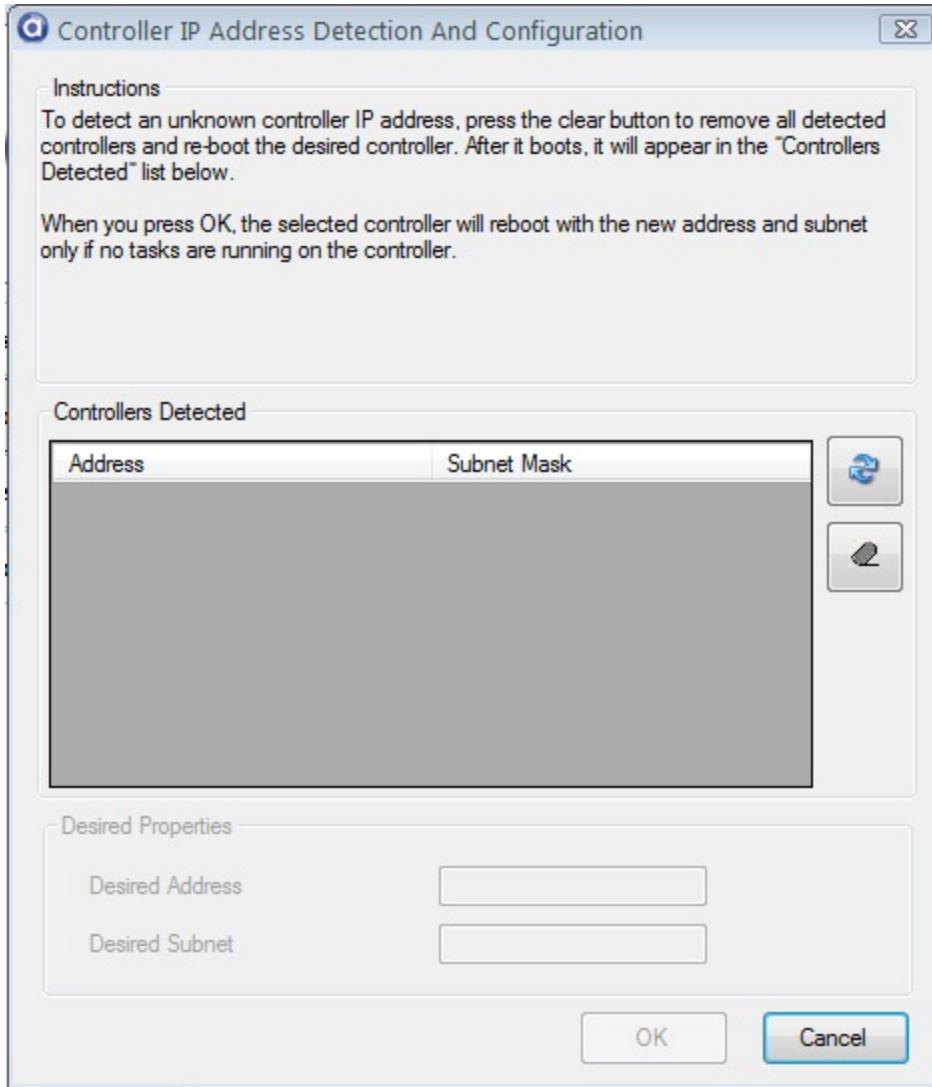
3. Connect the Omron Adept controller and your PC using the Omron Adept-supplied crossover cable. Optionally, you can connect the PC to the controller using a network hub and two Ethernet cables.

**NOTE:** DO NOT use the crossover cable with a network hub.

4. Install the ACE software on the PC. See [Installing the Software](#) on page 28 for details.
5. Determine the IP address and subnet mask of your PC by looking at the TCP/IP properties of Network Settings in the Windows Control Panel. Alternatively, you can determine the IP address of your PC by using the "ipconfig" command in the Windows Command Prompt window. See your Windows operating system documentation for details.
6. Based on the PC's IP address and subnet mask determined above, define a controller IP address, which will be configured in the next step.: If the PC's IP address is 192.9.225.80 and the subnet mask is 255.255.255.0, then the common network number is 192.9.225.xxx. In this case, the controller's IP address should be defined as 192.9.225.xxx, where xxx is a number between 0 and 255 that defines a unique IP address on the network. The subnet mask must be defined as 255.255.255.0.

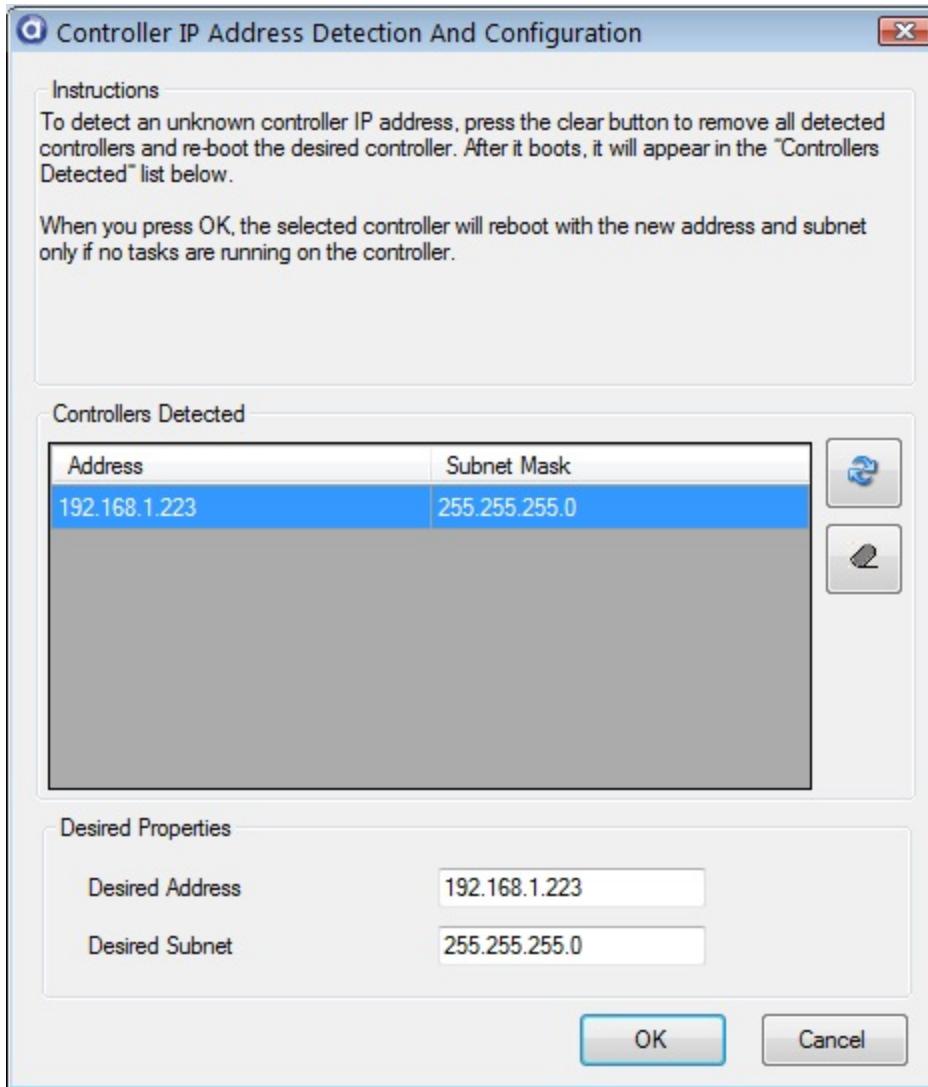
**NOTE:** If your controller will be connected to your corporate network, you may need to contact your network administrator to obtain a controller IP address.

7. Start the ACE software by selecting **Start > Programs > Omron > ACE > ACE** from the Windows task bar; or, double-clicking the ACE icon on the Windows desktop. The ACE Startup dialog box opens.
8. On the ACE Startup dialog box, click the Search () icon. The Controller IP Address Configuration dialog box opens.



*Controller IP Address Configuration Dialog*

9. Start the Omron Adept controller. After the controller start-up completes, the IP address will be detected and displayed in the Controllers Detected field.



*Detected Controller IP Address*

10. Enter the desired IP address and desired subnet mask into the corresponding fields.
11. Click **OK** to restart the controller. The controller restarts and the new IP address and subnet mask are assigned. After the controller restarts, the ACE Startup dialog box opens with the assigned controller IP address.

*Assigned Controller IP Address*

12. Click **Open** to connect to the Omron Adept controller. The ACE software completes the connection to the specified controller. The Omron Adept controller is now ready for use.

## Configuring Serial Communications

This section discusses the serial communication link between your PC and the Adept Cobra i600/i800 robot.

**NOTE:** If your system requires an Ethernet communications link, see Configuring Ethernet Communications on page 32 for connection details.

### Required Cabling and Hardware

Please see the [\*Adept Cobra i600/i800 Robot User's Guide\*](#) for details.

### Configuring the Serial Protocol

On the Cobra i600/i800 robot, the serial port is automatically configured by the MicroV+ operating system. Therefore, you do not need to manually configure that serial protocol.

You are now ready to run ACE. For details, see Running ACE for the First Time on page 37.

## Running ACE for the First Time

This topic describes how to start the ACE software and load a workspace.

### Starting the ACE Software

To start the ACE software:

1. Verify that the Omron Adept controller you want to connect to is on the same network as your computer. Note that you can also run the ACE software in simulation mode (without a controller).
2. From the Windows task bar, select:

**Start > Programs > Omron > ACE > ACE**

Optionally, you can double-click the "ACE" icon on the Windows desktop.



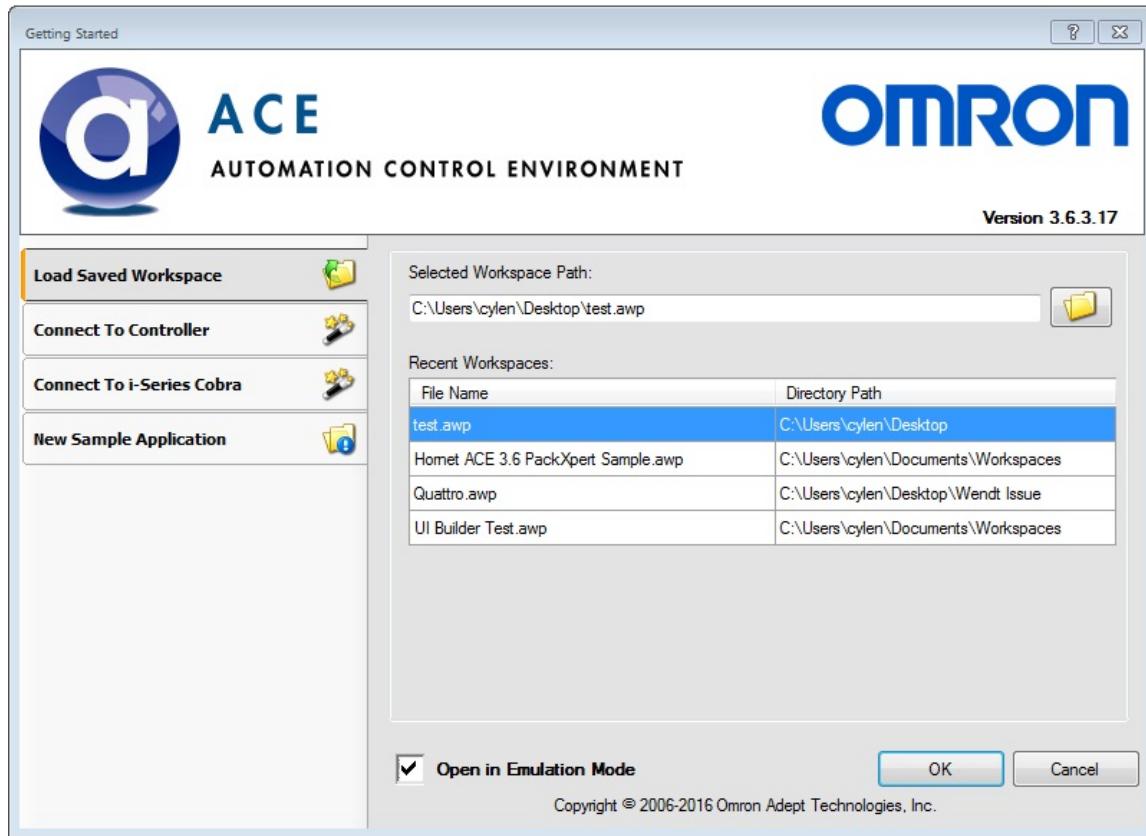
*ACE Icon*

The ACE splash screen opens, which shows the software version number and Status messages.



*ACE Splash Screen*

The ACE Getting Started dialog box opens.



ACE Getting Started Dialog Box

## Starting a Localized Version of the ACE Software

The ACE software allows you to run a localized (translated) version of the interface. The available languages are shown on the Localized Shortcuts menu.

To start a localized version of the ACE software:

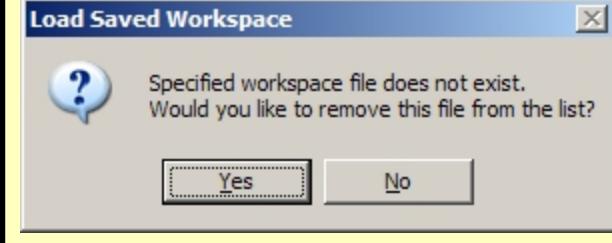
1. Verify that the Adept controller you want to connect to is on the same network as your computer. Note that you can also run the ACE software in simulation mode (without a controller).
2. From the Windows task bar, select:

**Start > Programs > Omron > ACE > Localized Shortcuts > [Select the desired language]**

## Loading or Creating a Workspace

This section describes how to load or create a workspace during the ACE software startup. For instructions on loading a workspace after another workspace has been loaded, see [Workspace Loading and Unloading](#) on page 52.

After the ACE Getting Started dialog box has displayed, you will need to load a new or previously-saved workspace, create an example workspace, or create an empty workspace. To load or create a workspace, select one of the following:

Item	Description
Load Saved Workspace	<p>Loads a previously-saved workspace from disk. After selecting this option:</p> <ul style="list-style-type: none"> <li>• Select one of the recently-used workspaces from the Recent Workspace grid list, which displays up to six of the recently-used workspaces,</li> <li>• Or, click the Folder icon (📁) to locate a previously-saved workspace on your computer.</li> <li>• Click OK to load the selected workspace.</li> </ul> <p><b>NOTE:</b> If the workspace file has been moved, renamed, or deleted, the ACE software asks if you would like to remove the file from the recently-used workspace list.</p> 
Connect to Controller	Connects to a controller on your network.
Connect to i-Series Cobra	Connects to a Cobra i600 or i800 robot on your network. You will be shown a list of available Com ports, and asked to select a Com port for the connection.
New Sample Application	<p>Creates a workspace based on the selected sample application.</p> <ul style="list-style-type: none"> <li>• Select a sample workspace from the Select a sample list.</li> </ul> <p>The workspace is populated with the objects required for the selected sample. The samples contain interview wizards that will guide you through the steps for creating a working application.</p>

Item	Description
Open in Emulation Mode	This is a special operating mode for ACE in which the physical workcell items, such as robots, cameras, conveyors, cameras, encoders, etc.) are virtualized. For more details, see Emulation Mode on page 1089.

## Controller Connection Errors

When you are connecting to a controller through Ethernet, if the controller cannot be found on the network (for example, you have specified the wrong IP address), you will see the following message:



*Connection to Controller Error Message*

To correct the error:

1. Click **OK** to acknowledge the error message.
2. Locate the Search (🔍) icon on the ACE Getting Started dialog box.
3. Click the Search icon to open the Controller IP Address Configuration tool, which is used to detect the connected controller and configure the IP address. For more details on using the Controller IP Address Configuration tool, see Configuring Ethernet Communications on page 32.

## Starting ACE from the Windows Command Prompt

For some applications, you may want to start the ACE software from the Windows Command Prompt. If you need to do this frequently, you may want to create a shortcut on the Windows Desktop, which contains the required startup parameters. For details, see Creating Shortcuts for Starting ACE on page 42.

---

To start the ACE software from the Windows Command Prompt:

1. Open the Windows Command Prompt. From the Window Start menu, select:

**Start > Programs > Accessories > Command Prompt**

The Command Prompt window opens.

2. Use the Windows "cd" command to navigate to the default ACE program folder:

```
cd c:\program files\Omron\ACE\bin\
```

**NOTE:** If you have installed the ACE software in a different folder, you will need to make the corresponding changes to the path shown.

3. To start the ACE software without any options, type:

```
ace.exe
```

Optionally, you can add one or more parameters to the command (parameter options are shown in square brackets). The syntax is:

```
ace.exe culture=myCulture datafile=path\filename.awp client_server[=name@port]
server[=name@port]
client[=name@ipAddress:port] loadui=uiformname nosplash noopc timeout
```

The parameters and their descriptions are shown in the following table.

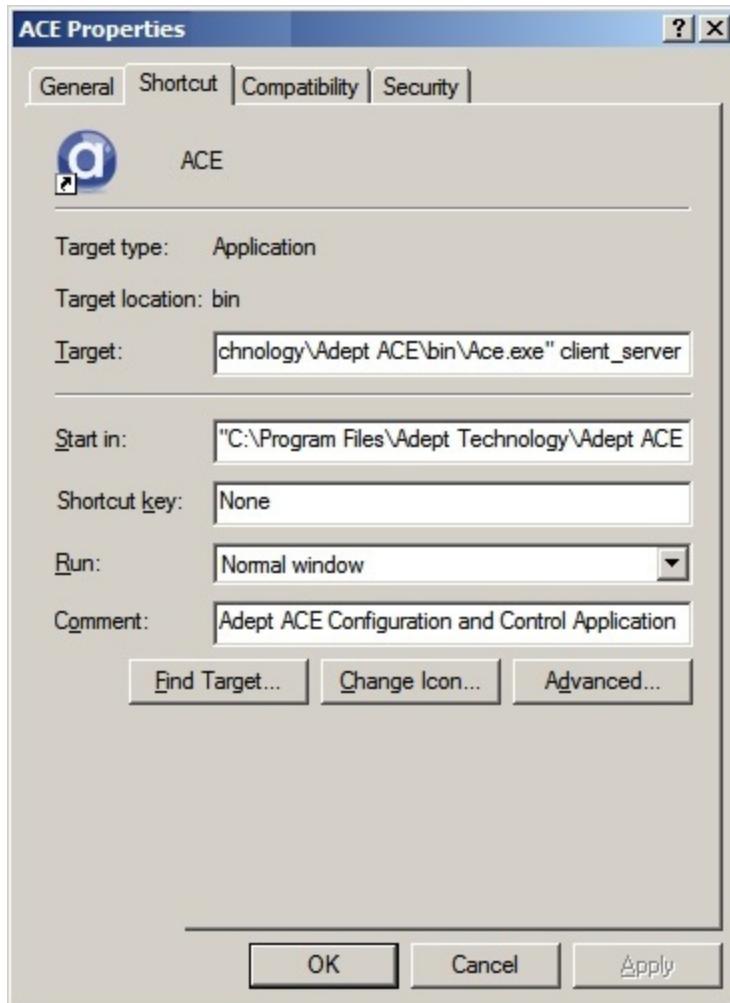
Parameter	Description
culture	Configures the ACE user interface for a specific culture (language), such as "de" (German). For example, to start ACE with the German interface, use:  <code>ace.exe culture=de</code>
datafile	Loads the specified data file. For example, to load the data file "c:\myfile.awp", use:  <code>ace.exe datafile=c:\myfile.awp</code>  <b>NOTE:</b> If your path or file names contain spaces, you must put the string in quotes:  <code>ace.exe datafile="c:\my file.awp"</code>
client_server[=name@port]	Starts the ACE software in dual-process, client-server mode. An optional name and port can be included. For example:  <code>ace.exe client_server=ace@43434</code>
server[=name@port]	Starts the ACE software in server mode without a user interface. An optional name and port can be included. For example:

Parameter	Description
	<code>ace.exe server=aces@43434</code>
<code>client[=name@ipAddress:port]</code>	<p>Starts a GUI to a remote ACE instance. An optional name, IP address, and port can be included. For example:</p> <pre>ace.exe client=ace@192.168.1.63:43434</pre> <p><b>NOTE:</b> If an argument is not provided, the client connects to a local instance.</p>
<code>loadui</code>	Loads the specified user interface form. For details, see Deploying the Custom User Interface on page 1082.
<code>nosplash</code>	Starts the ACE software but does not show the splash screen. For example:
	<code>ace.exe nosplash</code>
<code>noopc</code>	<p>Starts the ACE software but does not start OPC. For example:</p> <pre>ace.exe noopc</pre> <p>For details on OPC, see OPC Data Access and Process Control on page 785.</p>
<code>timeout</code>	Overrides the default connection timeout of 120 seconds.
	<code>ace.exe client_server timeout=200</code>
<code>help</code>	Displays the help file for the command-prompt startup modes.

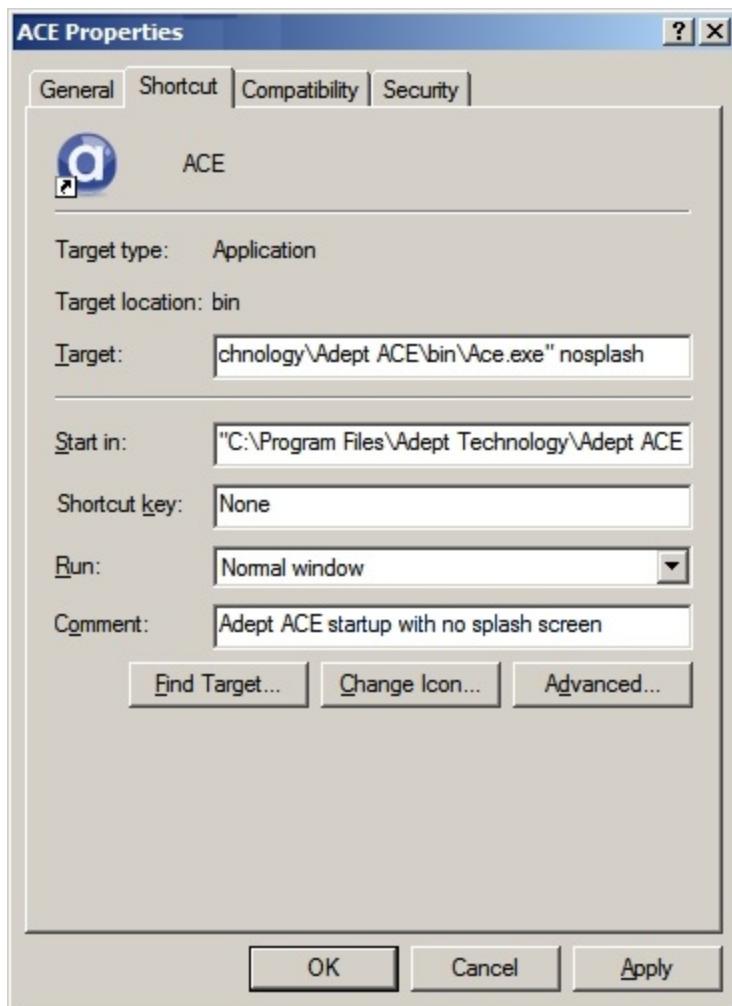
## Creating Shortcuts for Starting ACE

If you are frequently using the Windows Command Prompt for starting ACE, you may want to create shortcuts on your Windows Desktop to make the startup process easier. This section briefly describes the process for creating a shortcut. For more details on creating shortcuts, refer to the Microsoft Windows online help documentation.

1. Copy the existing ACE shortcut on the Windows Desktop. To copy the shortcut:
  - a. Right-click on the shortcut and select Copy from the menu.
  - b. Right-click on the Windows Desktop and select Paste from the menu. A copy of the shortcut is created on the Windows Desktop.
2. Customize the properties to start ACE with the desired options.
  - a. Right-click on the copied shortcut and select Properties from the menu. The shortcut Properties dialog opens.



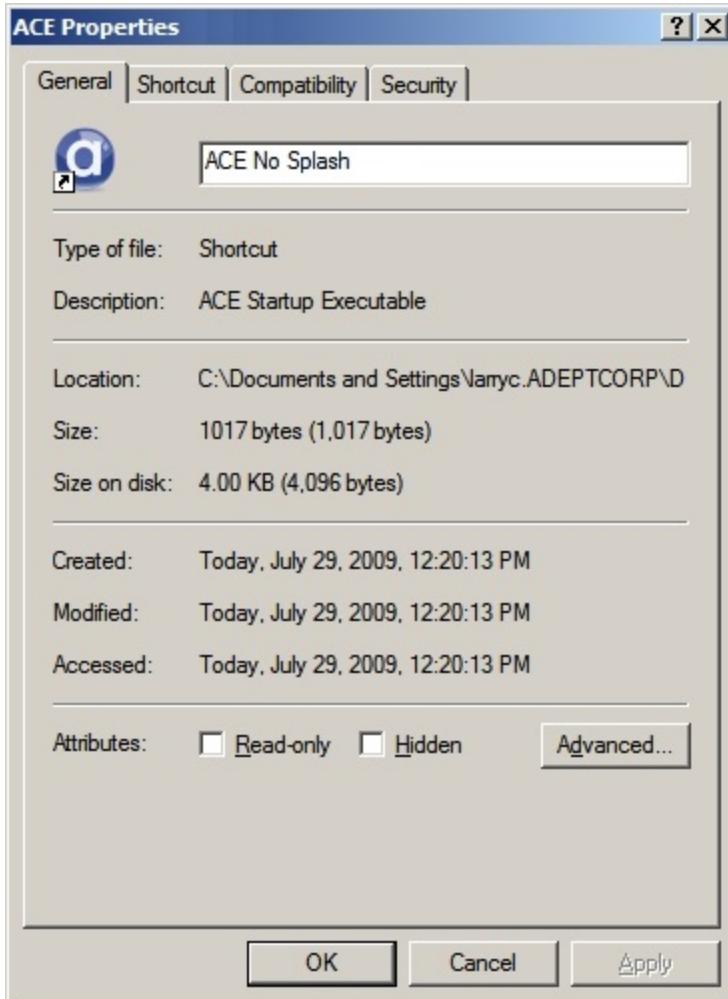
- b. In the Target field, edit the startup command parameters with the desired options, as described in the previous section. For example, to start ACE without the splash screen, you would edit the Target field to:  
`"C:\Program Files\Omron\ACE\bin\Ace.exe" nosplash`
- c. Optionally, you can edit the Comment field with a description for the shortcut.
- d. Click **Apply** to save your changes. The edited Properties page is shown in the following figure.



3. Rename the shortcut.
  - a. Click the General tab. The General page opens.
  - b. Edit the field next to the icon with a name you would like to use for the new shortcut. For example, for the "no splash" example above, you could use:

ACE No Splash

The edited General page is shown in the following figure.



4. Click **OK** to save the changes. The shortcut Properties dialog closes. The finished shortcut is updated on the Windows Desktop, as shown in the following figure.



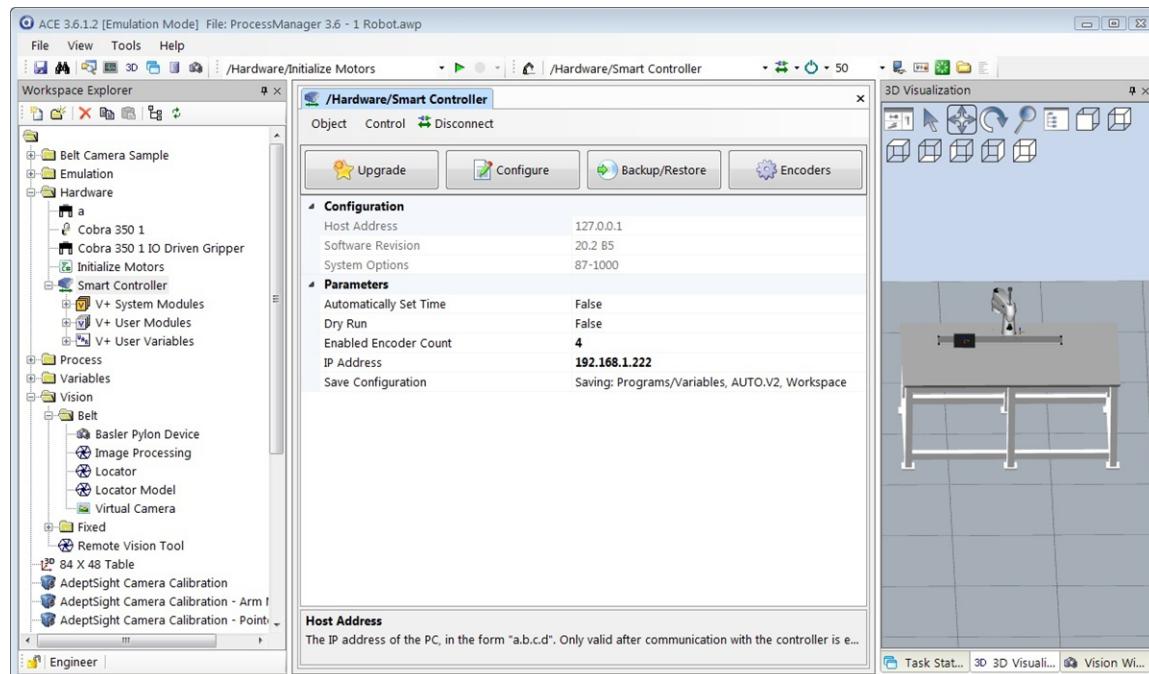
## Understanding the Interface

This topic describes the ACE main user-interface items.

### The Main User Interface

After you have selected a method for creating a new workspace, click **Open** to access the ACE main user interface shown in the following figure. The actual appearance may vary, depending on how the elements of the user interface are enabled and docked in the main window. Note that you can dock and undock the windows (for example, the Workspace Explorer window or the Task Status window) by dragging the title bar or tab.

Details on the main components of the user interface can be found in the topic [User Interface](#).



ACE Main User Interface

You are now ready to start exploring object properties, configuring the system, executing programs, and so on. For more details, see the topic [Workspace Explorer](#).

### Icons in the Workspace

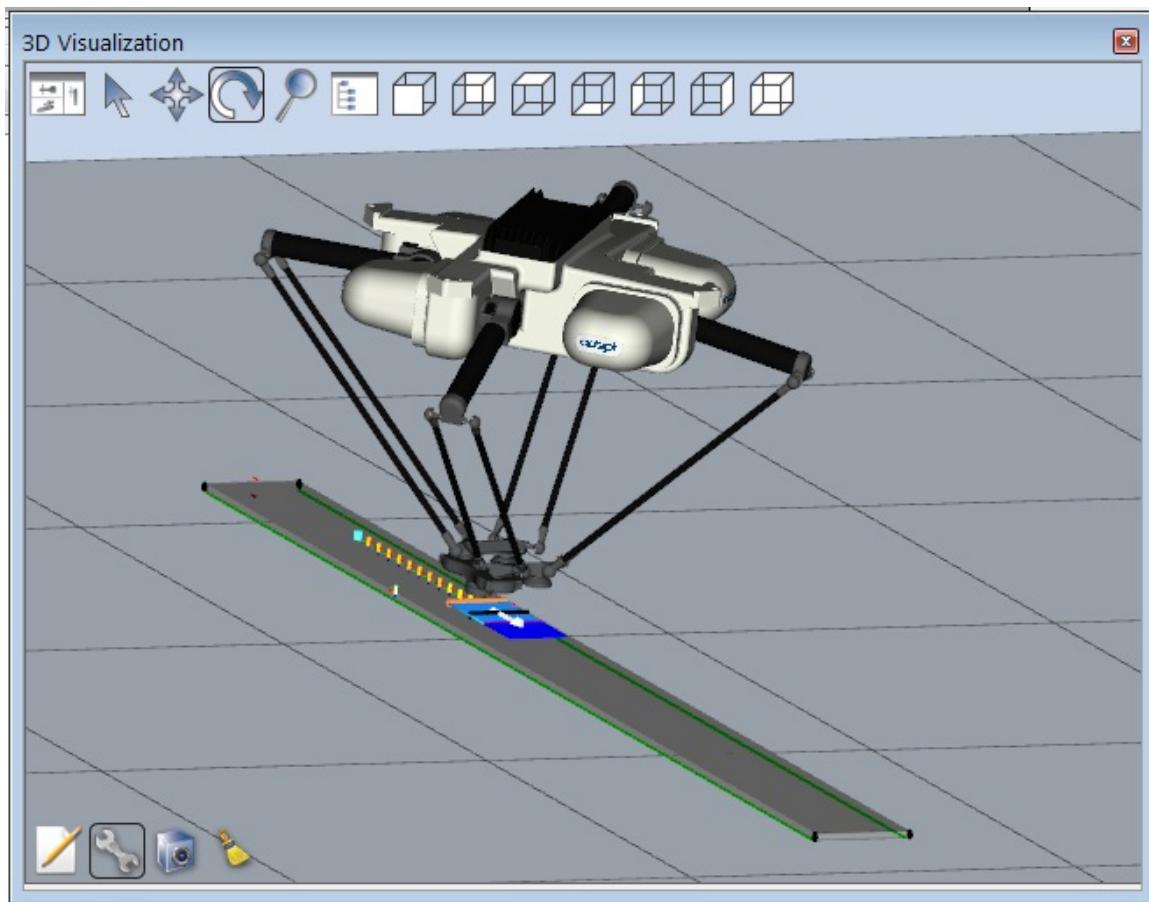
All Workspace Explorer objects and many menu and toolbar functions have corresponding icons. The state of the object or function is represented by its icon, as follows:

<b>State</b>	<b>Example</b>	<b>Description</b>
<b>Workspace Object Icons</b>		
Normal (colored) icon		The feature or object is available.
Dimmed (gray) icon		The feature or object is not available. (This typically means that the required license was not detected for the object or function.)
<b>Toolbar Icons</b>		
Normal (colored) icon		The feature or object is available.
Dimmed (gray) icon		The feature or object is not available. (This typically means the ACE software is not connected to a controller.)
<b>Controller and Cobra i600/i800 Icons</b>		
Green underlined icon		The controller or robot object is connected.
Red underlined icon		The controller or robot object is not connected.
<b>Process Manager Icons</b>		
Exclamation mark on icon		The process needs to be taught or has configuration errors.
Green arrowhead on icon		The process is running.

## The 3D Visualization Window

**NOTE:** When the ACE software is started, it will check to see if hardware acceleration is available. If it is not available, it will display an icon on the toolbar indicating that 3D Visualization is not available.

The 3D Visualization window allows you to see a 3D visualization of your workcell. To access the 3D Visualization window, click the 3D ( icon on the toolbar. A 3D view of your workspace is displayed, as shown in the following figure. Note that when the system is connected to a live controller, the robot position is shown real-time in the display.

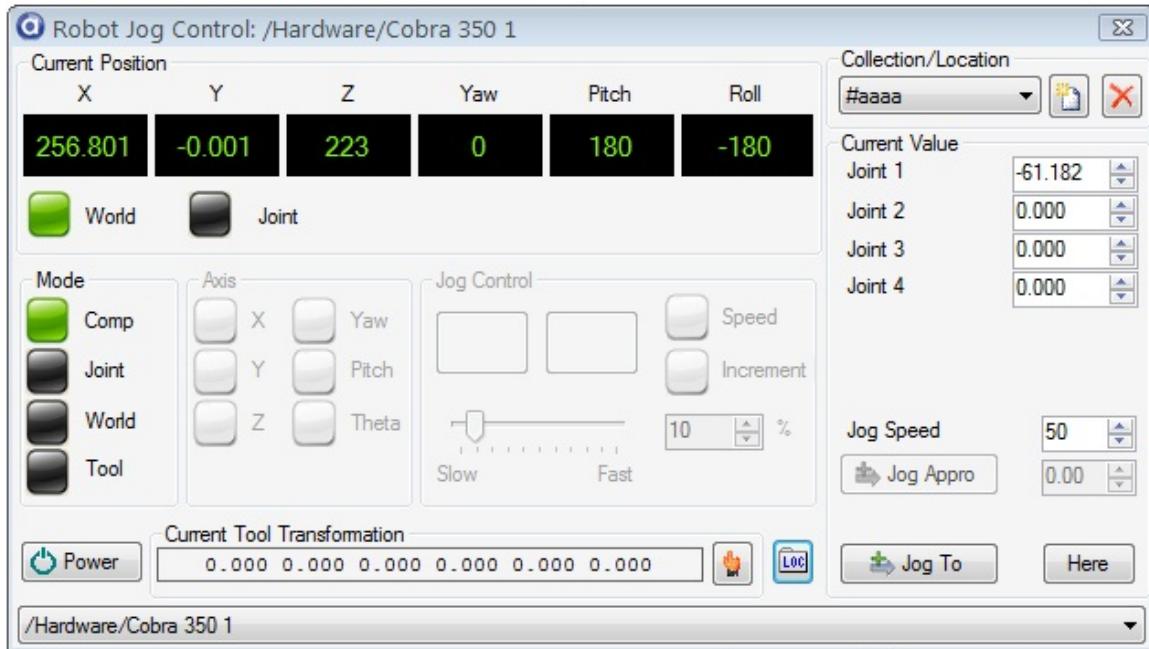


*Simple 3D Visualization*

For more details, see 3D Visualization on page 176.

### **The Robot Jog Control**

The Robot Jog Control, shown in the following figure, is a "virtual" pendant, which is used to position the robot.



### *Robot Jog Control*

You can access the Robot Jog Control in several ways:

- Click the Robot (  ) icon on the toolbar,
- Right-click the robot in the 3D display and select the Robot (  ) icon from the shortcut menu, or
- Choose "Pendant" from any object that represents a location. For more details, see Robot Jog Control on page 163.

## Programming Tools and Editors

The following programming tools and editors are available in the ACE software.

Item	Description
Process Manager	The Process Manager is the central control point for developing sophisticated ACE PackXpert packaging application. For details, see Process Manager on page 843.
V+ Editor	The V+ Editor/Debugger is an online, interactive V+ program editor and debugger. The V+ Editor/Debugger performs syntax checking and formatting while you are programming. For details, see V+ Editor Tool on page 765.

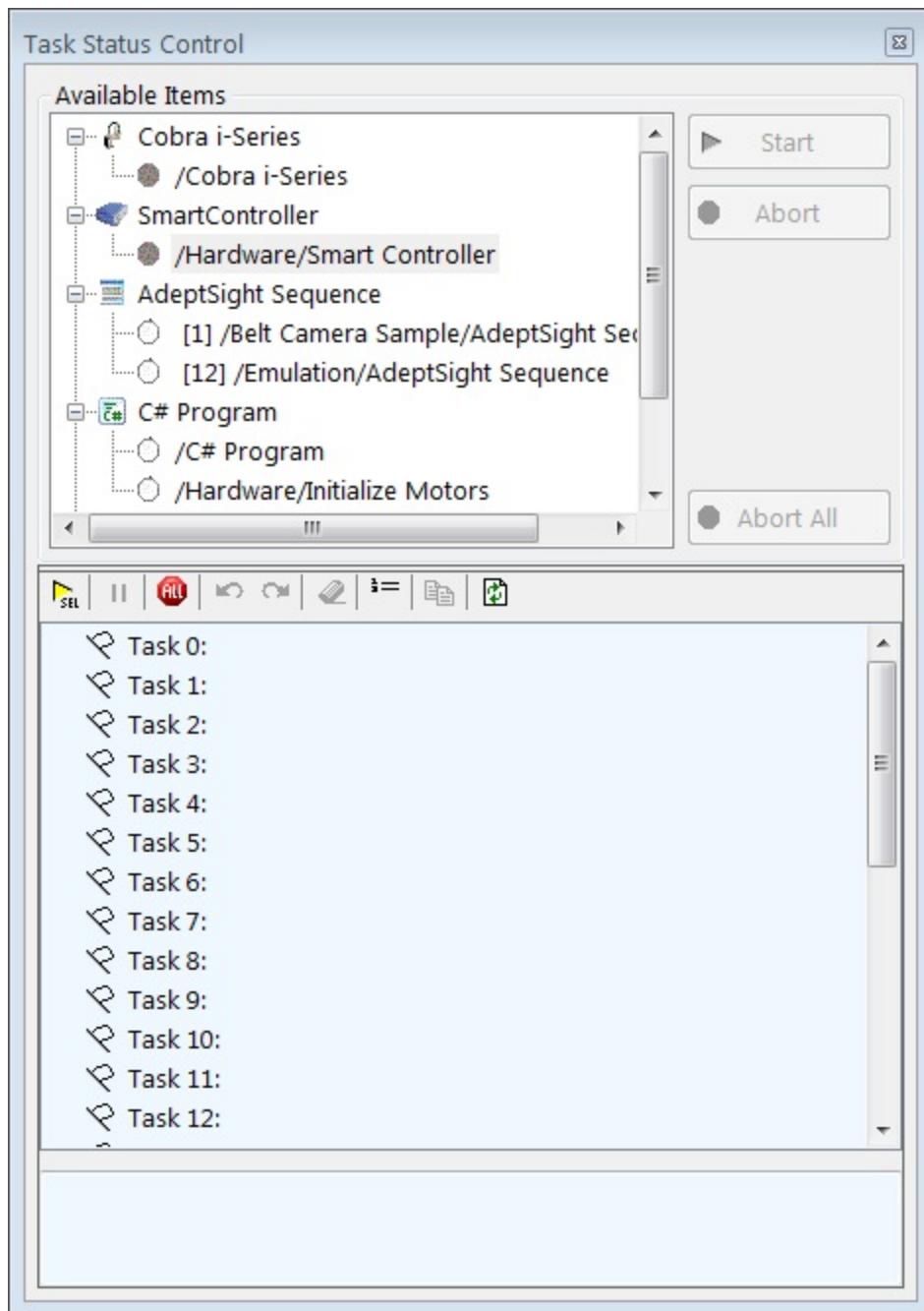
Item	Description
C# Editor	The ACE software provides a basic C# program editor, which can be used to create and edit programs. For details, see C# Language Programming on page 750.
OPC Data Communications	The OPC Data Communications tool can be used to communicate the values of specified V+ global and local variables. For details, see OPC Data Access and Process Control on page 785.

## Task Status Control

The Task Status Control provides an interface for controlling and monitoring the execution of one or more programs (such as C# Script objects) that run on the PC or the controller.

## Task Status Control

---



### Task Status Control

For more details, see Task Status Control on page 747.

## Workspace Loading and Unloading

ACE allows you to load or unload the entire contents of the workspace, which makes it possible to quickly switch between applications or product lines.

Before a new workspace can be loaded, the current workspace must first be saved. This is handled automatically, through user prompts, during the loading procedure. Files can also be saved through the File menu. For details, see the section [File Menu](#).

### Loading a Workspace

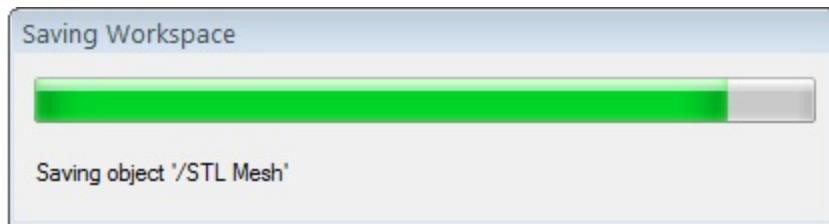
To load an ACE workspace:

1. From the ACE menu bar, select **File > Load**. You are prompted to Save the Current Workspace.

**NOTE:** If there are no changes to the workspace, you will not be prompted to Save the Current Workspace.

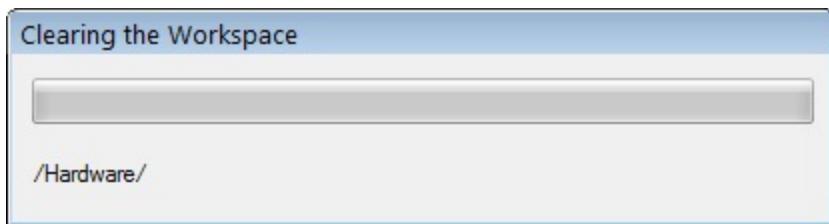
2. Select **Yes** to save the workspace; select **No** to proceed without saving the workspace.

While the workspace is saving, a progress indicator shows the status of the operation.



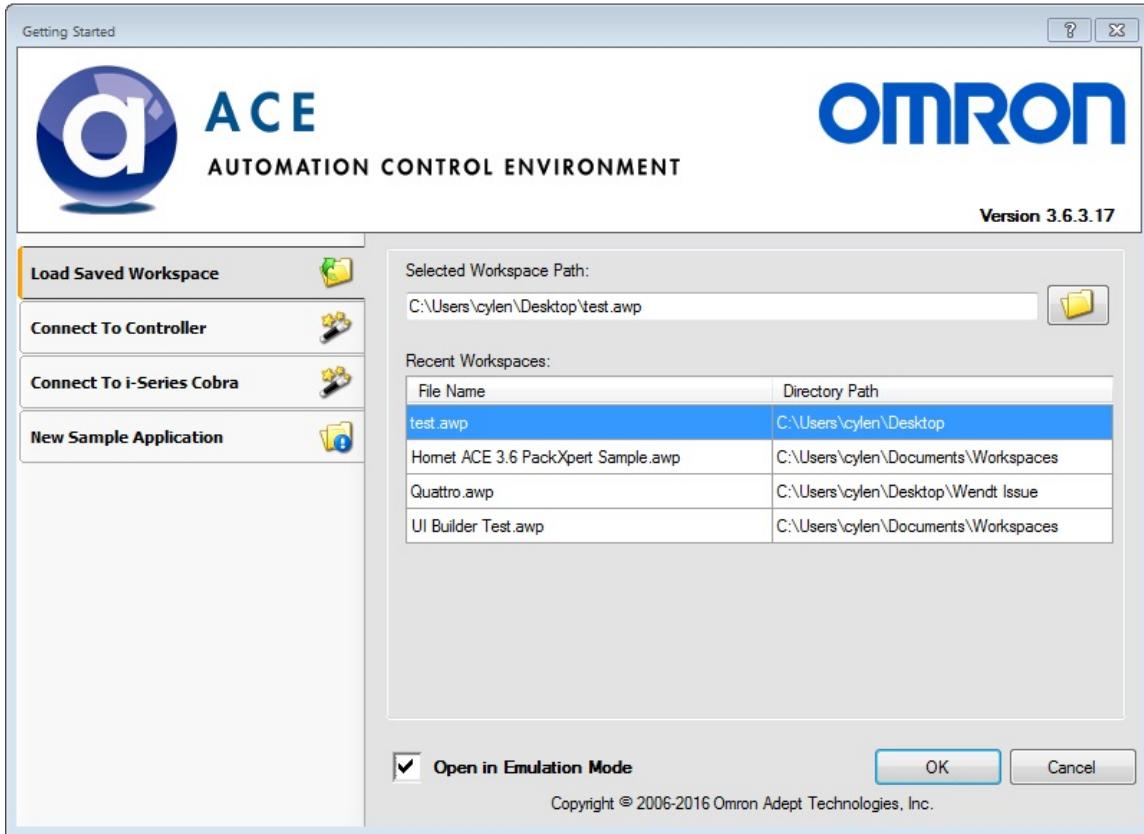
*Saving Workspace Progress Indicator*

While the workspace is unloading, a progress indicator shows the status of the operation.



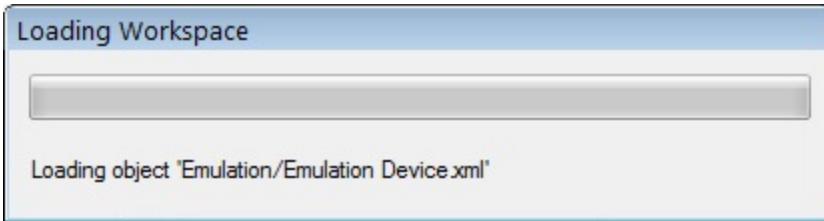
*Clearing the Workspace Progress Indicator*

3. The ACE Getting Started dialog opens.



*ACE Getting Started Dialog Box*

4. On the Getting Started dialog, select Load Saved Workspace.
5. Use the Recent Workspaces list to select the desired workspace file from the list of recently-used files. You can use your mouse to select the file, or use the up/down arrow keys on your keyboard to move to the desired file.  
If the desired workspace file is not on the list, click the Folder ( icon to locate the file.
6. If you wish to open the workspace in emulation mode, select the Open in Emulation Mode checkbox. The ACE software remembers your selection, and applies it the next time you load that workspace.
7. Click OK (or press the Enter key). The selected ACE workspace is loaded. If a V+ module conflict is found, a notification prompt is displayed. For details, see the section Workspace Loading and Unloading on page 52. While the workspace is loading, a progress indicator shows the status of the operation.



*Loading Workspace Progress Indicator*

## Dragging an AWP File into the Workspace

You can open or add the contents of a saved workspace (AWP file) by dragging the file into an open ACE workspace.

1. Open an existing workspace using the instructions in the previous section.  
OR  
Open a new default workspace. For details, see Running ACE for the First Time on page 37.
2. Click the File Explorer (📁) icon on the ACE toolbar and navigate to the folder where you store your ACE workspace (AWP) files. For details on the File Explorer, see File Explorer on page 168.
3. Click the desired file and drag it onto the folder pane of the ACE workspace.

**NOTE:** If an object in the added AWP file uses the same name as one in the current workspace, that object will not be loaded. If this condition occurs, the ACE software displays a message stating the number of objects that are unable to load.

## Saving a Workspace

After you have made changes to your ACE workspace, you will want to save it periodically. You can save the ACE workspace using:

- the Save icon (💾) on the ACE toolbar
- the **File > Save** or **File > Save As** menu items.

For details on saving your ACE workspace, see Save Workspace on page 161.

In addition, individual folders and objects in the workspace can also be saved as ZIP or AWP (ACE Workspace Project) files and reused in other ACE projects. For more details, see the section Shortcut Menus on page 76.

**NOTE:** If you attempt to close a workspace, or to close the ACE software, but have not saved the changes, you will be prompted to save before closing.

## User and Startup Configuration

The ACE software contains tools for configuring system user access and system startup.

### Configuring User Access

System user access is configured through the User Manager editor. This tool allows you to create a list of system users and assign an access level to each user. To access the User Manager Editor, you must first create a User Manager object in the Workspace Explorer.

To create a User Manager object, right-click on a folder and select **New > Configuration > User Manager**. A User Manager object is added to the Workspace Explorer and the User Manager Editor opens. For more details, see the topic [User Manager Editor](#).

**NOTE:** Only one User Manager object is permitted in the Workspace Explorer.

### Configuring System Startup

System startup is configured through the System Startup editor, which allows you to specify various ACE software start-up options. For example, you can specify an auto-run file that will be executed when the ACE software is connected to a controller. To access the System Startup Editor, you must first create a System Startup object in the Workspace Explorer.

To create a System Startup object, right-click on a folder and select **New > Configuration > System Startup**. A System Startup object is added to the Workspace Explorer and the System Startup editor opens. For more details, see the topic [System Startup Editor](#).

**NOTE:** Only one System Startup object is permitted in the Workspace Explorer.



## Robot Concepts

The topics in this chapter describe basic robot concepts that you should be familiar with when using the ACE software.

The Coordinate System .....	58
Basic Robot-Motion Parameters .....	61
Understanding Conveyors (Belts) .....	66
Defining a Pallet Layout .....	68

## The Coordinate System

This section describes the coordinate system used by Adept robots.

### Locations and Precision Points

A robot location specifies the position and orientation of the robot tool tip in three-dimensional space. A robot location is a predefined position at which a robot will acquire or place a part or move through en route to acquiring or placing a part. Locations have controllable characteristics that produce smooth, efficient robot operation. You can define robot locations using Cartesian coordinates (transformations), or Joint values (Precision Points).

#### *Transformations*

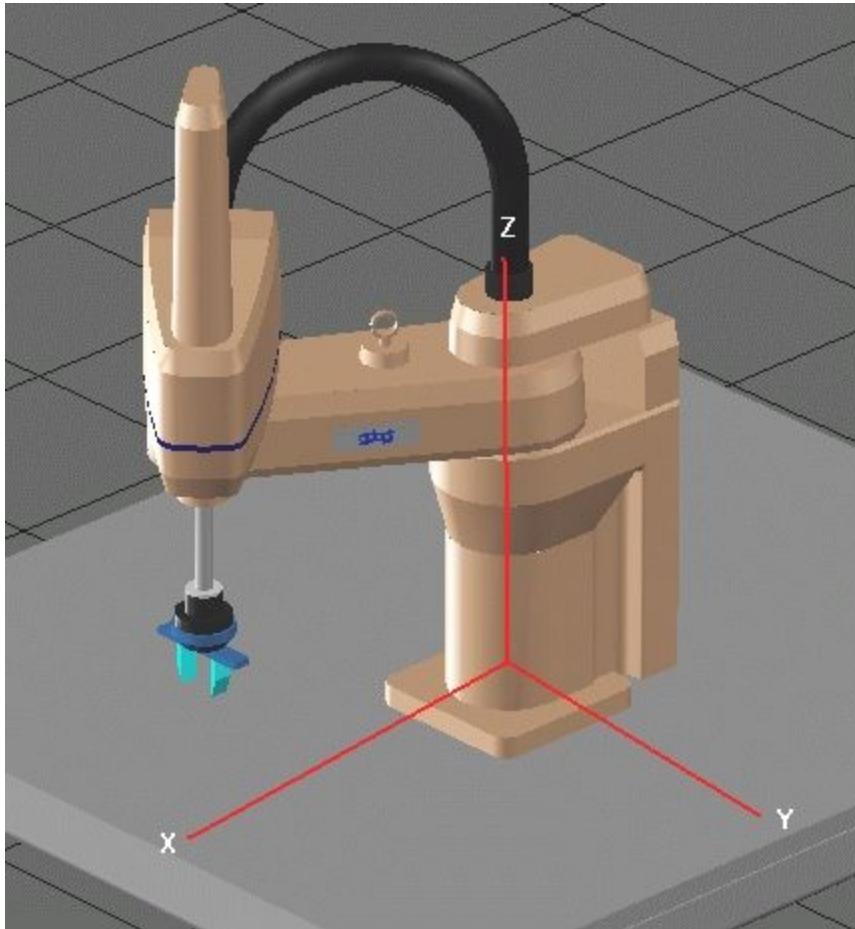
A transformation is a set of six components that uniquely identifies a location in Cartesian space and the orientation of the robot tool flange. A transformation can also represent the location of an arbitrary local reference frame (also known as a "frame of reference").

#### *Precision Points*

A precision point allows you to define a location by specifying a value for each robot joint. These joint values are absolute with respect to the home sensors of the robot, and cannot be made relative to other locations or coordinate frames. Precision point locations are useful for jointed-arm applications and with applications that require full rotation of Adept arms with Joint 4. They are also helpful where joint orientation is critical to the application, or when you want to move an individual joint.

#### *Location Origin*

Locations are, by default, relative to the base of the robot. See the following figure for an example using an Adept SCARA robot. For a default (World) location, the coordinate offsets are from the origin of the World coordinate system (which is located at the base of the robot).

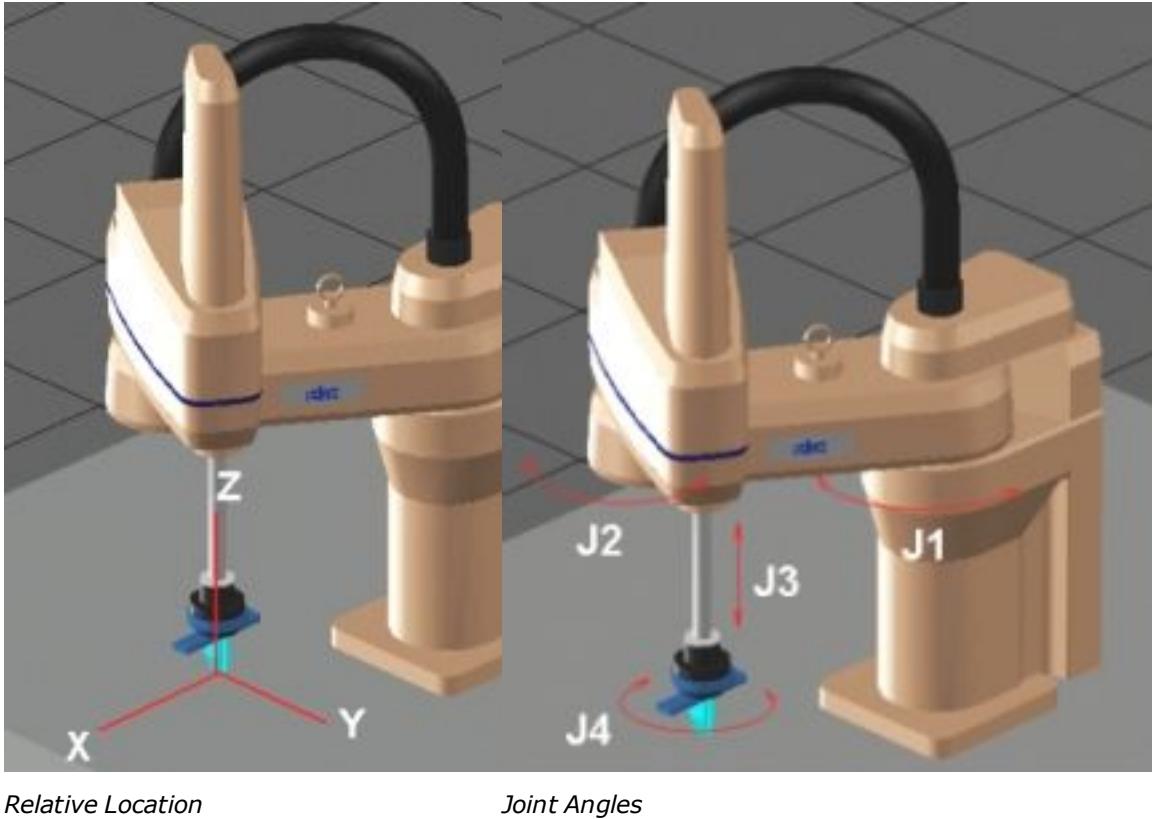


*World Coordinate System*

## Defining a Location

Locations can be entered manually or generated and stored automatically by ACE. To store a location, you can use the Robot Tool (see Robot Jog Control on page 163 for details) to place the robot at the correct location and orientation and then click **Here**. Location values are automatically calculated (based on the current robot location) and then stored.

**NOTE:** If you create a location that is incorrect, it can be changed by placing the robot in the correct location and clicking **Here**. If you are teaching a location from within an ACE wizard, you can typically click **Back** and then reteach the location.



*Relative Location*

*Joint Angles*

## Workspace Coordinates

The ACE software introduces a new feature called "workspace coordinates". The workspace coordinates use the offset defined in the Offset From Parent item for each robot. You can pick a "parent" (a point in the system that all robots will be relative to) and set the Offset From Parent item for each robot in the system. This allows you to have a common workspace coordinate system for all robots in the system.

Additionally, the ACE software contains a Workspace Reference calibration wizard, which relates the positions of all robots and conveyor belts used in an application to a common workspace coordinate system. For details, see Workspace Positioning on page 945.

## Basic Robot-Motion Parameters

When programming a robot, there are several factors that play an important part in performance. In order to achieve optimum performance from your robot, it is helpful to have an understanding of how these factors work.

### Speed, Acceleration, and Deceleration

Robot speed is usually specified as a percentage of normal speed, not as an absolute velocity. The speed for a single robot motion is set in the Speed parameter of the Pick Motion Parameters or Place Motion Parameters dialogs for each Part or Part Target location. Note that the result obtained by the speed value depends on the operating mode of the robot—joint-interpolated versus straight-line (for details, see Joint-Interpolated Motion vs. Straight-Line Motion on page 64). Whether in joint-interpolated mode or straight-line mode, the maximum speed is gated by the slowest-moving joint during the motion, since all the joints are required to start and stop at the same time. For example, if a given motion requires that the tool tip is rotated during the motion, which requires Joint 4 to rotate, that joint could gate the maximum speed achieved by the other joints, since Joint 4 is the slowest-moving joint in the mechanism. Using the same example, if Joint 4 was not rotated, the motion could be faster without any change to the speed value.

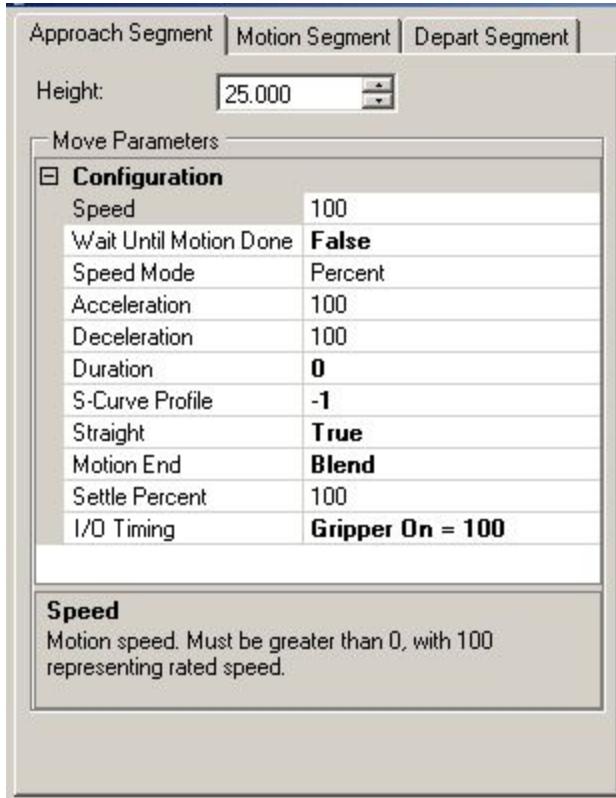
**NOTE:** The motion speed specified in the Pick Motion Parameters or Place Motion Parameters dialogs must always be greater than zero for a regular robot motion. Otherwise, an error will be returned.

You can use the Acceleration parameter to control the rate at which the robot reaches its designated speed and stops. Like speed, the acceleration/deceleration rate is specified as a percentage of the normal acceleration/deceleration rate. To make the robot start or stop smoothly (using lower acceleration and deceleration for a less-abrupt motion), set the Acceleration parameter to a lower value. To make the robot start or stop quickly (using higher acceleration and deceleration for a more-abrupt motion), set the Acceleration parameter to higher values.

The Speed and Acceleration parameters are commonly modified for cycle-time optimization and process constraints. For instance, abrupt stops with a vacuum gripper may cause the part being held to shift on the gripper. This problem could be solved by lowering the robot speed. However, the overall cycle time would then be increased. An alternative is to slow down the acceleration/deceleration rate so the part does not shift on the gripper during motion start or stop. The robot can still move at the maximum designated speed for other movements. Another case would be a relatively high payload and inertia coupled with tight positioning tolerances. A high deceleration rate may cause overshoot and increase settling time. Higher acceleration/deceleration rates and higher speeds don't necessarily result in faster cycle times.

### Approach and Depart

Approach and depart heights are typically specified for pick and place locations. The Approach Segment parameters are shown in the following figure:



### Approach Segment

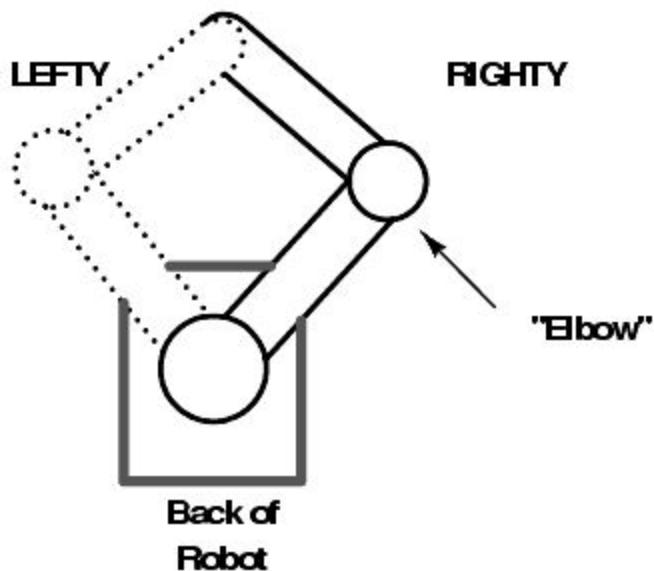
When approach and depart heights are specified, the robot moves in three distinct motions. In the first motion (Approach segment), the robot moves to a location directly above the specified location. For the second motion, the robot moves to the actual location and the gripper is activated. In the third motion (Depart segment), the robot moves to a point directly above the location.

Approach and depart heights are used to make sure that the robot approaches and departs from a location without running into any other parts in the assembly being built (or any other obstructions in the robot envelope). Approaches and departs are always parallel to the Z-axis of the tool coordinate system. Notice that all the motion parameters that apply to a motion to a location also can be applied to approach and depart motions. This allows you to move at optimum speed to the approach height above a location, then move more slowly when actually acquiring or placing the part, and finally depart quickly.

### Arm Configuration

Another motion characteristic that you can control is the configuration of the robot arm when moving to a location. However, configuration options apply only to specific types of robots. For example, the lefty/righty option applies to SCARA-type robots (such as the Adept Cobra robots), but the above/below option does not apply to those robots. The arm configuration is specified in the Move Configuration parameters group of the Configuration Items. For details, see Configuration Items on page 856.

The following figure shows how a SCARA robot can reach a point with the elbow pointing in two different directions.



*Left/Right Robot Arm Configuration*

## Location Precision

When a robot moves to a location, it actually makes several moves, each of which is a closer approximation of the exact location. You can control the precision with which the robot moves to a location using the Motion End (Settle Fine/Settle Coarse) parameter. If coarse is selected, the robot will spend less time attempting to reach the exact location. In many cases, this setting will prove completely adequate, and robot cycle times can be improved by setting location precision to coarse.

## Continuous-Path Motion

When a single motion instruction is processed, the robot begins moving toward the location by accelerating smoothly to the commanded speed. Sometime later, when the robot is close to the destination location, the robot decelerates smoothly to a stop at the location. This motion is referred to as a single motion segment, because it is produced by a single motion instruction.

When a continuous-path series of two motion instructions is executed, the robot begins moving toward the first location by accelerating smoothly to the commanded speed just as before. However, the robot does not decelerate to a stop when it gets close to the first location. Instead, it smoothly changes its direction and begins moving toward the second location. Finally, when the robot is close to the second location, it decelerates smoothly to a stop at that location. This motion consists of two motion segments since it is generated by two motion instructions.

Making smooth transitions between motion segments without stopping the robot motion is called continuous-path operation. If desired, the robot can be operated in a non-continuous-path mode, which is also known as "breaking continuous path" operation. When continuous-path operation is not used, the robot decelerates and stops at the end of each motion segment before beginning to move to the next location. The stops at intermediate locations are referred to as "breaks" in continuous-path operation. This method is useful when the robot must be stopped while some operation is performed (for example, closing the gripper or applying a dot of adhesive). The continuous or non-continuous path motion is set using the Wait Until Motion Done parameter and Motion End parameter in the Pick Motion Parameters or Place Motion Parameters dialogs. To enable continuous-path operation, you must set both parameters as follows:

- Wait Until Motion Done = False
- Motion End = Blend

**NOTE:** Breaking continuous-path operation affects forward processing (the parallel operation of robot motion and program execution). Program operation is suspended until the robot reaches its destination.

Continuous-path transitions can occur between any combination of straight-line and joint-interpolated motions (see "Joint-Interpolated Motion vs. Straight-Line Motion" in the following section).

## Joint-Interpolated Motion vs. Straight-Line Motion

The path a robot takes when moving from one location to another can be either a joint-interpolated motion or a straight-line motion. A joint-interpolated motion moves each joint at a constant speed (except during the acceleration/deceleration phases—see Speed, Acceleration, and Deceleration on page 61). With a rotationally-jointed robot, the robot tool tip typically moves along a curved path during a joint-interpolated motion. Although such motions can be performed at maximum speed, the nature of the path can be undesirable.

Straight-line motions ensure that the robot tool tip traces a straight line. That is useful for cutting a straight line, or laying a bead of sealant, or any other situation where a totally predictable path is desired.

**NOTE:** For X, XY, XYZ, or XYZT devices, straight-line motion and joint-interpolated motion result in identical paths, because the (positioning) joints all move in straight lines themselves.

The joint-interpolated or straight-line motion is set using the Straight parameter in the Pick Motion Parameters or Place Motion Parameters dialogs.

## Performance Considerations

Things that may impact performance in most applications include robot mounting, cell layout, part handling, and programming approaches.

### ***Robot Mounting Considerations***

The mounting surface should be smooth, flat and rigid. Vibration and flexing will degrade performance. Therefore, it is recommended that you carefully follow the robot-mounting procedures described in your robot user's guide. When positioning a robot in the workcell, take advantage of moving multiple joints for

faster motions. On a SCARA robot, the "Z" and "theta" axes are the slowest, and motion of these joints should be minimized whenever possible. This can be accomplished by positioning the robot, and setting conveyor heights and pick and place locations, to minimize Z-axis motion.

#### ***Cell Layout Considerations***

Regarding cell layout and jointed arms, the same point-to-point distance can result in different cycle times. Moving multiple joints combines the joint speeds for faster motion. If the same distance is traversed using motion of a single joint, the motion of that joint will be longer, and therefore will take more time.

#### ***Part Handling Considerations***

For part handling, settling time while trying to achieve a position can be minimized by centering the payload mass in the gripper. A mass that is offset from the tool rotation point will result in excess inertia that will take longer to settle. In addition, minimizing gripper mass and tooling weight will improve settling time. This could include using aluminum versus steel, and removing material that is not needed on tooling.

#### ***Programming Considerations***

The use of joint-interpolated versus straight-line motion has to be evaluated on a case-by-case basis. In general, joint-interpolated motion is more efficient. Nulling tolerances should be as loose as the application will permit. This has a direct impact on cycle time. Lastly, on jointed arms, changing the arm configuration (for example, lefty versus righty for a SCARA robot) generally requires more time than maintaining the current configuration during a motion.

## Understanding Conveyors (Belts)

This section describes some basic conveyor (belt) concepts.

**NOTE:** In the ACE interface, conveyors are referred to as "belts".

### **Indexing versus Tracking Conveyors**

There are two basic types of conveyor systems: indexing and tracking. In an indexing conveyor system (also referred to as a noncontinuous conveyor system), you specify either control signals or a time interval for the belt to move between stops. When the conveyor stops, the robot removes parts from the belt and the conveyor is signaled to move again. The conveyor must be equipped with a device that can use digital output to turn the conveyor on and off.

#### ***Indexing Conveyors***

Indexing conveyor systems are configured as either nonvision or vision. With a nonvision indexing system, the part must be in the same location each time the belt stops. In a vision-equipped indexing system, a fixed-mount camera takes a picture when the belt stops and the robot accesses any objects found.

#### ***Tracking Conveyors***

In a tracking conveyor system, the belt moves continuously and the robot tracks parts until the speed and location of the robot gripper match those of a part on the belt. The robot then accesses the part. Tracking conveyors must be equipped with an encoder, which reports the movement of the belt, and distance moved, to the ACE software.

Tracking conveyor systems are also configured as either nonvision or vision. With a nonvision tracking conveyor, a sensor signals that a part has passed a known location. The ACE software tracks the progress of the belt and accesses the part when it comes into the robot working area (belt window). Parts must always be in the same location with respect to the center line of the belt.

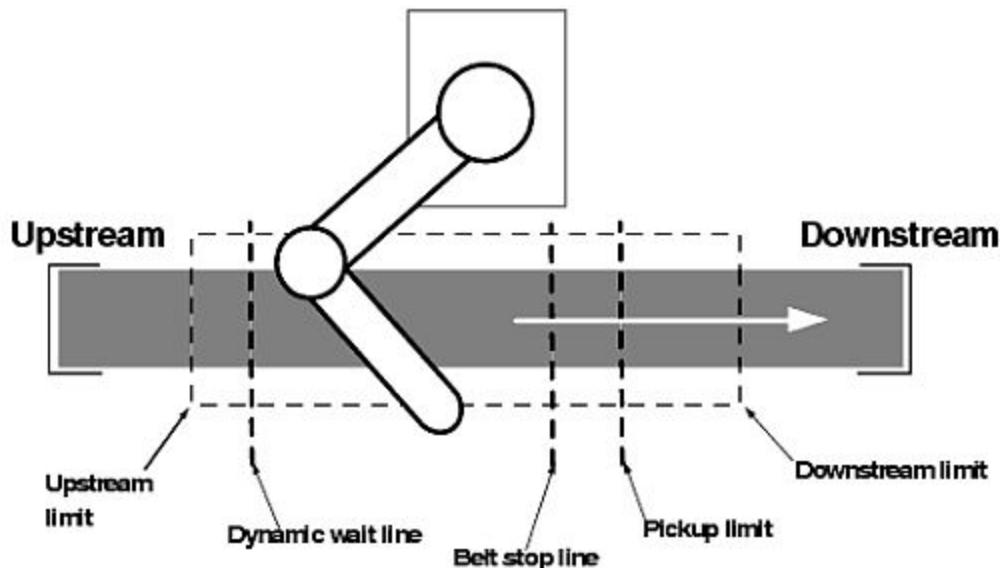
With a vision-equipped tracking conveyor, a fixed-mount camera takes pictures of the belt based on the distance the conveyor travels, and returns the location of parts it locates. These part locations are queued and accessed by the robot. The vision system allows detection of parts that are randomly positioned and oriented on the belt.

#### ***Conveyor Calibration***

Belt calibration, a simple procedure performed from within the ACE software, relates belt direction of travel to the robot work space. Calibration establishes the relationship between an encoder tick and the distance moved by the belt. Once a belt is calibrated, it can be used by the application. For more details on belt calibration, see Belt Calibrations on page 848.

### **Conveyor Terminology**

The following figure shows a typical robot cell equipped with a conveyor belt.

*Conveyor Terminology*

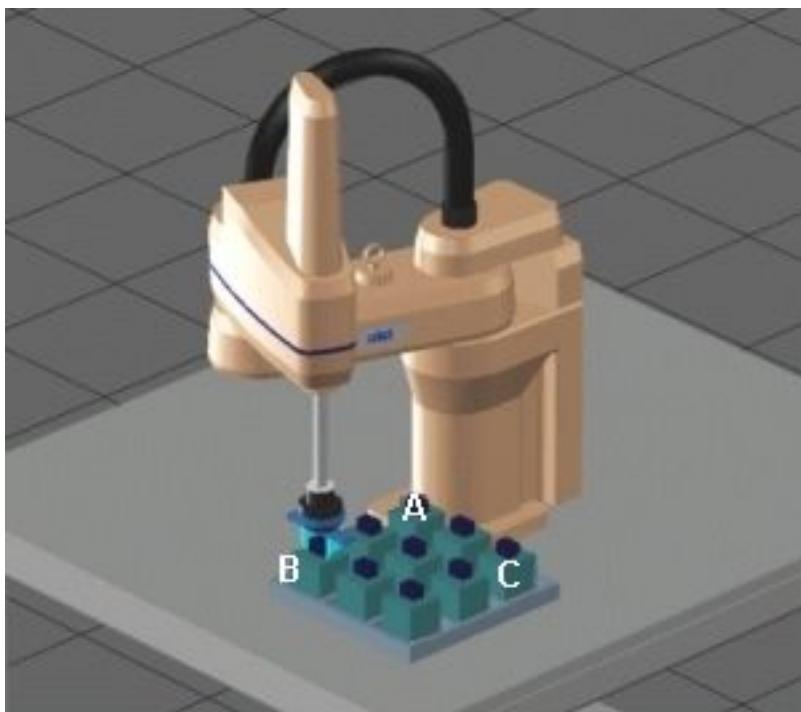
The direction of travel determines the upstream and downstream ends of the conveyor. In addition to the normal limits on the robot range of motion, a belt window must be created for a tracking belt to define the belt area to be accessed by the robot. The following items in the window limit belt access and tracking of parts along the conveyor.

Item	Description
Upstream limit	Specifies the point on a conveyor belt that must be passed before a robot will begin tracking a part.
Dynamic wait line	Specifies where the robot will wait if there are objects in the queue but not yet within the belt window.
Belt stop line	Stops the belt when an object reaches this point. This allows you to stop a belt before objects leave the belt window. (The belt must use an On/Off digital signal.)
Pickup limit	Specifies the point farthest downstream where the robot can begin accessing an object. It helps control belt-window violations by ensuring that the robot has sufficient time to track and access an object before it reaches the downstream limit.
Downstream limit	Specifies the farthest downstream point that the robot is allowed to reach. If the robot is tracking a part when it reaches this point, a belt-window violation will be issued.

## Defining a Pallet Layout

When defining a pallet layout, you are teaching points for the pallet, such as: the pallet origin (A), a point along the pallet X-axis (B), and a point along the pallet Y-axis (C). See the following figure for an example.

**NOTE:** The points labeled in the figures are only for example. You could define the pallet using any corner part as the origin, and using any row or column orientation. That is, the pallet rows do not need to be parallel to the robot World axes as shown in the example.



### *Defining a Pallet Layout*

For example, assuming a 40 mm part spacing, the pallet in the previous figure would be defined as follows:

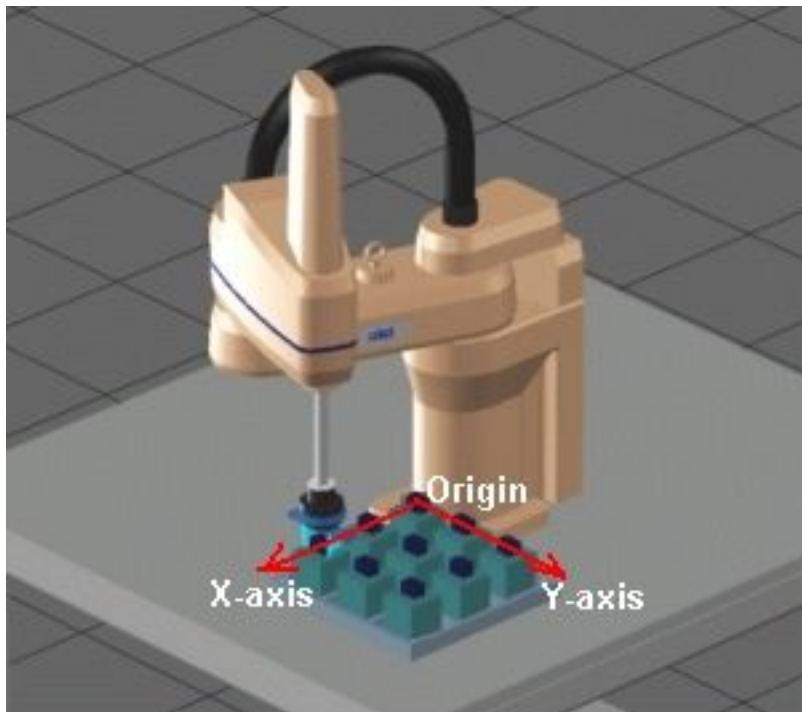
		Location Components					
		X	Y	Z	y	p	r
A	Pallet origin	220.0	220.0	54.0	0.0	180.0	0.0
B	Position along the X-axis	300.0	220.0	54.0	0.0	180.0	0.0
C	Position along the Y-axis	220.0	300.0	54.0	0.0	180.0	0.0

In addition to the above, there are other parameters to define, such as:

- the access order (XYZ, XZY, etc.)
- the number of parts on the X-axis
- the part spacing on the X-axis
- the number of parts on the Y-axis
- the part spacing on the Y-axis
- the number of parts on the Z-axis
- the part spacing on the Z-axis

**NOTE:** Make sure that you review the next section Why is Gripper Orientation Important? on page 70. It is necessary to understand this concept, so that your parts are picked/placed correctly

When teaching the pallet using the ACE software wizard, the system automatically computes the orientation and origin offset of the pallet. Then, the system has all of the information it needs to pick parts from positions in the pallet (or, to place parts into the pallet).



*Pallet Orientation*

### **Why is Gripper Orientation Important?**

When teaching locations, remember that the gripper orientation relative to the part is important. As you teach your pallet, you should check the gripper orientation to make sure you haven't changed it. This will ensure that the parts are picked and placed in the correct orientation.

## User Interface

The topics in this chapter describe the main components of the user interface.

Main Menu .....	72
Objects and Object Editors .....	75
Toolbar .....	157
Function Keys .....	180
Wizards .....	181

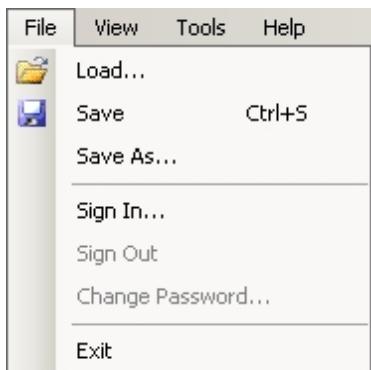
## Main Menu

The ACE main menu provides access to the following:

- a [File menu](#) for loading and saving the workspace, signing in and signing out (user access), and changing the password.
- a [View menu](#) for accessing items such as the Workspace Explorer, Task Status, and Controller Development Tools.
- a [Tools menu](#) for teaching Workspace Referencing, accessing Application Samples, and setting ACE options.
- a [Help menu](#) for viewing the About ACE information, the online help (this document) and related documents, the Event Log, and the Diagnostic Summary.

## File Menu

This section describes the items available on the File menu.



### Load...

Clears all objects from the workspace so that a new workspace can be created, and opens a dialog box allowing you to specify the workspace to load. For details, see [Workspace Loading and Unloading on page 52](#).

### Save

Saves the current workspace in the default file (the last file loaded or selected in the Save As dialog box). For details, see [Saving a Workspace on page 54](#).

### Save As...

Saves the current workspace in a file with a different name. For details, see [Saving a Workspace on page 54](#).

### Sign In...

Displays a dialog box that allows users to increase their access level by logging in (requires a user name and password). For details, see [User Manager Editor on page 125](#).

### Sign Out

(This item is enabled when a user has successfully signed-in.) Returns the access level to that specified by the User Manager when no user is logged in. For details, see [User Manager Editor on page 125](#).

### Change Password...

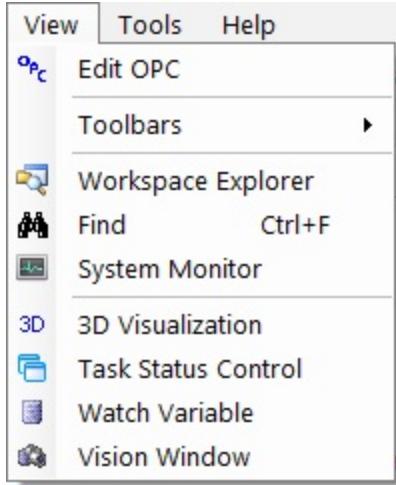
Displays a dialog box that allows users to change their password (access to this dialog box requires a successful login). For details, see [User Manager Editor on page 125](#).

### Exit

Closes ACE.

## View Menu

This section describes the items available on the View menu.



### Edit OPC

Opens the Edit OPC window, which allows you to enable OPC, communicate the values of specified V+ global and local variables, and monitor process status. For details, see OPC Data Access and Process Control on page 785.

### Toolbars

Opens the Toolbar menu, which is used to display (or hide) the ACE toolbars. For details, see Toolbar on page 157.

### Workspace Explorer

Opens the Workspace Explorer in the dockable portion of the window. For details, see Workspace Explorer on page 75.

### Find

Opens the Find dialog, which can be used to find (and replace) character strings in a text file or program within your workspace. For details, see Find Dialog on page 782.

### System Monitor

Opens the Robot Monitor tool, which is used to monitor specified parameters on the system. For details, see System Monitor on page 1117.

### 3D Visualization

Opens a windows that displays a 3D representation of the workcell. For details, see 3D Visualization on page 176.

### Task Status Control

Opens the Task Status Control in the dockable portion of the window. For details, see Task Status Control on page 747.

### Watch Variable

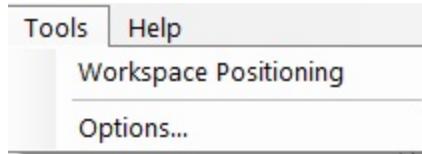
Opens the Watch Variable tool, which is used to monitor a list of user-selected V+ variables. For details, see Watch Variable Tool on page 794.

### Vision Window

Opens the Vision Window, which contains a tab displaying input from each virtual camera defined in the system. For details, see Vision on page 301.

## Tools Menu

This section describes the items available on the Tools menu.



### Workspace Positioning

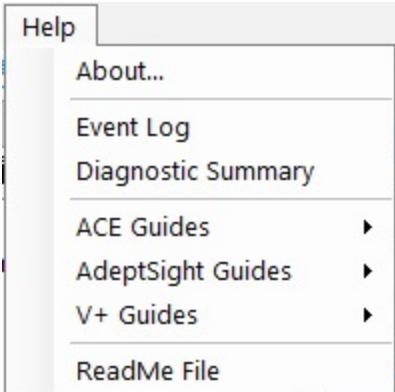
Opens a tool that is used to teach a reference point for the robot and conveyor belts used in a process-managed application, such as an ACE PackXpert packaging application. For details, see [Workspace Positioning on page 945](#).

### Options...

Displays a dialog box used for setting system options. For details, see [System Options on page 104](#).

## Help Menu

This section describes the items available on the Help menu.



### About...

Displays the ACE software version and information about the software components. For more details, see [About ACE on page 1106](#).

### Event Log

Displays the Event Log window, which contains event messages that have been logged since system startup. For more details, see [Event Log on page 1108](#).

### Diagnostic Summary

Displays the Diagnostic Summary window, which contains a list of information about the devices in the workspace. For more details, see [System Diagnostic Summary on page 1128](#).

### ACE Guides, AdeptSight Guides, V+ Guides

Displays this various guides for the product.

### ReadMe File

Displays the ACE ReadMe file, which contains the system requirements, installation instructions, distribution contents, usage considerations, and a version change history.

## Objects and Object Editors

As described previously, the ACE workspace uses objects to represent items that, when put together, make up a functioning workcell. The workspace is composed of a hierarchy of objects, both physical (like robots and end-effectors) and non-physical (like variable values).

Some objects are created automatically when you connect to a controller (for example, the controller, robot, and robot grippers). Other objects can be added to provide increased functionality to the workcell. For more details on adding objects to the ACE workspace, see the topic [Objects in the Workspace](#)

After objects have been added to the ACE workspace, the object editors are used to edit parameters for the objects. For more details on the object editors, see the topic [Object Editors](#).

**NOTE:** Most objects and object editors require an access level of Technician or higher. For details, see Appendix 2: User Access Item List on page 1168. For details on configuring user access levels, see User Manager Editor on page 125.

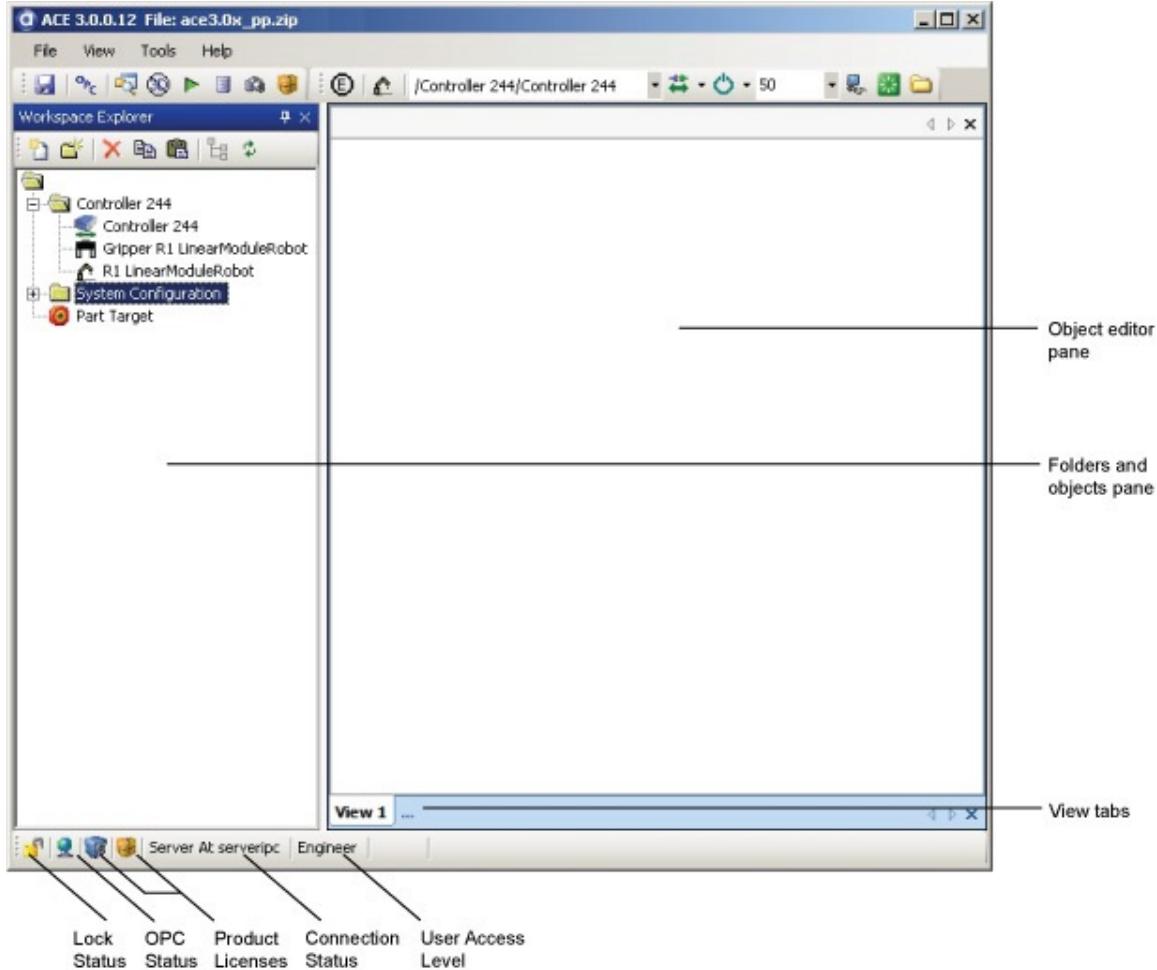
## Workspace Explorer

The Workspace Explorer is the main work area for the ACE software. You can use it to add and delete objects, access an object editor, access configuration utilities, and access program editors.

The Workspace Explorer displays the workspace objects in a tree structure on the left side of the display, and opened editors for those objects on the other side of the display (see the following figure).

## Workspace Explorer

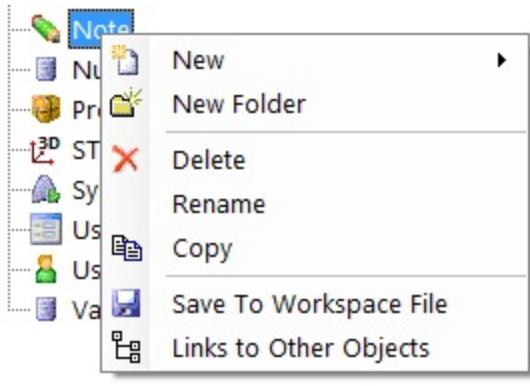
---



Workspace Explorer Window

### Shortcut Menus

Right-clicking on an object in the tree structure of the Workspace Explorer causes a shortcut menu to appear, as shown in the following figure.



#### *Example Object Shortcut Menus*

The shortcut menu items will vary, depending on the object that is selected. The following is a description of the shortcut menu items:

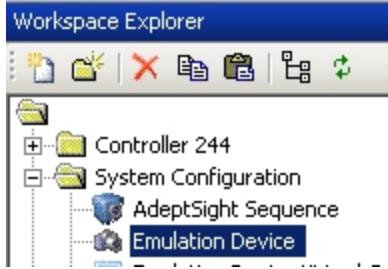
Item	Description
New...	Adds new objects.
New Folder	Creates a new folder under the current folder.
Delete	Deletes the selected object or folder.
Rename	Renames the selected object or folder.
Copy	Copies the selected object or folder to the clipboard as XML data.
Paste	Pastes an object or folder from the clipboard. A number appended to the end of the name for the pasted object.
Save To Workspace File	Saves the selected folder or object to a file. You can save the file in ZIP or AWP format.
Links to Other Objects	Shows all references to the specified object or folder in the workspace.

#### **Folder Pane Toolbar Icons**

The folder pane of the Workspace Explorer contains a set of toolbar icons, which are used for managing the objects and folders.

## Workspace Explorer

---



*Folder Pane Toolbar*

**NOTE:** In addition to the folder pane toolbar, the Workspace Explorer also contains a main toolbar. For details, see [Toolbar](#) on page 157.

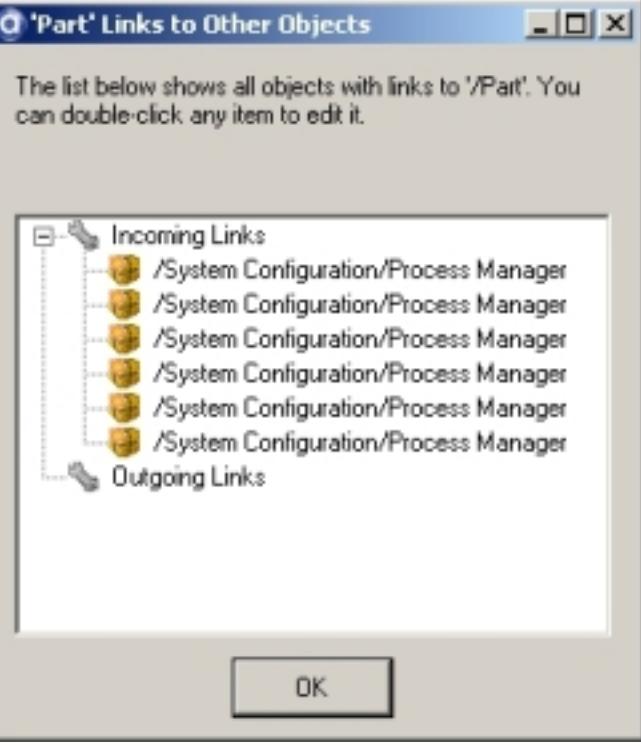
The following is a description of the toolbar items.

Icon	Description
	Opens the Create a New Object dialog, which is used to add a new object to the workspace. For more details on adding new objects to the workspace, see <a href="#">Objects in the Workspace</a> on page 85.

## Workspace Explorer

---

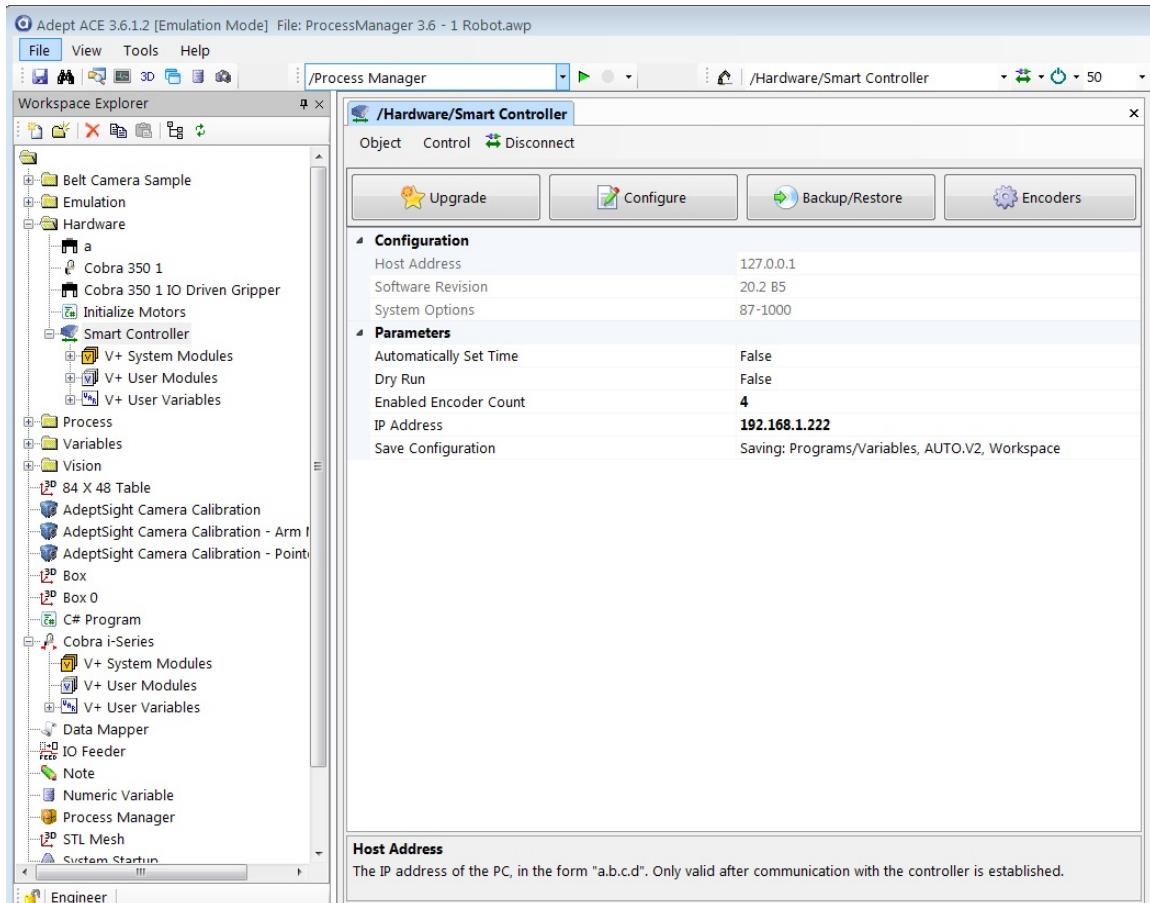
Icon	Description
	Adds a new folder to the workspace.
	Deletes the selected item.
	Copies the selected item.
	Pastes the previously-copied item. Greyed out when not available for the selected item.

Icon	Description
	<p>Opens the Links to Other Objects window, which displays the Incoming and Outgoing linked objects for the selected object. You can double-click any link to open the corresponding editor for the linked object. For more details on editing objects, see Objects and Object Editors on page 92. Greyed-out when not available for the selected object.</p> 
	<p>Refreshes (updates) the list of items in the folder pane of the Workspace Explorer. Only available for SmartControllers and their V+ Modules.</p>

### Object Editors

When an object in the tree structure of the Workspace Explorer is double-clicked, its editor opens. For example, in the following figure, if you double-click the Adept Controller object, the Workspace Explorer displays the Controller editor.

## Workspace Explorer



### Workspace Explorer with Controller Editor

Most object editors include one or more of the following items:

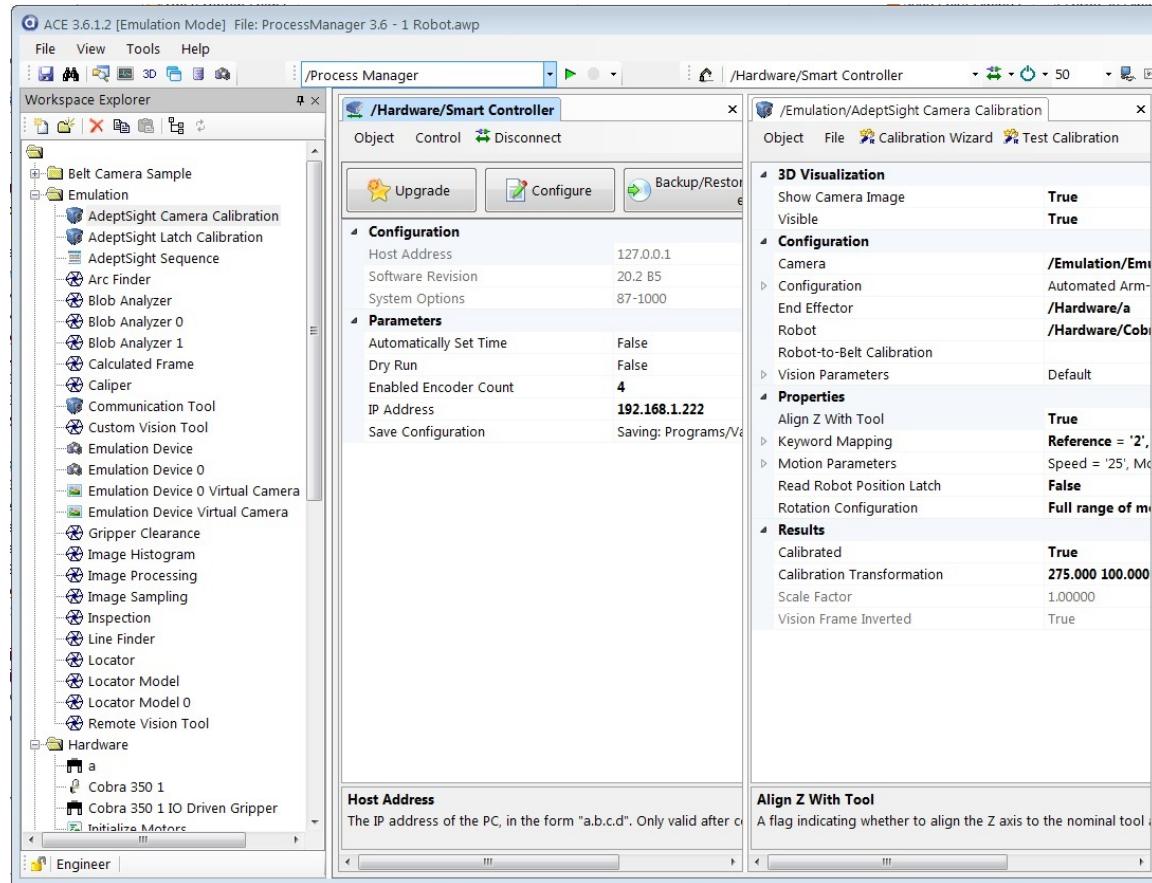
Item	Description
Editor name	Displays the name and path of the current editor
Menu	Menu items vary, depending on the particular editor that is opened
Editor parameters	Used to select and enter values for each item. For most editors, items are grouped; a group can be expanded or collapsed using the "+" or "-" symbol next to the group heading.
Online help	Provides help for the selected editor item.

For more details on the Workspace Explorer objects and object editors, see the section [Objects and Object Editors](#).

## Workspace Explorer

### Dockable Editor Windows

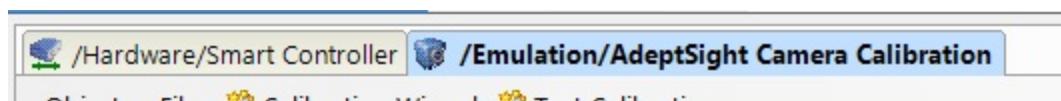
The ACE workspace is designed so you can position (dock) multiple editor windows within the same workspace, as shown in the following figure. This feature allows you to customize the editing environment for your application.



### Multiple Windows

To dock multiple editor windows in the workspace:

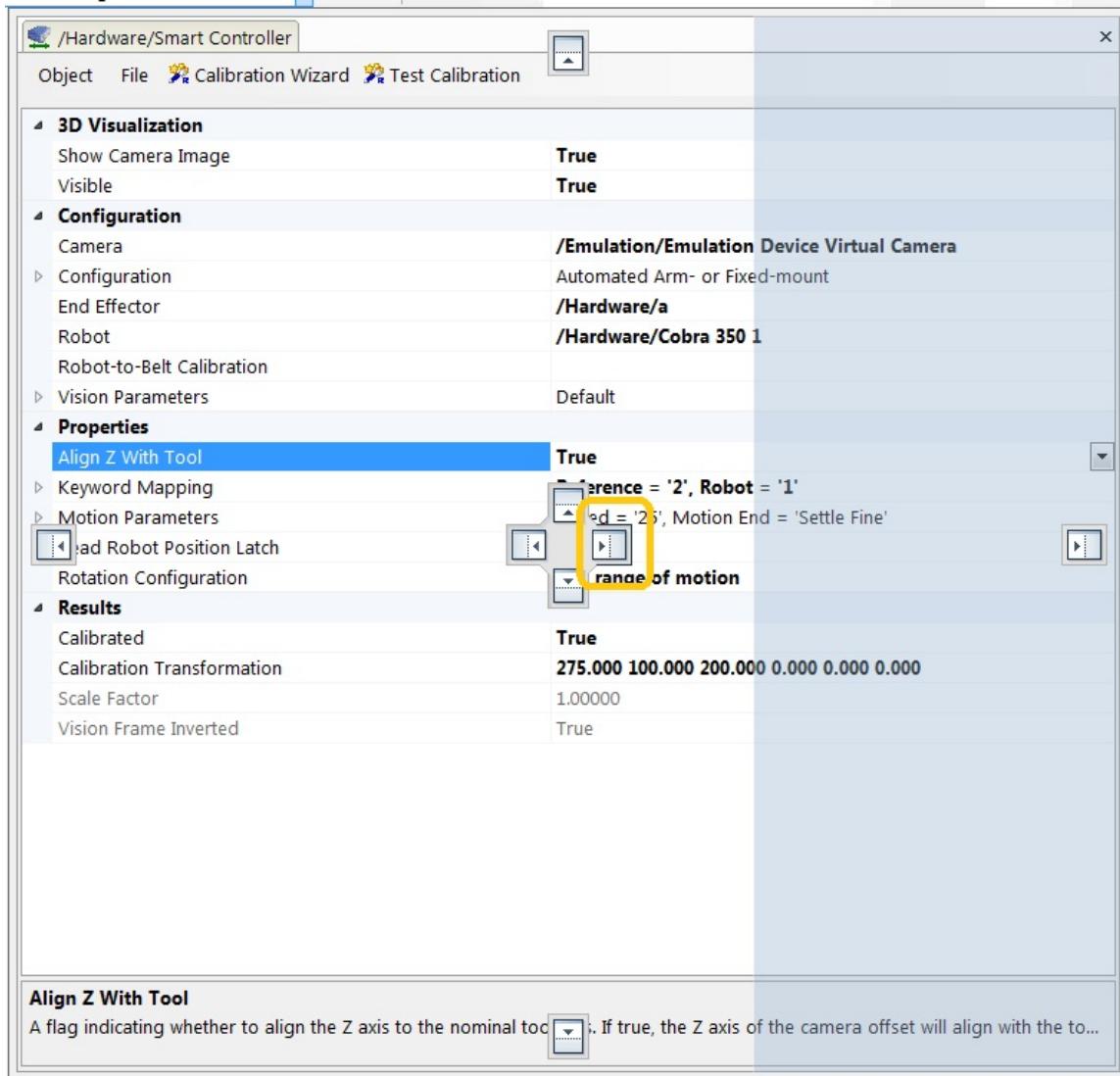
1. Open the object editors you wish to use. The object editors are shown in a "stacked" configuration; the tabs at the top of the window allow you to select the editor you wish to view.



*Editor Window Tabs*

2. Click and hold the tab at the top of the editor window to select it.

3. Drag the tab into the workspace. Window position locators are displayed that you can use to position the selected editor within the workspace window.



#### Window Position Locators

4. Drop the tab onto the desired position locator and release. The editor window is repositioned in the workspace window.
5. Repeat the steps above until all editor windows are positioned as desired.

### **Editor Window Management**

When working with multiple editor windows, you can right click on any object editor tab to display a menu containing window-management functions. The menu is shown below along with a description of each function.

Close	Ctrl+Shift+C
Close All	Ctrl+Shift+A
Close All But This	Ctrl+Shift+B
Prominent	Ctrl+Shift+T
Rebalance	Ctrl+Shift+R
New Horizontal Tab Group	Ctrl+Shift+H
New Vertical Tab Group	Ctrl+Shift+V
Move to Next Tab Group	Ctrl+Shift+N
Move to Previous Tab Group	Ctrl+Shift+P
Move To View	
Move to New View	

#### **Close**

Closes the editor.

#### **Close All**

Closes all object editors that are currently open.

#### **Close All But This**

Closes all object editors except for the one currently selected.

#### **Prominent**

Resizes the selected object editor to fill the current window.

#### **Rebalance**

Resizes the object editors within a tab group, so that all the editors are the same size; resizes the tab groups within the Workspace Explorer, so that all tab groups are the same size.

#### **New Horizontal Tab Group**

Splits the current tab group horizontally, and moves the selected tab to the new tab group.

#### **New Vertical Tab Group**

Splits the current tab group vertically, and moves the selected tab to the new tab group.

#### **Move to Next Tab Group**

When New Horizontal or New Vertical Tab Group is used and a tab is selected, you can use this menu item to move the selected tab into the next tab group (down for horizontal groups; right for vertical groups).

#### **Move to Previous Tab Group**

When New Horizontal or New Vertical Tab Group is used and a tab is selected, you can use this menu item to move the selected tab into the previous tab group (up for horizontal groups; left for vertical groups)

#### **Move to View**

When multiple View tabs are available, you can use this menu item to move the selected tab into a different View tab. For more details on using

View tabs, see [Workspace Explorer](#) on page 75.

### Move to New View

Creates a new view tab and then moves the selected object editor to that tab. For more details on using View tabs, see [Workspace Explorer](#) on page 75.

## Objects in the Workspace

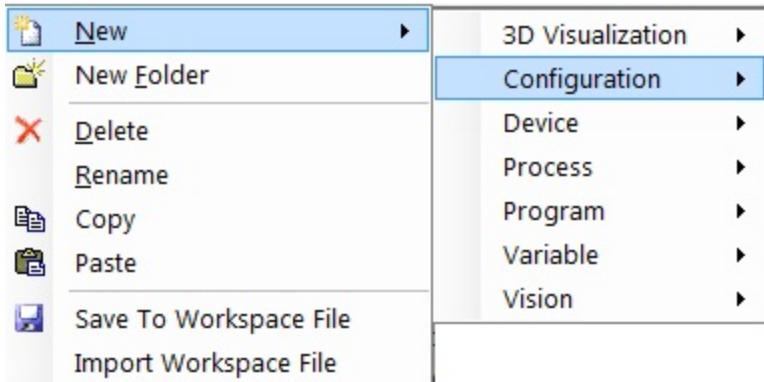
When you connect to a controller, objects representing the controller, the robots, and grippers attached to the robots are created in the workspace and shown in the [Workspace Explorer](#). These object provide access to core configuration capabilities for the controller and the robot. Optionally, you can add additional objects to the [Workspace Explorer](#), if you want to access more functionality or script the system from the PC.

### **Adding Objects to the Workspace**

To add new objects to the workspace:

### **Using the Right-Click Method**

1. Right-click in the folder area of the workspace and select **New** from the menu. A sub-menu is displayed that shows the object categories.



*Object Categories Menu*

2. Hover over a category to display a menu of items for that category.
3. Click on the desired item to add it to the workspace.

**NOTE:** When adding objects, it is a good idea to place your objects into folders . This will help you stay organized when you have multiple robots and many objects in the same workspace.

## Using the Create a New Object Dialog

1. On the Workspace Explorer folder pane toolbar, click the new (  ) icon. The Create a New Object dialog opens.
2. Use the scroll bar to scan through the list of objects until you find the one you want to create.
3. Select the object. The default object name is shown in the Name field.
4. Optionally, edit the name shown in the Name field.
5. Click **OK** to add the new object to the workspace.

### **Renaming Workspace Objects**

When you add a new object to the workspace, the ACE software assigns a default name to the new object. In cases where two similar objects are in the same folder (for example, two robots) the ACE software automatically assigns a different default name to each object (for example, "Robot" and "Robot 0"). However, if you have two robots in different folders (one robot in each folder), and have used the default name, you will have an object with the same name in each folder. In this case, it is recommended that you rename one of the objects so that each object has a unique name.

**NOTE:** It is a good programming practice to assign unique names to your objects, even if they are in different folders.

To rename an object:

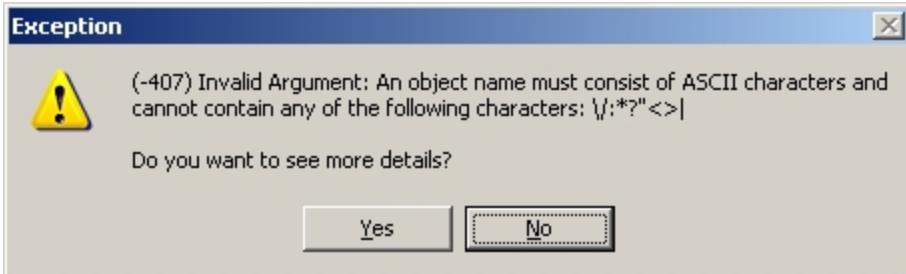
1. In the Workspace Explorer folder view, right-click on the object and select Rename from the menu. Or, as a shortcut, you can select the object and press F2. The object name is highlighted and a cursor is displayed.



### *Rename an Object*

2. Type the new name for the object.

**NOTE:** Only ASCII characters are permitted. Additionally, the following ASCII characters cannot be used:  
\\ : \* ? < > |  
If an illegal character is used, the following message is displayed:



*Illegal Character Message*

3. Press the Enter key or click in an empty area of the Workspace Explorer folder view. The new name is assigned to the object.

**Deleting Workspace Objects**



**CAUTION:** There is no "undo" for this operation. Therefore, once an object is deleted from the workspace, it cannot be automatically restored.

Workspace object can be deleted (removed) from the workspace. When an object is deleted, any references to that object are broken.

When you delete an object from the workspace, the ACE software automatically checks to see if the object being deleted is referenced by any other workspace objects. If so, you will see a warning message similar to the example in the following figure.

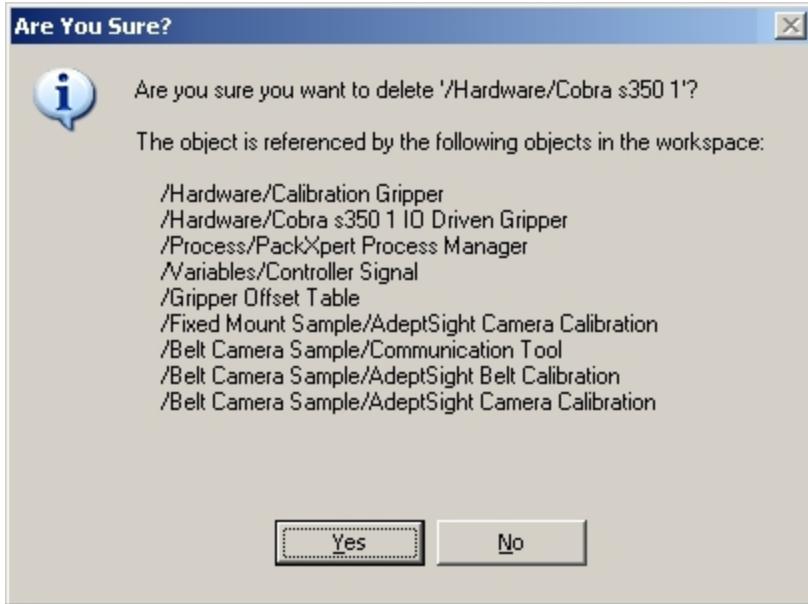
To delete an object:

1. In the Workspace Explorer folder view, right-click on the object and select Delete from the menu. Or, click the object to select it, and then press the Delete key.

If the object is not referenced by another workspace object, the following message is displayed:



If the object is referenced by one or more workspace objects, the following message is displayed:

*Delete Object Confirmation Messages*

- Click **Yes** to delete the object. The object is deleted (removed) from the workspace.

***Understanding Workspace Objects***

The table below describes the objects available in the software. Your system may show fewer objects depending on the options installed in your system.

**NOTE:** All workspace objects and many functions are represented with icons. Variations of the icons are used to indicate, for example, if the object or function is available or not, or to indicate when there is a problem. For details, see Icons in the Workspace on page 46.

Object	Description
Box	Creates an Box object, which represents a box shape (typically used in 3D visualization). For details, see 3D Visualization on page 176.
Cylinder	Creates an Cylinder object, which represents a cylinder shape (typically used in 3D visualization). For details, see 3D Visualization on page 176.
CAD Data	Imports CAD data file into the 3D visualization. For details, see CAD Data Object on page 152.

## Configuration

Object	Description
Data Mapper	Creates a Data Map object in the workspace. This can be used to map information between objects in the workspace.
Note	Creates a Note object (similar to Windows Notepad), which is used to type notes about the application. The Note objects are saved as part of the ACE workspace. For details, see Note Editor on page 113.
Recipe Manager	Creates a Recipe Manager object, which is used to create and manage production recipes. For details, see Recipe Manager on page 983.
System Startup	Creates a System Startup object, which is used to specify various ACE software start-up options. Only one System Startup object may exist in the workspace. For details, see System Startup Editor on page 119.
User Manager	Creates a User Manager object, which is used to configure users and access levels. Only high-level access users may edit user information. Only one user manager may exist in the workspace. For details, see User Manager Editor on page 125.

## Device

Object	Description
Cobra i-Series	Creates a Cobra i-Series robot object, which represents a robot. For more details on the Cobra i-Series, see Cobra i600/i800 Robot Configuration on page 265.
Controller Robot	Creates a Robot object, which represents a robot. These will typically not need to be created, since they are automatically added to the workspace when you connect to a controller.
<b>Feeders</b>	
AnyFeeder	Creates an Adept AnyFeeder object, which represents that feeder. The editor contains parameters and controls for setup and test of the feeder. For details, see Adept AnyFeeder Object on page 138.
IO Feeder	Creates an IO Feeder object, which represents that feeder. The editor contains parameters and controls for setup and test of the feeder. For details, see IO Feeder Object on page 148.
SmartController	Creates a SmartController object, which represents an Adept Smart Controller. These will typically not need to be created, since they are automatically added to the workspace when you connect to a controller. For details, see Controller

Object	Description
	Installation on page 190.
Tool > IO Driven Gripper	Creates an IO Driven Gripper object, which represents a gripper that uses specified inputs and outputs for activating the gripper. For details, see Gripper Editor on page 108.

## Process

Object	Description
Belt	Creates a Belt object, which represents a conveyor. For details, see <a href="#">Belt on page 809</a> .
Custom Allocation Script	Creates a Custom Allocation Script object, which contains a simple C# program editor. It is used to create and edit custom part-allocation programs for use with the Process Manager. For details, see <a href="#">Custom Allocation Script on page 825</a> .
Part	Creates a Part object, which represents a part that can be picked or placed by the robot. The Part object defines an object that is input for processing. For details, see <a href="#">Part on page 829</a> .
Part Buffer	Creates a Part Buffer object, which represents an “overflow” buffer where parts can be temporarily stored when an output conveyor belt (or feeder) is unavailable to accept the parts. For details, see <a href="#">Part Buffer on page 836</a> .
Part Target	Creates a Part Target object, which represents a placement destination for a part. For details, see <a href="#">Part Target on page 841</a> .
Process Manager	Creates a Process Manager object, which is used to manage the elements of a ACE PackXpert packaging application. For details, see <a href="#">Process Manager on page 843</a> .
Process Pallet	Creates a Pallet object, which represents a rectangular or radial pallet. For details, see <a href="#">Process Pallet on page 927</a> .
Vision Refinement Station	Creates a Vision Refinement Station object, which represents a location with an upward-looking camera that is used to improve the part-to-gripper orientation for more accurate placement of the part. For details, see <a href="#">Vision Refinement Station on page 941</a> .

## Program

Object	Description
C# Program	Creates a C# Program object, which provides a basic C# program editor that can be used to create and edit programs. For details, see C# Language Programming on page 750.
User Interface Form	Creates a User Interface Form object, which is used to create a custom user interface (UI) for your application. For details, see User Interface Designer on page 1000.

## Variable

Object	Description
Belt Calibration	Creates an Belt Calibration object, which uses an interview wizard to guide you through the robot-to-belt calibration process. It provides the belt-to-robot offset and belt-control information for conveyor-tracking applications. For more details, see the topic Belt Calibration in the <i>ACE Sight User's Guide</i> .
Numeric Variable	Creates a Numeric Variable object, which represents static or controller numeric values. For details, see Variable Editor on page 131.
String Variable	Creates a String Variable object, which represents static or controller string values. For details, see Variable Editor on page 131.

## Vision

Object	Description
ACE Sight	
ACE Sight Camera Calibration	<p>Creates an ACE Sight Camera Calibration object, which uses an interview wizard to calculate the offset of a camera target to the base of the robot. The wizard uses an interview process to collect the application details and guide you through the calibration procedure. For more details, see the <i>ACE Sight User's Guide</i>.</p> <p><b>NOTE:</b> If a conveyor belt is used to feed or remove parts in a vision-guided application, the ACE Sight Belt Calibration must be performed <i>before</i> the ACE Sight Camera Calibration.</p>

Object	Description
ACE Sight Sequence	Creates an ACE Sight sequence object, which shows the list of vision tools that will be executed as part of the vision sequence, and the Index associated with each one. The Index is the execution order of the vision sequence. For more details, see the <i>ACE Sight User's Guide</i> .
Communication Tool	Creates an ACE Sight Communication Tool object, which is a tool for vision-guided conveyor-tracking applications. The purpose of the tool is to provide instructions to the controller for the handling of objects that must be picked or manipulated by a robot. For more details, see the <i>ACE Sight User's Guide</i> .
Gripper Offset Table	Creates a Gripper Offset Table object, which is used to create and manage a table of gripper offsets. For more details, see the <i>ACE Sight User's Guide</i> .
Overlap Tool	Creates an ACE Sight Overlap Tool object, which is a tool for vision-guided conveyor-tracking applications. The purpose of the Overlap Tool is to make sure that parts moving on the belt are recognized only once. For more details, see the <i>ACE Sight User's Guide</i> .
Device ▾	
Basler Pylon Device	Creates a Basler Pylon Device camera object and opens the object editor, which is used to specify the Basler (1394) camera. For more details, see the <i>ACE Sight User's Guide</i> .
Emulation Device	Creates an Emulation Device object and opens the object editor, which is used to load image files into the camera emulation device. For more details, see the <i>ACE Sight User's Guide</i> .
Virtual Camera	Creates a Virtual Camera object and opens the object editor, which is used to specify the acquisition device (Basler camera or Emulation Device) and a calibration method. When the calibration method is specified, a dialog opens, which is used to perform the device calibration. For more details, see the <i>ACE Sight User's Guide</i> .
Tool ▾	
(Various vision tool objects)	Creates a vision tool object corresponding to the selected Vision Tool and opens the editor, which is used to define and configure the items in the vision tool. For more details, see the <i>ACE Sight User's Guide</i> .

## Objects and Object Editors

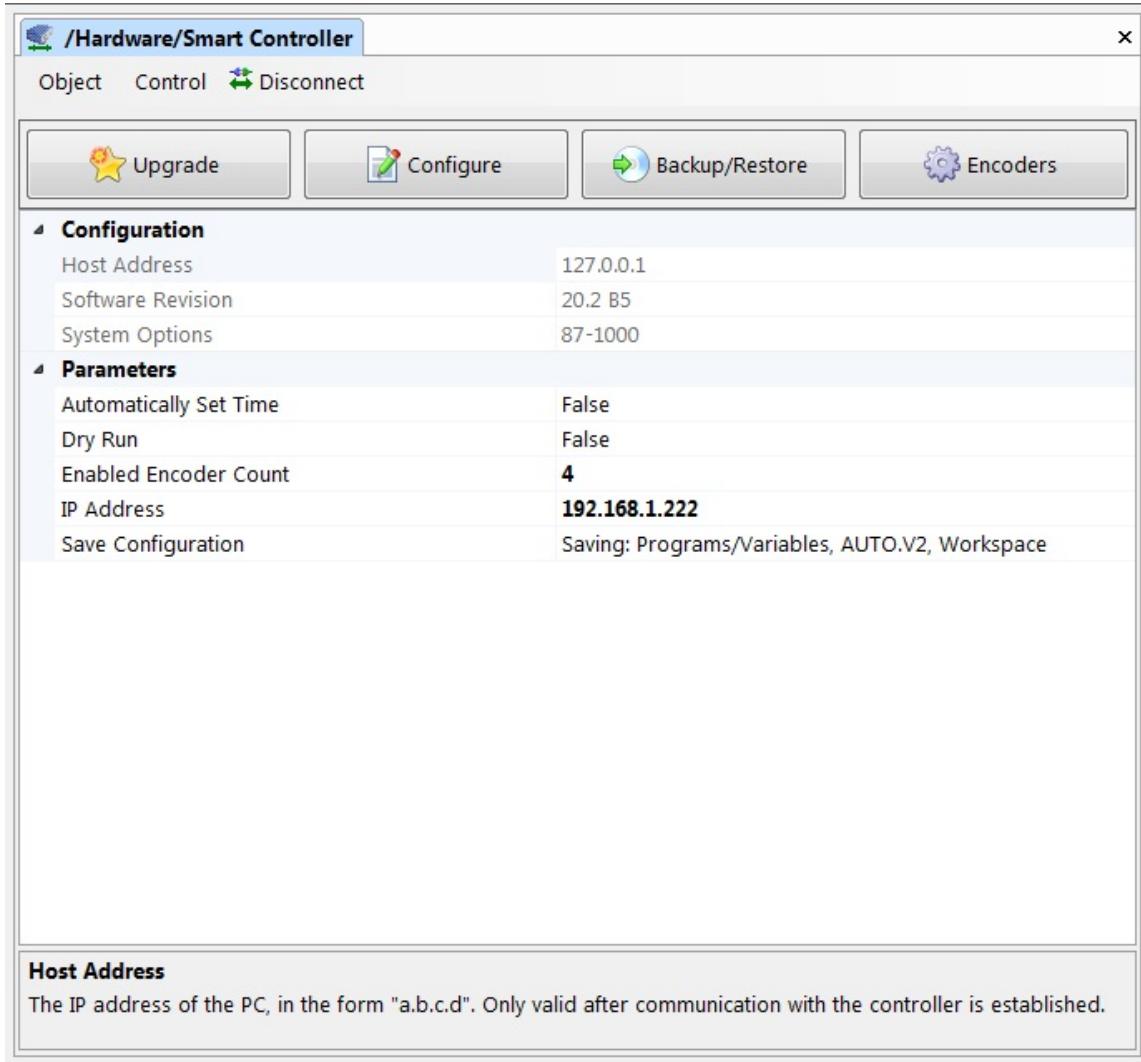
After objects have been added to the Workspace Explorer, the ACE object editors are used to edit the parameters for the objects. When a new object is added to the Workspace Explorer, the ACE software automatically opens the corresponding object editor, which is used to configure the new object. You can also access the editor by double-clicking an object in the Workspace Explorer.

Each of the object editors has a similar "look and feel", which consists of:

- a menu
- quick-access buttons (available on some editors, see the NOTE below)
- editor parameters
- online help

**NOTE:** Several editors, such as the Controller Editor, include an additional area containing quick-access buttons, which provide access to various additional functions. For more details, see the topic [Controller Editor](#).

These elements can be seen in the following figure, which shows an example of the Controller Editor.



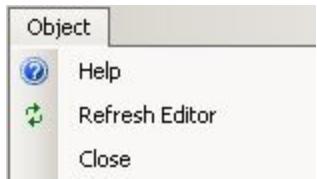
### *Controller Editor*

Most editors can be resized, as needed, by dragging the edges of the editor. If you change the size of an editor so that some fields/controls are hidden, scroll bars will be added so you can get to the hidden information.

### ***Understanding Editor Menus***

As shown in the figure above, each editor has a menu. The menus and their contents will vary, depending on which editor you're using. At a minimum, the editor will have an Object menu with the following selections:

## Object Menu (Sample)



### Help

Displays the online help for the editor.

### Refresh Editor

Refreshes the contents of the editor window.

### Close

Closes the editor.

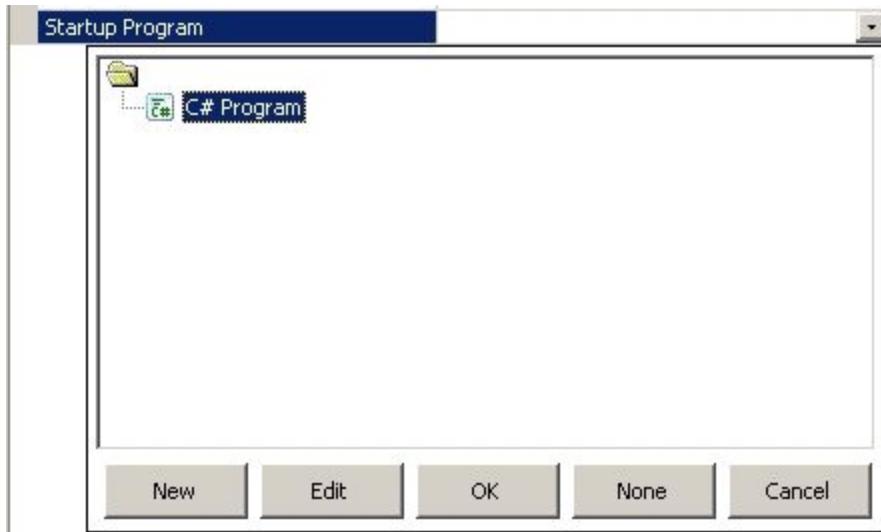
## Understanding Editor Parameters

As shown in the Controller Editor figure above, each editor has a parameters section with two columns: the left-hand column contains a parameter name, the right-hand column contains the current setting for that parameter. The parameters can be modified as follows:

Field Type	To use this field...
Pick list	Click the down arrow ( ) icon to display the list and then click the desired value. Note that for certain parameters, a resource dialog box is displayed, which is used to create the list of items. See Resource Dialog (below) for more details.
Direct entry / browse	Type the value directly into the field, or click the Browse ( ) icon to browse for the requested path / filename.
Direct entry	Type the value directly into the field. (For example, see the IP Address parameter in the figure above.)
Information only	These parameters are read-only; you can view the value for the parameter, but the value cannot be edited. (For example, the Software Revision parameter.)

## Resource Dialogs

For some parameters, when the down arrow ( ) is clicked, a resource dialog box displays, as shown in the following figure.



*Resource Dialog*

The resource dialog box displays a list of object types (such as program objects) that correspond to the parameter being edited. The resource dialog box buttons are described in the following table.

Button Name	Function
New	Used to create a new resource object. A "Create a New Object" dialog box displays, which allows you to create the type of object required for the parameter being edited.
Edit	Displays the object editor for the selected object.
OK	Used to save the object selection. Select the desired object and then click <b>OK</b> to save the selection.
None	Removes a previously-made selection. (The value field for the parameter will be blank.)
Cancel	Closes the dialog box without saving any changes.

#### ***Understanding Online Help for Parameters***

Online help is automatically displayed for the selected editor parameter. The online help gives a brief description of the function for the selected parameter. You can "hide" the online help area (this is handy if you're working with a small display) by doing the following:

To hide the online help bar:

1. Right click in the parameter area of the editor. A menu opens.



2. De-select (uncheck) "Description". The online help bar is hidden.

To show the online help bar:

1. Right click in the parameter area of the editor. A menu opens.



2. Select (check) "Description". The online help bar is displayed.

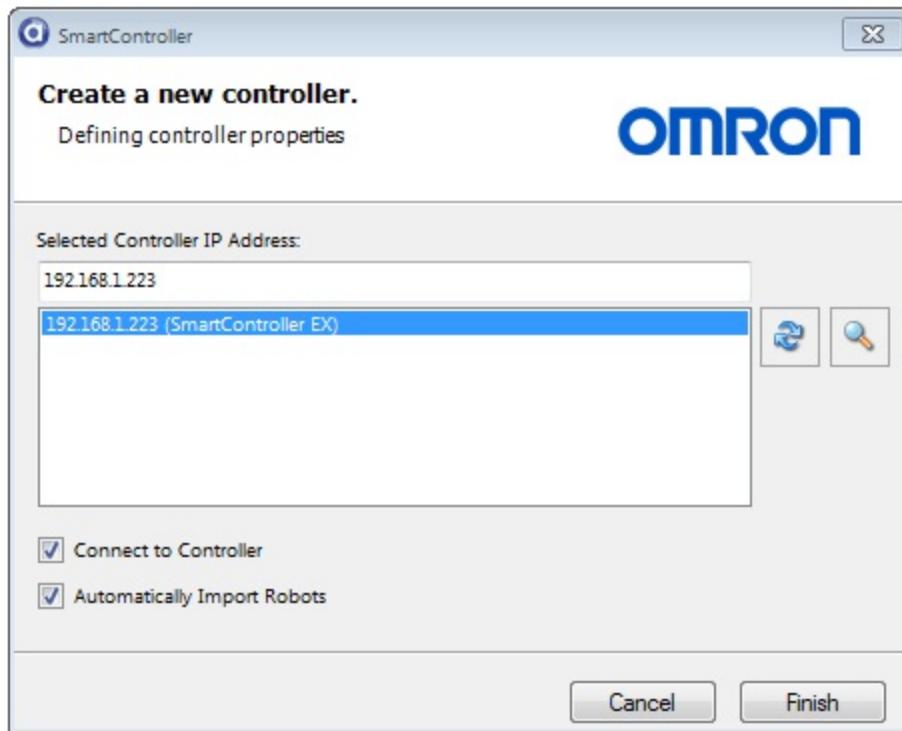
## Controller Editor

**NOTE:** When emulation mode is enabled, this item has reduced features.

The Controller Editor provides tools for installing and configuring your V+ system, controller, FireWire nodes, and robots.

To add an additional controller to the workspace, right-click in the folder area of the Workspace Explorer and select **New > Device > SmartController**.

A dialog box opens that helps you create a new SmartController object. The dialog box options allow you to connect to the controller and import any robots that are connected to that controller.

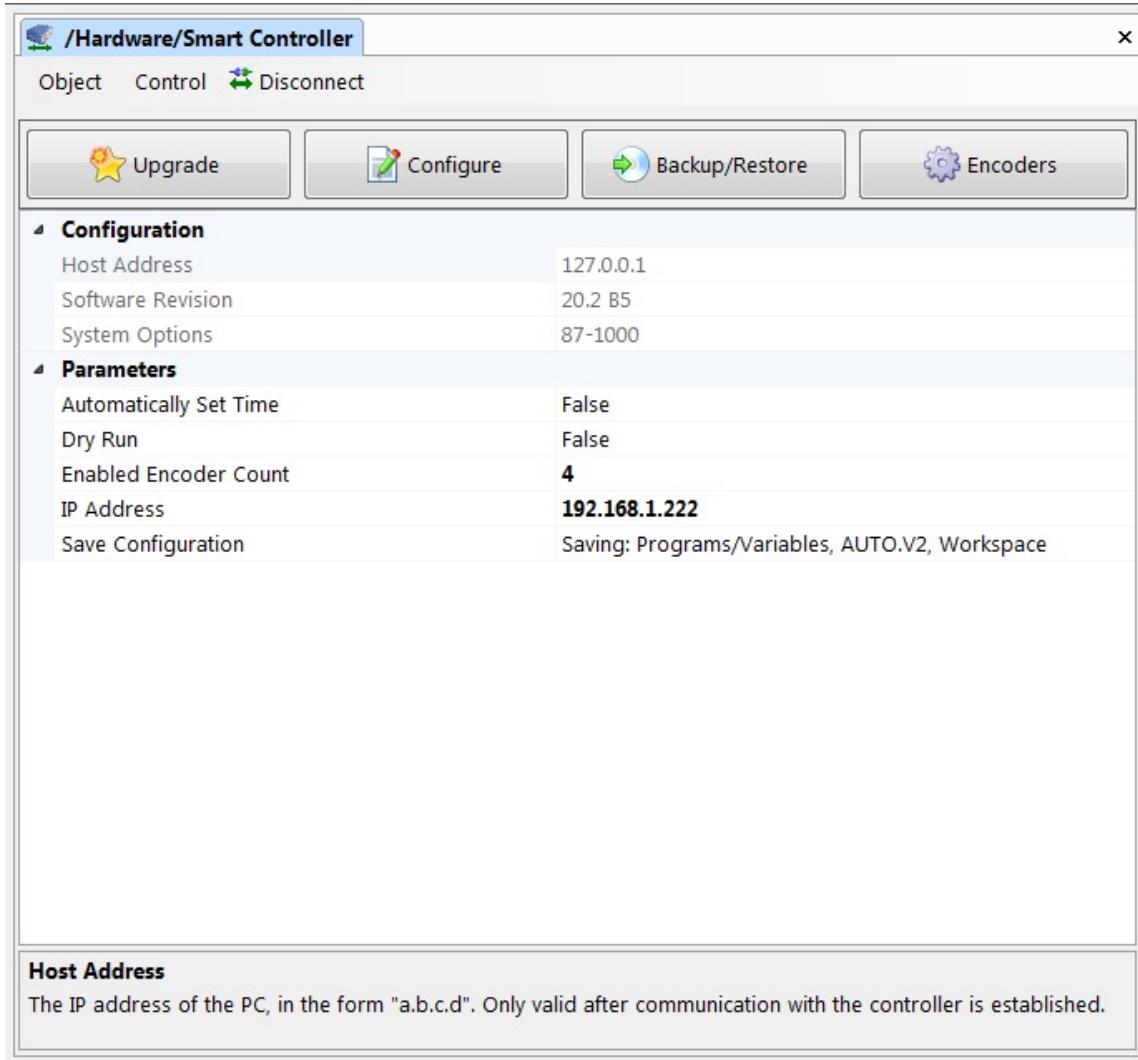


*Create a New Controller Dialog Box*

To open the Controller Editor, double-click the IP address of the controller object in the folder area of the Workspace Explorer.

## Controller Editor

---

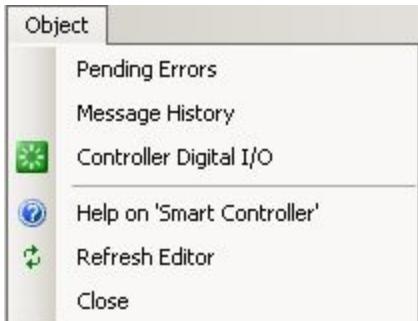


*Controller Editor*

### **Menu Items**

This section describes the selections available from the Controller Editor menu.

## Object Menu



### Pending Errors

Displays any errors that have occurred since the last ENABLE POWER command. For details, see the topic Pending Errors Log on page 1112.

### Message History

Displays the Message History window, which is used to view the system message log.

### Controller Digital I/O

Displays the controller digital I/O window. For details, see the topic Digital I/O Window on page 166.

### Help

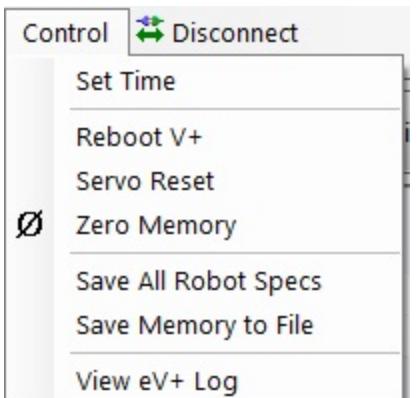
Displays the online help for the Controller Editor.

### Refresh Editor

Refreshes the contents of the Controller Editor window.

### Close

Closes the Controller Editor.



### Set Time

Displays a prompt to verify that you want to set the controller time to match the time detected on the connected PC. Click **OK** to proceed; click **Cancel** to quit.

### Reboot V+

Displays a prompt to verify that you want to save the system specifications to the boot disk before initiating a reboot of the controller. Click **Yes** to save the specifications and proceed; click **No** to proceed without saving the specifications; click **Cancel** to quit.

### Servo Reset

Resets the servos. A "Working...." message and progress bar is displayed while the command is processed.

### Save All Robot Specs

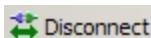
Saves all robot and motor specifications to the V+ boot disk. A "Working...." message and progress bar is displayed while the command is processed.

### Save Memory to File

Saves all V+ files, program files, and/or variable files to a specified location on the PC.

### View eV+ Log

Views the eV+ event log. This is only displayed if the controller



is a SmartController EX.

#### Connect (Disconnect)

Connects (initiates communication) to the controller specified in the IP Address field of the Configuration grid editor.

After the connection is established, the button name and function changes to Disconnect, which disconnects (stops communication) to the specified controller.

### Quick-Access Buttons

The upper area of the Controller Editor contains quick-access buttons, which are grouped by function. These buttons provide a convenient way to access the configuration tools (V+, controller, FireWire, and robot utilities). The function of each button is described below:

#### Upgrade

This button provides access to the following upgrade options:

- Upgrade V+
- Firewire Firmware
- Controller FPGA

##### Upgrade V+

Displays the V+ Upgrade dialog box. For details, see the topic V+ Upgrade on page 228.

##### FireWire Firmware

Displays the FireWire Node Firmware Update dialog box. For details, see the topic FireWire Node Firmware Upgrade on page 200.

#### Configure

This button provides access to the following configuration options:

- Configure Controller
- Configure Licenses
- Configure Robots
- Configure Encoder Latches
- Configure FireWire Nodes
- DeviceNet Scan

##### Configure Controller

Displays the V+ System Configuration editor, which provides editing access to: NVRAM settings, the V+ Configuration file. For details, see the topic V+ System Configuration Editor on page 221.

**NOTE:** When emulation mode is enabled, the emulated controller is not restarted after making configuration changes, it is simply reset.

##### Configure Licenses

Displays the Controller License Configuration manager, which is used to install and configure the V+ licenses for the controller. For details, see the topic Licenses on page 230.

##### Configure Robots

Displays the Robot Installation Wizard, which is used to install and configure robots in the system. For details, see the topic Robot Installation on page 211.

**NOTE:** This feature operates differently in emulation mode. For details on emulation mode, see Emulation Mode on page 1089.

#### **Configure Encoder Latches**

Displays the Configure Encoder Latches tool, which is used to configure the belt encoder latch signals in the system. For details, see the topic Encoder Latches on page 198.

#### **Configure FireWire Nodes**

Displays the FireWire Nodes configuration dialog box. For details, see the topic Robot and Motor FireWire Configuration on page 205.

**NOTE:** When emulation mode is enabled, this item is not available. For more details on emulation mode, see Emulation Mode on page 1089.

#### **DeviceNet**

Displays the DeviceNet Scan dialog box. For details, see the topic DeviceNet Scan on page 193.

**NOTE:** When emulation mode is enabled, this item is not available. For more details on emulation mode, see Emulation Mode on page 1089.

## **Backup/Restore**

This button provides access to the following backup and restore options:

- Backup V+**
- Restore V+**
- Backup or Restore Controller Files**

#### **Backup V+**

Displays the Backup V+ to PC dialog box. For details, see the topic V+ Backup on page 214.

#### **Restore V+**

Displays the Restore V+ from PC dialog box. For details, see the topic V+ Restore on page 215.

#### **Backup or Restore Controller Files**

Allows for backing up and restoring files from the entire controller.

## **Encoders**

This button provides access to the Encoder diagnostics dialog. For details, see External Encoder Diagnostics on page 1115.

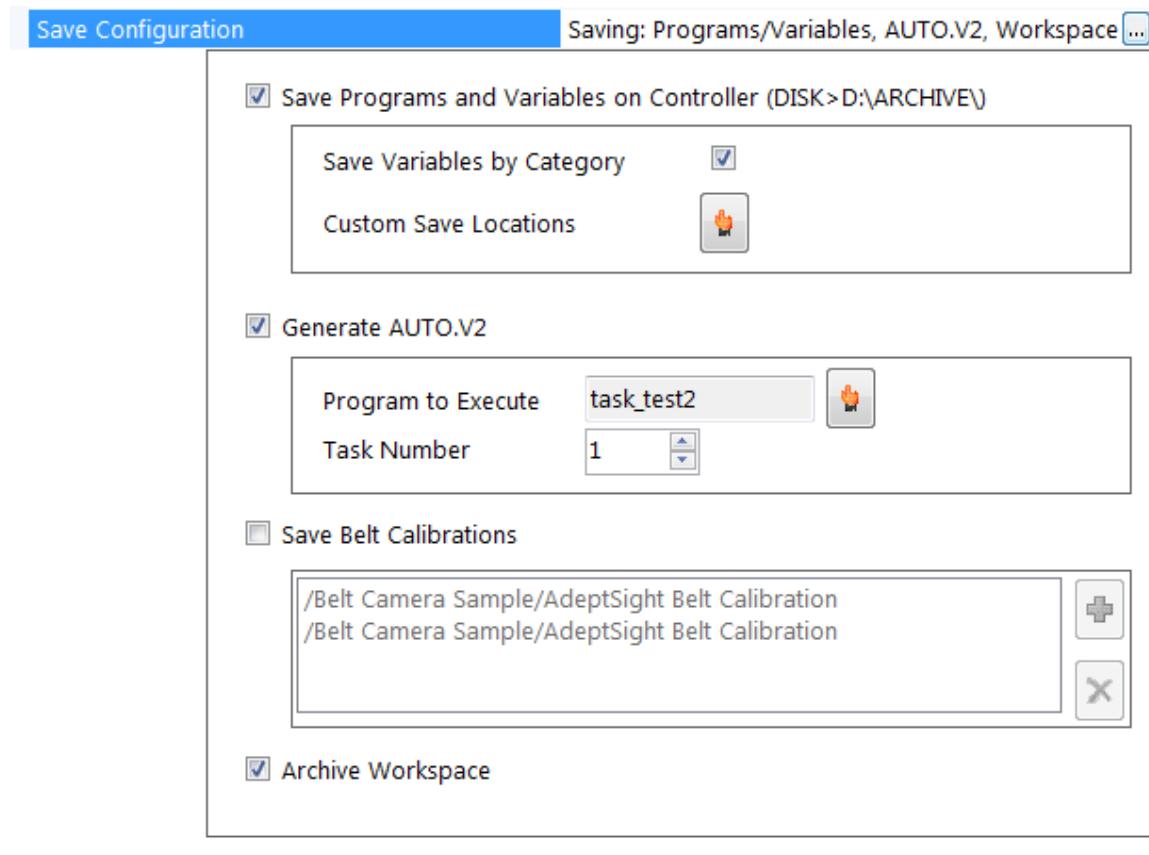
### **Editor Parameters and Online Help**

The middle portion of the Controller Editor contains the editor parameters. These are used to configure various settings on the connected controller. The bottom area of the Controller Editor displays online help for the selected parameter.

**NOTE:** Because online help for each parameter is provided directly in the editor, it is not duplicated in this documentation.

### **Save Configuration**

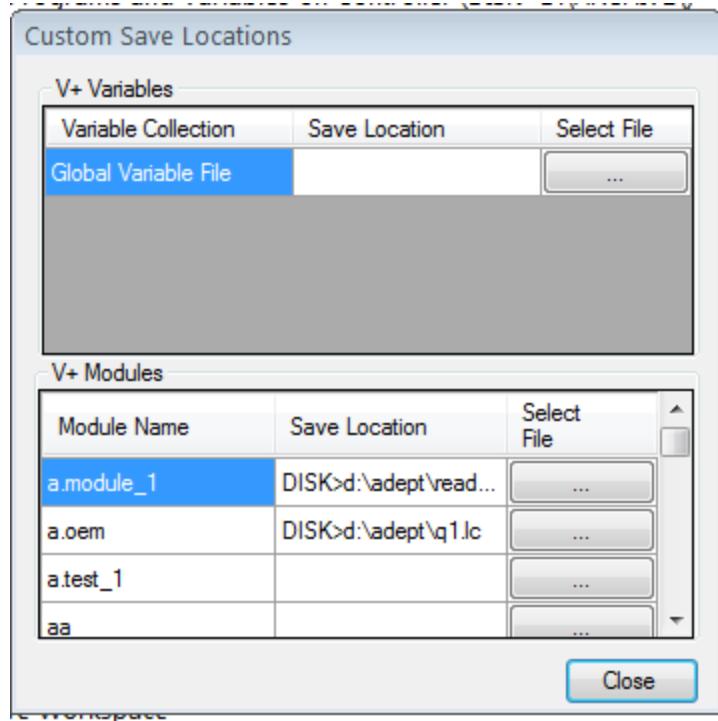
The save configuration section defines what information is saved locally on the controller when the workspace is saved. The options are as follows:



#### **Save Programs and Variables on Controller (DISK>D:\ARCHIVE\)**

When enabled, all V+ User variables and V+ User modules will be saved to the controller in the **DISK>D:\ARCHIVE\** directory. By default, all variables will be saved in a file called GLOBALS.VAR and all modules will be saved in a file called {MODULE NAME}.PG. If the **Save Variables by Category** option is selected, only variables assigned to a category are saved and the file name is based on the category name.

The names and locations of these files can be changed by accessing the **Custom Save Locations** button:



### Generate AUTO.V2

When selected, an AUTO.V2 file will automatically be generated. If **Save Programs and Variables on Controller** is enabled, the files will be loaded as part of the AUTO program. Additionally, you can specify a **Program to Execute on a Task Number**. These will be included in the AUTO program.

### Save Belt Calibrations

When selected, you can identify belt-calibrations in the workspace that will be saved to the controller. These calibrations will be saved into the DISK>D:\ARCHIVE\ folder in a file called BELT.DAT. The V+ Belt calibration utility can be used to load and access the information.

### Archive Workspace

When enabled, the workspace will be saved on the controller in the DISK>D:\ARCHIVE\ folder.

## System Options

The Edit Options windows provides an interface for enabling one or more of the following startup options:

- V+ Module Settings: enables the automatic insertion of a program header when a new V+ program is created. For details, see V+ Module Settings on page 105.
- Vision Server: enables the remote vision server access and specify a port number for the vision

server. For details, see Vision Server on page 106.

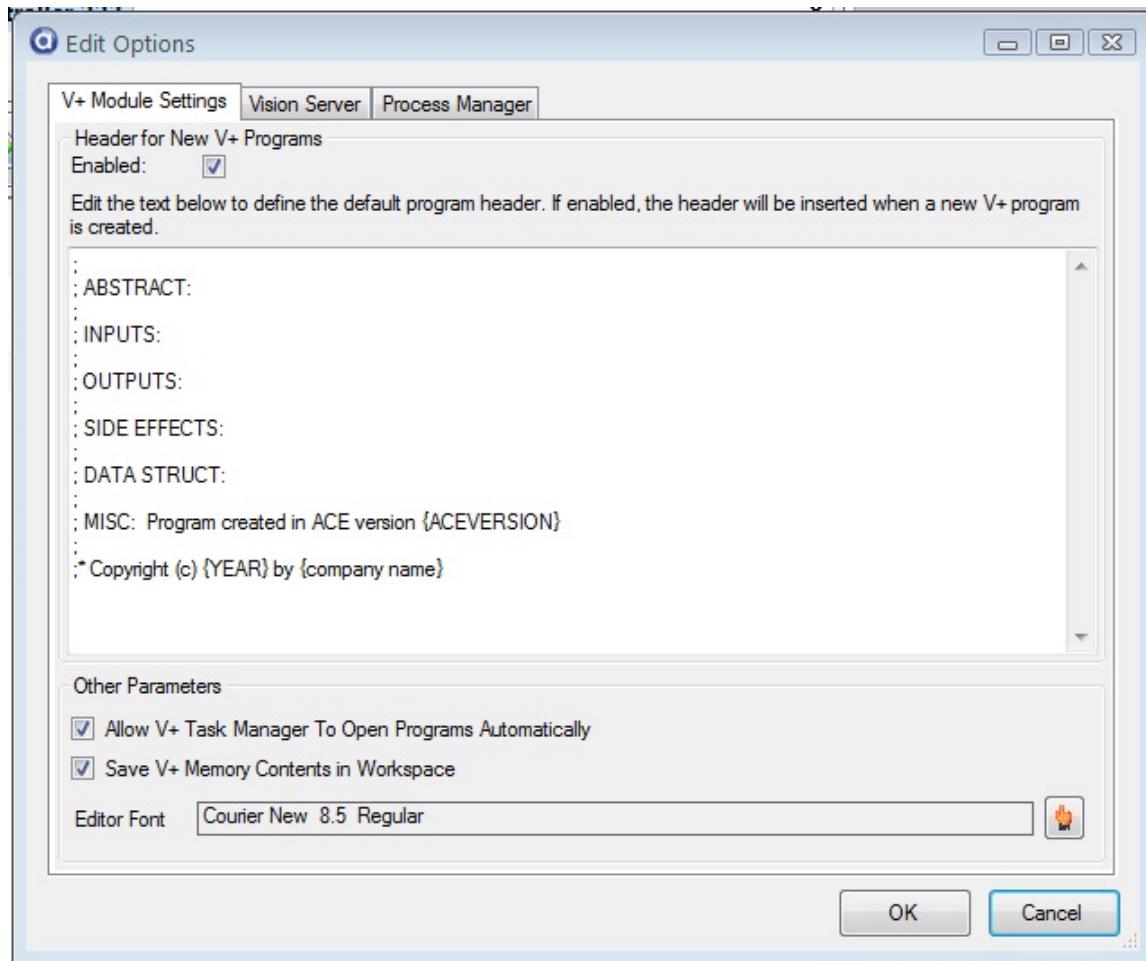
**Process Manager Settings:** configures what process manager information is saved locally on the current PC when the workspace is saved. For details, see Process Manager on page 108.

To open the Edit Options window, from the ACE menu, select **Tools > Options**. The Edit Options window opens, as shown in the following figures. Each of the startup options is described in the following sections.

### **V+ Module Settings**

This startup option allows you to enable the automatic insertion of a program header when a new V+ program is created. To enable the automatic program header:

1. Click the V+ Module Settings tab. The V+ Module Settings options are displayed.



*Edit Options - V+ Module Settings*

2. Select the Enabled check box. If enabled, the program header is automatically inserted when a new V+ program is created.
3. Edit the default text to define the default V+ program header.
4. Optionally, select the check box to allow the V+ Task Manager to open programs automatically. If this is checked, when a V+ task stops, the ACE software opens that task's program in the V+ editor.
5. Optionally, click the browse () icon to select a different font (type face and size) for the V+ program editor.
6. Optionally, select the check box to allow or disable the saving of V+ variables and modules in the workspace.
7. Click **OK** to store the settings and close the window. Click **Cancel** to close the window without storing the settings.

For more details on creating a V+ program, see V+ Editor Tool on page 765.

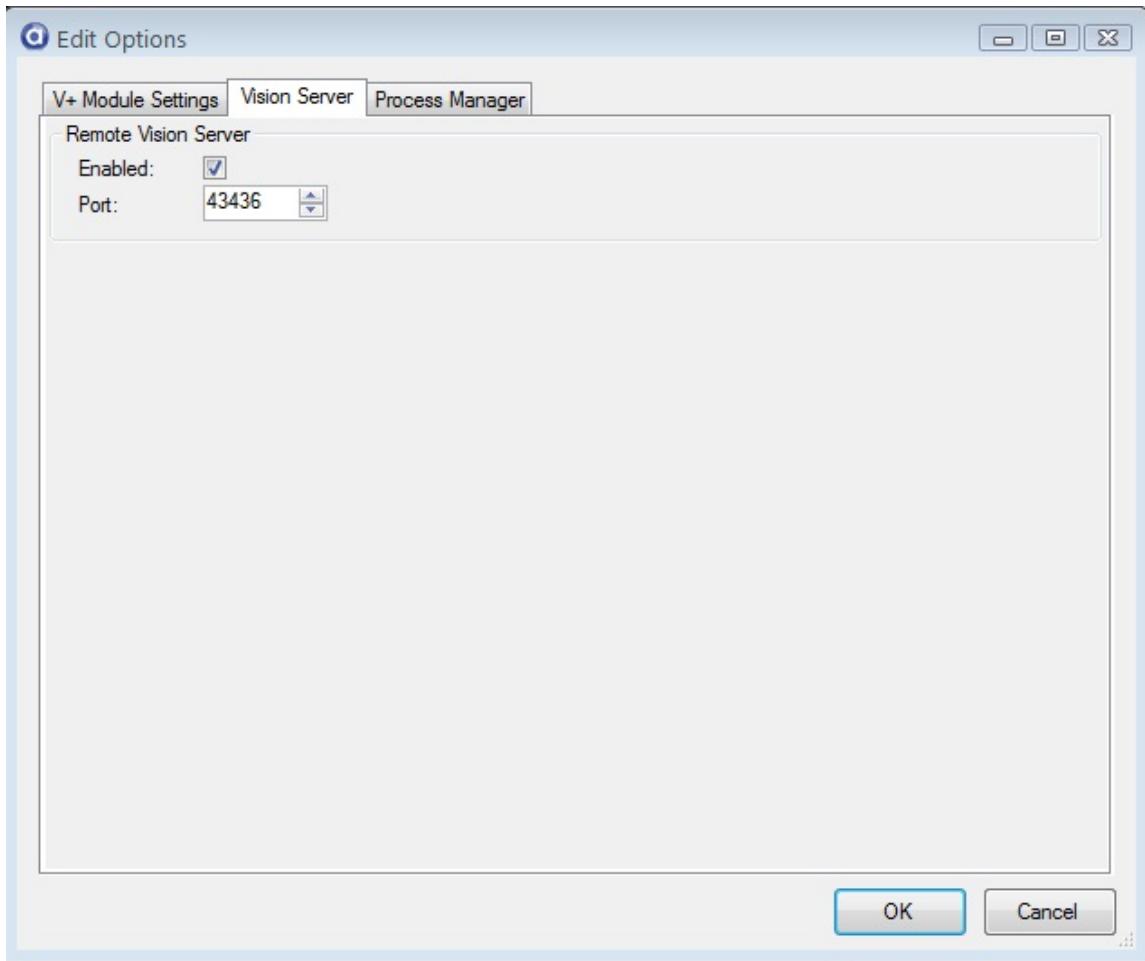
### **Vision Server**

This option allows you to enable the remote vision server, which is used to communicate with a second PC that is running one or more vision tools. To enable the remote vision server:

## System Options

---

1. Click the Vision Server tab. The Remote Vision Server options are displayed.



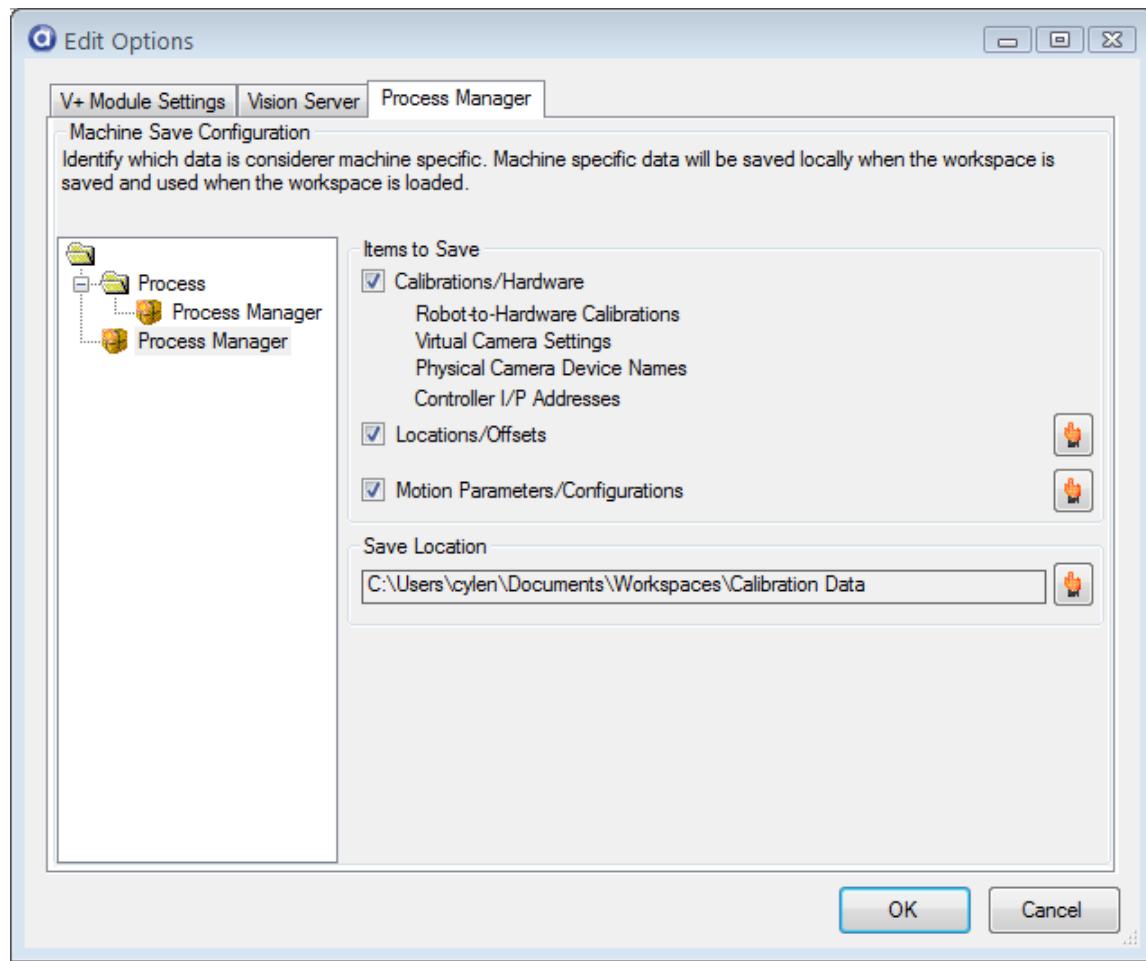
### Edit Options - Remote Vision Server

2. Select the Enabled check box.
3. Specify the port number. This is the port (socket) that the remote vision server will open for providing the requested vision results. Note that the remote vision server does not "push" the data to the clients, it simply allows data requests to be accepted.
4. Click **OK** to store the settings and close the window. Click **Cancel** to close the window without storing the settings.

For more information on using the Remote Vision option, see the *ACE Sight User's Guide*.

### **Process Manager**

This option allows you to configure what information is saved locally by each process manager in your workspace:



You can select a process manager and define what information is saved locally when the workspace is saved. When the workspace is saved, a copy of the data is placed in the directory specified by the user. In the future, when the workspace is loaded, it will overwrite the data in the workspace with the data saved in the specified directory. This can enable a single workspace to be used on multiple systems. Items which are specific to an individual machine will be used when the workspace is loaded.

### **Gripper Editor**

End-effectors, also called tools and grippers, can be used in pick, place, dispensing or other activities. End-effectors are often driven by digital outputs to grip, release, or dispense. Most robots have a single end-effector, but others may have multiple end-effectors to pick and place multiple objects at the same time. ACE supports all these variations.

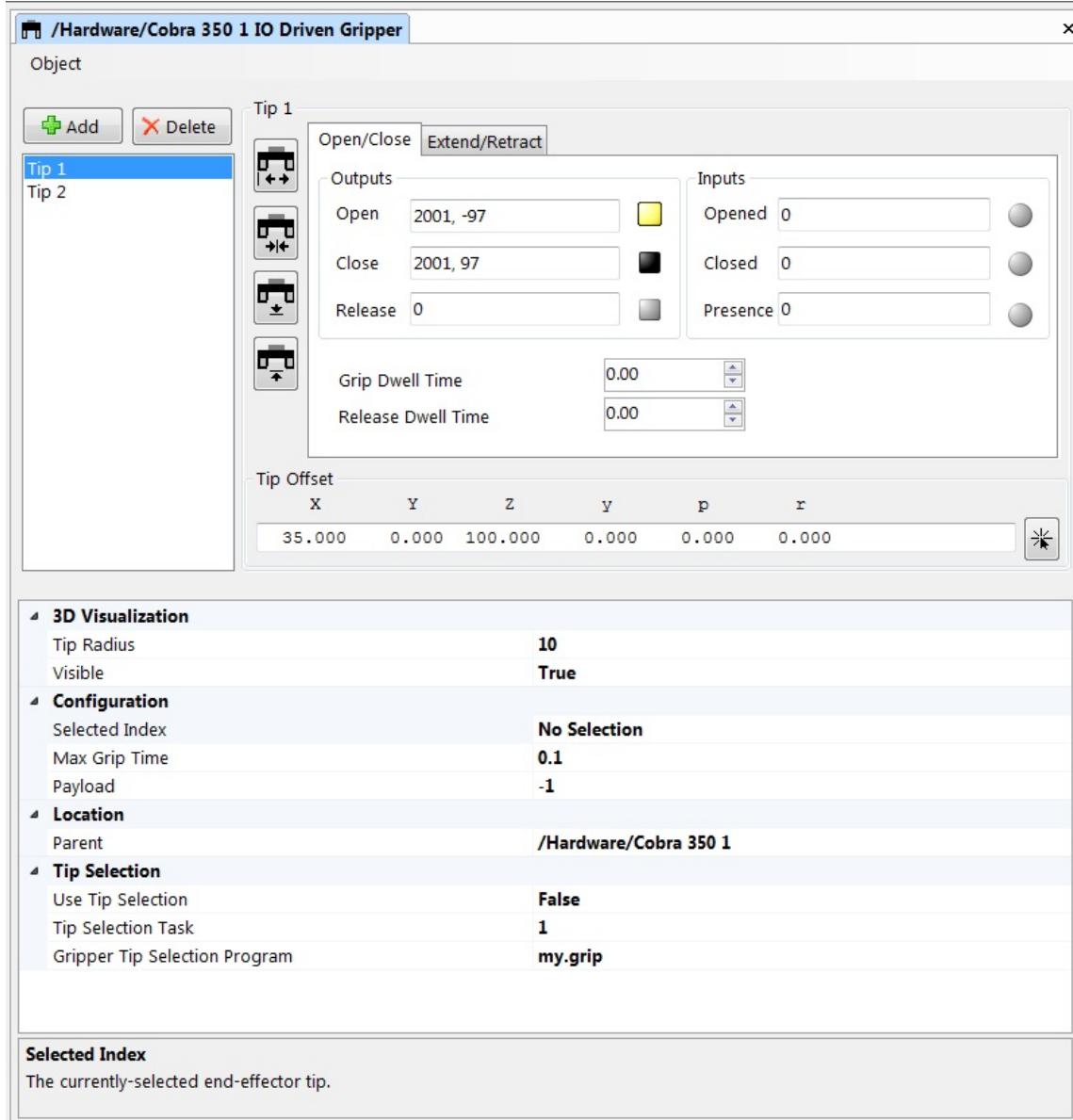
By default, one end-effector object, IO Driven Gripper, is provided. This is an I/O driven gripper with single or multiple end-effector tips. It uses digital input and output signals to control each tip. Additional gripper objects may be added, as needed, for your application.

To add an additional gripper to the workspace, right-click in the folder area of the Workspace Explorer and select **New > Device > Tool > IO Driven Gripper**.

The IO Driven Gripper editor provides an interface for setting various gripper-related parameters, such as end-effector tips, minimum grip time, and maximum grip time. To open the IO Driven Gripper editor, double-click the Gripper object in the folder area of the Workspace Explorer.

The Gripper properties can be exposed for view/control through OPC. For details, see OPC Data Access and Process Control on page 785.

## Gripper Editor

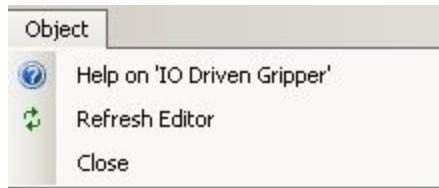


*IO Driven Gripper Editor*

### Menu Items

This section describes the selections available from the IO Driven Gripper editor menu.

## Object Menu



### Help on 'IO Driven Gripper'

Displays the online help for the Gripper editor.

### Refresh Editor

Refreshes the contents of the Gripper editor window.

### Close

Closes the Gripper Editor.

## Editor Parameters and Online Help

The middle portion of the IO Driven Gripper editor contains the editor parameters. These are used to configure various settings on the selected robot. The bottom area of the IO Driven Gripper editor displays online help for the selected parameter.

Several of the important IO Driven Gripper editor parameters are described below:

Item	Description
Add/Delete buttons	Used to add a new gripper tip or to delete an existing gripper tip.
Tip list	<p>Displays the list of defined gripper tips.</p> <p><b>NOTE:</b> For robot-to-belt or robot-to-camera calibrations, if a gripper has multiple tips defined, the entire tool (all tips) will be used to pick up and release the part during the calibration procedure. The centerline of the gripper will be used as the reference TOOL in the calibration procedure.</p>
Input/Output signals	<p>Defines the Open/Close or Extend/Retract activation signals, and Opened/Closed or Extended/Retracted status signals for the selected tip. You can define multiple signals by entering the signal numbers separated by a comma (for example: 97, 98). When multiple signals are defined, the following icon colors apply:</p> <ul style="list-style-type: none"> <li><span style="color: green;">■</span> on</li> <li><span style="color: black;">■</span> off</li> <li><span style="color: yellow;">■</span> multiple signals, not all are on or off</li> </ul> <p><b>NOTE:</b> When emulation mode is enabled, the Input signals are ignored; if the Output signals are not valid signals, they are ignored.</p>
Presence	Through the specified signal from a part-presence sensor, indicates the presence (on) or absence (off) of a part in the gripper. If specified, the status of the presence sensor is checked when picking or placing a part: before the robot places a part at a target, the signal must be on; before the robot picks a part, the signal must be off.

Item	Description
Grip Dwell Time Release Dwell Time	The time (in seconds) to wait, when gripping or releasing, before continuing operation; this should be the actuation time for the gripper.
Open/Close Tab	Click this tab to access the Open/Close signal settings for the selected tip.
Extend/Retract Tab	Click this tab to access the Extend/Retract signal settings for the selected tip.
Grip/Release buttons	Use the grip (  ) and release (  ) buttons to send a grip (close) or release (open) signal to the selected tip.
Extend/Retract buttons	Use the extend (  ) and retract (  ) buttons to send an extend (move out) or retract (move in) signal to the selected tip.
Extend/Retract Dwell Time	The time (in seconds) to wait, when extending or retracting, before continuing operation. The value represents the minimum time to dwell. After the specified dwell time, the input signals are checked.
Tip Offset	Shows the current offset for the selected tip. To change the offset, click the  button, which starts the <a href="#">Tool Offset Wizard</a> .
Tip Radius	The radius of the tip when drawing the end-effector in the 3D virtual display.
Visible	Enables the display of the end-effector tip positions in the 3D virtual display.
Selected Index	0 for an end-effector with only one tip; up to n-1 for an end-effector with 'n' tips.
Max Grip Time	Only used when a grip input signal is specified, this value specifies the maximum time in seconds for the grip input signal to become true when picking a part, or false when placing a part.
Payload	The weight of the gripper in kg.
Parent	Because the gripper is mounted to the robot, the parent of the gripper is typically the robot. In the case of a tool-changer, the parent changes when the gripper is put on the tool stand.
Gripper Tip Selection Program	These parameters allow you to enable and call a V+ tip-selection program and specify a task for executing the program. This is used in the case where the gripper has some additional operation required to switch the tips. For example, there might be some additional I/O that needs to be set (or a motor moved) to physically move the tips

**NOTE:** Help for the remaining parameters is provided directly in the editor.

## Note Editor

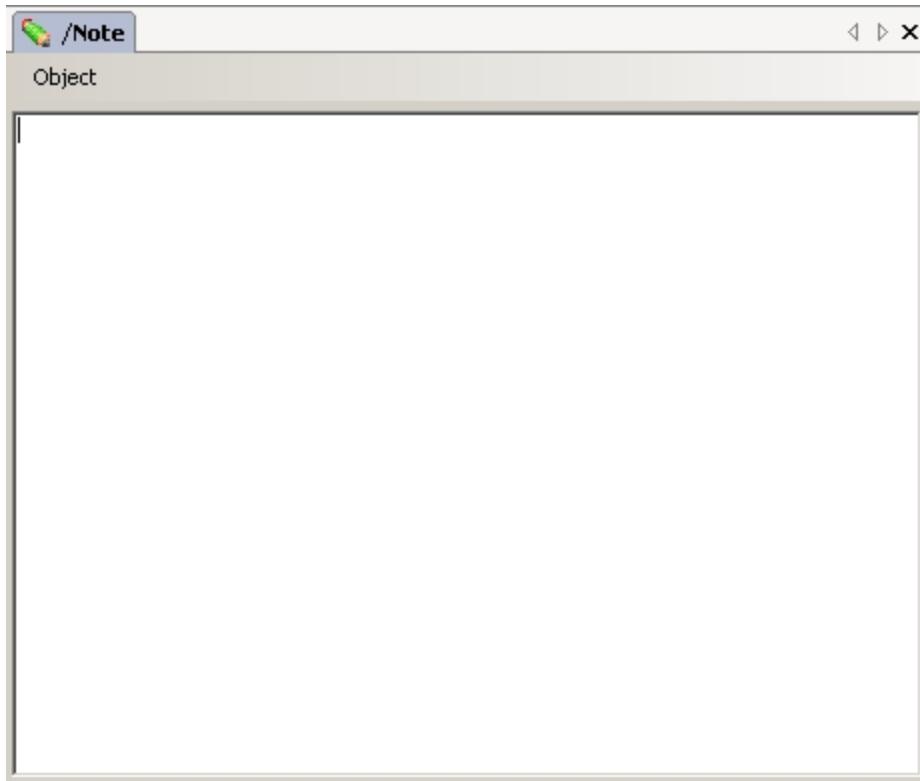
The Note editor allows you to type information about your ACE software application. For example, you can create entries about recent changes you've made or new features you've added. The Note objects are saved as part of your ACE workspace.

### ***Creating a Note Object***

To create a new Note object, right-click in the folder area of the Workspace Explorer, and select:

**New > Configuration > Note**

To open the Note editor, double-click the Note object in the folder area of the Workspace Explorer.

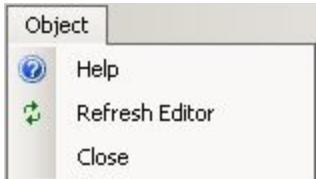


*Note Editor*

### ***Menu Items***

This section describes the selections available from the Note editor menu.

## Object Menu



### Help

Displays the online help for the Note editor.

### Refresh Editor

Refreshes the contents of the Note editor window.

### Close

Closes the Note editor.

## ***Creating and Editing Notes***

To create a new note:

1. Create a Note object. The Note object is added to the folder area of the Workspace Explorer and the Note editor opens.
2. Click inside the Note editor and begin typing your notes.

To edit a note:

1. Locate the Note object you wish to edit in the folder area of the Workspace Explorer.
2. Double-click the Note object to open it. The previously-typed notes are displayed in the Note editor.
3. Add your new notes or edit the existing notes.

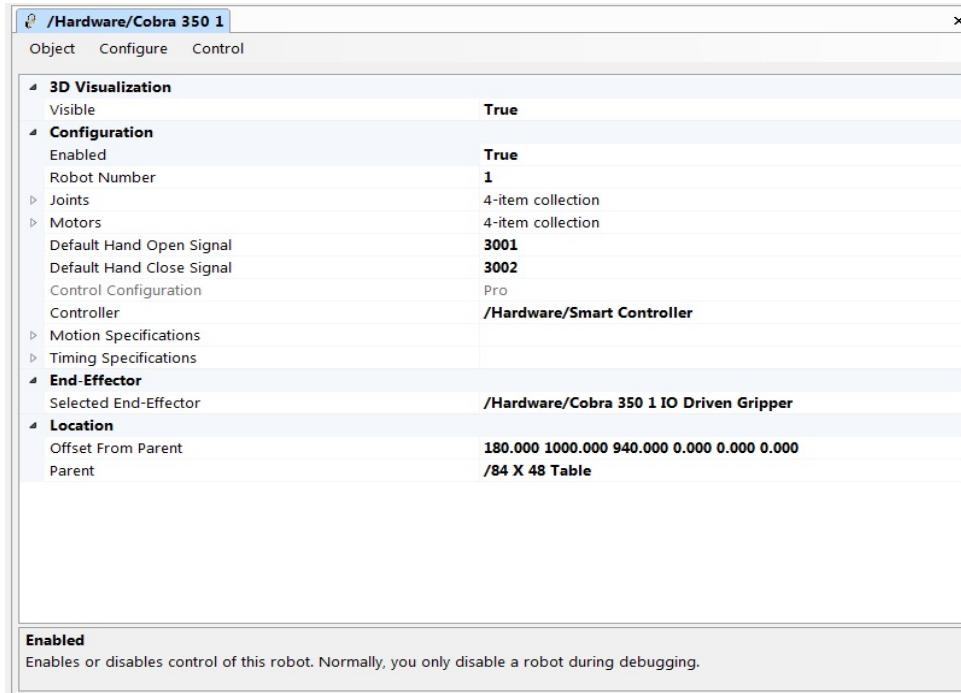
## Robot Editor

The Robot editor provides an interface for setting various robot-related parameters, such as joint parameters, motor parameters, and the selected end-effector.

To add an additional robot to the workspace, navigate to the Controller Editor: See "Controller Editor" To open the Robot editor, double-click the robot object in the folder area of the Workspace Explorer.

## Robot Editor

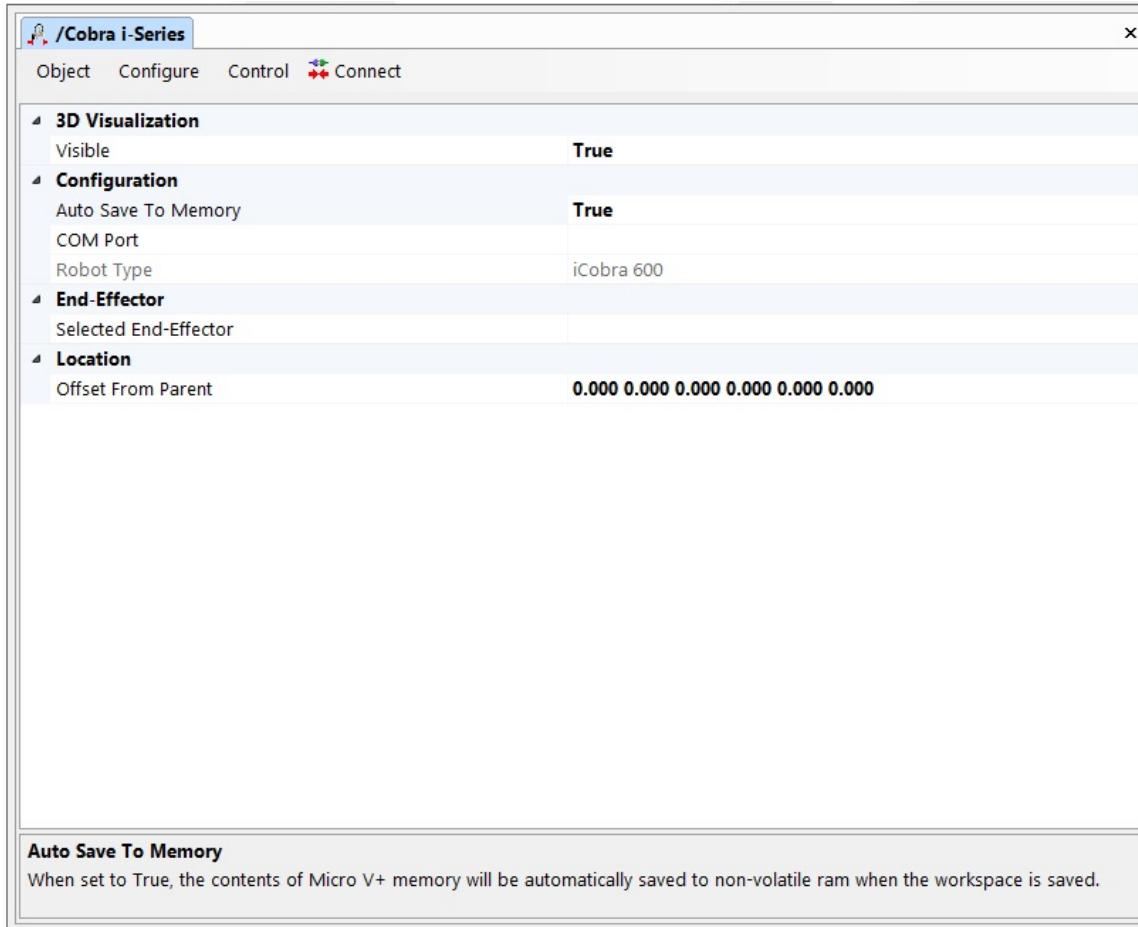
---



*Robot Editor*

## Robot Editor

---

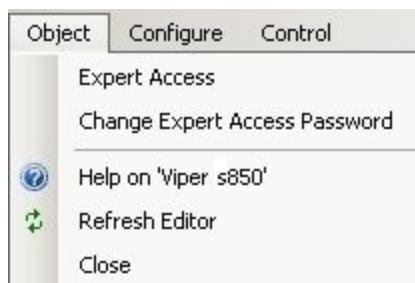


Cobra i-Series Robot Editor

### Menu Items

This section describes the selections available from the Robot editor menu.

#### Object Menu



#### Expert Access

Used for controlling “Expert Access” to robot parameters. When Expert Access is enabled, it permits writing to robot parameters that are normally protected from the ACE API.

#### Change Expert Access Password

Displays the Change Password dialog, which is used to change the password for accessing Expert Access mode.

#### Help on [Robot]

Displays this user guide.

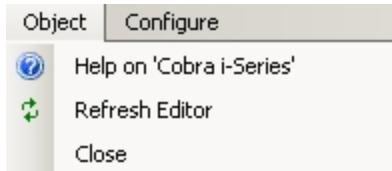
**Refresh Editor**

Refreshes the contents of the Robot editor window.

**Close**

Closes the Robot editor.

**Object Menu (Cobra i-Series Robot)**



**Help on [Robot]**

Displays this user guide.

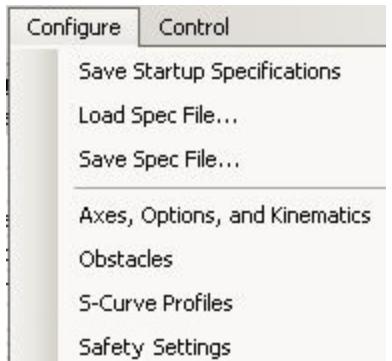
**Refresh Editor**

Refreshes the contents of the Robot editor window.

**Close**

Closes the Robot editor.

**Configure Menu**



**Save Startup Specifications**

Saves the robot startup specifications to a disk file.

**Load Spec File...**

Displays the Load Robot/Motor Specification File dialog box. For details, see the topic [Loading and Saving Robot/Motor Specification Files](#).

**Save Spec File...**

Displays the Save Robot/Motor Specifications dialog box. For details, see the topic [Loading and Saving Robot/Motor Specification Files](#).

**Axes, Options, and Kinematics**

Displays the Axes, Options, and Kinematics dialog box. For details, see the topic [Configuring Axes, Options, and Kinematic Parameters](#).

**Obstacles**

Displays the Save Robot/Motor Specifications dialog box. For details, see the topic [Editing Obstacles](#).

**S-Curve Profiles**

Displays the S-Curve Profile Configuration dialog box. For details, see the topic [Configuring S-Curve Profiles](#).

**Configure Menu (Cobra i-Series Robot)**

**Configuration Manager**

Displays the Configuration Manager dialog for the Cobra

i600/i800 robot. For details, see the topic [Cobra i600/i800 Robot Configuration](#) on page 265.

#### **License Manager**

Displays the License Manager dialog for the Cobra i600/i800 robot. For details, see the topic [License Manager](#) on page 296.

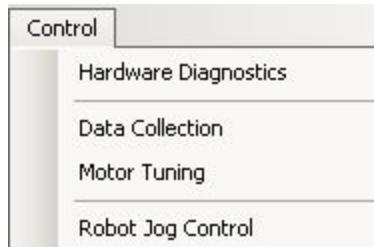
#### **Diagnostics Wizard**

Displays the Cobra i600/i800 robot diagnostics wizard. For details, see the topic [Cobra i600/i800 Robot Diagnostics](#) on page 1130.

#### **Switches and Parameters**

Displays the Switches and Parameters dialog for the Cobra i600/i800 robot. For details, see the topic [Switches and Parameters](#) on page 299.

### **Control Menu**



#### **Hardware Diagnostics**

Displays the Hardware Diagnostics dialog box. For details, see the topic [Robots and Encoders](#).

#### **Data Collection**

Displays the Robot Data Collection dialog box. For details, see the topic [Collecting Data](#).

#### **Motor Tuning**

Displays the Robot Motor Tuning dialog box. For details, see the topic [Tuning the Motors](#).

#### **Robot Jog Control**

Displays the Robot Jog Control. For details, see the topic [Robot Control](#).

### **Connect (Cobra i-Series Robot)**



#### **Connect**

Connects (initiates communication) to the COM port specified in the COM Port field of the Configuration grid editor.

After the connection is established, the button name and function changes to Disconnect, which disconnects (stops communication) to the specified COM port.

### ***Editor Parameters and Online Help***

The middle portion of the Robot editor contains the editor parameters. These are used to configure various settings on the selected robot. The bottom area of the Robot editor displays online help for the selected parameter.

**NOTE:** Because online help for each parameter is provided directly in the editor, it is not duplicated in this documentation.

## System Startup Editor

The System Startup editor allows you to specify various ACE software start-up options. For example, you can specify an auto-run file that will be executed when the ACE software is connected to a controller.

### ***Creating a System Startup Object***

To create a new System Startup object, right-click in the folder area of the Workspace Explorer, and select:

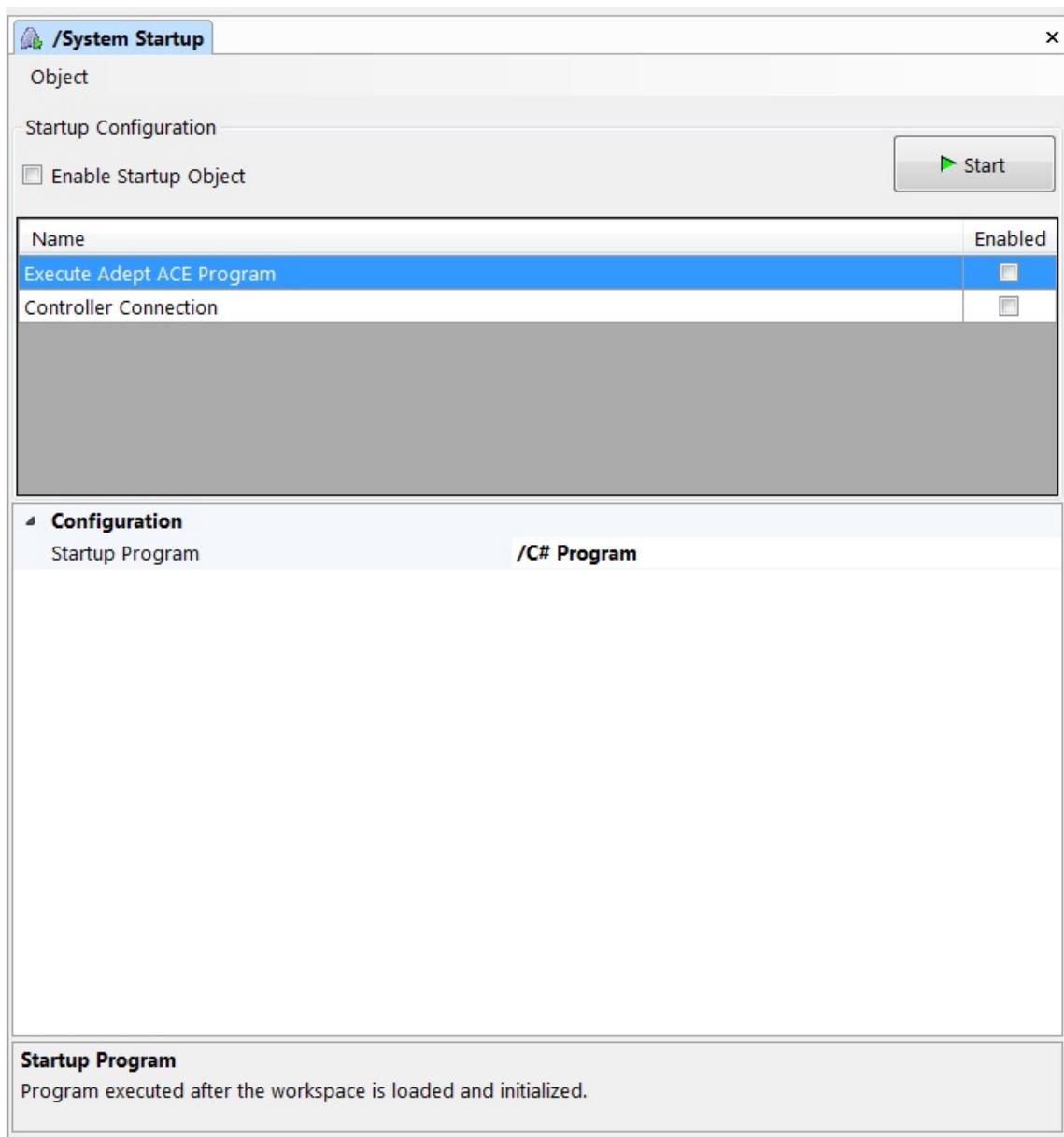
**New > Configuration > System Startup**

To open the System Startup editor, double-click the System Startup object in the folder area of the Workspace Explorer.

**NOTE:** Only one System Startup object is permitted in the Workspace Explorer.

## System Startup Editor

---

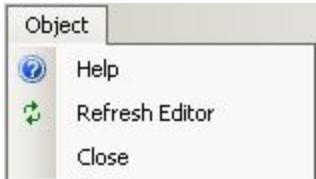


*System Startup Editor - Execute ACE Program Option*

### **Menu Items**

This section describes the selections available from the System Startup editor menu.

## Object Menu



### Help

Displays the online help for the System Startup editor.

### Refresh Editor

Refreshes the contents of the System Startup editor window.

### Close

Closes the System Startup editor.

## ***Selected Configuration***

The top region is used to select a startup option for configuration:

- [Execute ACE Program](#) - Executes an ACE program at system startup.
- [Controller Connection](#) - Establishes communications with the selected controller(s) at system startup.

The Enabled check box must be selected (checked) to enable the specified configuration to load at system startup.

The Start button invokes the startup operation. This can be used to test a startup configuration without reloading the workspace.

### ***Execute ACE Program Option***

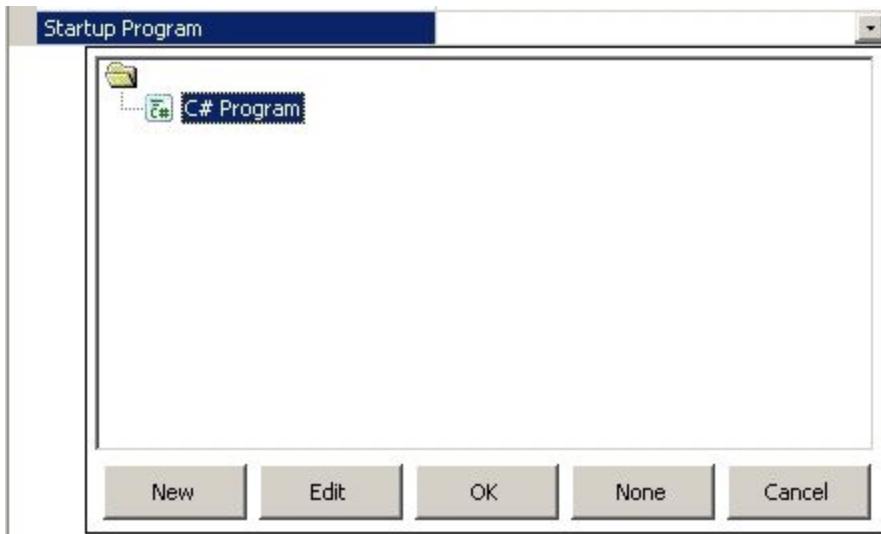
This section describes the Execute ACE Program option.

## **Specifying a Startup Program**

The Startup Program is a program that executes when the ACE software is started.

To specify a Startup Program:

1. Click the field to the right of the Startup Program parameter.
2. Type the program name directly into the field, or use the down-arrow icon to display a resource list. You can select a program from the list and then click **OK**, or click **New** to create a new program.



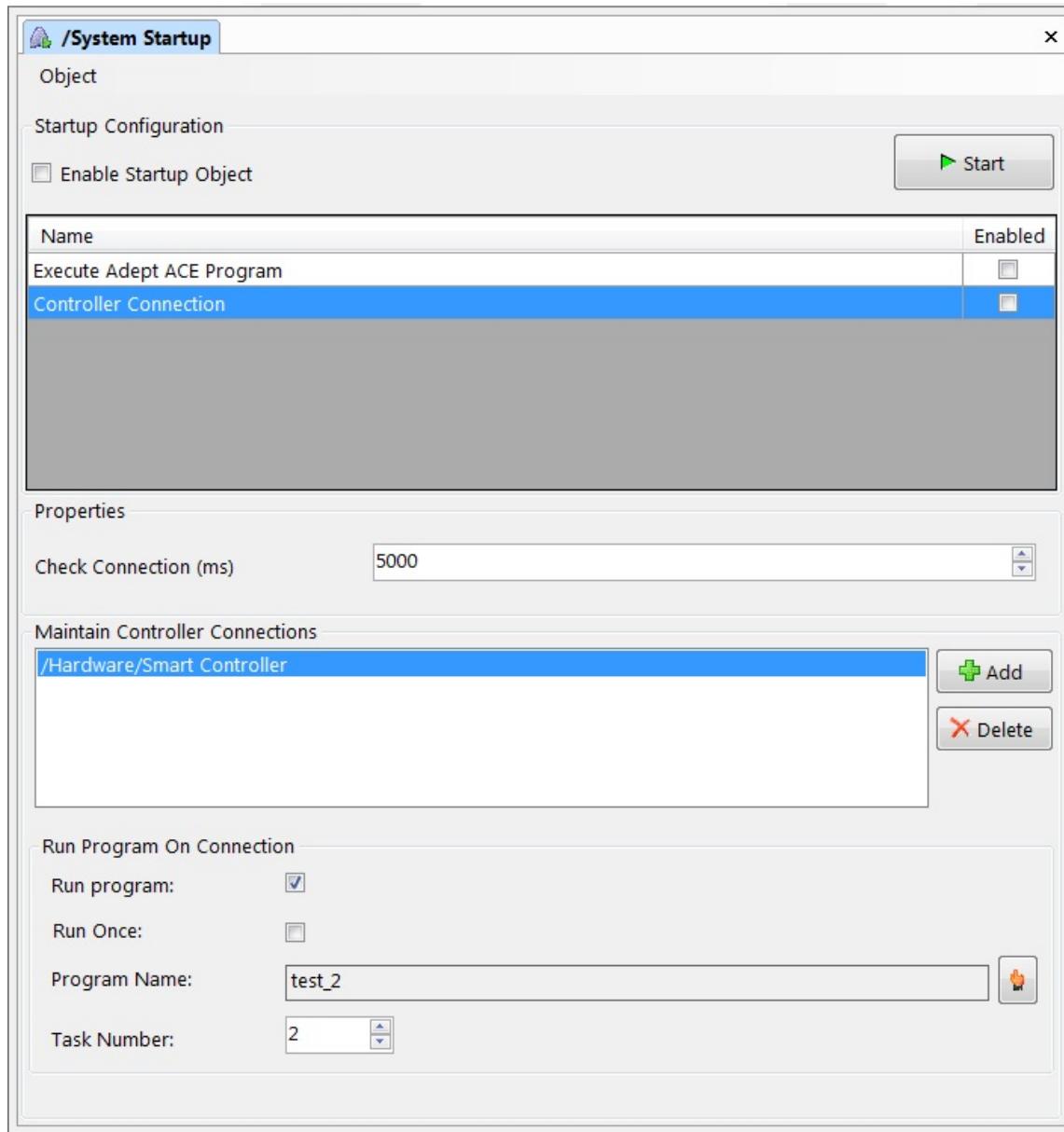
*Resource List*

***Controller Connection Option***

This section describes the Controller Connection option.

## System Startup Editor

---



*System Startup Editor - Controller Connection Option*

The Controller Connection startup option is used to establish a connection with (and then monitor and maintain the connection) to one or more controllers in the workspace.

Item	Description
Properties - Check Connection	Specifies the interval (in milliseconds) for checking the connections to the controllers specified on the Maintain Controller Connections list.
Maintain Controller Connections	Shows the list of controllers being monitored. Use the Add/Delete buttons to add or delete controllers.
Add	Click to add a controller to the Maintain Controller Connections list.  <b>NOTE:</b> You can only add a controller that already exists in the workspace. You cannot create a new controller connection.
Delete	Click to remove the selected controller from the Maintain Controller Connections list.
Run Program on Connection	This group is used to specify a V+ program to run when a connection is made to the selected controller. For details, see Run a Program on Connection on page 124.
Run Program	Select to enable a specified V+ program to run when the selected controller is connected.
Run Once	When selected, the specified program will be run one time, when ACE is first connected to the controller. It will not re-run when disconnected and reconnected. If you need to re-run the program, you will need to shutdown ACE and restart it.
Program Name	Specifies the program to run when the selected controller is connected.
Task Number	Specifies the V+ task number for the V+ program.

**NOTE:** Although the Controller Connection option and the AUTO Controller Startup option share some similarities, the startup file for the Controller Connection option is stored on the PC, whereas the AUTO Controller Startup option files are stored on the controller. This allows the AUTO Controller Startup option to provide a "headless" (no PC required) operation of the application.

## Run a Program on Connection

To run a V+ program on connection:

1. Select a controller from the Maintain Controller Connections list.
2. In the Run Program on Connection group:
  - a. Select Run Program on Connection.

- b. Specify the V+ program name or use the Browse () icon to select the program.
- c. Specify the V+ task number for the program.

**NOTE:** The specified V+ task number must be idle. Otherwise, the program will not run on connection.

## User Manager Editor

The User Manager provides an interface for controlling users and access levels to your ACE-controlled system. It allows you to create a list of users and assign an access level to each user. A user then has access to those features permitted by his/her assigned access level. For a listing of the items that are available to each access level, see Appendix 2: User Access Item List on page 1168.



**CAUTION:** The User Manager implements a basic level of user-access security. For applications that require a higher level of security, you will need to implement a security scheme within a custom user-interface. This could be based on the network login credentials from Windows (or similar access-control method).

To create a new User Manager object, right-click in the folder area of the Workspace Explorer and select:

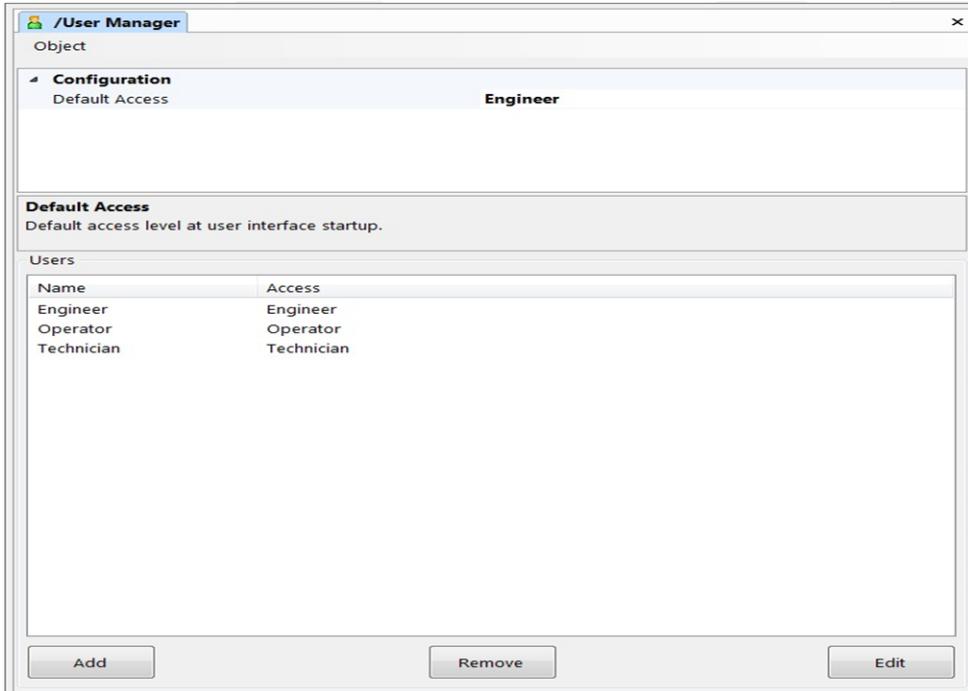
**New > Configuration > User Manager**

To open the User Manager editor, double-click the User Manager object in the folder area of the Workspace Explorer.

**NOTE:** Only one User Manager object is permitted in the Workspace Explorer.

## User Manager Editor

---

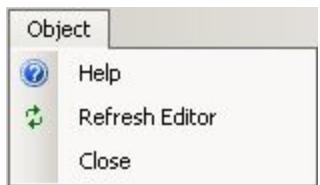


User Manager Editor

### Menu Items

This section describes the selections available from the User Manager editor menu.

#### Object Menu



#### Help

Displays the online help for the User Manager editor.

#### Refresh Editor

Refreshes the contents of the User Manager editor window.

#### Close

Closes the User Manager editor.

### Editor Parameters, Online Help, and Users List

The middle portion of the User Manager editor contains the editor parameters. There is one parameter available, which is used to define the default access level. The default access level is used when ACE is started or when no one is signed into the system.

The area below the editor parameters displays online help for the selected parameter.

The bottom portion of the User Manager editor contains the list of defined users. You can add, edit, or remove users through the steps described in the following sections.

### ***Adding Users and Assigning Access Levels***

To add a new user:

1. Click **Add**. The Add User dialog box opens.
2. In the User Name field, type the name of the new user.
3. Use the Access drop-down menu to view and select the access level for the new user. The following table provides a list of available access levels and the corresponding description:

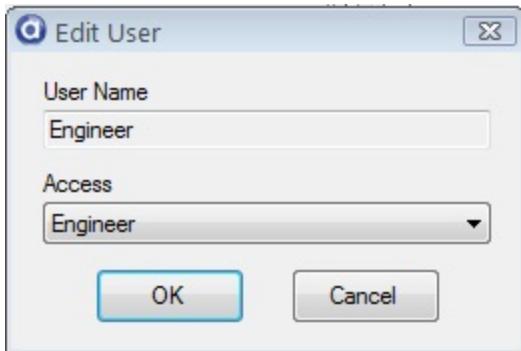
Access Level	Description
Operator	This user can start and stop execution.
Technician	This user can teach basic values.
Engineer	This user can create and edit many values.

4. Click **OK** to save the information and close the dialog box. Or, click **Cancel** to close the dialog box without saving the information.
5. For password-protected access, you will need to add a password to each user account that you want to protect. For details, see Creating or Changing a Password on page 129.

### ***Editing Users***

There may be times when you need to change the access level for an existing user. To edit a user:

1. From the Users list, select the user you wish to edit.
2. Click **Edit**. The Edit User dialog box opens.



3. Use the Access drop-down list to select the new access level for the user.

4. Click **OK** to save the change and close the dialog box. Or, click **Cancel** to close the dialog box without saving.

### **Removing Users**

To remove a user:

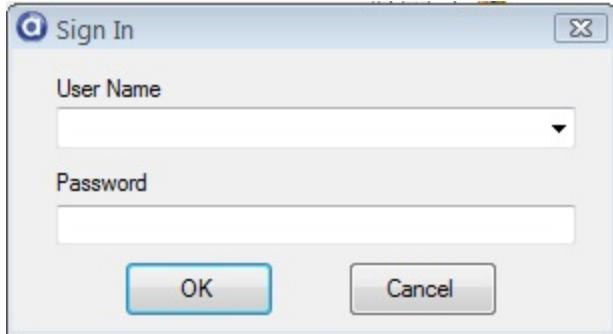
1. From the Users list, select the user you wish to remove.
2. Click **Remove**. The selected user is removed from the list.

### **Signing In**

After you have created your user list, you can sign-in to and sign-out of the available user accounts.

For example, to sign-in to the Engineer account:

1. From the ACE menu bar, select **File > Sign In**. The Sign In dialog box opens.



2. Use the User Name drop-down list to select the user account.
3. If the user account is password-protected, enter the corresponding password in the Password field.
4. Click **OK** to sign in. Or, click **Cancel** to close the dialog box without signing in.

**NOTE:** When a user signs in or out, the event is logged to the Windows operating system Event Log.

### **Signing Out**

**NOTE:** When a workspace is loaded or unloaded, the current user is automatically logged out.

To sign-out of a user account, from the ACE menu bar, select **File > Sign Out**. The ACE software access level returns to the system default.

**NOTE:** When a user signs in or out, the event is logged to the Windows operating system Event Log.

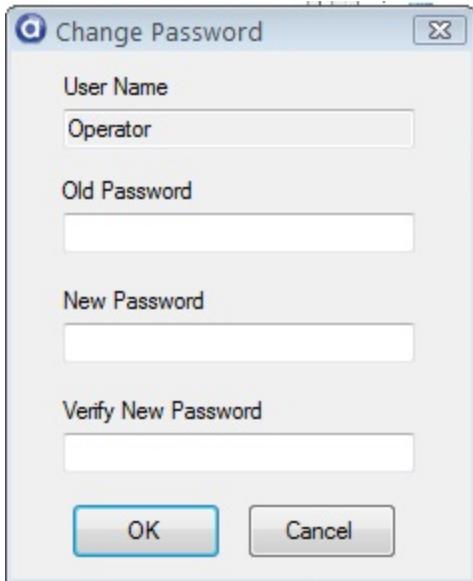
### ***Creating or Changing a Password***

After you have created your user list, you can use the Change Password dialog box to add a password, or change an existing password, for one or more of your user accounts.

**NOTE:** Access to the Change Password dialog box requires a successful login.

For example, to add a password to the Engineer account:

1. Sign in to the Engineer account.
2. Open the Change Password dialog box. From the ACE menu bar, select **File > Change Password**. The Change Password dialog box opens.



3. If the user account is already password-protected, enter the corresponding password in the Old Password field.
4. Enter the new password in the New Password field.
5. Verify the new password by entering it into Verify New Password field.
6. Click **OK** to save the changes. Or, click **Cancel** to close the dialog box without saving.

### **Data Mapper Editor**

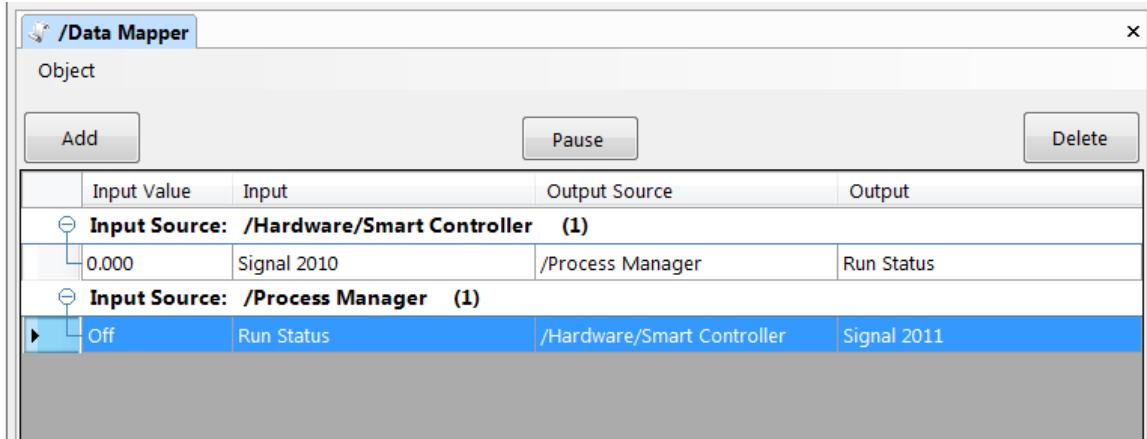
The Data Map allows the user to associate different data items in the workspace as inputs for other data items configured as outputs. For example, you can associate a V+ digital input signal as an input triggering an ACE PackXpert object to run when the signal is turned on.

When entries are added to the list, the conditions are continuously checked.

## Data Mapper Editor

---

To add an additional Data Map object to the workspace, right-click in the folder area of the Workspace Explorer and select **New > Configuration > Data Mapper**.



*Data Mapper Editor*

In the above screen shot, we have 2 different entries defined. The first entry associates **Soft Signal 2010** on **/Hardware/Smart Controller** to the **Run Status** for **/Process Manager**. When signal 2010 is asserted, the process manager will start running. When signal 2010 is turned off, the process manager will stop running.

The second entry associates the **Run Status** of **/Process Manager** with **Soft Signal 2011** on **/Hardware/Smart Controller**. When the process manager is running, soft signal 2011 will be turned on. When the process manager stops running, soft signal 2011 will be turned off.

### ***Editor Items***

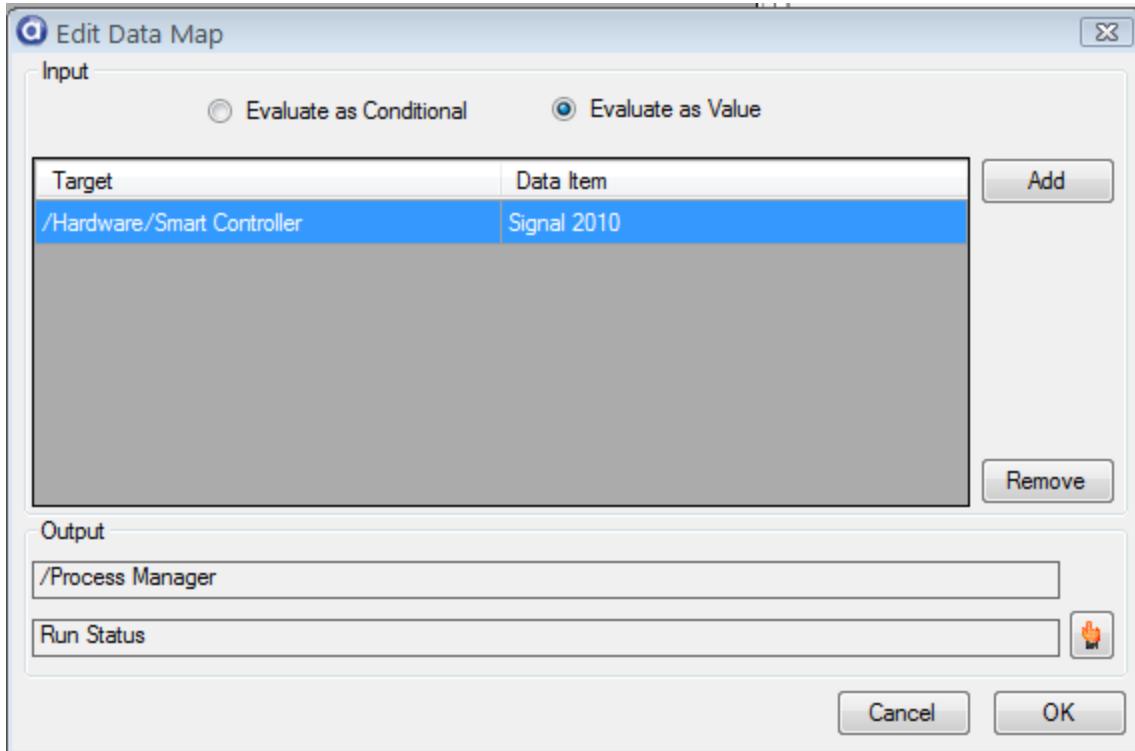
#### **Buttons**

<b>Button</b>	<b>Description</b>
Add	Adds a new data mapping from an input to an output condition to the list.
Pause	Stop the processing of the Data Mapper.
Delete	Removes the selected data mapping from the list.

In addition to the buttons, you can double-click on an existing data map in the list to edit it.

#### ***Data Map***

When creating or editing a data map, the data map editor is displayed:



Multiple input conditions can be added to the input list. The handling of the input values depends on the setting of **Evaluate as Conditional** or **Evaluate as Value**.

### Evaluate As Conditional vs. Evaluate as Value

When **Evaluate as Conditional** is selected, it will interpret each input item as a Boolean item. If the value of the input item is 0, the condition is considered off. If the value is non-zero, the condition is considered on. If all items in the input list are on, then the output condition is asserted. If any item in the input list is off, the output condition is not-asserted.

Additionally, when **Evaluate as Conditional** is selected, you can invert the expected value of an input item. In that case, if the value is 0, the condition is considered to be on.

When **Evaluate as Value** is selected, the value of all input conditions are added together and written to the output value.

### Variable Editor

The Variable editor provides an interface for setting various variables and their related parameters. To create a new Variable object, right-click in the folder area of the Workspace Explorer and select:

**New > Variable > Numeric Variable**

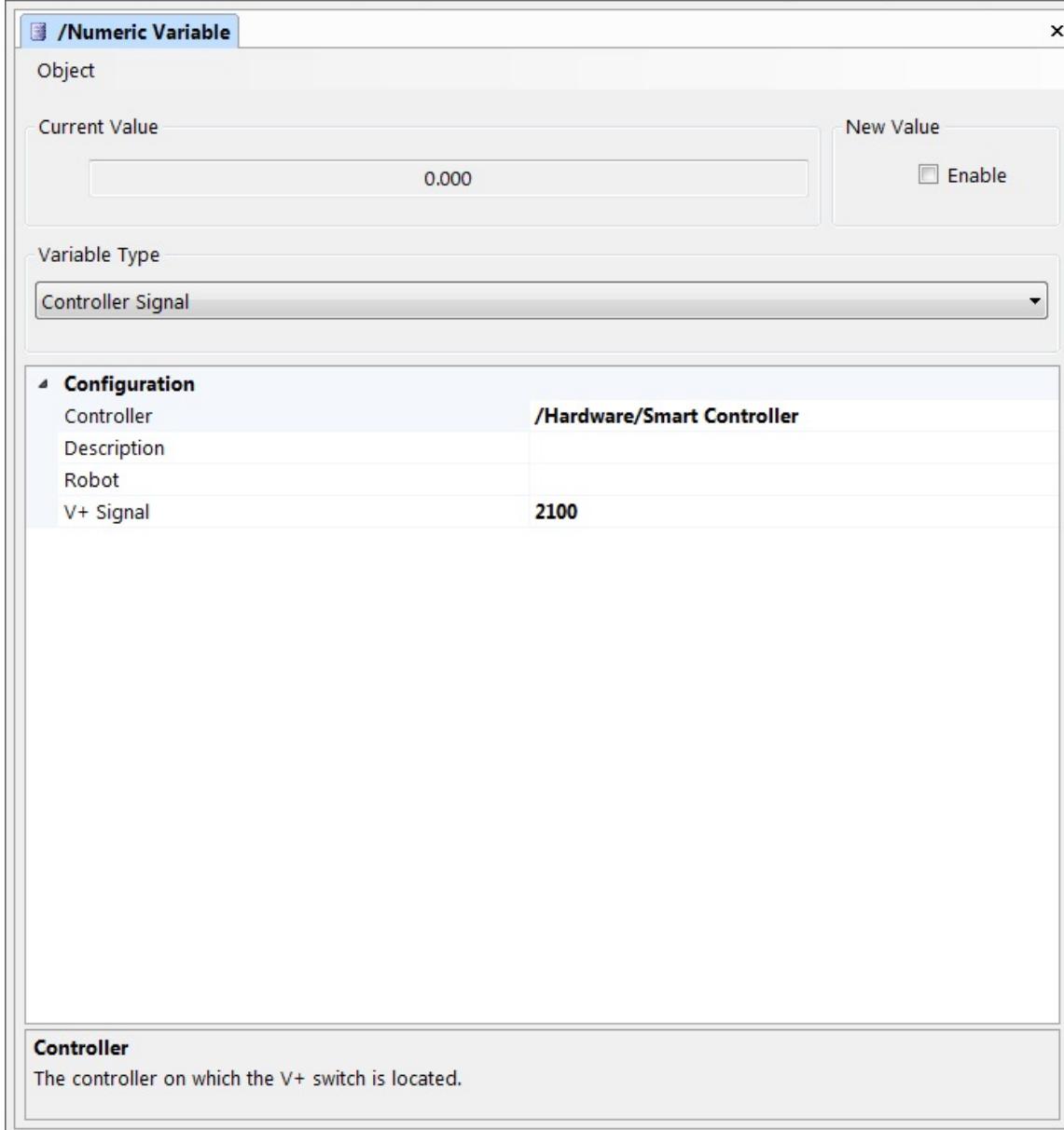
--OR--

**New > Variable > String Variable**

## Variable Editor

---

To open the desired editor, double-click the Variable object in the folder area of the Workspace Explorer.

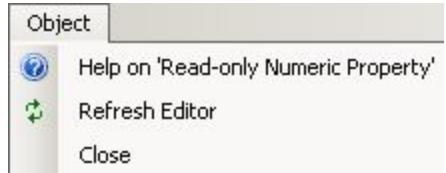


*Variable Editor*

### **Menu Items**

This section describes the selections available from the Variable editor menu.

## Object Menu



### Help on

Displays the online help for the Variable editor.

### Refresh Editor

Refreshes the contents of the Variable editor window.

### Close

Closes the Variable editor.

## ***Editor Parameters and Online Help***

The middle portion of the Variable editor contains the editor parameters. These are used to configure various settings for the variable being edited. The bottom area of the Variable editor displays online help for the selected parameter.

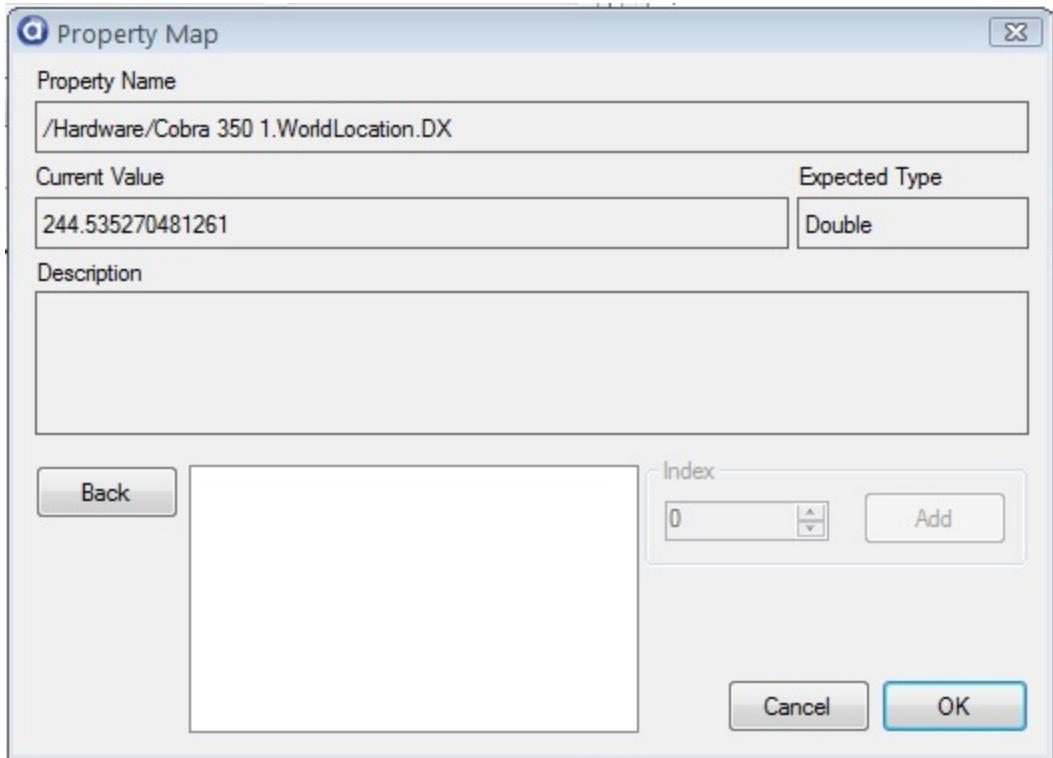
**NOTE:** Because online help for each parameter is provided directly in the editor, it is not duplicated in this documentation.

## ***Using the Property Map Parameter***

The Property Map parameter allows you to specify the property to extract from the target object. In other words, you can make a variable out of any property of any object in the system. This replaces the property list in earlier "property value" variables. Further, it allows you to drill down to access embedded properties.

For example, suppose you want to collect the robot X-coordinate once per second. That can be accomplished with this feature, as follows:

1. Create a variable by right-clicking in the folder area of the Workspace Explorer and selecting **New > Variable > Numeric Variable**.
2. Set the Variable Type to "Object Property"
3. Select the robot as the Target Object.
4. Click the Property Map parameter and then click the Browse (...) icon. The Property Map dialog opens.



5. Double-click the "WorldLocation" property
6. Double-click the "DX" coordinate.
7. Click **OK** to record the selection

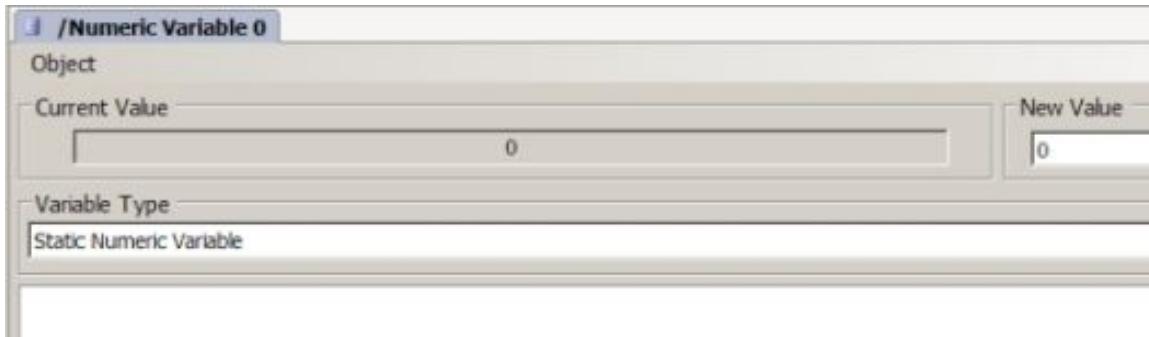
**NOTE:** If you want to change any of your selections, click the Back button to "undo" the previous selection.

8. Optionally, use the Description field to enter a description for the variable. If used, the Description text is exposed in other parts of the ACE interface (for example, through OPC).
9. Create a data collector in your program that references the new variable.

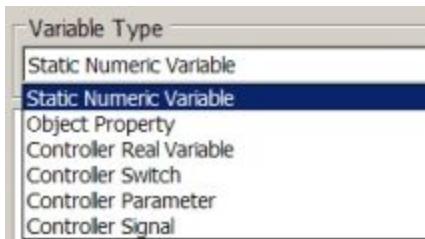
**NOTE:** By design, you are not permitted to have two variables with the same name assigned to the same controller. However, you are permitted to have two variables with the same name on two different controllers. This allows you to reuse programs across controllers and maintain consistency within variable names.

### Variable Types

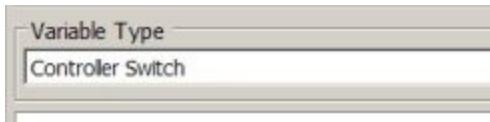
When you create certain ACE variables, the corresponding editor will have a Variable Type drop-down list box. For example, when you create a Numeric variable, the following editor is displayed:



In this editor, you can change the target of the variable by selecting a different Variable type, using the drop-down list box to display the list of available types, as follows:



After the desired Variable Type is selected, it is displayed in the Variable Type field.



### **Variable Type Descriptions**

The following table lists the available variable types and a description for each one.

<b>Category</b>	<b>Variable Type</b>	<b>Description</b>
Numeric	Static Numeric Variable	A user-defined numeric value (e.g., 15)
	Program Task Status Variable	A numeric value that represents the current status of a specified "abstract program" object (for example, a C# script object).
	Object Property	A numeric value for any ACE object property (e.g., GripDwellTime)
	Controller Power and Calibration	Represents the controller POWER switch. When power is enabled, it checks the robot calibration status and calibrates the robot, if needed. This is equivalent to the power/calibrate button on the

<b>Category</b>	<b>Variable Type</b>	<b>Description</b>
		ACE toolbar.
	Controller Real Variable	A V+ real value (e.g., 12.635). For example, if you've declared myvar=1, then entering "myvar" in the V+ variable field returns the value "1".
	Controller Signal	A V+ input or output signal. This value must represent a valid, installed digital signal (e.g., 1003) or a soft signal.
	Controller Switch	A V+ system switch (e.g., POWER)
	Controller Parameter	A V+ system parameter (e.g., HAND.TIME).
	Local I/O Signal	A numeric variable that represents a specified local I/O signal. This provides a way to reference the I/O signal in a script.
	Process Manager Clear Tracking Source	A numeric value that represents a Process Manager tracking source. When a non-zero value is written to the variable, it will issue a clear command to the selected source.
	Process Manager Hardware Status Code	The status (error code) associated with a given Process Manager control source. This is the code associated with a source error in the Process Manager runtime control. Writing a value to this variable is interpreted as a response to the error, where: 1 = retry, 4 = skip, and 8 = abort.
	Process Manager Robot Statistics	A numeric value that represents a robot statistic (for example, PPM).
	Process Manager Summary Statistics	A numeric variable that maps to runtime statistics of the Process Manager, like the Parts per Minute rate or the Idle Time.
	Process Manager Process Status	A numeric value that represents a Process Manager process index, which can be used to enable/disable that process. Writing a 0 to this variable disables the process; a non-zero value enables the process.
	Process Manager Control	A numeric value that represents a Process Manager application, which can be used to start/stop that application. Writing a 0 to this variable stops the Process Manager; a non-zero

Category	Variable Type	Description
		value starts the Process Manager.
String	Static String Variable	A user-defined string value (e.g., "Process Running")
	Controller String Variable	A V+ string value (e.g., "Product"). For example, if you've declared \$myvar="Product", then entering "\$myvar" in the V+ variable field returns the value "Product".
	Process Manager Hardware Status Text	The string associated with the status (error code) for a Process Manager control source. This is the code associated with a source error in the Process Manager runtime control.

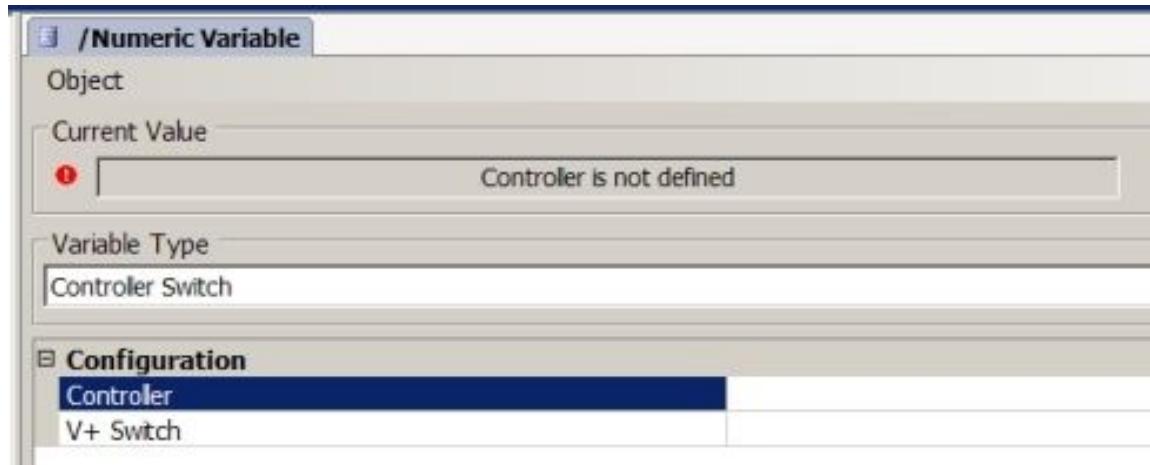
### Exporting Variable Data Through OPC

For details on exporting data through OPC, see OPC Data Access and Process Control on page 785.

### Configuration Error Notification

The Variable editor is designed to notify you when there is a configuration error. The alert icon ( ) is displayed next to the item containing the error.

For example, in the following editor, the Controller has not been selected, so the editor displays a "Controller is not defined" message in the Current Value field, as shown:



### Adding Variables to the Watch Variable List

ACE variables can be added to the Watch Variable list by dragging and dropping the variable from the Work-space Explorer. For more details on the Watch Variable list, see [Watch Variable Tool](#).

## Adept AnyFeeder Object

The ACE software contains a control panel for the AnyFeeder object, which allows you to set and test various Adept AnyFeeder parameters. For Adept AnyFeeder hardware installation details, see the [Adept AnyFeeder User's Guide](#).

**NOTE:** When emulation mode is enabled, all feed-operation durations are now emulated. However, this does not apply to: error reset, initialization, operation abort, or Firmware restart, as these operations are not supposed to be called during the feed cycle. For more details on emulation mode, see Emulation Mode on page 1089.

### ***Creating an AnyFeeder Object***

To create a new AnyFeeder object, right-click in the folder area of the Workspace Explorer, and select:

**New > Device > Feeders > AnyFeeder**

The Create New AnyFeeder interview wizard opens, which is used to create and position the AnyFeeder object in the workcell. Simply follow the interview wizard screens to complete the operation.

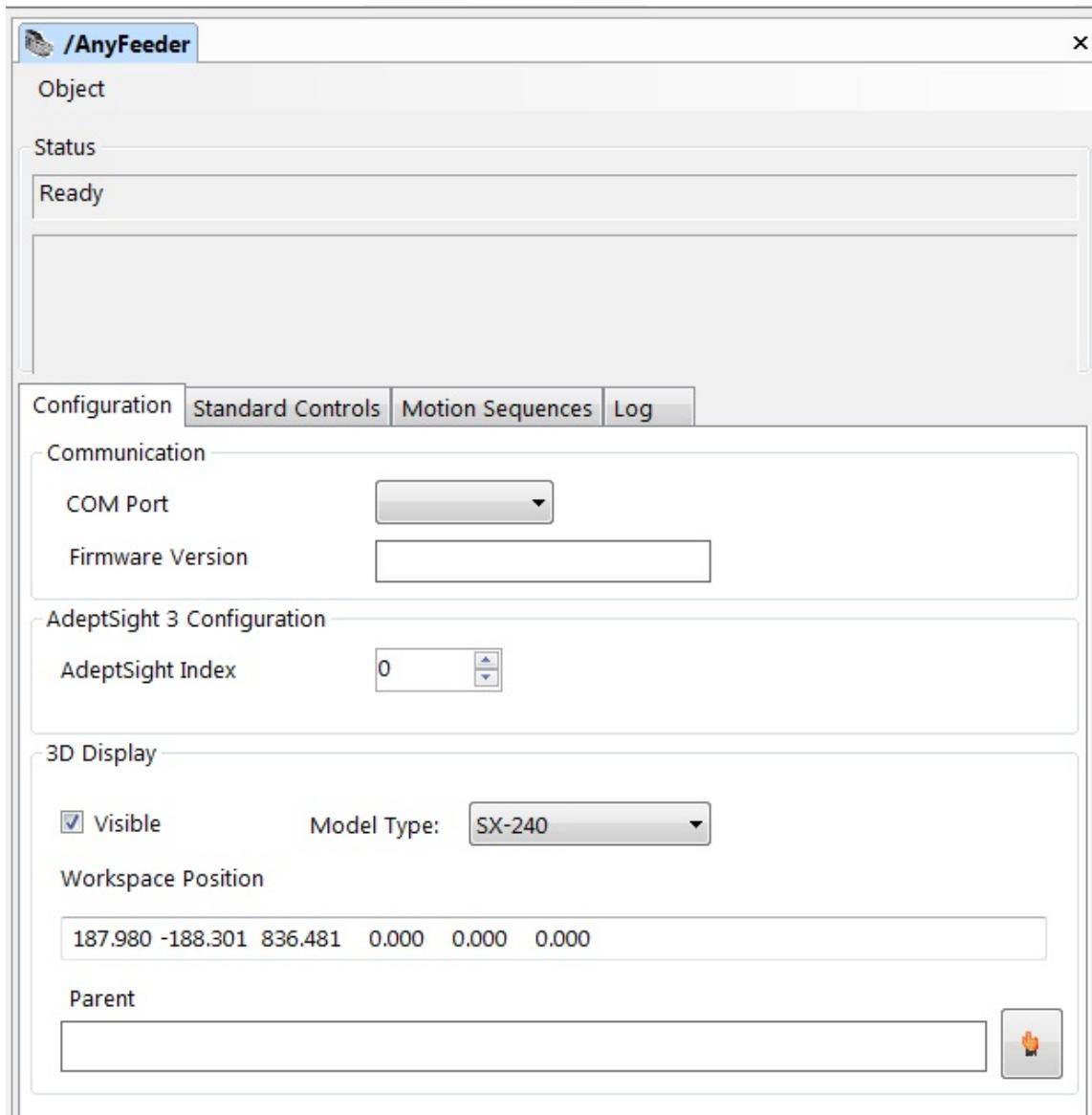
**NOTE:** This feature operates differently in emulation mode. For details on emulation mode wizards, see Emulation Mode Wizards on page 185.

Once the AnyFeeder object is created, use the AnyFeeder object editor to set the operating parameters for the AnyFeeder.

To open the AnyFeeder object editor, double-click the AnyFeeder object in the folder area of the Workspace Explorer.

## Adept AnyFeeder Object

---

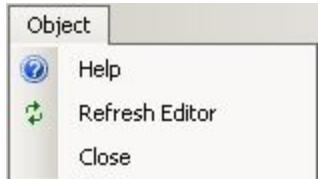


*Adept AnyFeeder Object Editor (Configuration Page)*

### **Menu Items**

This section describes the selections available from the System Startup editor menu.

## Object Menu



### Help

Displays the online help for the AnyFeeder object.

### Refresh Editor

Refreshes the contents of the AnyFeeder object.

### Close

Closes the AnyFeeder object.

## Status Group

The Status group provides operation and error information on the AnyFeeder object. The current status is shown in the top field; a description of the status is shown in the lower field.

## Configuration Page

The Configuration Controls page (shown in the previous figure) provides a set of Adept AnyFeeder configuration controls. The following table describes the function of each control.

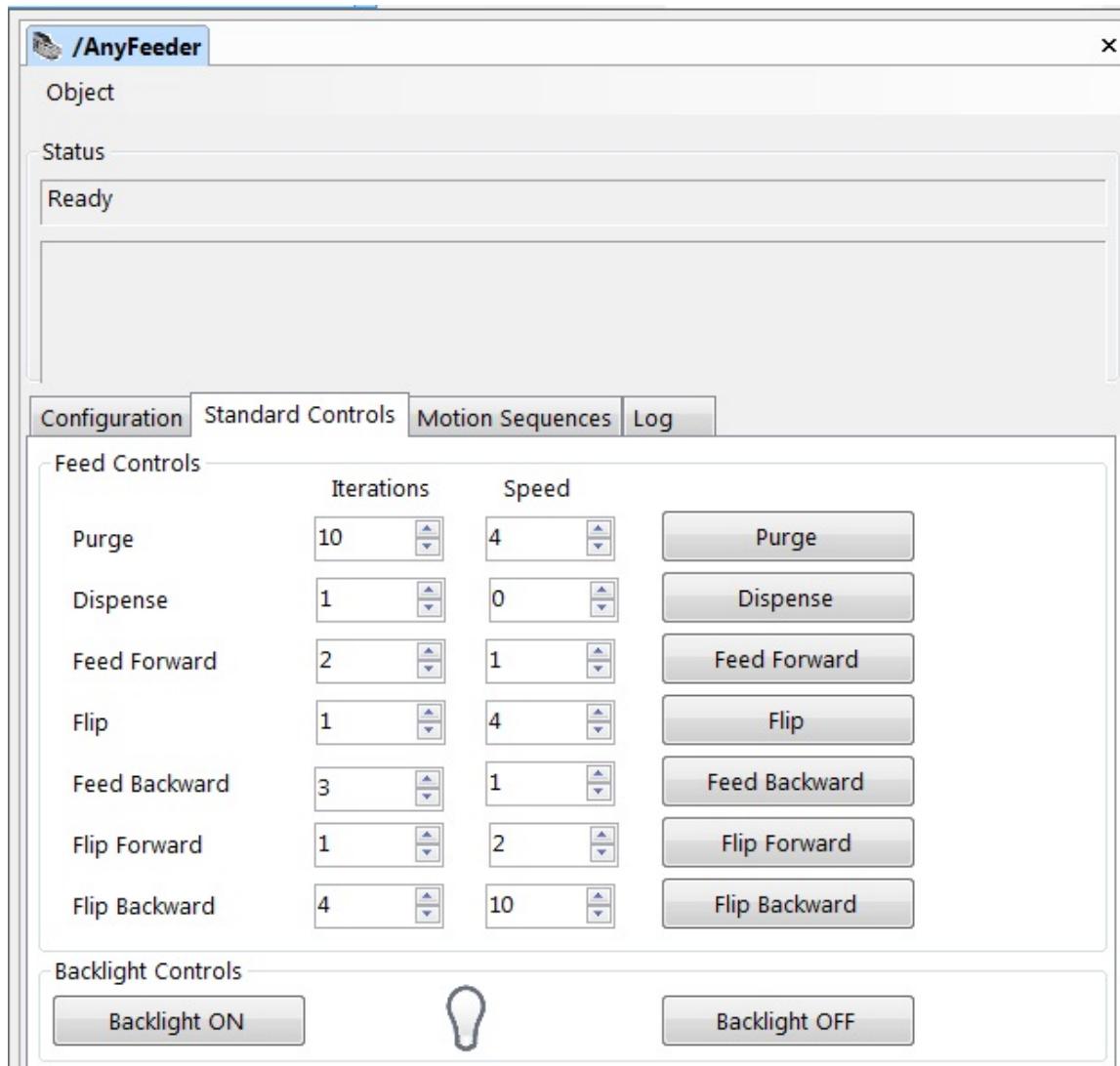
Item	Description
Communication	The Communication group is used to set the communication parameters for the AnyFeeder.
COM Port	Specifies the COM (serial) port that is used to send commands to the AnyFeeder.
Firmware Version	The firmware version as returned from the feeder
ACE Sight Configuration	This is an identifier used to allow driving the AnyFeeder directly from V+ using commands like VPARAMETER, VRUN, etc. For more details, see the <a href="#">Adept AnyFeeder User's Guide</a> .
ACE Sight Index	There is an index to identify an ACE Sight sequence, this is the corresponding identifier for the AnyFeeder. Specify a unique value for the AnyFeeder—it <i>must</i> be different than that used for the ACE Sight sequence.  <b>NOTE:</b> When an AnyFeeder and an ACE Sight sequence have the same index value, the PC will only try to run the ACE Sight sequence.
3D Display	Provides controls for showing the AnyFeeder object in the 3D Visualization window, and for adjusting the position of the AnyFeeder in the workcell.

Item	Description
	<p><b>NOTE:</b> You can visually reposition the AnyFeeder using the Workspace Referencing Wizard. For more details on the Workspace Referencing wizard, see Workspace Positioning on page 945.</p>

#### ***Standard Controls Page***

The standard controls page provides a set of commonly-used controls for adjusting the Adept AnyFeeder.

## Adept AnyFeeder Object



*Adept AnyFeeder Standard Controls Page*

The following table describes the function of each control.

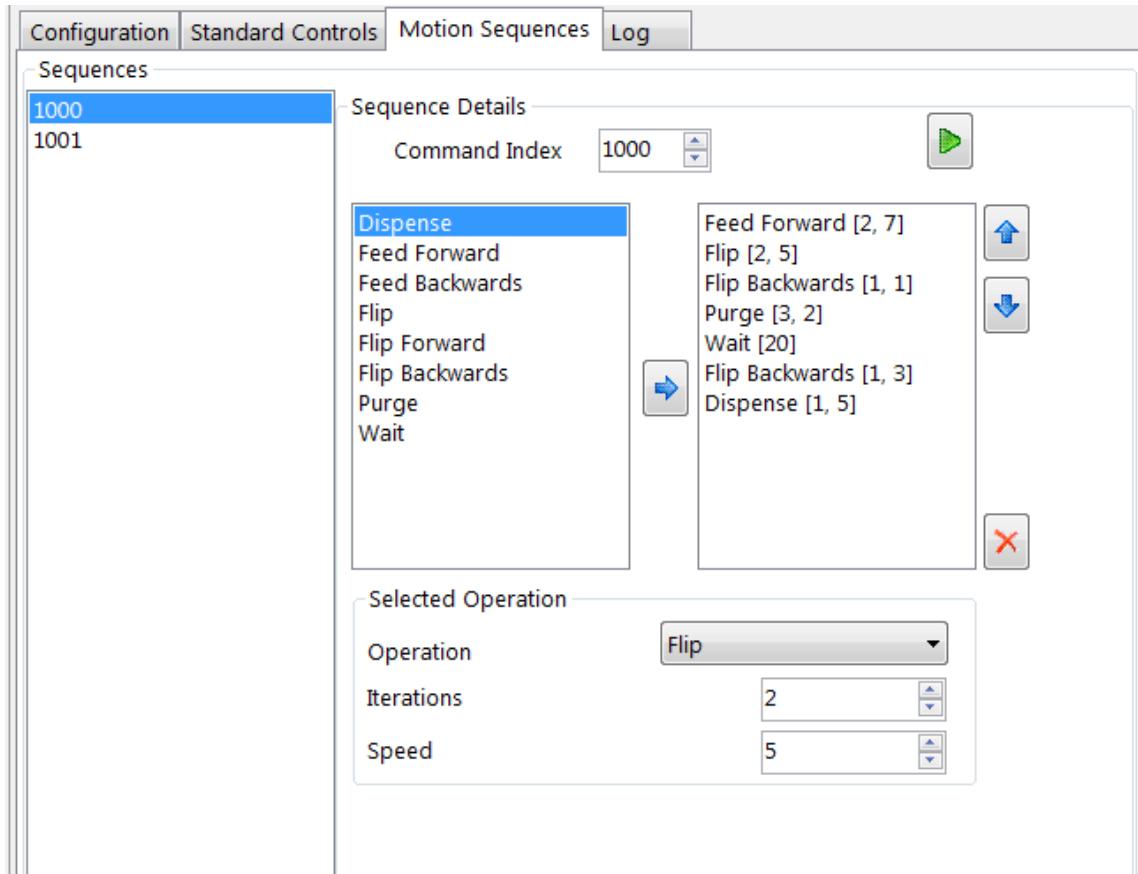
Item	Description
Purge	Feeds parts backwards and through the purge gate. <b>NOTE:</b> The purge gate must be manually opened before performing the operation.
Dispense	Dispenses parts from the part hopper onto the feed sur-

Item	Description
	face.
Feed Forward	Moves parts forward along the feed surface.
Flip	Flips parts that are on the feed surface.
Feed Backwards	Moves parts backwards along the feed surface.
Flip Forward	Flips parts forward on the feed surface.
Flip Backwards	Flips parts backwards on the feed surface.
BackLight Controls	The BackLight Controls group is used to set the state of the optional backlight.
BackLight ON (  <td>Activates (turns on) the backlight. The illuminated backlight icon () displays to show the ON state.</td>	Activates (turns on) the backlight. The illuminated backlight icon (  ) displays to show the ON state.
BackLight OFF (  <td>Deactivates (turns off) the backlight. The non-illuminated backlight icon () displays to show the OFF state.</td>	Deactivates (turns off) the backlight. The non-illuminated backlight icon (  ) displays to show the OFF state.

### ***Motion Sequences Page***

The Motion Sequences page shows a listing of high level motion sequences associated with the AnyFeeder. The user can define a sequence to be a collection of individual operations. When a sequence is selected and run, all the operations are performed in the order defined by the user.

## FlexiBowl Feeder Object



Motion sequences can be triggered through the AnyFeeder user interface, a C# script, or using the V+ ACE Sight protocol.

### **Log Page**

The log page shows a summary of the communications between the AnyFeeder and the PC.

### **FlexiBowl Feeder Object**

The ACE software contains a control panel for the FlexiBowl object, which allows you to set and test various FlexiBowl parameters. For hardware installation details or information about the functionality of the FlexiBowl feeder, see the [Adept FlexiBowl Feeder User's Guide](#).

**NOTE:** When emulation mode is enabled, all feed-operation durations are now emulated. For more details on emulation mode, see Emulation Mode on page 1089.

### **Creating a FlexiBowl Feeder Object**

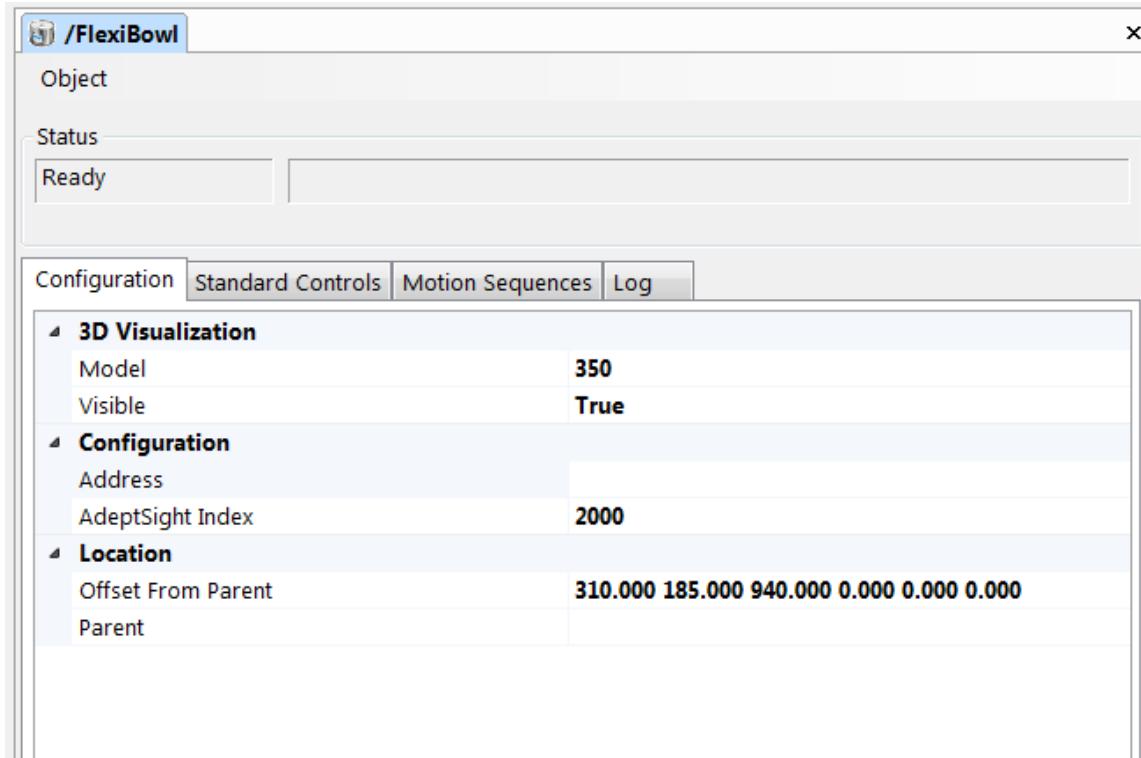
To create a new FlexiBowl feeder object, right-click in the folder area of the Workspace Explorer, and select:

**New > Device > Feeders > FlexiBowl**

The Create New FlexiBowl Feeder interview wizard opens, which is used to create and position the feeder object in the workcell. Simply follow the interview wizard screens to complete the operation.

Once the feeder is created, use the feeder editor to set the operating parameters for the feeder.

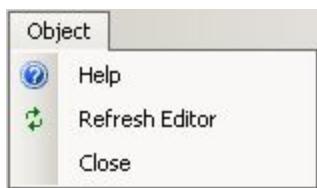
To open the feeder editor, double-click the FlexiBowl Feeder object in the folder area of the Workspace Explorer.



Adept FlexiBowl Object Editor (Configuration Page)

**Menu Items**

This section describes the selections available from the System Startup editor menu.

**Object Menu****Help**

Displays the online help for the AnyFeeder object.

**Refresh Editor**

Refreshes the contents of the object.

**Close**

Closes the AnyFeeder object.

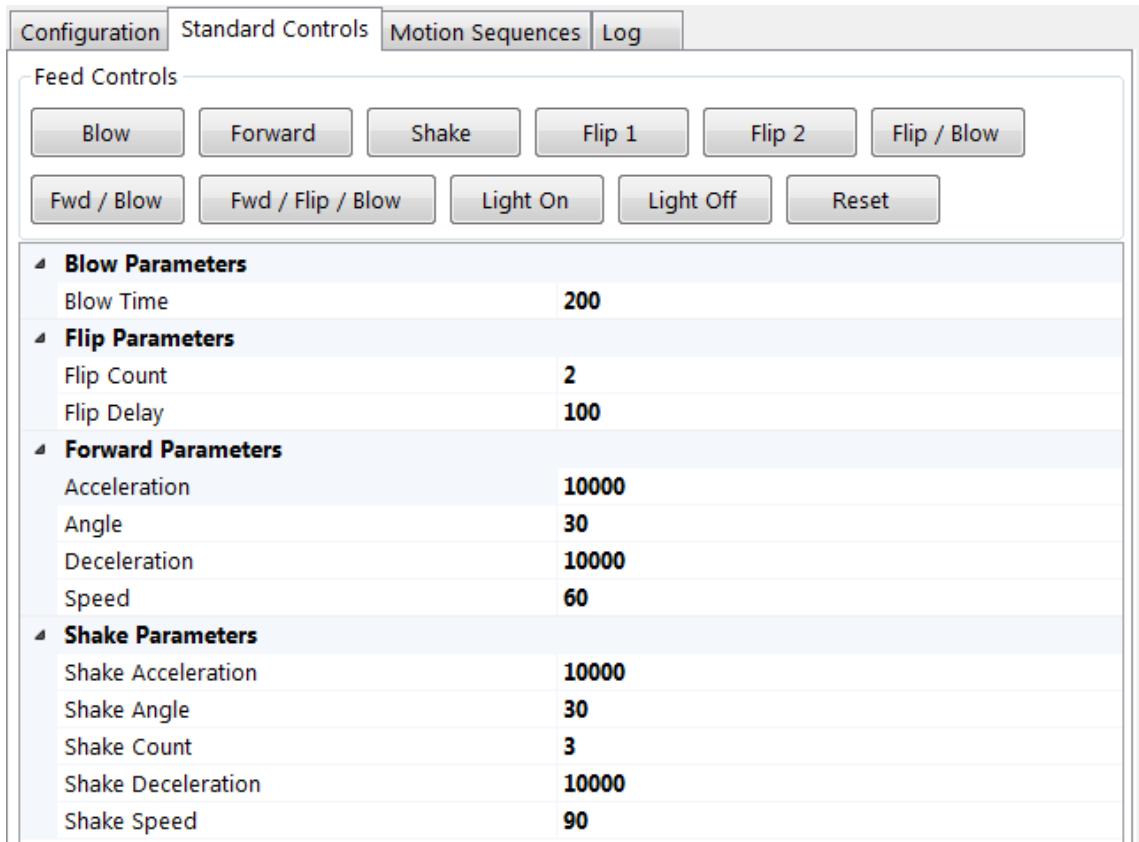
### **Configuration Page**

The Configuration Controls page (shown in the previous figure) provides a set of configuration controls. The following table describes the function of each control.

Item	Description
3D Visualization: Model and Visible	Identifies if the feeder is displayed in the 3D virtual display and the model number of the feeder to render.
Configuration: Address	The IP Address of the FlexiBowl feeder
Configuration: ACE Sight Index	There is an index to identify an ACE Sight sequence, this is the corresponding identifier for the feeder. Specify a unique value for the feeder—it <i>must</i> be different than that used for the ACE Sight sequence,
Location: Offset from Parent and Parent	The position of the FlexiBowl feeder relative to a parent in the workspace.

### **Standard Controls Page**

The standard controls page provides a set of commonly-used controls for adjusting the feeder.



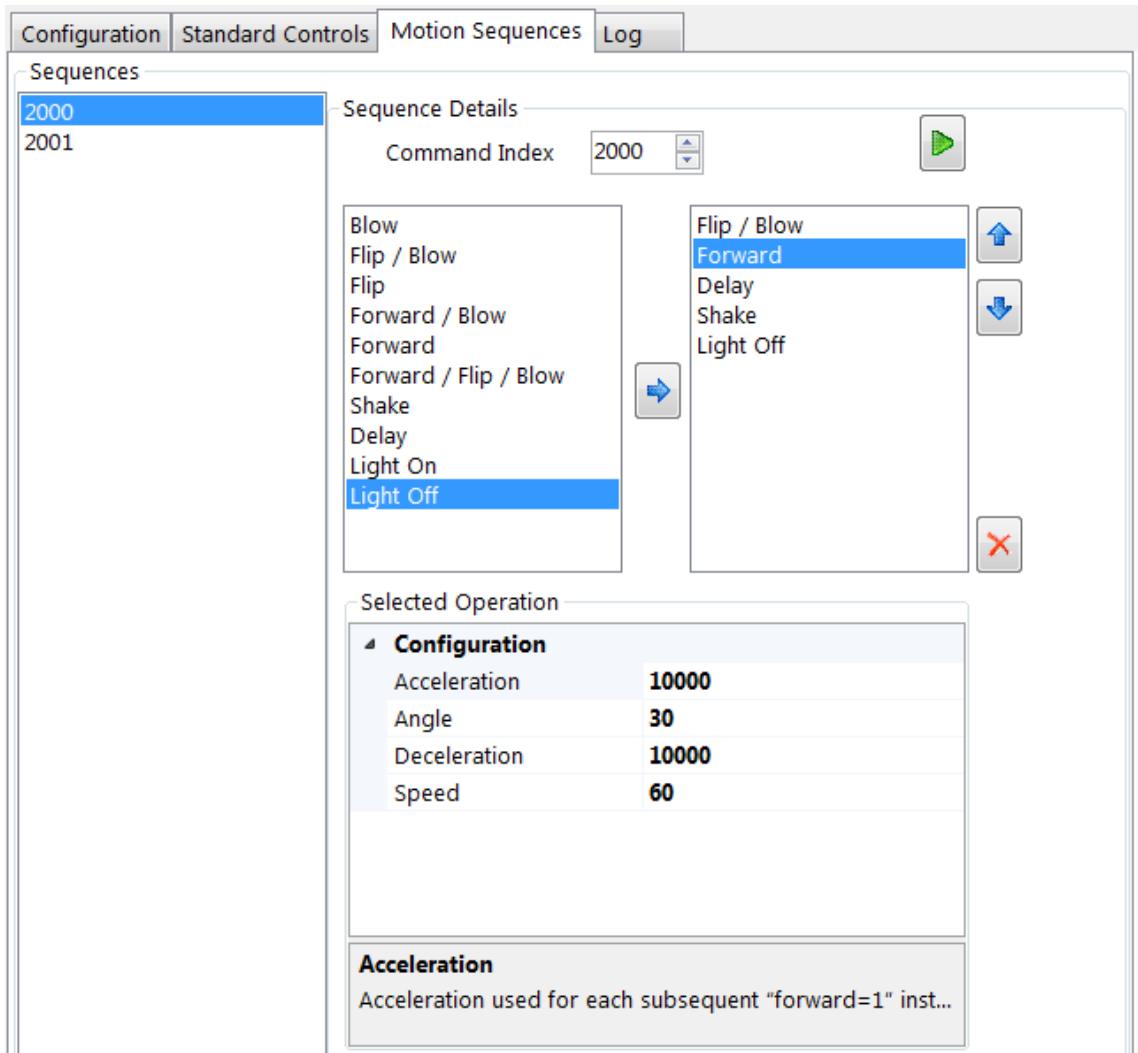
*Adept FlexiBowl Standard Controls Page*

### ***Motion Sequences Page***

The Motion Sequences page shows a listing of high level motion sequences associated with the feeder. The user can define a sequence to be a collection of individual operations. When a sequence is selected and run, all the operations are performed in the order defined by the user.

## IO Feeder Object

---



Motion sequences can be triggered through the FlexiBowl Feeder user interface, a C# script, or using the V+ ACE Sight protocol.

### ***Log Page***

The log page shows a summary of the communications between the feeder and the PC.

### **IO Feeder Object**

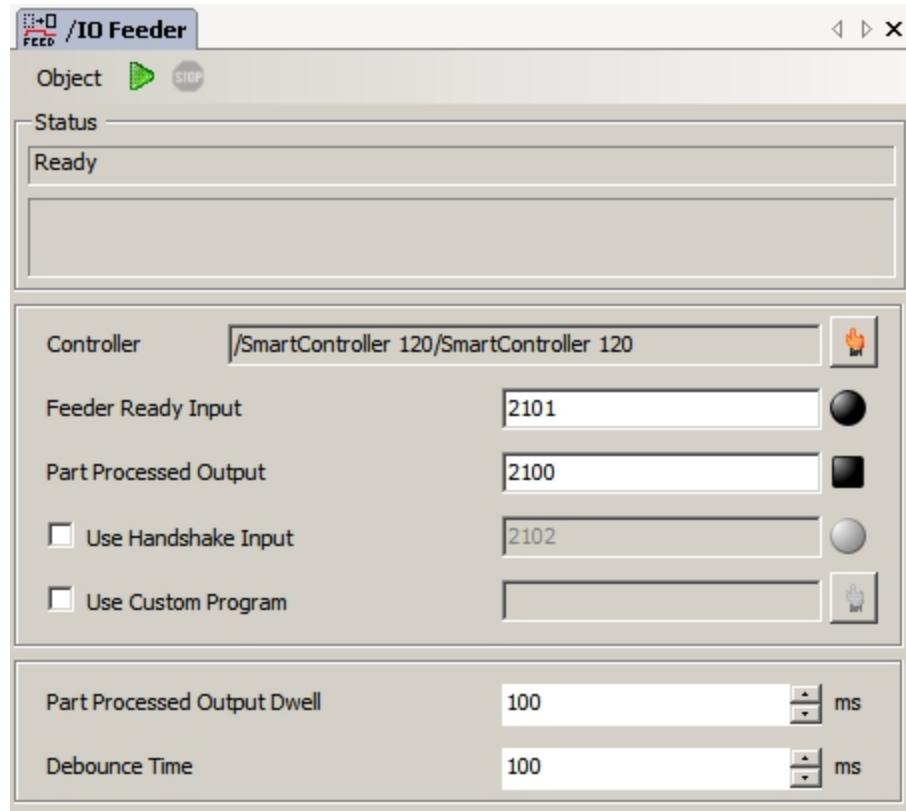
The ACE software contains a control panel for an IO Feeder object, which allows you to set and test various parameters for a generic IO-controlled feeder. For Adept AnyFeeder control, see [Adept AnyFeeder Object](#) on page 138.

### ***Creating an IO Feeder Object***

To create a new IO Feeder object, right-click in the folder area of the Workspace Explorer, and select:

**New > Device > Feeders > IO Feeder**

To open the IO Feeder control, double-click the IO Feeder object in the folder area of the Workspace Explorer.

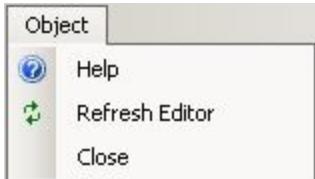


*IO Feeder Object Editor*

### ***Menu Items***

This section describes the selections available from the System Startup editor menu.

## Object Menu



### Help

Displays the online help for the IO Feeder object.

### Refresh Editor

Refreshes the contents of the IO Feeder object.

### Close

Closes the IO Feeder object.

## Icons

The following icons are available for controlling the feeder.

Icon	Description
	<p>Start - Used to perform one test cycle of the feeder. The operation stops when the cycle has completed or if the Stop icon is clicked before the end of the cycle. To repeat or restart the cycle, click Start again.</p> <p><b>NOTE:</b> When clicked, this icon dims until the Stop icon is clicked (feeder test cycle has been interrupted) or the cycle has completed .</p>
	<p>Stop - Stops (interrupts) the test cycle. The test cycle can be restarted by clicking the Start icon.</p> <p><b>NOTE:</b> This icon is dimmed until the Start icon is clicked (feeder test cycle has started).</p>

## Status Group

The Status group provides operation and error information on the IO Feeder object. The current status is shown in the top field; a description of the status is shown in the lower field.

## Controller and IO Group

The Configuration Controls page (shown in the previous figure) provides a set of Adept IO Feeder configuration controls. The following table describes the function of each control.

Item	Description
Controller	Specifies the controller that will process the feeder signals. Click the Browse icon () to display the list of available controllers, and then select the desired controller from that list.

Item	Description
Feeder Ready Input	<p>Specifies the input signal that indicates the feeder is ready and available to present a part instance.</p> <p><b>NOTE:</b> When emulation mode is enabled, these signals are ignored. For more details on emulation mode, see Emulation Mode on page 1089.</p>
Part Processed Output	<p>Specifies the output signal that indicates the instance has been processed (acquired) by the robot. The feeder should cycle and present a new part instance.</p> <p><b>NOTE:</b> When emulation mode is enabled, these signals are ignored. For more details on emulation mode, see Emulation Mode on page 1089.</p>
Use Handshake Input	<p>If enabled, the feeder will assert a signal indicating it has acknowledged the part processed signal.</p> <p><b>NOTE:</b> When emulation mode is enabled, these signals are ignored. For more details on emulation mode, see Emulation Mode on page 1089.</p>
Use Custom Program	<p>The feeder interface code runs as a V+ program on the specified controller. This program can be overridden, if some custom logic needs to be applied. For details, see IO Feeder Object on page 148.</p> <p>If you enable this option, you must enter a valid program. Otherwise, an error message is displayed.</p>

### Part Handling Group

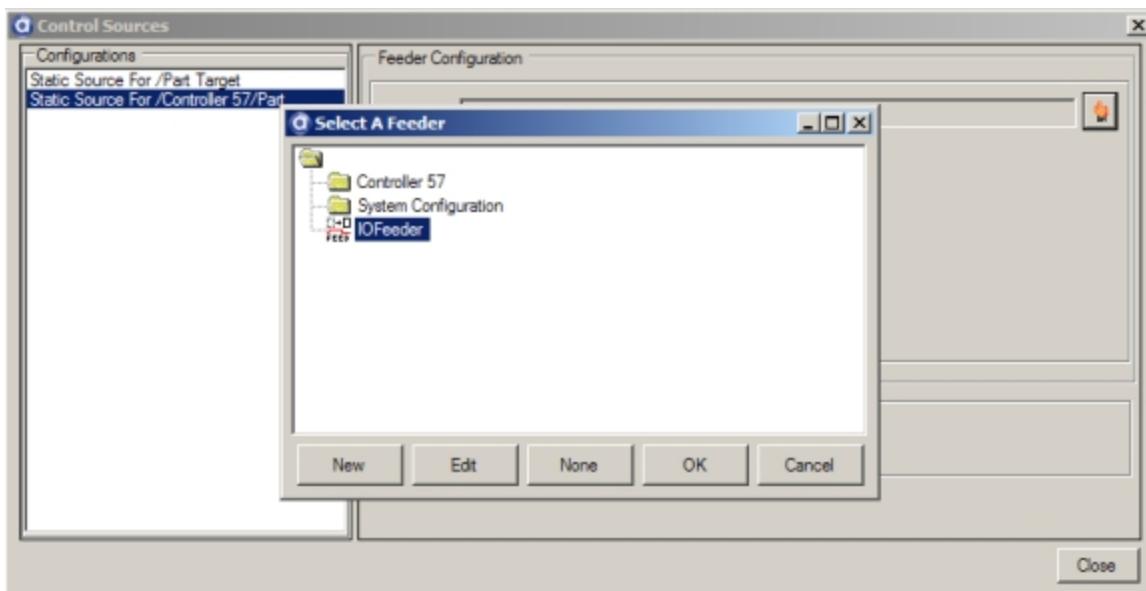
The following table describes the function of each control.

Item	Description
Part Processed Output Dwell	Specifies the dwell time (time to wait) in milliseconds after the Part Processed output signal is turned on before turning it off.
Debounce Time	Specifies the amount of time (in milliseconds) that a signal must be seen in the "on" (electrically on) state, before it is considered "on" (logically on).

### ***Using the IO Feeder with the Process Manager***

The IO Feeder acts as a part feeder for your ACE application. The Process Manager allows you to include the IO Feeder in your process through the Control Sources dialog. After you have installed an IO Feeder object in your workspace and created a process with the Process Manager, you can assign the IO Feeder to a Part or Part Target. To do this:

1. Click the Control Sources on the Process Manager.
2. Select the Part or Part Target that you want to assign to the IO Feeder.
3. Click the Browse () icon in the Feeder Configuration group to select the IO Feeder, as shown in the following figure.



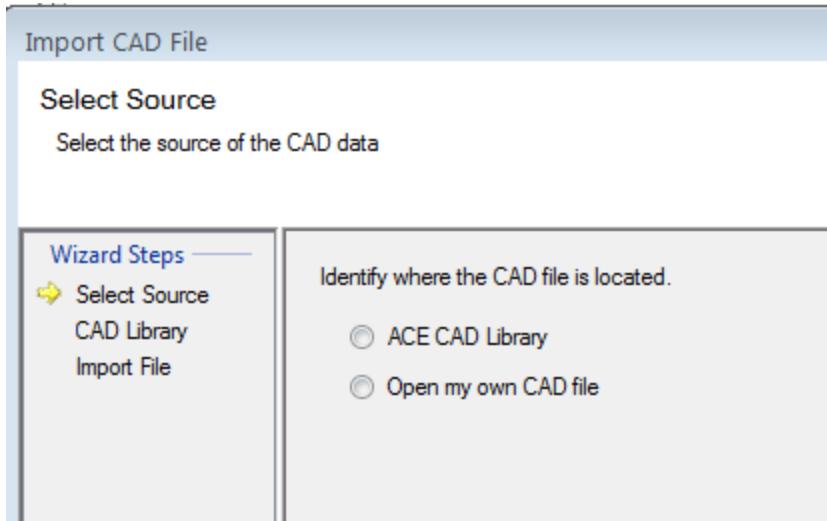
*Control Sources - Assigning the IO Feeder to a Part*

For more details on using Process Manager Control Sources, see Control Sources on page 869.

## **CAD Data Object**

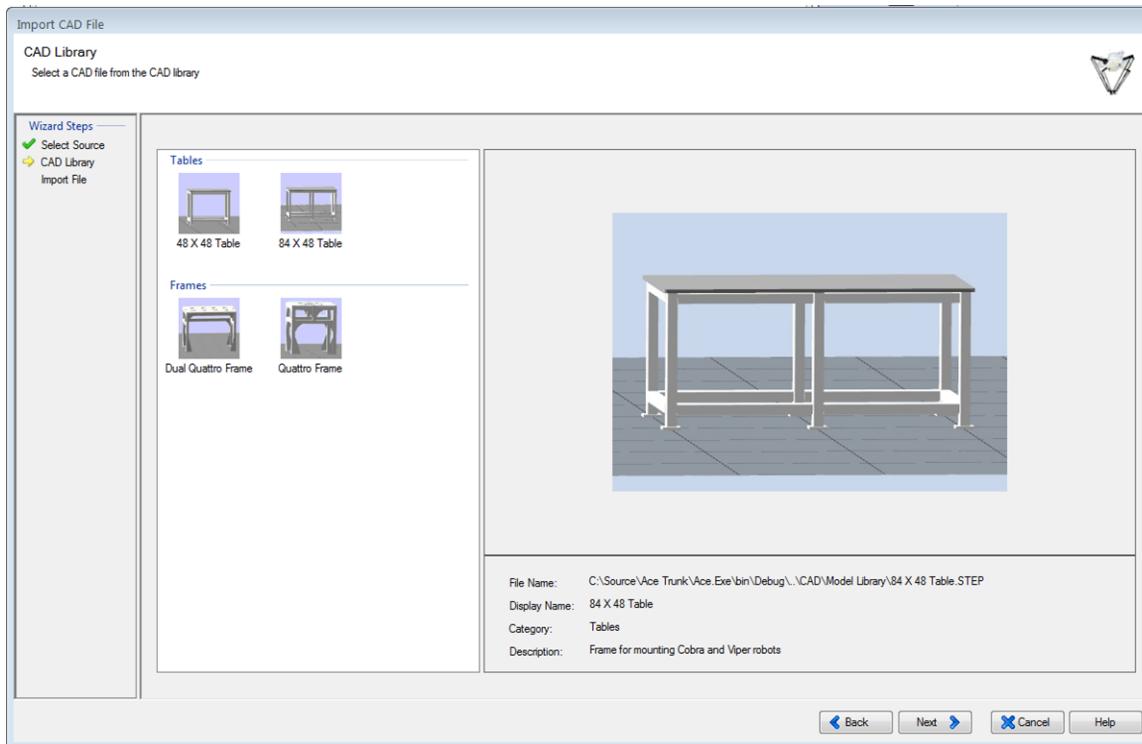
The ACE software contains a CAD data object that represents a CAD file imported into the 3D virtual display. When a CAD Data object is created, an import wizard is created that guides you on the process of importing the file.

---

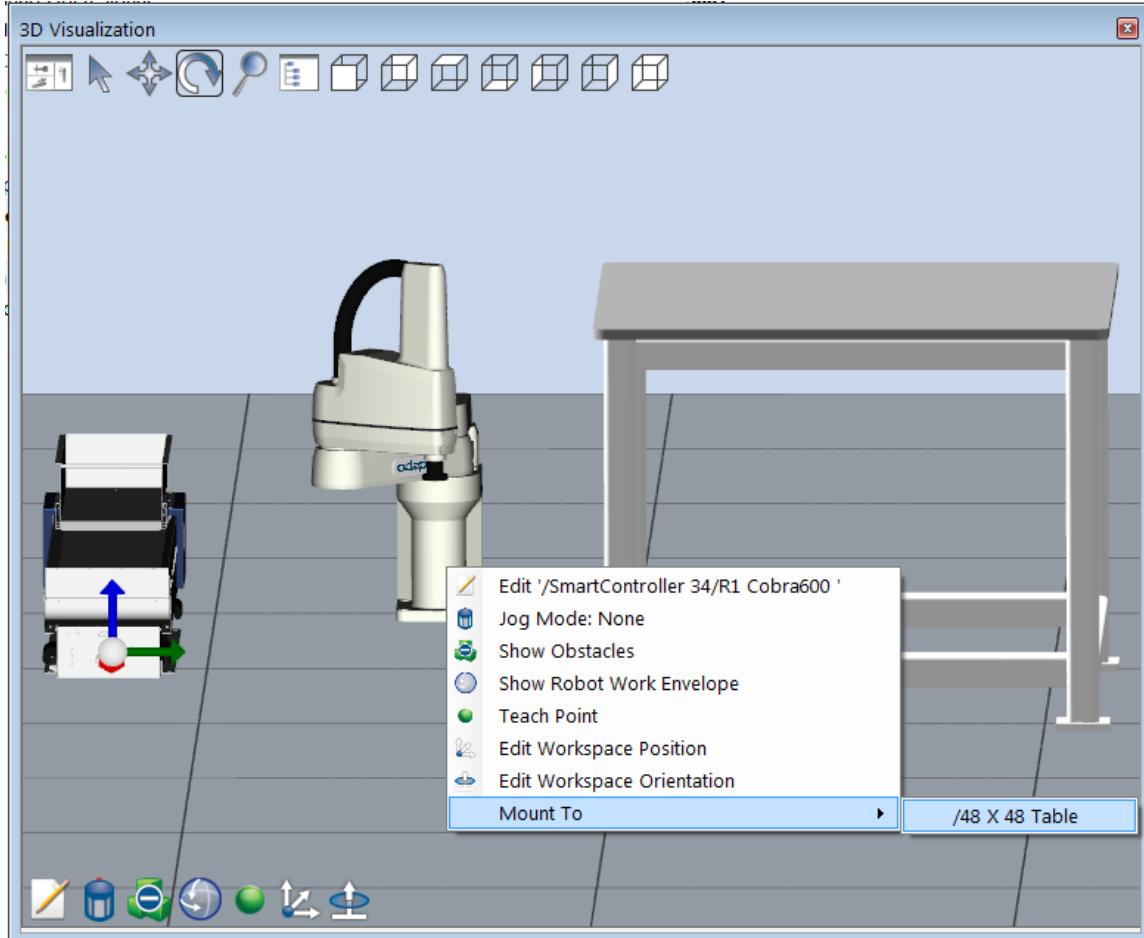


### **ACE CAD Library**

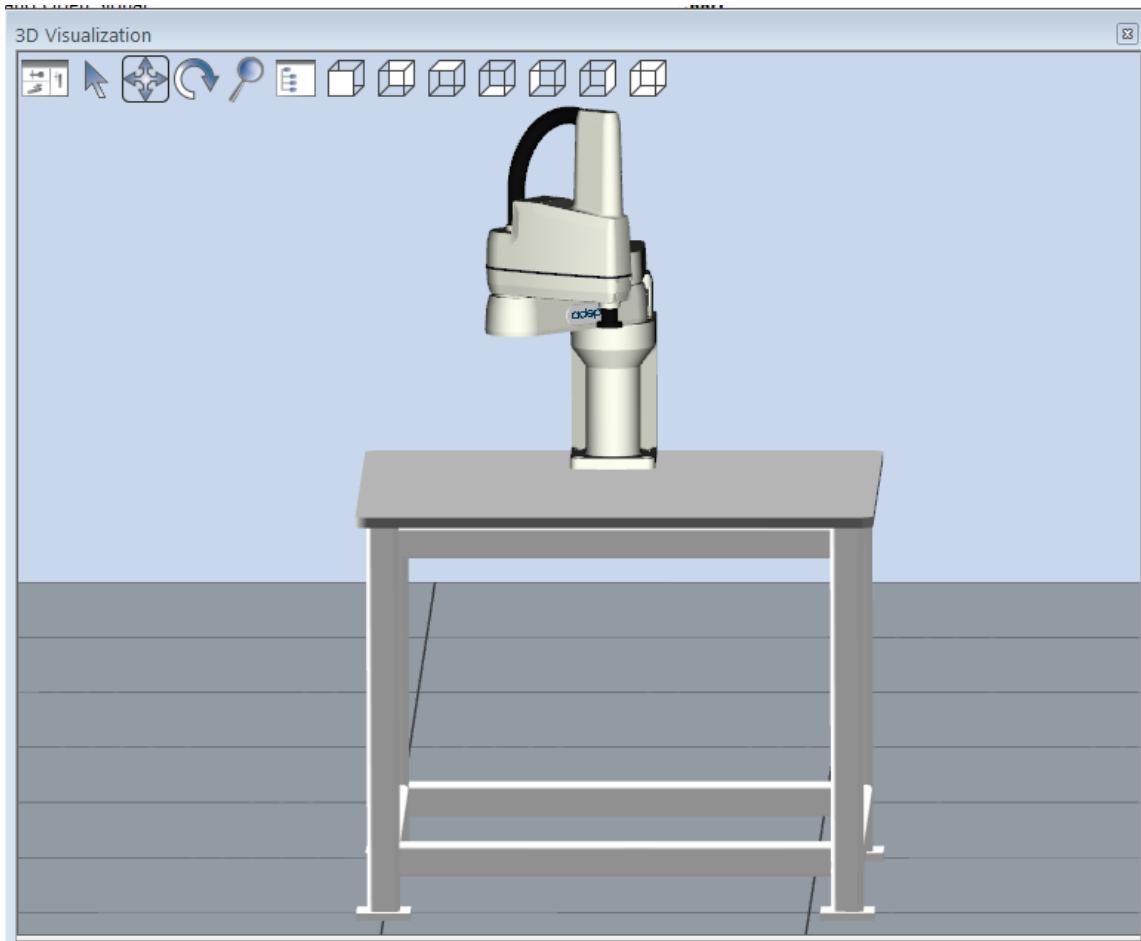
When this option is selected, the user is presented with a list of standard CAD files which ship with ACE. The user simply picks a picture of what they want to import and it is automatically added to the workspace.



After a standard ACE CAD file is imported into the workspace, you can select objects in the workspace and right-click for a context menu to associate objects with the imported CAD file. For example, if you import a base, you can right-click on a robot or feeder in the workspace and mount the robot to the base:



Doing so will move the robot and associate the table as the parent of the robot in the 3D display:



Looking at the robot editor will show that the Parent of the robot is now the CAD data table object and the position of the robot relative to the table is specified in the **Offset From Parent** property.

Selected End-Effector	/SmartController 34/Gripper R1 Cobra600
Location	
Offset From Parent	180.000 525.000 940.000 0.000 0.000 0.000
Parent	/48 X 48 Table

#### ***Open my own CAD file***

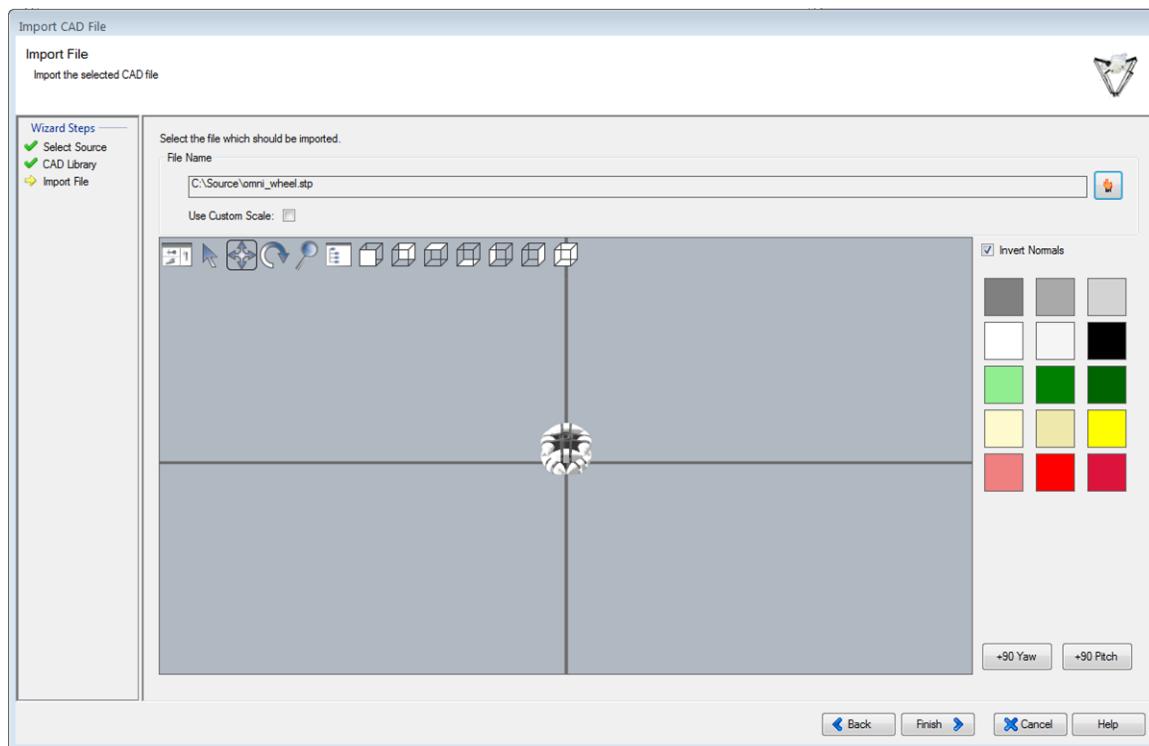
When this option is selected, the user must select a CAD file which they want to import into ACE. Currently, the supported file formats are STEP, IGES, and STL.

Once the user identifies a file name, the file is opened and imported into ACE.

---

## CAD Data Object

---



## Toolbar

The ACE toolbars provide quick access to frequently-used items (see the following figures). The toolbars will be shown (or hidden) based on the licenses available on the USB hardware key.



Toolbar - General



Toolbar - Connections



Toolbar - Task Status Control



Toolbars Menu

The toolbar items are described below.

**NOTE:** If a toolbar item is not available, its icon will be dimmed (shown as a gray image).

### Toolbar - General



Item	Description
	Saves the current workspace in the default file (the last file loaded or selected in the "Save As" dialog box). See the topic Save Workspace on page 161 for details.
	Opens the Find window. For details, see Find Dialog on page 782.

## Toolbar - Connections

---

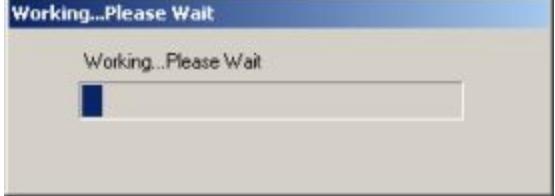
Item	Description
	Opens the Workspace Explorer window. See the topic <a href="#">Workspace Explorer</a> on page 75 for details.
	Opens the System Monitor tool. See the topic <a href="#">System Monitor</a> on page 1117 for details.
(3D available) (3D not available)	Opens the 3D Visualization window. See the topic <a href="#">3D Visualization</a> on page 176 for details.  <b>NOTE:</b> This feature requires hardware that supports DirectX 9.0c (or later) and 3D-graphics processing. Otherwise, the feature will be disabled.
	Opens the Task Status Control. See the topic <a href="#">Task Status Control</a> on page 747 for details.
	Opens the Watch Variable Tool. See the topic <a href="#">Watch Variable Tool</a> on page 794 for details.
	Opens the ACE Sight Vision window. See the <i>ACE Sight User's Guide</i> for details.

## Toolbar - Connections



Item	Description
	Opens the Robot Jog Control. See the topic <a href="#">Robot Jog Control</a> on page 163 for details.
Controller 22	Displays the currently-selected controller. A list of available controllers can be viewed by clicking the down arrow ().
(not connected) (connected) All	Connects (initiates communication) to the selected controller. When connected, click to disconnect (stop communication) from the selected controller. A progress indicator displays the status of the operation:

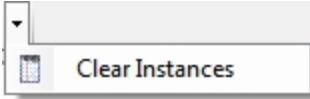
Item	Description
	<p><b>Connecting to '/Controller 22'</b></p>  <p><i>Connecting Progress Indicator</i></p> <p><b>Disconnecting from '/Controller 22'</b></p>  <p><i>Disconnecting Progress Indicator</i></p> <p>You can connect/disconnect from ALL controllers in the system by selecting the "All" option.</p>
	 <p><b>DANGER:</b> The robot may move during the enable power/calibration procedure. Make certain that the work cell is clear of all personnel and obstacles so that the robot does not cause death/injury or crash when moving.</p> <p>Enables high power to the controller/robot and indicates the calibration state of the robot.</p> <p><b>NOTE:</b> For most systems, after the icon is clicked, you are required to press and release the High Power button (a physical button located on the external front panel) within 10 seconds. If this is not done in the required time, High Power is <i>not</i> enabled and an error message is displayed.</p> <p>See your Adept robot user's guide for more details on enabling high power.</p> <p>A progress indicator displays the status of the operation:</p>

Item	Description
	 <p><i>Enable/Disable Power Progress Indicator</i></p> <p>After enabling high power, the system will automatically calibrate the robot, if needed.</p> <p>When enabled, click to disable high power to the controller/robot (icon turns red).</p> <p>You can enable/disable high power to ALL controllers/robots in the system by selecting the "All" option.</p>
<input type="text" value="50"/> ▾	Displays the current monitor speed. To change the monitor speed, select it from the drop-down list, or type the desired value into the field, and press Enter.
	Opens the Monitor windows. See the topic Monitor Window on page 777 for details.
	Opens the Digital I/O control. See the topic Digital I/O Window on page 166 for details.
	Opens the File Explorer, which is used to view files/folders on the controller. See the topic File Explorer for details.
	Opens the profiler. See See "Profiler" for details.

## Toolbar - Task Status Control



**NOTE:** This toolbar provides quick-access to specific Task Status controls. For additional controls, status messages, and other task control features, use the Task Status Control. For details, see Task Status Control on page 747.

Item	Description
 JC# Program	Displays the currently-selected task. A list of available processes can be viewed by clicking the down arrow (▼). See the topic Process Manager Control on page 920 for details.
	Starts (executes) the selected task.
	Enabled during execution, pauses or exceptions, this button causes the execution to stop for the selected task.
	Accesses the Clear Instances selection, which is used to clear all Part and Part Target instances from the Process Manager.

## Save Workspace

The Save icon (■) on the ACE toolbar displays the Save As dialog box, which provides an interface for saving the current workspace.

When the workspace is saved, the V+ configuration and hardware configuration are saved with the workspace.

**NOTE:** The Save As dialog box is displayed when there is no name assigned to the workspace file. If the file has been saved previously, clicking the Save icon simply saves the workspace using the current file-name. The previous version of the file is backed-up with "\_1" in front of the .awp extension (e.g., myfile\_1.awp). Each time the file is saved, the backup files are renamed (the number increments by 1). Only the last four backup files are retained.

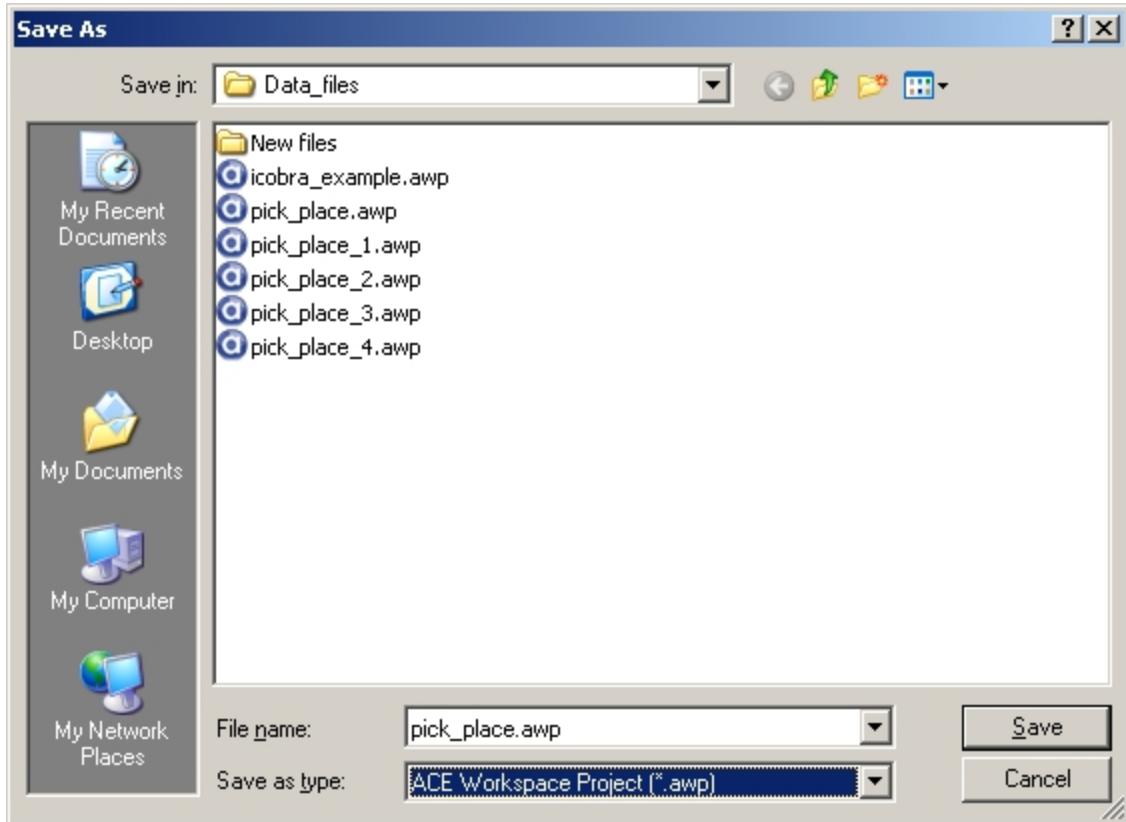
You can also access the Save As dialog box using:

- the **File > Save** menu item (if the current workspace has not been saved previously)
- the **File > Save As** menu item

The Save As dialog box opens.

## Save Workspace

---



*Save As Dialog Box*

The default folder and list of previously-saved files are displayed in the Save As dialog box. The dialog box items provide the following functions:

Item	Description
Save in	Selects a folder where the file will be stored (saved).
File name	Used to specify the name for the file. If the file extension is not specified, it will be automatically added when the file is saved.
Save as type	Used to specify the file format. The default format is .AWP file type.
Save	Saves the file to the folder and filename specified and then closes the dialog box.
Cancel	Closes the dialog box without saving the file.

For details on loading and unloading the workspace, see [Workspace Loading and Unloading on page 52](#).

### Saving a Workspace

To save an ACE workspace using the current file name:

1. From the ACE menu bar, select **File > Save**.

--OR--

Click the Save (  ) icon on the ACE toolbar.

2. The ACE workspace is saved with the current file name.

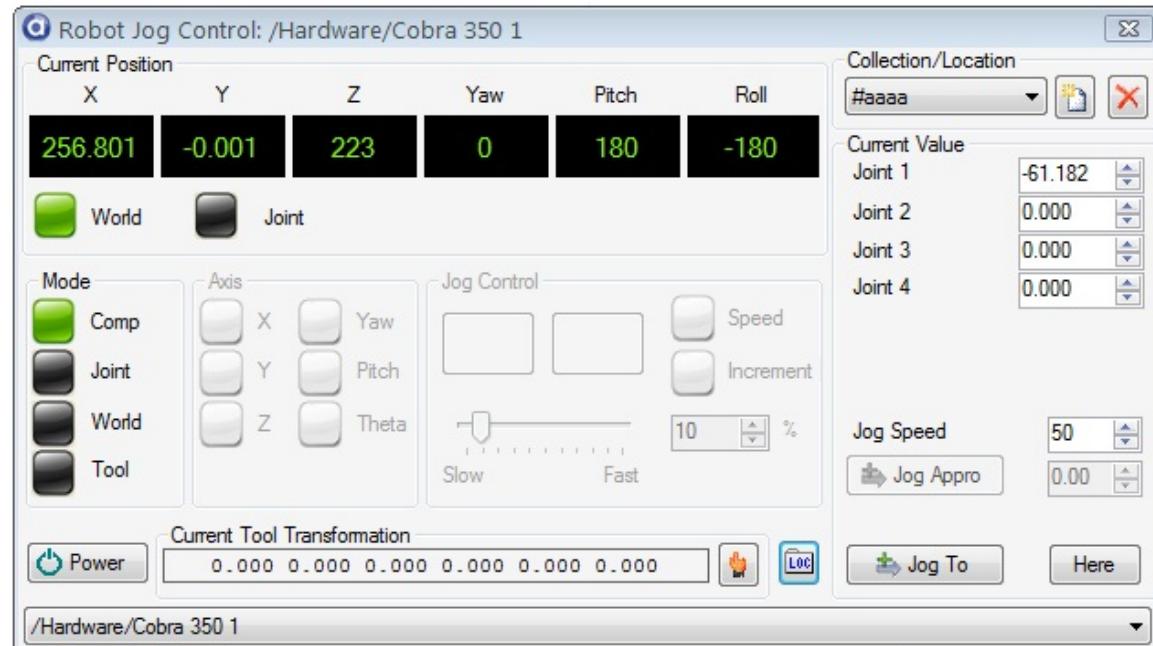
To save an ACE workspace using a new file name:

1. From the ACE menu bar, select **File > Save As**. The Save As dialog box opens.
2. Enter a file name and click **Save**. The ACE workspace is saved with the specified file name.

### Robot Jog Control

The Robot Jog Control provides an interface for calibrating, positioning, and monitoring the position of the selected robot. It is also used when teaching robot locations.

To access the Robot Jog Control, on the ACE toolbar, click the Robot (  ) icon. The Robot Jog Control opens (see the following figure).



*Robot Jog Control*

### Description of Robot Jog Control Items

The following sections describe the items on the Robot Jog Control.

#### Current Position Group

The Current Position group displays the current position of the robot, as follows:

Item	Description
World	Displays the current position in the World coordinate system.
Joint	Displays the current position in Joint coordinates.
Power	Toggles the power On/Off and calibrates the robot (when power is turned On). This functionality is also provided by the Power (  )icon on the ACE toolbar.
Align	(Six-axis robots only) Aligns the robot tool Z-axis with the nearest World axis.

#### Mode Group

The Mode group is used to select the control mode:

Item	Description
Comp	Also called Computer mode. In this mode, the Jog Control commands and settings are disabled. When Comp mode is enabled, the robot can be controlled by an executing program or the system terminal; the operator cannot move the robot.
Joint	Enables the Jog Control to move the robot about the axis of the joint selected in the Axis group.
World	Enables the Jog Control to move the robot in the selected direction: X, Y, or Z axes, of the World frame of reference, or rotated around these axes: RX, RY, or RZ, in the World coordinate system.
Tool	Enables the Jog Control to move the robot in the selected direction X, Y, or Z axes of the Tool coordinate system, or rotated around these axes: RX, RY, or RZ, in the Tool coordinate system.

**NOTE:** If the Robot Jog Control is open when Move is clicked while teaching a process, the Robot Jog Control temporarily switches to Comp (computer control) mode while the move is executed. When the move has completed, the Robot Jog Control switches back to its previous mode.

#### Axis Group

The Axis group is used to select the desired axis/joint based on the selection in the Mode group.

## Jog Control Group

The Jog Control group is used to manually position the robot:

Item	Description
-/+ Buttons	Activates movement of the selected joint/axis in a positive (+) or negative (-) direction. The movement is performed based on the Increment/Speed setting.
Increment	Sets the robot to move in step mode; the steps are specified millimeters. The step range is from 0.1 mm to 10.0 mm. The value is specified using the slide control or entering the value directly in the "mm" combo box.
Speed	Sets the speed at which the robot moves as a percentage of the maximum robot speed. The speed range is from 1% to 10%. The value is specified using the slide control or entering the value directly in the "%" combo box.

## Other Items

The remaining items have the following functions:

Item	Description
Power	Toggles the power On/Off and calibrates the robot (when power is turned On). This functionality is also provided by the Power (  ) icon on the ACE toolbar.
Align	(Six-axis robots only) Aligns the robot tool Z-axis with the nearest World axis.
Current Tool Transformation	Displays the current tool transformation for the selected gripper. The gripper is selected by clicking the browse (  ) icon.
Robot	When there is more than one robot in the ACE workspace, selects the robot that will be controlled. As robots are added to the workspace, this list is automatically updated.
Loc	Displays or hides the location listing on the right side of the display.
Cancel	Cancels the operation closes the Robot Jog Control.

## Locations Display

Locations are displayed on the right side of the jog control panel. The available items are:

Item	Description
Collection/Location	A listing of all V+ Locations and Precision Point variables associated with the robot selected in the jog pendant. Users can also add or remove locations and precision points.

Item	Description
Current Value	Shows the current values of the location transformation or precision point variable.
Jog Speed	The speed applied when jogging to the location
Jog Appro	When pressed, the robot will jog to the position at an approach height defined by the user.
Jog To	When pressed, the robot will jog to the position
Here	Records the current location of the robot into the selected variable.

## Digital I/O Window

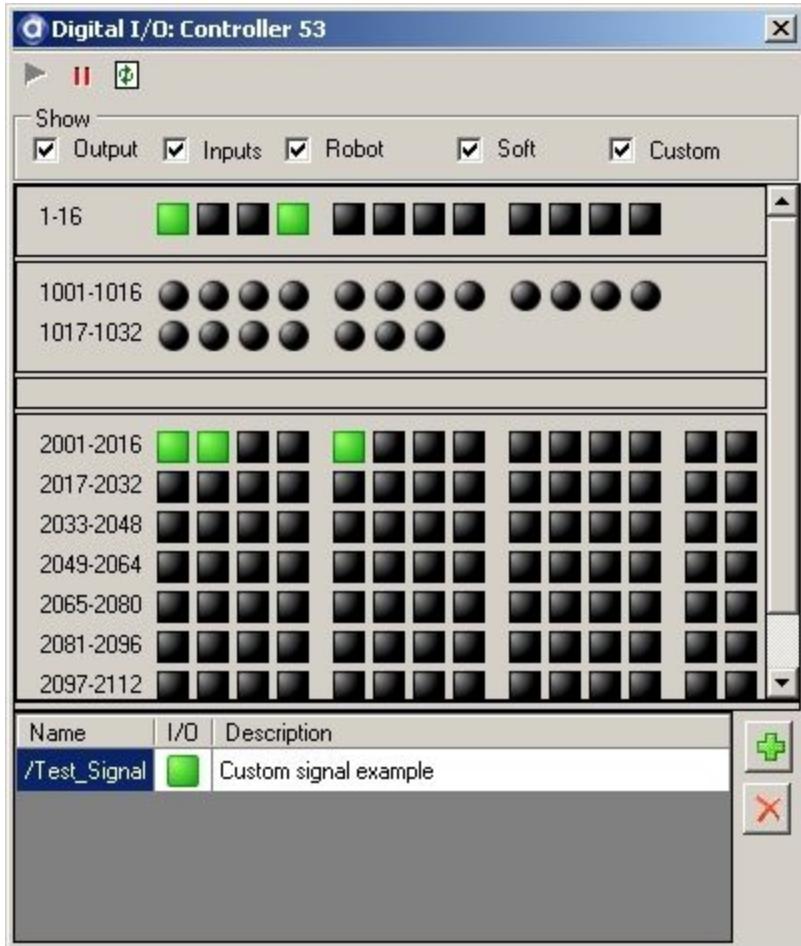
The Digital I/O window provides an interface for monitoring the state of digital I/O signals (inputs, outputs, soft signals, and robot signals) on the connected controller. Additionally, you can use this window to manually toggle (turn on/off) digital output signals and soft signals.

**NOTE:** When emulation mode is enabled, you can also toggle digital input signals.

To access the Digital I/O window, on the ACE toolbar, click the Digital I/O ( ) icon. The Digital I/O window opens.

**NOTE:** Many of the ACE wizards contain an I/O button, which is used to open the Digital I/O window. For details, see Wizards on page 181.

## Digital I/O Window



## Digital I/O Window

The Show check boxes are used to select the digital I/O signal types to display, as follows:

Item	Description
Outputs	Displays the digital I/O output signals.
Inputs	Displays the digital I/O input signals.
Robot	Displays the digital I/O robot signals.
Soft	Displays the digital I/O soft signals.
Custom	Allows you to watch I/O variables defined in the workspace. For details on creating variables, see Variable Editor on page 131.

When a signal-type check box is selected (checked), the corresponding group of signals is displayed in the window. For Outputs and Soft Signals, you can manually toggle (turn on/off) the state of a signal by clicking it. The signal states are color coded, as follows:

- Green indicates the signal is on
- Black indicates the signal is off

For details on configuring digital I/O, see the topic [Digital I/O Configuration](#).

### File Explorer



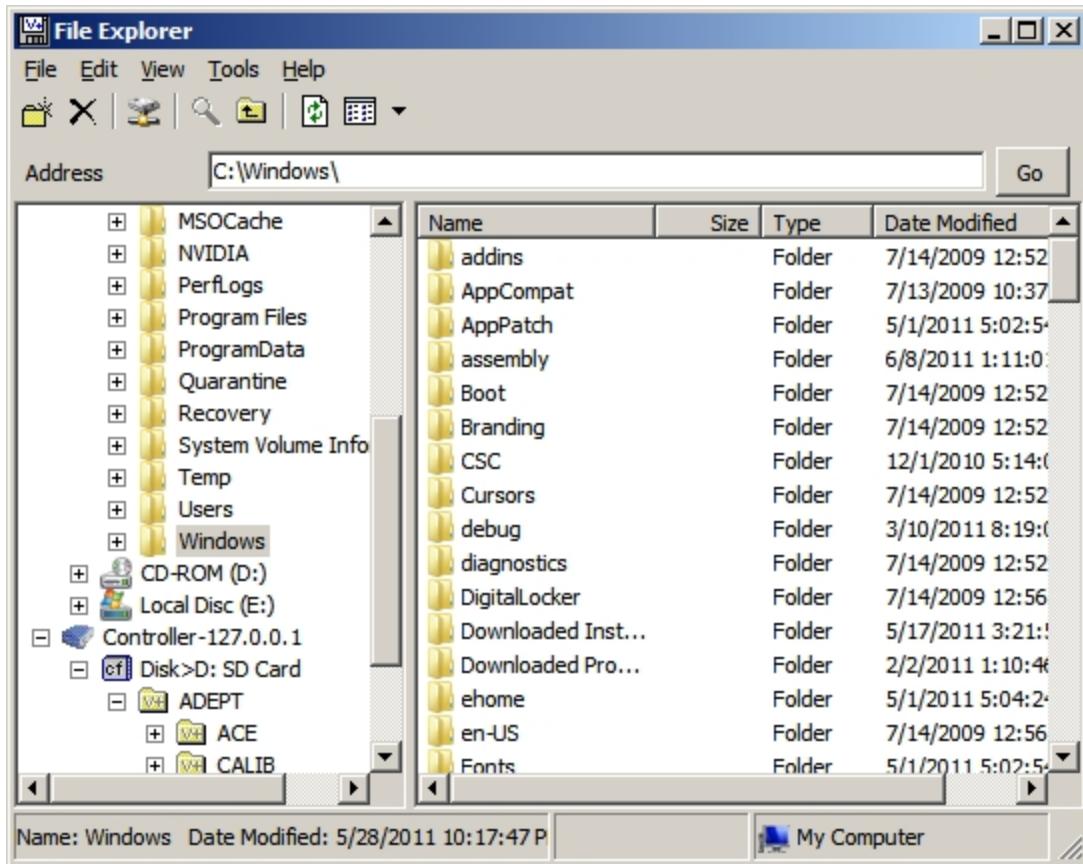
**CAUTION:** When emulation mode is enabled, *do not* save any data to the emulated controller (Disk>D: SD Card). When using emulation mode, ACE creates a new "fresh" emulated-controller file system in a temporary folder. When ACE is shut down, that file system is destroyed, which means the contents of any user-created folders, files, or data will be deleted. Therefore, you should save your data in a PC folder and/or with the ACE workspace.

The File Explorer provides an interface for viewing and managing the files and folders on the connected controller. You can also use this window to: copy/paste, move, rename, and delete files and folders on the connected controller, and drag-and-drop files from your PC to the connected controller. This functionality is similar to that provided by Windows Explorer.

To access the File Explorer, on the ACE toolbar, click the File Explorer ( ) icon. The File Explorer opens.

## File Explorer

---

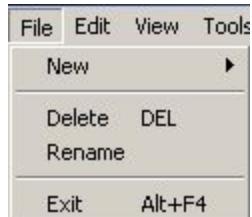


### File Explorer

#### Menu Items

This section describes the selections available from the File Explorer menu.

#### File Menu



##### New

Creates a new file or folder.

##### Delete

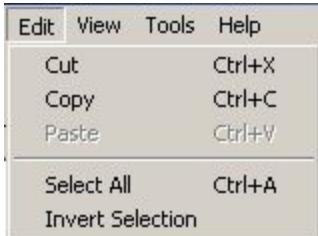
Deletes the selected file or folder.

##### Rename

Renames the selected file or folder.

##### Exit

Closes the File Explorer.

**Edit Menu****Cut**

Copies the selected file or folder to the Windows Clipboard and removes it from the list.

**Copy**

Copies the selected file or folder to the Windows Clipboard.

**Paste**

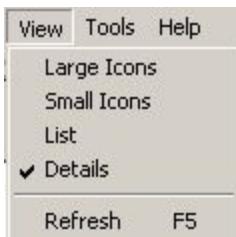
Copies a file or folder from the Windows Clipboard to the selected location.

**Select All**

Selects all items in the current folder.

**Invert Selection**

Inverts the currently-selected items in a folder. The currently-selected items become not-selected; the not-selected items become selected.

**View Menu****Large Icons**

Displays the file and folder items as large icons.

**Small Icons**

Displays the file and folder items as small icons.

**List**

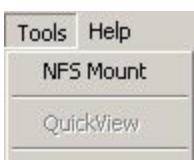
Displays the file and folder items in a list.

**Details**

Displays the file and folder items in a detailed list (includes information on Size, file Type, and Date Modified).

**Refresh**

Updates the File Explorer display to show any recent changes.

**Tools Menu****NFS Mount**

Displays the Mount NFS Drive dialog, which is used to mount an NFS drive.

**QuickView**

Displays the selected file in an ASCII text viewer.

## Help Menu



### Help

Displays the online help for the File Explorer.

## ***Folder and File Tools***

The File Explorer contains a number of tools that are used to perform various operations on the folders and files. To access the folder and file tools:

- click an icon on the File Explorer toolbar
- right-click on a drive, folder, file, or in an empty area of the File Explorer window. Note that the list of available operations will vary, depending on where you right-click.

## Toolbar Folder and File Tools

Use these icons to perform various folder and file operations. The function of each icon is described below.

Icon	Description
	Create a new folder on the selected drive. <a href="#">Details...</a>
	Delete the selected folder/file(s) from the disk.
	Mount an NFS drive on a networked PC. <a href="#">Details...</a>
	View the selected file in an ASCII file viewer.
	Display the folder/directory that is one level above the currently-displayed folder-/directory.
	Refresh the drive/folder/file lists.
	Display the drives/folders/files in icon or list format.

## Shortcut Menu Folder and File Tools

The File Explorer provides tools for the following file or folder operations. Right-click on items in the File Explorer to display a shortcut menu, which contains these options.

Operation	Description
Cut	Deletes the selected file or folder and copies it to the Windows Clipboard.

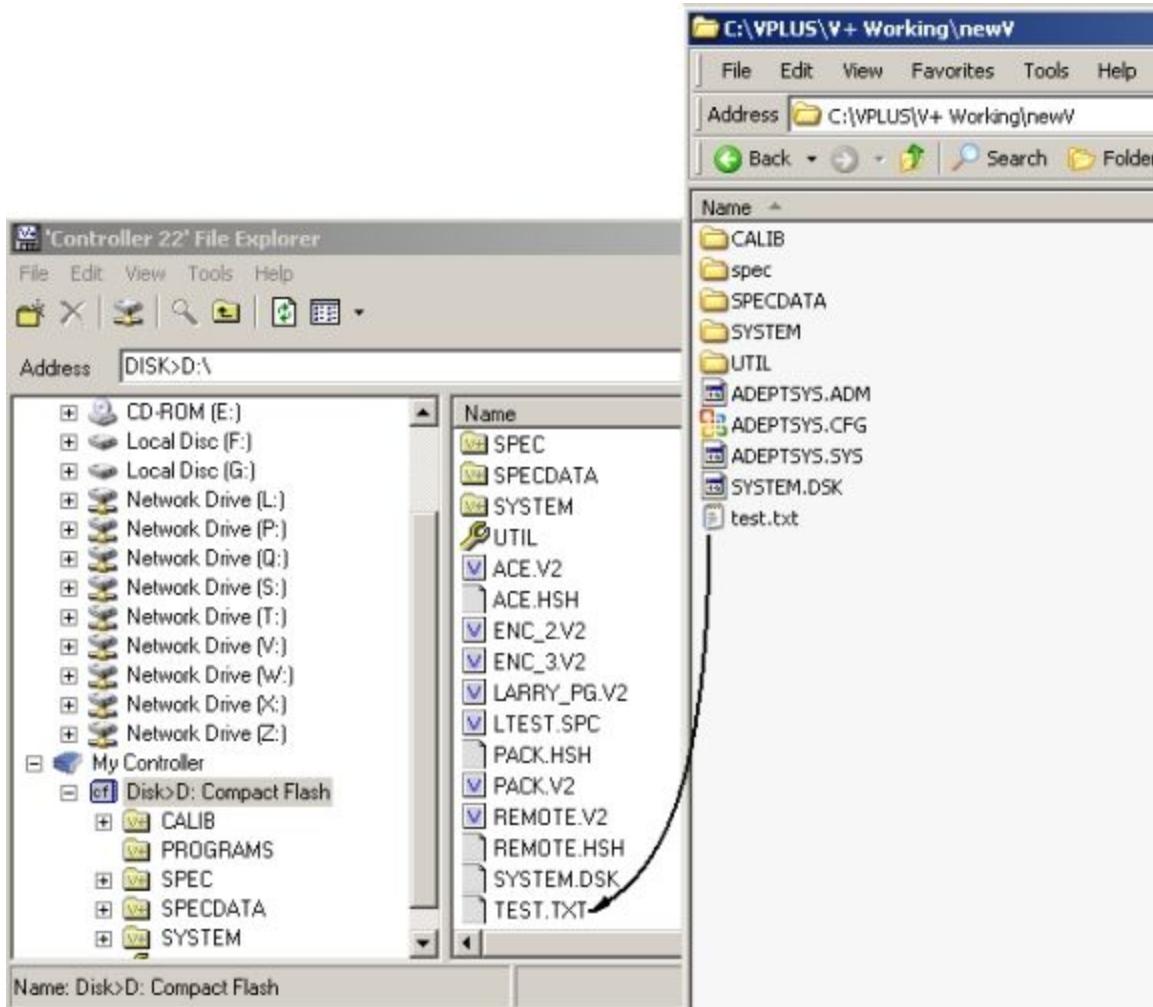
Operation	Description
Copy (Ctrl+C)	Copies the selected file to the Windows Clipboard.
Paste (Ctrl+V)	Pastes the file from the clipboard to the selected location.
Delete (Del)	Deletes (removes from disk) the selected file.
Rename	Allows you to rename the selected file.
Refresh (F5)	Refreshes the File Explorer display.
New Folder (Ctrl+N)	Creates a new folder within the currently-selected folder.

#### ***Drag-and-Drop***

In addition to the tools described in the previous section, the File Explorer also allows you to "drag-and-drop" files or folders (copy a file or folder from its current location to a new location within a lower-level or higher-level folder). Using this feature, you can also copy files from your PC to the connected Adept controller, as shown in the following figure:

## File Explorer

---



*File Explorer (Drag-and-Drop Operation)*

### ***Creating a New Folder***

To create a new folder:

1. Click the Create a New Folder icon ( ) on the File Explorer toolbar to display the Enter New Folder Name dialog:



2. Type a name for the new folder in the field. The folder name must be eight characters or less, and must use alpha-numeric characters or underscores (spaces or other characters are not allowed).
3. Click **Create** to create the new folder, or click **Cancel** to cancel the operation. The dialog closes.

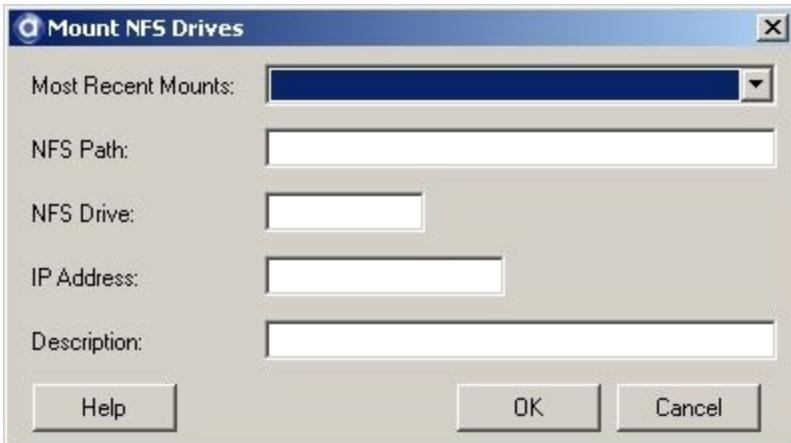
### ***Mounting an NFS Drive***

The Mount NFS Drives dialog box is used for defining and mounting one or more NFS drives.

**NOTE:** In order to mount an NFS drive, you must have an NFS server utility, such as [OmniNFS Server](#) by [XLink Technology, Inc](#), running on the PC. This type of utility allows the PC to "share" a drive or directory with the controller through NFS. Additionally, the desired drive(s) and directories must be exported by the NFS server utility before they can be mounted by the controller.

To mount an NFS drive:

1. Install and run an NFS server utility on your PC. See the online help in the NFS server utility for details on exporting drives or directories for mounting.
2. Select the Mount NFS Drive icon ( from the ACE File Explorer toolbar. The Mount NFS Drives dialog box is displayed.



*Mount NFS Drives Dialog Box*

3. Enter the information in the Mount NFS Drives dialog box, as follows:

Item	Description
Most Recent Mounts	Click the down arrow to display a list of the recently-used NFS-mounted drives.
NFS Path	Enter the physical path of the exported drive/directory. The drive/directory names in the path must be eight characters or less, and must use alpha-numeric characters or underscores (spaces or other characters are not allowed).
NFS Drive	Enter a name for the mounted drive/directory. The name must be eight characters or less, and must use alpha-numeric characters or underscores (spaces or other characters are not allowed).
IP Address	Enter the IP address of the PC that has the exported drive/directory.
Description	Enter a brief description of the drive/directory being mounted.

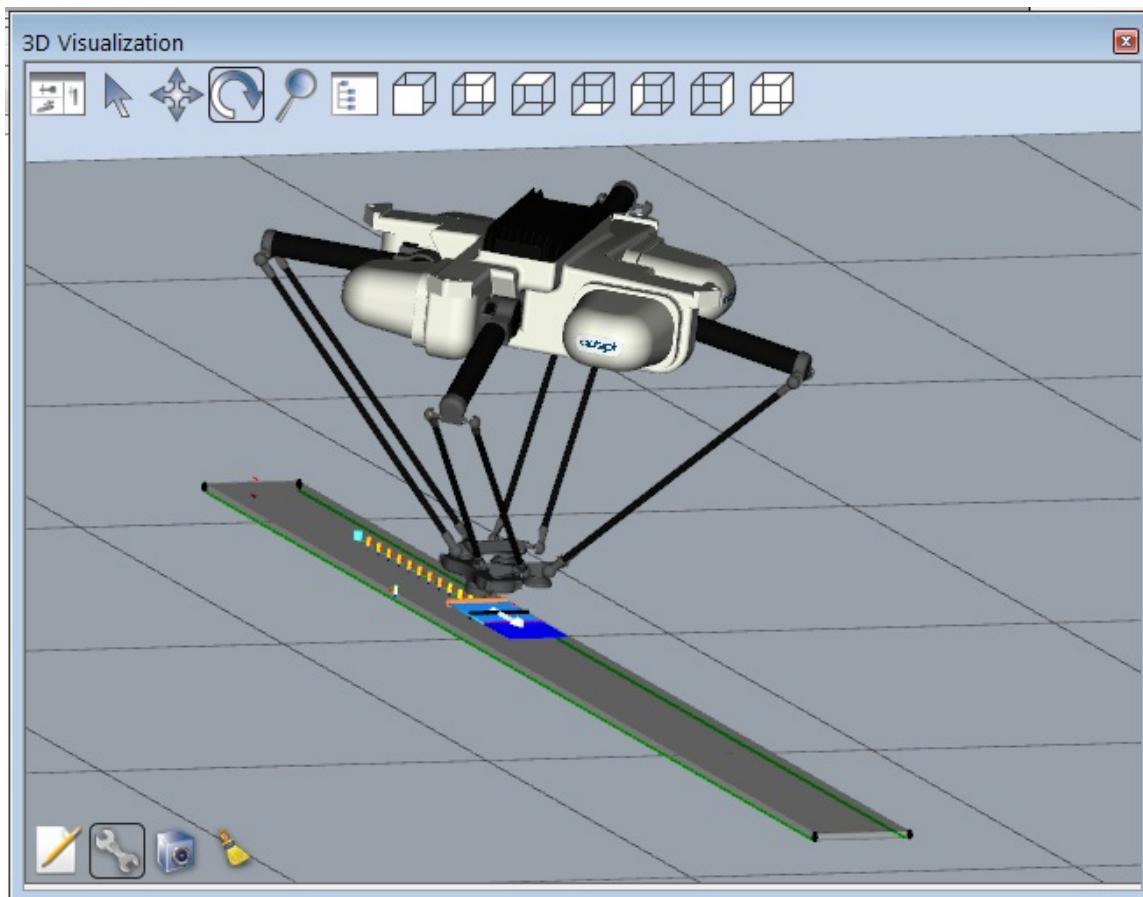
4. Click **OK** to store the information. The new entry is added to the Most Recent Mounts list and the NFS drive is mounted to the controller.

**NOTE:** If you are unable to mount the selected drive/directory, you may have entered an incorrect IP address or NFS Path.

The mounted NFS drive is displayed in the ACE File Explorer directory tree. You can now browse the files and folders on the mounted NFS drive, and access files on the mounted drive(s) from the controller, as if the files/folders on the mounted drives were located directly on the controller.

### 3D Visualization

The ACE software incorporates Adept's "3D Visualization" technology, which automatically creates a 3D visualization (simulation) of your system, as shown in the following figure. For example, when you first connect to the Adept controller, the robot automatically appears in the 3D Visualization window. If you add a gripper to your robot, the tool offset is added to the 3D display. If you add other objects to the Workspace Explorer and then teach the locations of the objects, they automatically move to the correct locations in the 3D display. You can even see representations of your move locations in the 3D Visualization window.



*3D Visualization with robot, conveyor, camera FOV, belt window, and found/allocated instances*

**NOTE:** The default status of all 3D objects is set to "visible". Therefore, any 3D objects in your workspace (like a robot or belt) will be shown in the 3D Visualization window. To hide an object, go to that

object's editor and deselect the "Show in visualization" option or set the "3D Visualization > Visible" property to false.

In addition to the robot, the 3D Visualization window also shows additional information, such as:

- The graphical representation of all belts in the workspace.
- The robot belt windows
- The robot belt dynamic wait line
- The belt lane widths  
(This applies only to Process Manager Belt Calibrations. For details on Process Manager Belt Calibrations, see Belt Calibrations on page 848.)
- The location of latch sensors
- The field of view (FOV) of each camera.
- The Process Manager objects and Part / Part Target instances. Instances are shown as follows:
  - Part allocated = yellow
  - Part not allocated = orange
  - Part Target allocated = light green
  - Part Target not allocated = green

#### **Toolbar Items**

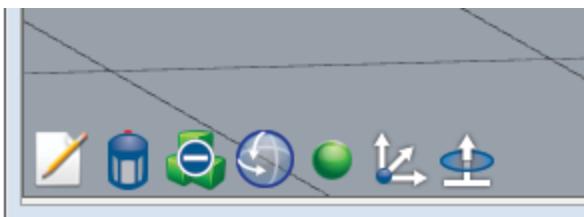
This section describes the items available on the 3D Visualization toolbar:

Item	Description
  	<b>Zoom/Translate/Rotate</b> These functions are used to change your view of the workcell. Select the desired function from the toolbar and then use the left mouse button to control the selected function. You can also use: <ul style="list-style-type: none"> <li>• the Shift (Zoom), Alt (Translate), and Control (Rotate) keys along with the left mouse button, to access and control the functions.</li> <li>• the middle mouse button (pressed) to rotate the view.</li> <li>• the mouse scroll wheel (not pressed) to zoom in or out.</li> </ul> For rotation: <ul style="list-style-type: none"> <li>• If you click on an object to start the rotation, it will rotate around the origin of the selected object.</li> <li>• If you do not click on the object, it will rotate around the scene center (average X, Y, Z position of all objects in the scene).</li> </ul>

Item	Description
	<p><b>Scene Graph</b> Opens the Scene Graph dialog box, which shows the objects and their parent-child connections in the 3D display. Use the check box next to each object to show (checked) or hide (unchecked) the object.</p>
      	<p><b>Front/Back/ Top/Bottom/Left/Right/Iso View</b> Allows you to set the camera location to Front, Back, Top, Bottom, Left, Right or Iso view. The view selections are useful for quickly accessing different views of your workcell. If you are "lost" in the 3D virtual world, you can simply select the desired view position to return to a default view of the workcell.</p> <p><b>NOTE:</b> The Iso (isometric) view is not a true isometric view. Rather, it is a front view that is tilted for an angled view of the workspace.</p>

### Other 3D Visualization Features

When you select an object in the 3D display using the mouse, a series of icons will be displayed at the bottom left hand side of the 3D window. These options allow you quick access to different functions associated with that object:



Some of the functionality available is:

Item	Description
	<p><b>Edit</b> Opens the workspace editor for the selected object.</p>

Item	Description
	<b>Pendant</b> Allows you to move the robot using a pendant in the 3D display. Only valid for robot objects in the 3D display.
	<b>Display Obstacles</b> Allows you to display robot obstacles. Only valid for robot objects in the 3D display.
	<b>Display Robot Work Volume</b> Allows you to display the work volume of the robot. Only valid for robot objects in the 3D display.
	<b>Teach Point</b> Allows you to teach a new robot point at the current position of the robot. Only valid for robot objects in the 3D display.
	<b>Workspace Position</b> Allows you to move the object in the virtual display by clicking and dragging on the orientation adjustment graphic.
	<b>Workspace Orientation</b> Allows you to adjust the orientation of the object in the 3D display by clicking and dragging on the orientation adjustment graphic.
	<b>Show Images</b> Allows you to display camera images in the 3D virtual display. Only valid for process manager or ACE Sight camera calibrations in the 3D display.
	<b>Show Instances</b> Allows you to display instances being tracked in the 3D virtual display. Only valid for process managers in the 3D display.
	<b>Clear Instances</b> Allows you to clear all instances being tracked by a process manager object. Only valid for process managers in the 3D display.

Many of these options are also available if you right-click on the object in the 3D Visualization window. A context menu is displayed with access to the features.

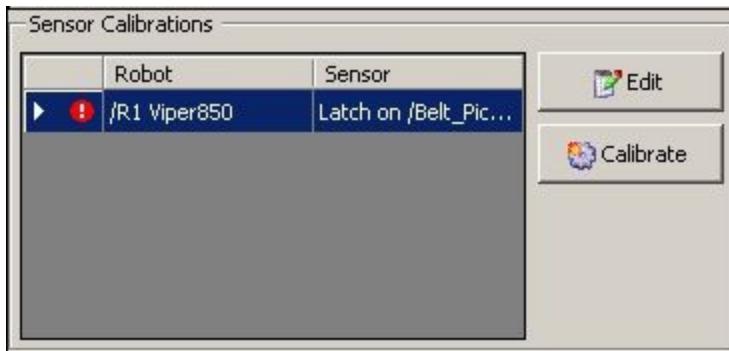
## Function Keys

The ACE software provides special operations through the following function keys:

Key	Description
F1	Displays the online help for the current object.
F7	(Code editors only) Compiles the program.
F10	(Code editors only) Step over the next line in the program.
F11	(Code editors only) Step into the next line in the program.
Ctrl+Shift+B	Close all windows except the current window.
Ctrl+F	Opens the Find dialog. For details, see Find Dialog on page 782.
Ctrl+S	Saves the current workspace. If the workspace has not been saved previously, the Save As dialog opens; if the workspace has been saved previously, it is saved to the current file name. For details, see Save Workspace on page 161.
Ctrl+V	Paste the contents of the Windows clipboard at the cursor position.
Ctrl+X	Cut the selected item (a copy is stored in the Windows clipboard).
Ctrl+Y	Revert the last undo operation.
Ctrl+Z	Undo the last operation.

## Wizards

The ACE software uses "wizards" to simplify the configuration of various items and processes in the work-space. For example, the Sensor Calibrations group in the Process Manager editor uses a wizard for calibrating the selected sensor. (The wizard is started by clicking the Calibrate button.) For details on the Process Manager, see Process Control on page 799.

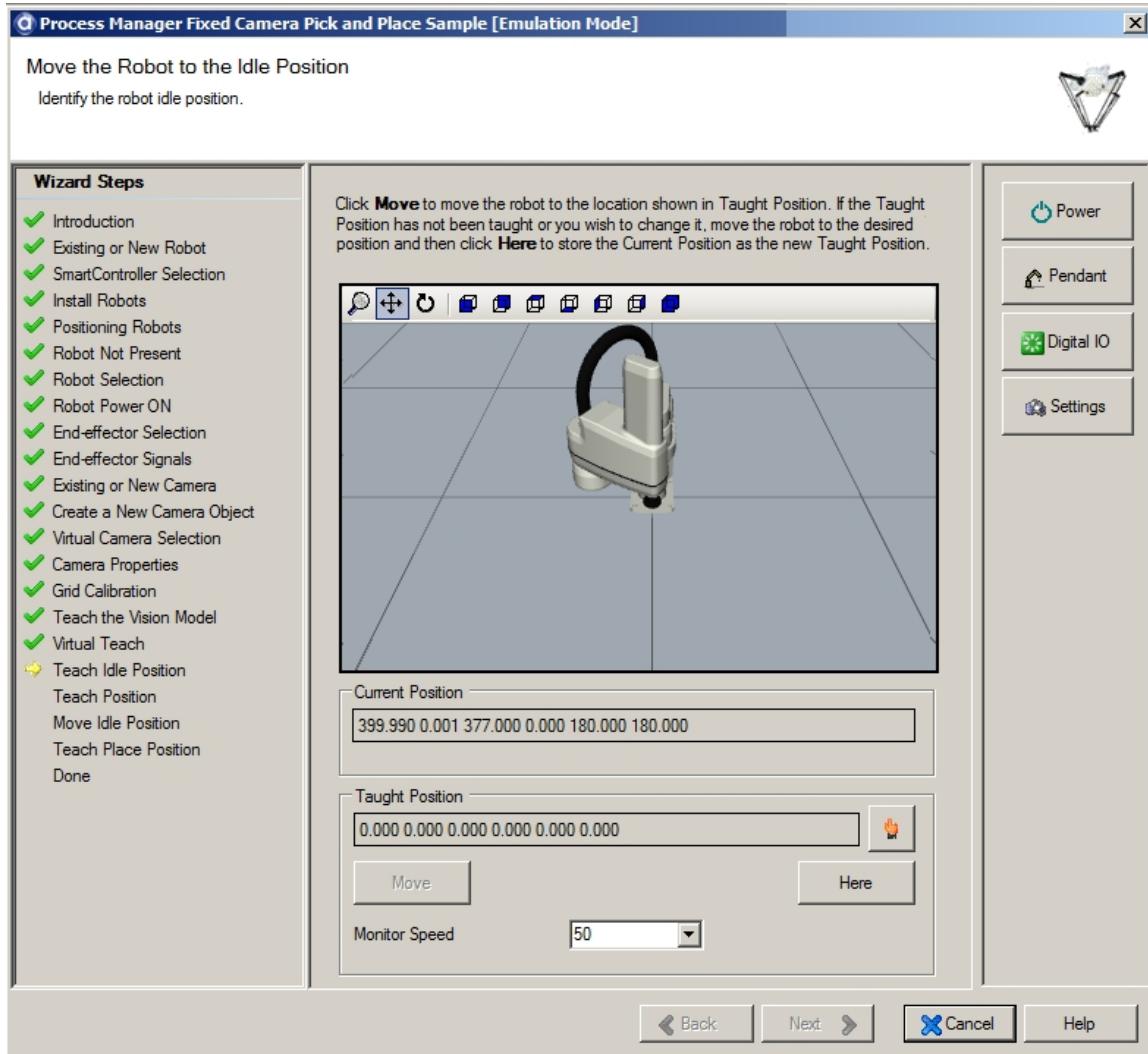


*Sensor Calibrations Group*

An example wizard dialog is shown in the following figure.

## Wizard Elements

---



Example Wizard Dialog

## Wizard Elements

Many of the wizards share common elements (buttons, fields, etc.) The following tables describe the items you may see in the wizard interface.

### Wizard Navigation

Item	Description
Back	Opens the previous screen in the wizard.

Item	Description
	<p><b>NOTE:</b> For certain procedures, you cannot go back to repeat the previous screen. In those cases, the Back button is dimmed (not available).</p>
Cancel	Cancels the operation and closes the wizard.
Help	Opens the online help.
Next	<p>Opens the next screen in the wizard.</p> <p><b>NOTE:</b> The Next button will not be available until the current screen is completed.</p>

#### **Dialog-Access Controls**

Item	Description
Pendant	Opens the Robot Jog Control. For details, see Robot Jog Control on page 163.
Power	Toggles high power for the robot.

#### **Robot and Position Controls**

Item	Description
Approach	Moves the robot to the approach position (the taught position plus the approach height)
Current Position	Displays the current position of the robot.
Depart	Moves the robot to the depart position (the taught position plus the depart height)
End Effector	Displays the selected end-effector (gripper) for the robot.
Gripper	Activates/deactivates the gripper (end effector). Click the signal (■ / □) icon to toggle the state.
Here	Records the current position of the robot. The recorded position is displayed in the Taught Position field.
Monitor Speed	Adjusts the monitor speed (% of full speed) for the robot movements. <b>NOTE:</b> This item changes the monitor speed on the ACE toolbar. For details, see Toolbar on page 157.
Move	Moves the robot to the recorded (taught) position using the speed specified in Monitor Speed.

Item	Description
Taught Position	Displays the taught (recorded) position.

### **Conveyor Belt Controls**

The following controls require that:

- the belt is under active control by the Adept controller,
- the conveyor supports the selected control (for example, to use Reverse/Forward, the conveyor must have a motor that supports operation in reverse direction),
- the appropriate control signals have been defined in the Belt object editor.

For details, see [Belt](#) on page 809.

Item	Description
Fast/Slow	Selects fast or slow speed. Click the Signal (■ / □) icon to toggle the signal state.
On/Off	Starts and stops the conveyor belt. Click the Signal (■ / □) icon to toggle the signal state.
Reverse/Forward	Selects forward or reverse direction. Click the Signal (■ / □) icon to toggle the signal state.

### **Vision Controls**

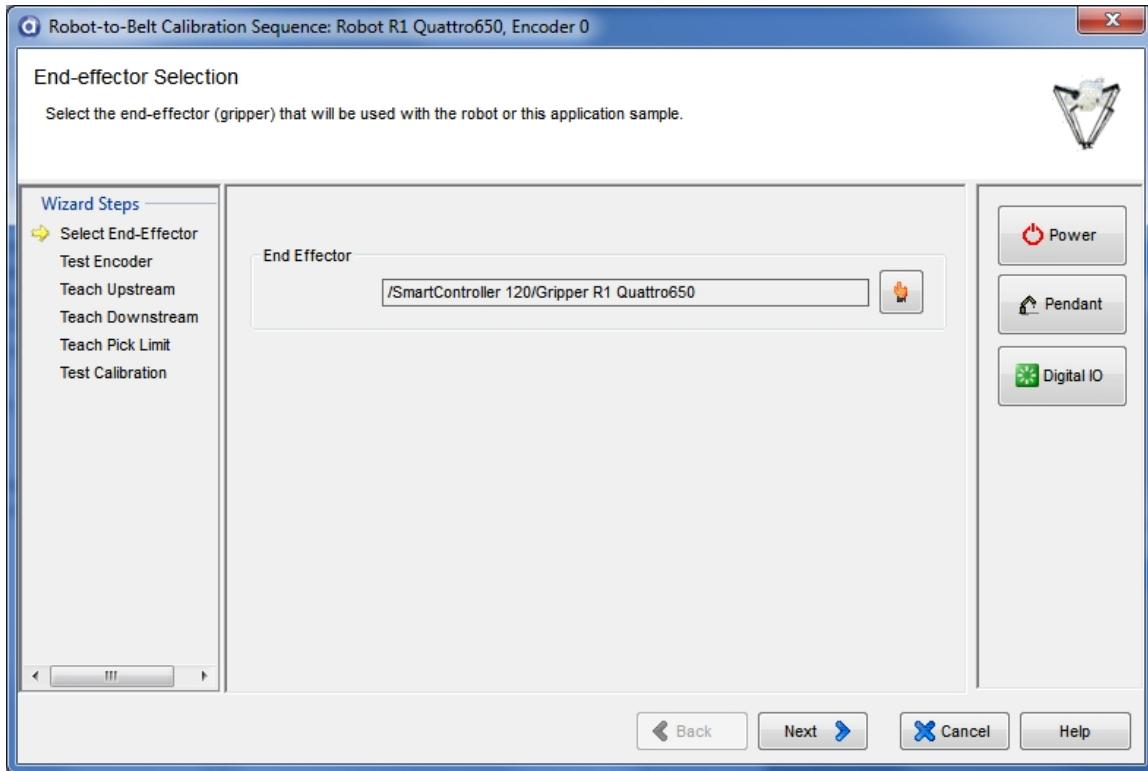
For details on the following items, see the *ACE Sight User's Guide*.

Item	Description
Edit	Opens the Vision Tools Properties window, which is used to edit various parameters of the vision tool.
Live	Displays a live image from the camera input.
Picture	Acquires a still image from the camera input.
Stop	Stops the currently-running vision tool or process. (This is only active in Live mode.)

## **Application Sample and Calibration Wizards**

For application samples, belt calibrations, and sensor calibrations, there are special "interview" wizards that step you through these tasks. These wizards contain a "step" pane, on the left-hand side of the wizard screen, which shows the list of steps and the step you're currently on, as shown in the following figure.

---



*Calibration Wizard with Steps Pane*

For more details on application samples, see Application Samples on page 947.

For more details on belt calibrations, see Belt Calibrations on page 848.

For more details on sensor calibrations, see Sensor Calibrations on page 909.

### Emulation Mode Wizards

When emulation mode is enabled, some of the ACE software wizards contain differences from their operation in standard mode. This section describes those differences. For more details on other features of the emulation mode, see Emulation Mode on page 1089.

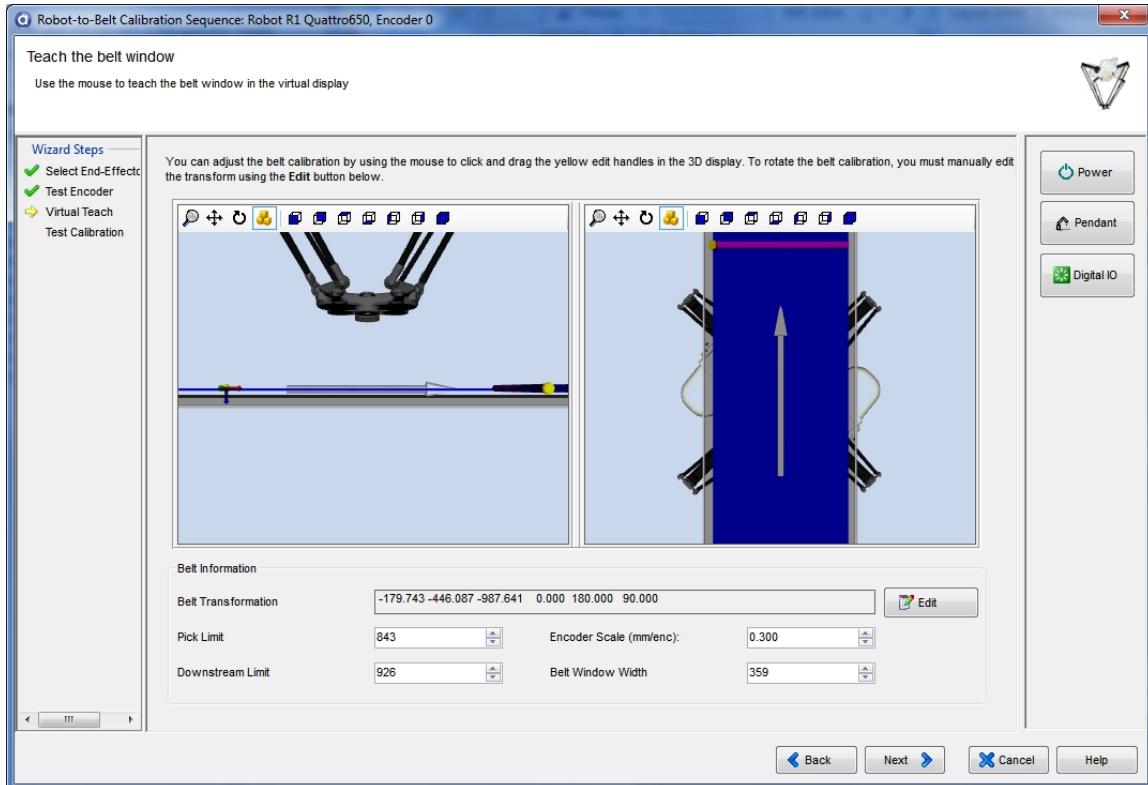
#### Calibration Wizards

**NOTE:** For emulation-mode calibrations, the belt controls in the Calibration wizards will allow you to operate the belt, even when the Active Control option of the Belt object is not enabled.

When performing a belt calibration or sensor calibration in emulation mode, those wizards include special interactive 3D visualization windows, which allow you to interactively position the elements being cal-

ibrated. This feature allows you to see what is being changed, and how the change affects the calibration. An example is shown in the following figure.

**NOTE:** When multiple robots are present that access the same belt in the workspace, if a belt has not been taught (through the Workspace Referencing wizard) it is not displayed in the 3D teach processes. For more details on the Workspace Referencing wizard, see Workspace Positioning on page 945.



### Interactive 3D Windows

For these wizard pages, there are two ways to change the settings:

- Use the interactive 3D windows to drag the elements to the desired positions. After positioning the elements, you can see the changes to the values in the fields below the 3D windows.
- Use the fields below the interactive 3D windows to enter the values. After entering the values, you can see the changes in the 3D windows.

The following table describes the items available on the 3D Visualization toolbar:

<b>Item</b>	<b>Description</b>
	<p><b>Zoom/Translate/Rotate</b> These functions are used to change your view of the workcell. Select the desired function from the toolbar and then use the left mouse button to control the selected function.</p>
	<p><b>Selection/Position</b> Changes the cursor to the selection/position tool (), which allows you to select and position (by dragging) objects in the 3D display. Drag handles are indicated by yellow spheres, as shown in the following figure.</p> <div data-bbox="306 699 899 973"> </div> <p><i>Drag Handles</i></p>
	<p><b>Scene Graph</b> Opens the Scene Graph dialog box, which shows the objects and their parent-child connections in the 3D display. Use the check box next to each object to show (checked) or hide (unchecked) the object.</p>
	<p><b>Front/Back/Top/Bottom/Left/Right/Iso View</b> Allows you to set the camera location to Front, Back, Top, Bottom, Left, Right or Iso view. The view selections are useful for quickly accessing different views of your workcell. If you are "lost" in the 3D virtual world, you can simply select the desired view position to return to a default view of the workcell.</p> <p><b>NOTE:</b> The Iso (isometric) view is not a true isometric view. Rather, it is a front view that is tilted for an angled view of the workspace.</p>



## V+ System Configuration Tools

The V+ operating system configuration has historically been performed through a series of command-line V+ utilities. The ACE software improves this through a set of configuration tools, which provide both programmatic and GUI access to the V+ system configuration utilities, such as upgrading V+, restoring V+, backing up V+, and digital I/O configuration. The topics in this chapter provide details on these configuration tools.

Controller Installation .....	190
DeviceNet Scan .....	193
Digital I/O Configuration .....	195
Encoder Latches .....	198
FireWire Node Firmware Upgrade .....	200
Python Module/Smart Servo Kit Configuration .....	202
Robot and Motor FireWire Configuration .....	205
Robot Installation .....	211
V+ Backup .....	214
V+ Restore .....	215
DeviceNet Configuration .....	216
MACID Statement .....	218
V+ System Configuration Editor .....	221
V+ Upgrade .....	228
V+ License Configuration .....	230

## Controller Installation

The ACE software can communicate with multiple Adept SmartControllers. When you first create a workspace and connect to a controller, that controller, and its corresponding robots and grippers, are added to a folder in the workspace explorer, as shown in the following figure.



*Controller with Connected Robot/Gripper*

For some applications, you may have several controllers in your system. The ACE software allows you to add additional controllers. A Controller Installation wizard is provided for this purpose. As each controller is added, it will be placed in a separate folder.

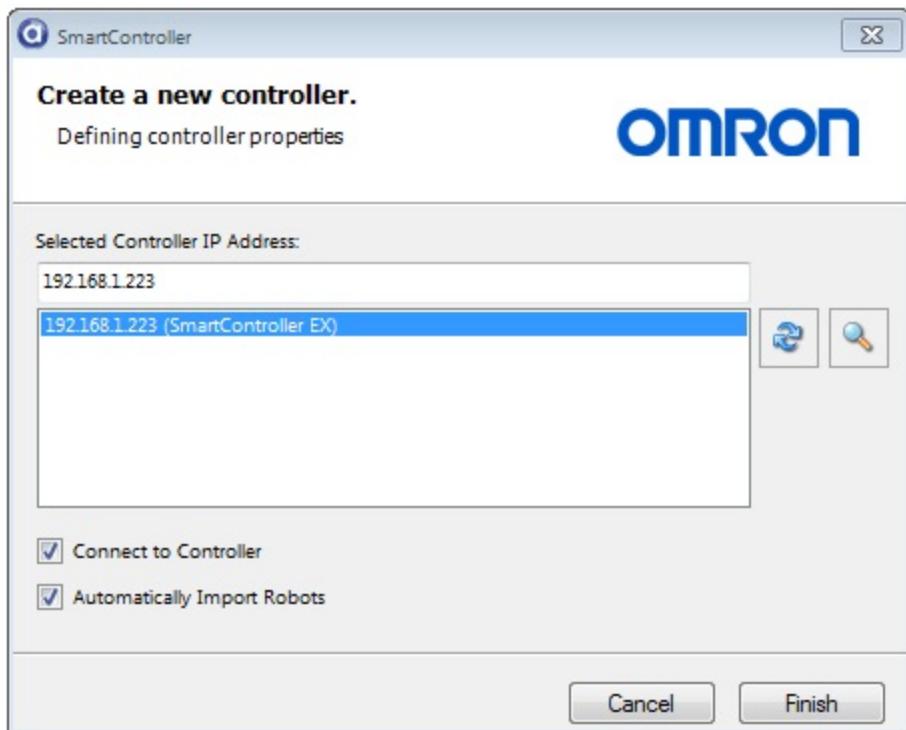
**NOTE:** Any robots (and grippers) attached to the new controller will be automatically detected and added into the same folder.

After the controller (and attached robots and grippers) are added to the workspace, they are available for use in your ACE application.

## Using the Controller Installation Wizard

To use the Controller Installation wizard:

1. Right-click on the top-level folder in the Workspace Explorer and select **New > Device > SmartController**. The Controller Installation Wizard opens.



*Controller Installation Wizard*

2. A default Controller Name and IP Address are supplied. You can change these to meet your system requirements, as described in the following table:

Item	Description
Controller Name	A descriptive name for the controller. This name will be used to for the controller object in the Workspace Explorer.
IP Address	The IP address of the controller you wish to add. Select a controller from the drop-down list. <ul style="list-style-type: none"><li>• To refresh the controller list, click the  (refresh) icon.</li><li>• To detect, and optionally change, the IP address of a controller to which you are connected, click the  (binoculars) icon. For more details on detecting a new controller, see <a href="#">Configuring Ethernet Communications</a>.</li></ul>
Connect to Controller	Select this option to automatically connect to the controller when

Item	Description
	it is added to the Workspace Explorer.
Automatically Import Robots	Select this option to automatically import the robots (and corresponding grippers) that are connected to the controller.

3. Click **Finish** to exit the wizard. A new folder is created. The new controller and (optionally) any attached robots/grippers are added to the Workspace Explorer.



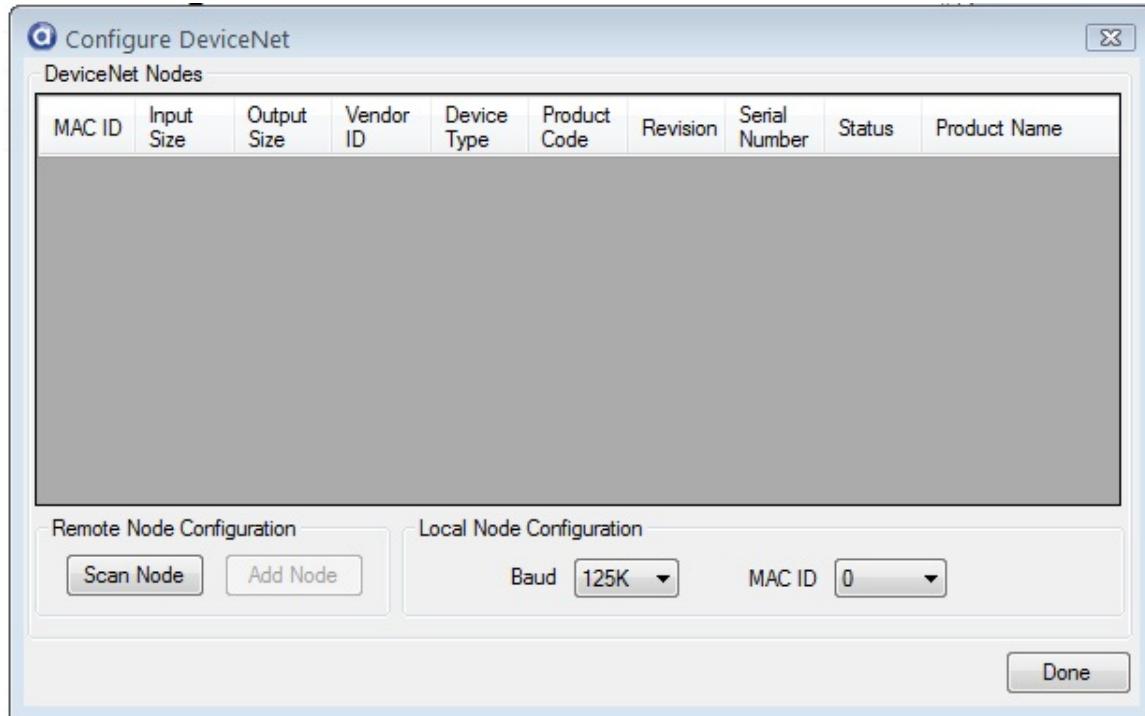
New Controller in Workspace Explorer

**NOTE:** The folder name and controller name are based on the Controller Name field in the Controller Installation wizard.

## DeviceNet Scan

**NOTE:** When emulation mode is enabled, this item is not available. For more details on emulation mode, see Emulation Mode on page 1089.

The DeviceNet scanner, shown in the following figure, allows you to scan for DeviceNet nodes and view information about them (product name, input size, output size, product id, status, etc). The DeviceNet scanner is accessed from the Configure button on the Controller Editor object. For more details, see Controller Editor on page 97.

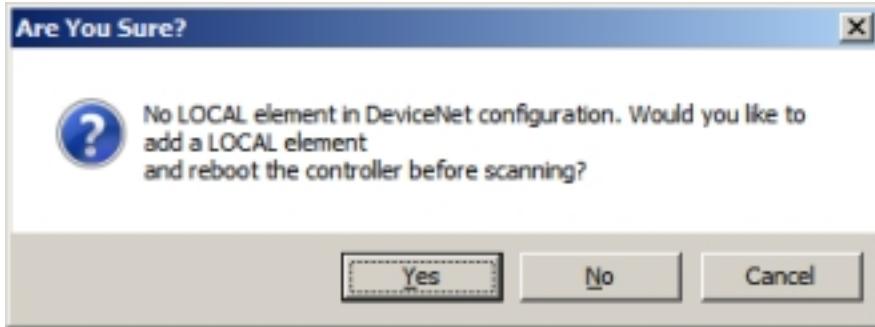


### DeviceNet Scanner

When using the DeviceNet scanner to scan for DeviceNet nodes on your system:

- Use the Baud Rate drop-down list box to set the desired baud rate for your system. This must be set to the same value as defined in the DeviceNet area on the [Configuration](#) tab (described earlier in this topic).
- Use the Local MAC ID drop-down list box to set the MAC ID for your system. This must be set to the same value as defined in the DeviceNet area on the [Configuration](#) tab (described earlier in this topic).
- Click **Scan** to initiate the scanning process.

If there is no LOCAL statement, you are prompted to add one before scanning, as follows:



*No LOCAL Statement Prompt*

Otherwise, the scan is initiated.



*DeviceNet Scan Progress Indicator*

**NOTE:** The scanning process will take several minutes, depending on the number of nodes on your system.

- Click **Add** to:
  - Clear the old MAC ID configurations
  - Add the new entries
  - Write the changes to the controller

For more information on DeviceNet, see DeviceNet Configuration on page 216.

## Digital I/O Configuration

The digital I/O for the V+ operating system uses numeric signal numbers, with possible outputs in the range 0001 through 0512, and inputs in the range 1001 through 1512. The mapping from signal numbers to inputs and outputs is a two-stage process:

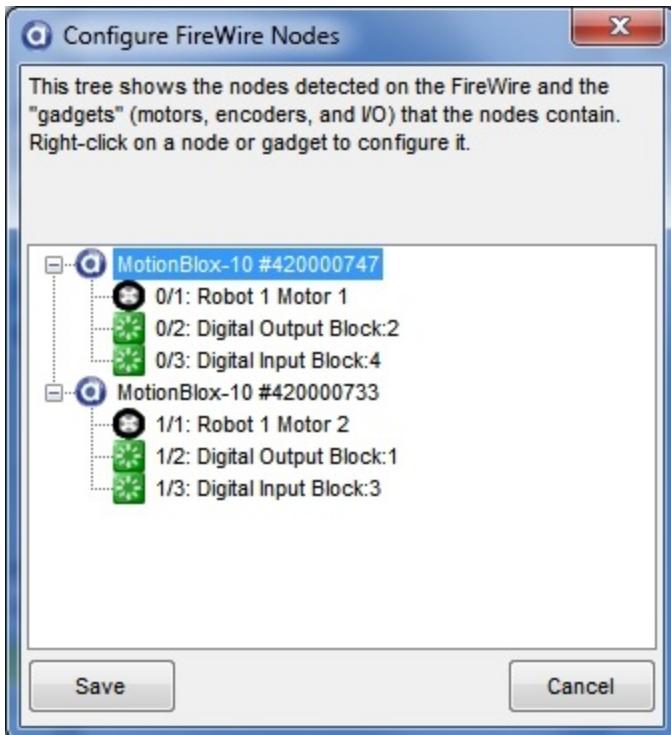
1. Map blocks of FireWire I/O to "output blocks" and "input blocks" in the range of 1 to 31, using the FireWire Configuration dialog box.
2. Map the output and input blocks to I/O numbers through the V+ System Configuration dialog box.

Additional configuration, such as position latching or I/O changes, can be done using the V+ System Configuration dialog box.

## Using the Configure (FireWire) Nodes Dialog

The FireWire configuration is accessed from the Configure button on the Controller Editor object. For more details, see Controller Editor on page 97. This dialog is used to configure (map) blocks of FireWire I/O, as described in the following steps:

1. Access the On the [Controller Editor](#), click on the **Configure** button and access the **Configure FireWire Nodes**. The Configure Nodes dialog box opens.



*Configure Nodes Dialog*

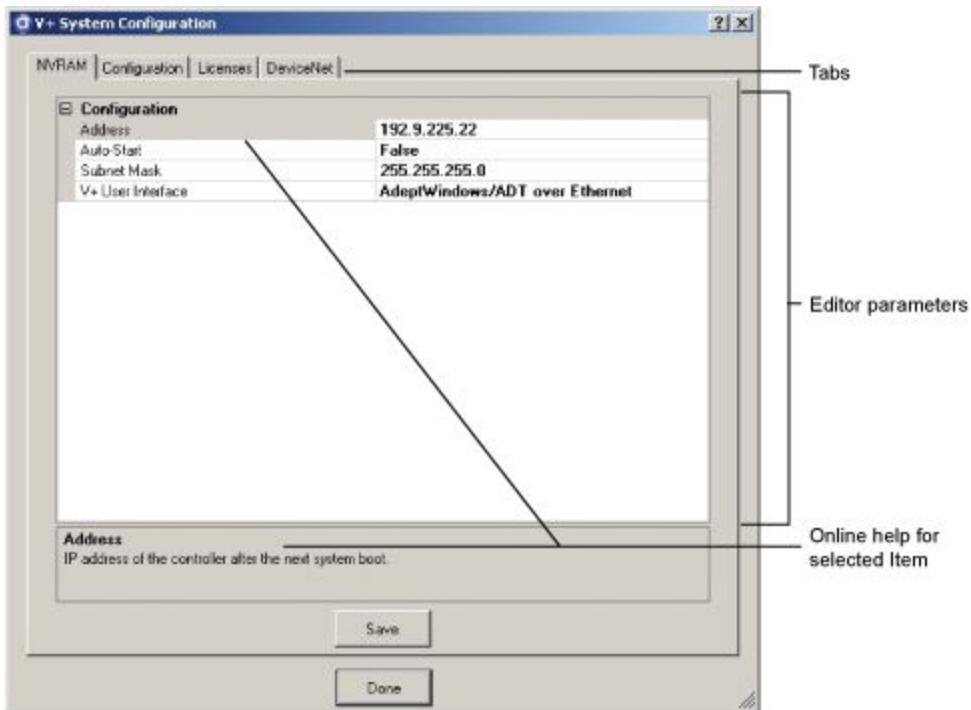
2. Right-click on any digital input or output node. Choose the output block or input block numbers. Ensure that only one instance of each output and input block numbers exists on the network.
3. Click **Save** to write changes to the FireWire nodes.

After completing the above steps, you must map the output and input blocks to I/O numbers, as described in the next section.

## Using the V+ System Configuration Dialog

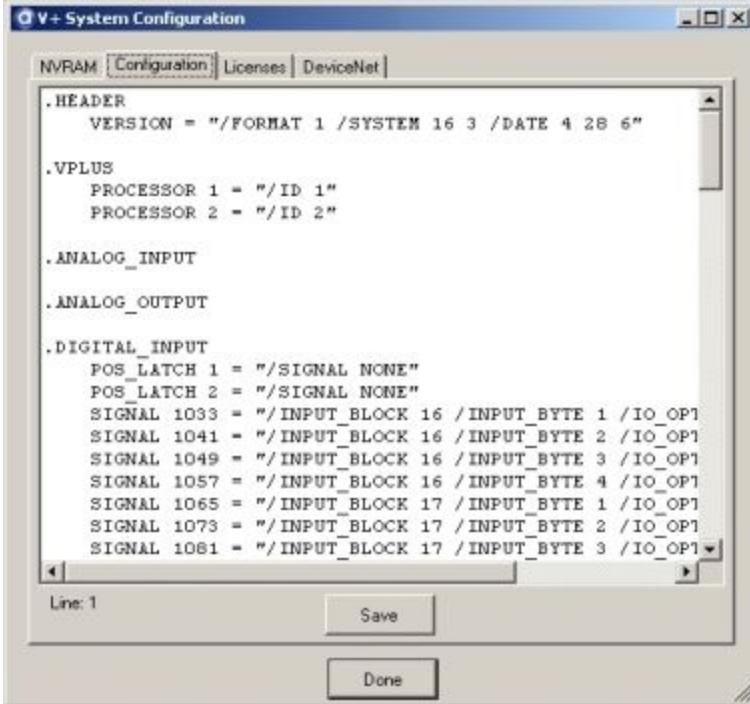
The V+ System Configuration Dialog is used to map the output and input blocks to I/O numbers, as described in the following steps:

1. From the [Controller Editor](#), click **Configure V+**. The V+ System Configuration dialog box opens.



*V+ System Configuration Dialog*

2. Click the Configuration tab to access the text editor.

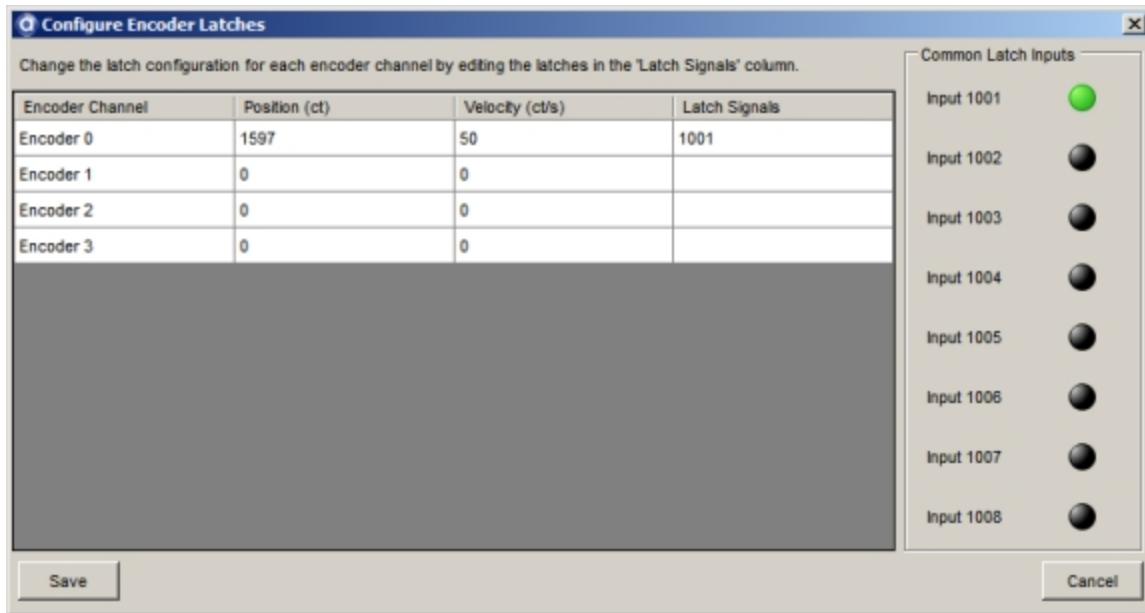


*V+ Configuration Text Editor*

3. Edit the configuration file to map the input and output blocks to I/O numbers.
  - You can press Ctrl+F to display a Find/Replace dialog box, which can be used to find and/or replace specified text.
  - You can click **Help** to display help on the configuration statement syntax.
4. Click **Save** to write the changes to the Adept controller.
5. Click **Done** to close the dialog box.

## Encoder Latches

The Configure Encoder Latches tool, shown in the following figure, allows you to access the information needed to configure the belt encoders. The Configure Encoder Latches tool is accessed from the Configure button on the Controller Editor object. For more details, see Controller Editor on page 97.



*Configure Encoder Latches*

The following table describes the items available in the Configure Encoder Latches tool:

Item	Description
Encoder Channel	Shows the belt encoder channels for the system.
Position (ct)	Shows the current position of the corresponding belt encoder. When the belt is moving, the position reading will change based on the current velocity setting.
Velocity (ct/s)	Shows the current velocity in counts per second for the corresponding belt encoder. You can temporarily change the specified velocity here, and see the effect on the Position reading.
Latch Signals	Shows the latch signal assignments for the corresponding encoder. You can change the latch signal assignments here. You must exit the tool by clicking Save to store the changes.
Common Latch Inputs	Shows the input signals that are activated.
Save	Saves the changes and closes the tool. You are prompted to reboot the controller for the changes to take effect.

## Encoder Latches

---

Item	Description
Cancel	Closes the tool without saving the changes. You are shown a confirmation screen, if you have made changes and you click Cancel.

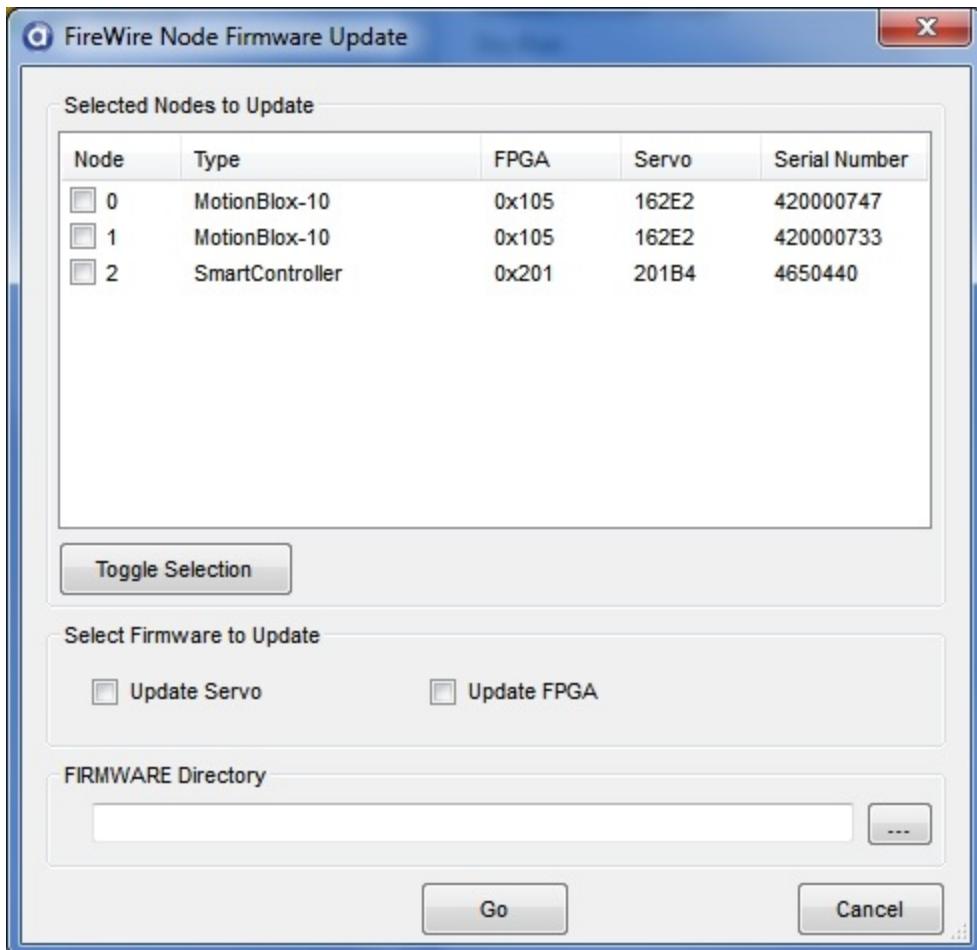
## FireWire Node Firmware Upgrade

Servo firmware is included in the standard V+ operating system's FIRMWARE directory. While the proper version of servo firmware is typically downloaded to the servo nodes during the V+ operating system startup, there are sometimes reasons for upgrading that firmware. The FireWire Node Firmware Update dialog box, shown in the following figure, is used to upgrade the FireWire node firmware.

**NOTE:** The FireWire and FPGA firmware can also be upgraded during the V+ Upgrade process. For details, see [Upgrading V+](#).

To upgrade the FireWire node firmware:

1. On the [Controller Editor](#), click **Upgrade > FireWire Firmware** and then click **Next**. The Firmware Update dialog box opens.



*Firmware Update Dialog*

2. Select the nodes to update from the Select Nodes to Update list. For convenience, you can click **Toggle Selection** to check/uncheck the items in the list.

**NOTE:** The upgrade operation may take several minutes per selected node.

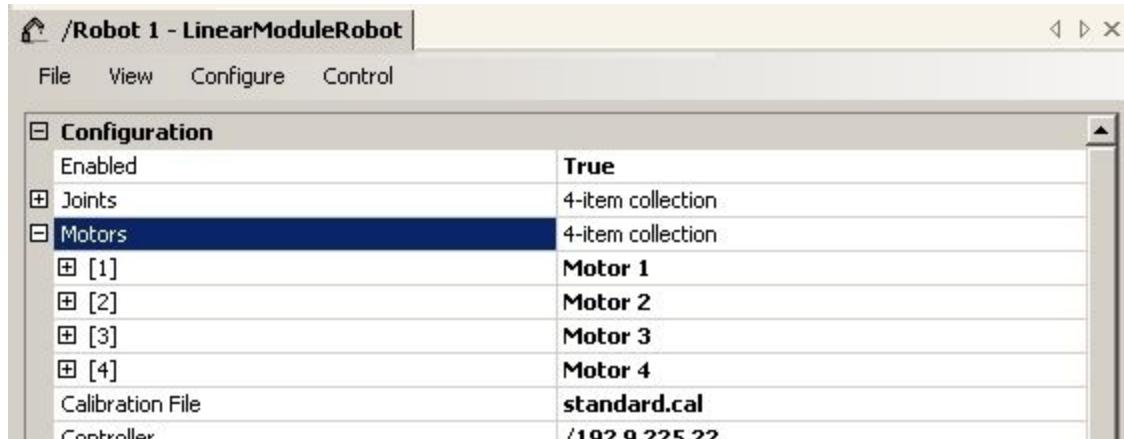
3. Use the "Select Firmware to Update" group to select the Servo Firmware and/or the FPGA Firmware.
4. In the Firmware Directory text box, specify the directory on the PC where the files are located. You can use the browse button () next to the text box to locate the directory on your PC.
5. Click **Go** to start the update.

## Python Module/Smart Servo Kit Configuration

Python modules and Smart Servo Kits have part numbers embedded in the encoders. This allows ACE to automatically load the appropriate motor specifications into the system. The procedure below describes how to teach the calibration for your Python modules and Smart Servo Kit systems.

To configure your system:

1. Ensure that your FireWire nodes are properly configured with the correct robot and motor numbers. To open the FireWire Nodes Configuration dialog box, on the [Controller Editor](#), click **Configure Nodes**. For details, see the topic Robot and Motor FireWire Configuration on page 205.
2. Use the Robot Append, Replace and Select Wizard, if you need to change the robots in your system. To open the wizard, on the [Controller Editor](#), click on the **Configure** button and access the **Configure Robot** option. For details, see the topic Robot Installation on page 211.
3. Use the Load Spec File dialog box to load the default specification (SPEC) files for each Python module or Smart Servo Kit. To open the dialog box, from the File menu of the Robot Object Editor, choose "Load Spec File...". For details, see the topic Loading and Saving Robot/Motor Specification Files on page 251.
4. Teach the calibration for each motor, as follows:
  - a. Display the robot motor list. From the Robot Object Editor, click the '+' next to Motors to display the list of motors for the robot.



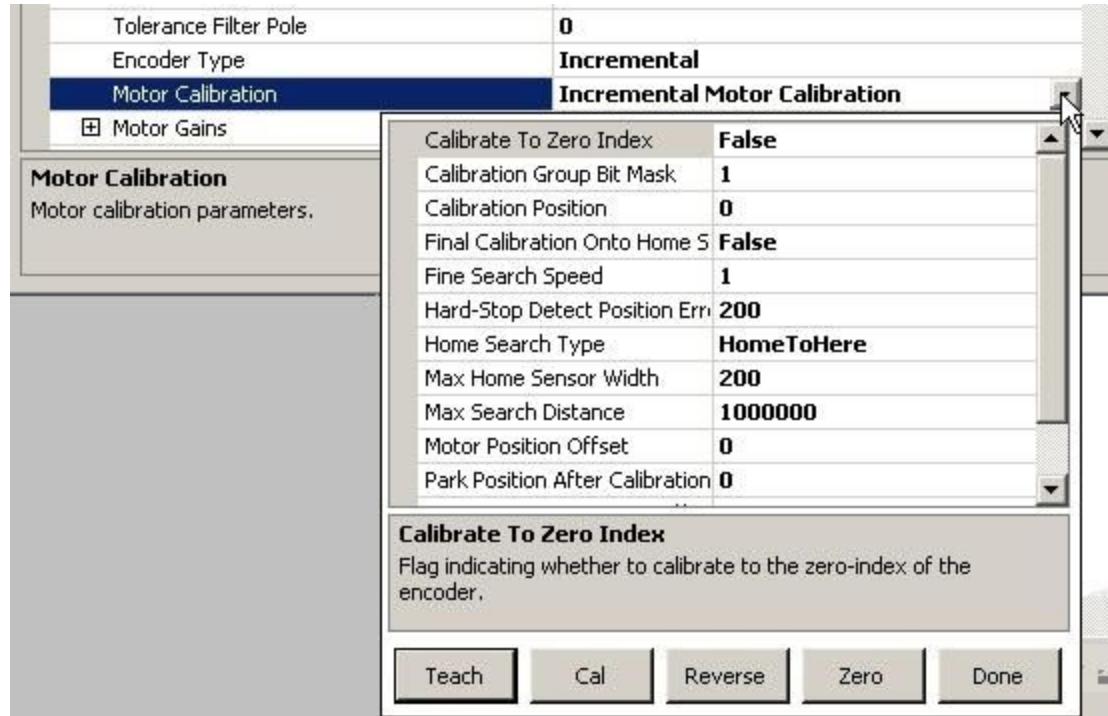
*Robot Motor List*

- b. Double-click the first motor in the list to display the motor-related parameters.

☐ Motors	4-item collection
☒ [1]	<b>Motor 1</b>
Maximum Output Level	<b>5000</b>
Maximum Manual Mode Output Level	<b>5000</b>
Duty-Cycle Output Limit	<b>5000</b>
Duty-Cycle Filter Pole	<b>9</b>
Soft Envelope Error Limit	<b>6144</b>
Hard Envelope Error Limit	<b>12288</b>
Manual Mode Envelope Error Limit	<b>2048</b>
Velocity Envelope Error Limit	<b>2E+09</b>
Default Motion Speed	<b>4</b>
Fine Nulling Tolerance	<b>10</b>
Coarse Nulling Tolerance	<b>400</b>
Required Time In Tolerance	<b>0.016</b>
Tolerance Filter Pole	<b>0</b>
Encoder Type	<b>Incremental</b>
Motor Calibration	<b>Incremental Motor Calibration</b>
☒ Motor Gains	

*Motor Configuration Parameters*

- c. Select the Motor Calibration item and use the down arrow to display the motor calibration parameters.

*Motor Calibration Parameters*

- d. Select Home Search Type and use the drop-down list to set the value for that parameter.
  - If your motor has a hard stop, choose "HomeToStop".
  - If your motor has no stop (like a conveyor), choose "HomeToHere".
- e. Click **Teach** to teach calibration. The hardware front panel power button may flash—in that case push the button to enable power. When the teach operation has completed, a message displays. Click **OK** to continue.
- f. Repeat the steps above for the other motors.
5. After you have taught the calibration for all motors, from the Robot Object Editor menu, select **File > Save All Specs to V+ Boot Disk** to save the specifications.
6. You can now calibrate your system at startup, or when needed, by clicking **Calibrate** on the Robot Control. For details, see the topic [Robot Control](#).

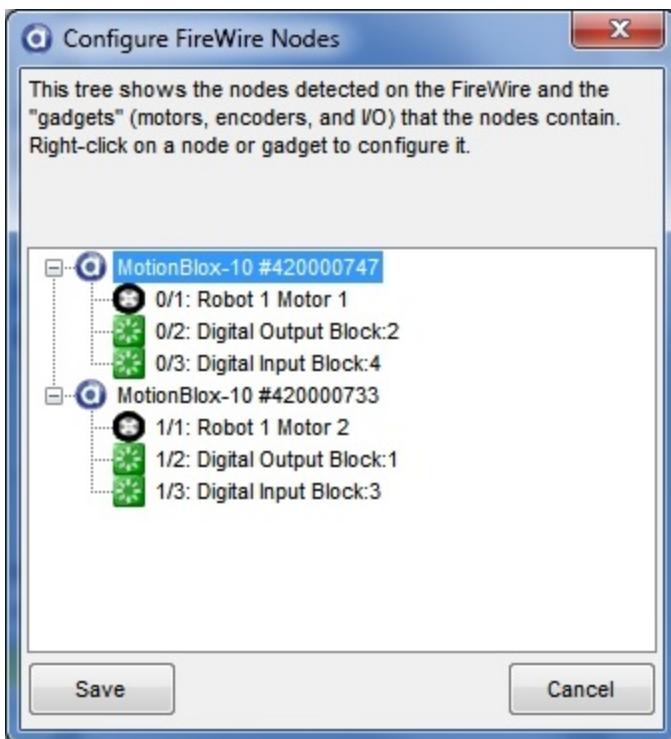
## Robot and Motor FireWire Configuration

**NOTE:** When emulation mode is enabled, this item is not available. For more details on emulation mode, see Emulation Mode on page 1089.

Robot and motor numbers are stored in each FireWire node. At startup, the V+ operating system uses this information to ensure that each motor is properly configured. The Configure Nodes dialog box, shown in the following figure, is used to configure the FireWire nodes.

To configure the FireWire nodes:

1. On the [Controller Editor](#), click **Configure > FireWire Nodes** and then click **Next**. The Configure Nodes dialog box opens.



*Configure Nodes Dialog*

2. Right-click on a node to configure or identify the node. A shortcut menu opens, which provides the following options:

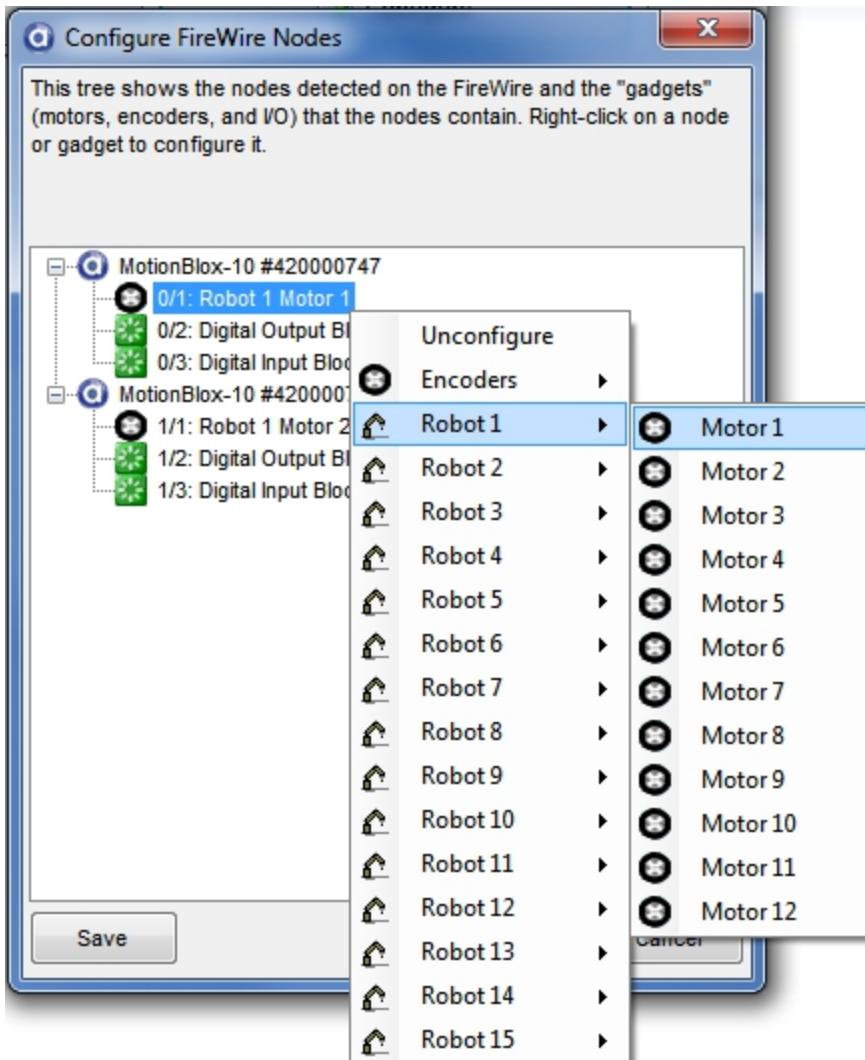
Item	Description
Flash LED	Flashes the indicator LED for the selected node, and allows you to specify the color and flash rate.

Item	Description
View Event Log	Opens the FireWire Event Log window, which displays a record of the FireWire events. For details, see FireWire Event Log on page 1110.
Clear Event Log	Clears the FireWire event log. For details, see FireWire Event Log on page 1110.
Set Node Clock	Opens a window that is used to synchronize the node clock to the PC clock.
View Node Information	Opens a window that provides: Node Time, Node Version, Servo Version, FPGA Version, Serial Number, and Servo Rate.

3. After you have identified the FireWire node, right-click on a robot/motor "gadget" and assign the robot and motor numbers. For details, see Setting the Robot and Motor Numbers on page 206.
4. Right-click on a Digital Input or Output block "gadget" and assign the block number. For details, see Setting the sDIO Block Numbers on page 208.
5. Click **Save** to store the changes and close the dialog box.

### Setting the Robot and Motor Numbers

1. After you have identified the FireWire node, right-click on a robot/motor "gadget" and assign the robot and motor numbers, as shown in the following figure.



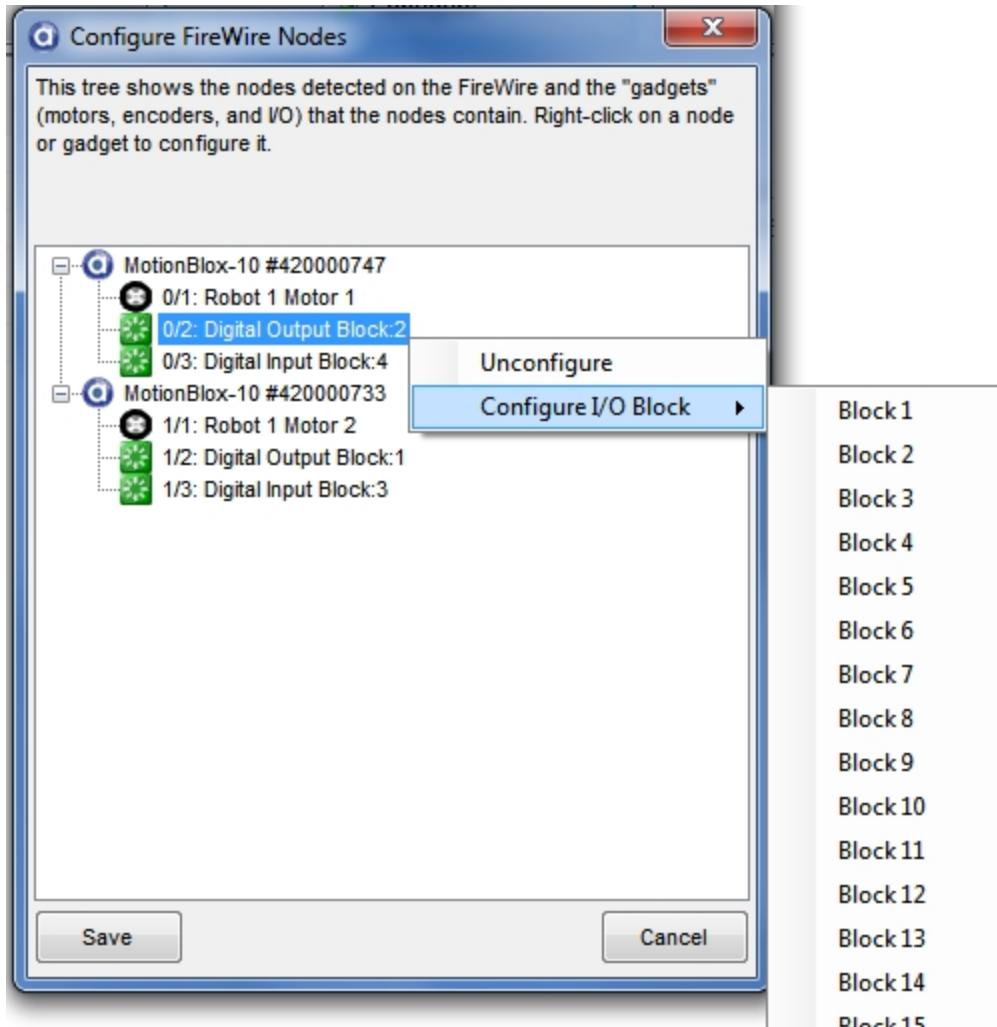
*Configure Robot and Motor*

Note the following:

- On Adept robots, you can only select the robot number; the motor numbers are pre-configured.
  - On MotionBlox or sMI-6 axes, you must select both the robot and motor number for each axis.
2. Click **Save** to store the changes and close the dialog box.

## Setting the sDIO Block Numbers

1. After you have identified the FireWire node, right-click on a Digital Input or Output block and assign the block number, as shown in the following figure.



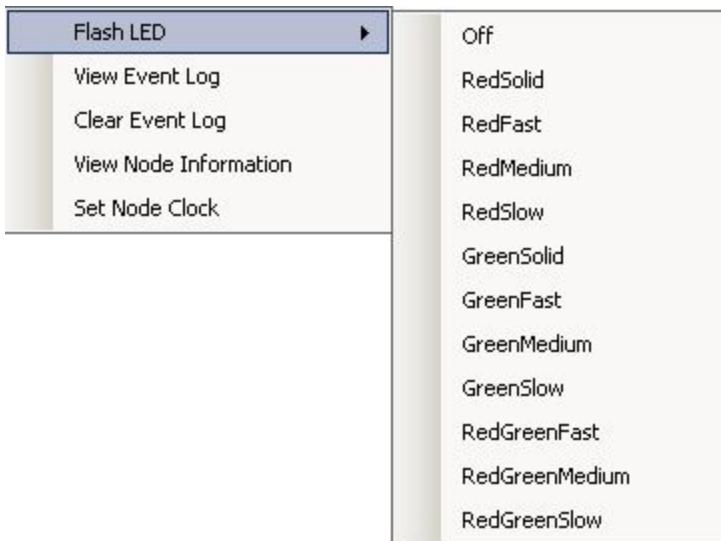
Configure sDIO Block

2. Click **Save** to store the changes and close the dialog box.

## Flashing the Node LED

To flash the LED for the selected node:

1. Right-click on a node. The following menu opens:



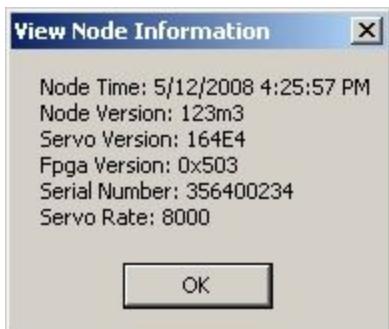
*FireWire Shortcut Menu*

2. From the menu, select **Flash LED** and then select a flash option to help identify the node.

## Viewing the Node Information

To view the details about the selected node:

1. Right-click on a node to display the shortcut menu.
2. From the menu, select **View Node Information** to display the View Node Information window.



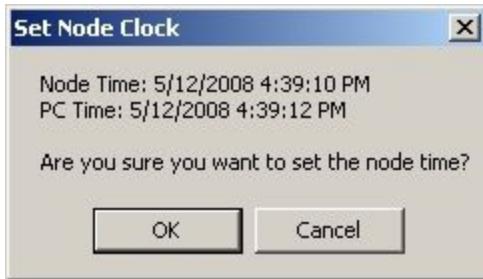
*View Node Information Window*

3. Click **OK** to close the window.

## Setting the Node Clock

To set the clock for the selected node:

1. Right-click on a node to display the shortcut menu.
2. From the menu, select **Set Node Clock** to display the Set Node Clock dialog.



*Set Node Clock Dialog*

3. Click **OK** to synchronize the node clock to the PC clock, or click **Cancel** to cancel the operation. The dialog closes.

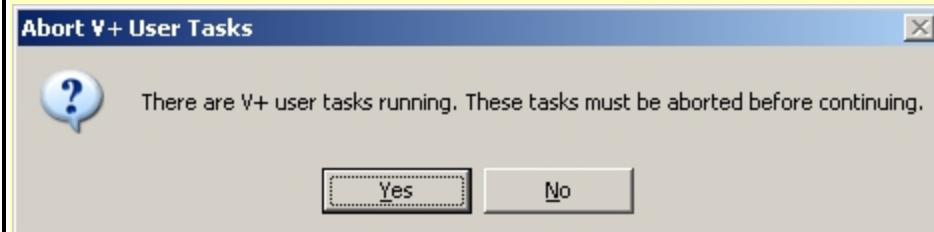
## Robot Installation

The V+ operating system can control multiple robots of various types, such as SCARA, Cartesian (linear modules), and six-axis. For most Adept robots, no additional robot configuration is required. However, if linear modules or OEM robots are used, a multi-step Robot Installation Wizard is provided, which helps you install and configure the robots that will be controlled by the system (see the following figure).

To use the Robot Installation Wizard:

1. On the [Controller Editor](#), click **Configure > Robots** and then click **Next**. The Introduction page opens.

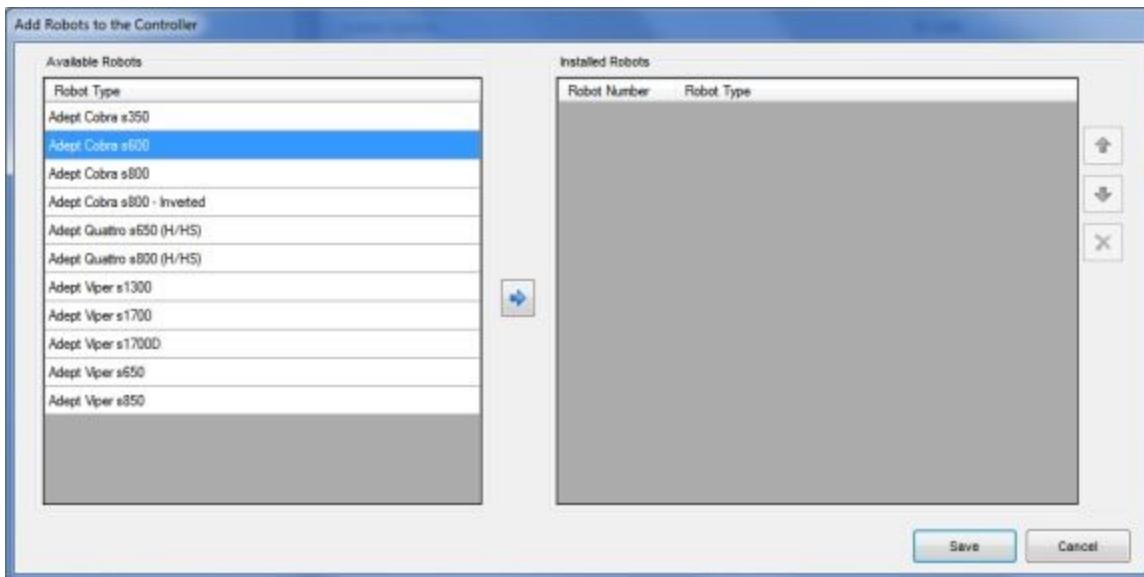
**NOTE:** If any user tasks are running, you will see the following notification dialog. Click **Yes** to stop the tasks. After all V+ user tasks are stopped, the wizard will proceed.



*Abort V+User Tasks Dialog*

*Create Controller Robots (Instructions)*

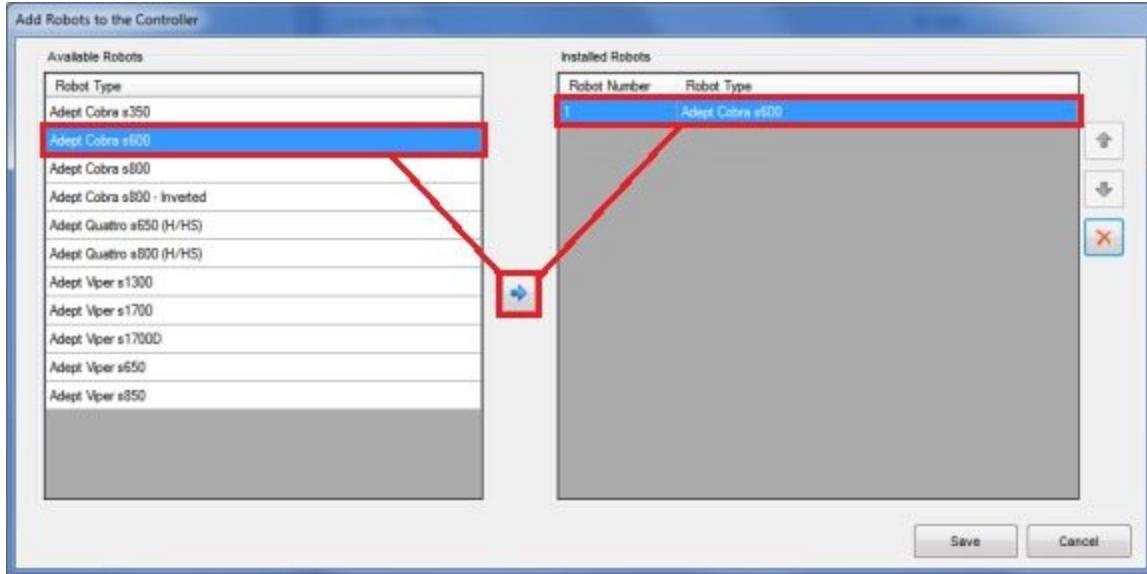
2. The following dialog opens, which allows you to select one or more robots for the application.



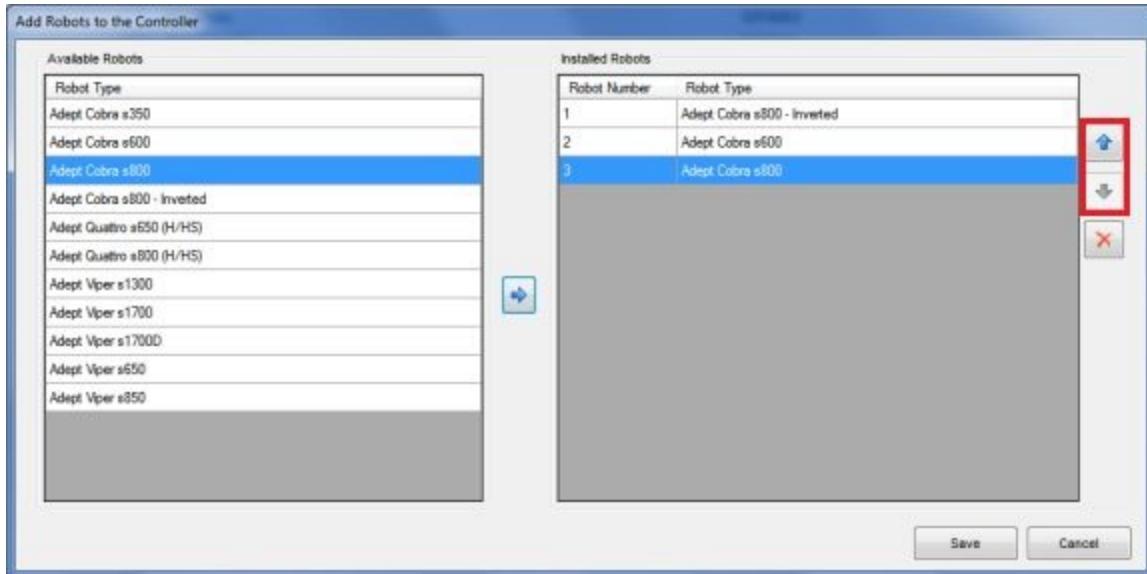
## Robot Installation

---

3. Select a robot from the Available Robots list and then click the blue arrow to move it to the Installed Robots list. Optionally, you can double-click the desired robot to add it to the Installed Robots list. Repeat this process for each robot you wish to add to the application.



*Selecting a Robot*



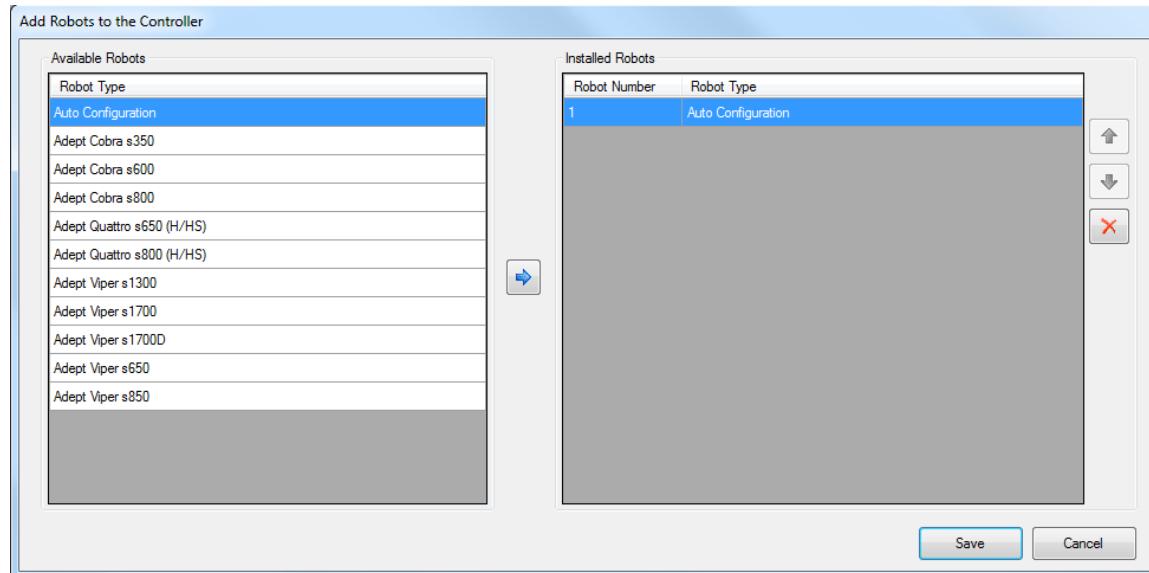
*Selecting Multiple Robots*

- To change the order of the Installed Robots list, select a robot and then use the up/down arrows to move that robot to the desired position on the list.
  - To remove a robot from the Installed Robots list, select the robot and then click the delete (X) icon.
4. Click Save to complete the robot-selection process.

## Adding a Robot With Auto Configuration

**NOTE:** This feature is only available on SmartController EX systems.

Beginning with ACE version 3.3.x, a new feature was added, called "autoconf" (Auto Configuration), which automatically configures a robot attached to the 1394 (FireWire) network. This feature is automatically enabled on all SmartController EX systems that ship from the factory. The Auto Configuration robot type is the first item in the Available Robots list, as shown in the following figure.



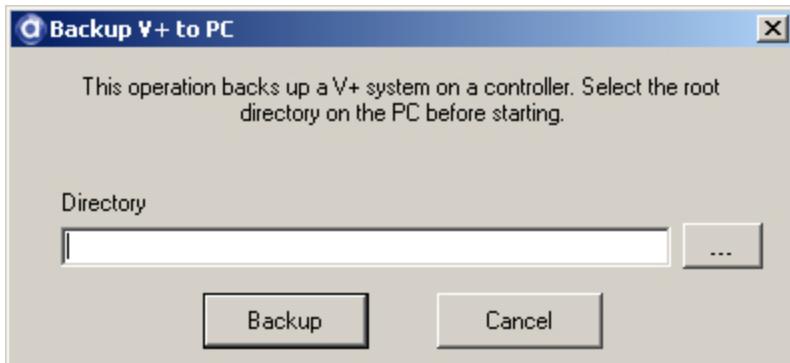
### Auto Configuration Robot Type

The Auto Configuration feature takes the convenience of the Add Robots to the Controller dialog one step further, by automatically detecting the product type of the connected robot(s) rather than you having to select the proper robot(s) from the Robot Type list.

## V+ Backup

The V+ backup utility allows you to back up the V+ operating system and key V+ directories. To perform a V+ backup:

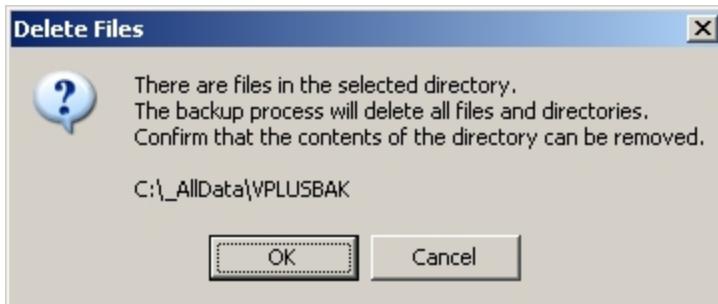
1. On the [Controller Editor](#), click **Backup/Restore > Backup V+** and then click **Next**. The Backup V+ System dialog box opens.



*Backup V+ System Dialog*

2. In the Directory text box, specify the directory into which the existing V+ system will be backed up. You can use the browse button (...) next to the text box to locate the directory on your PC.
3. Click **Backup**. The entire process will take several minutes.

If there are files present in the selected directory, the following message is displayed:



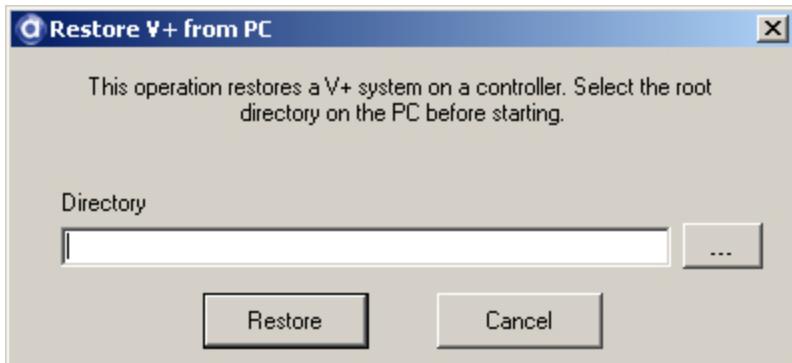
Click **OK** to continue, the existing files will be deleted. Or, click **Cancel** to select a different folder for storing the backup.

---

## V+ Restore

The V+ restore utility allows you to install a "plain vanilla" V+ distribution or to restore a backed-up V+ installation. To restore the V+ system:

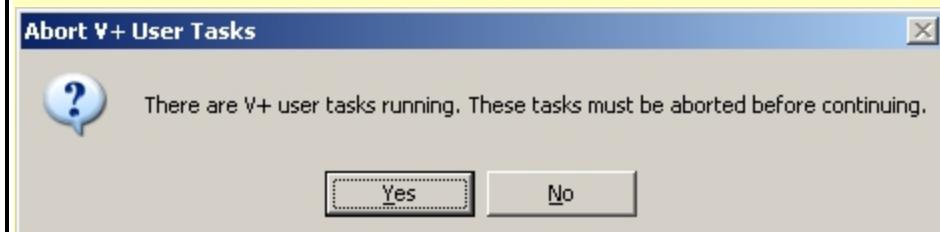
1. On the [Controller Editor](#), click **Backup/Restore > Restore V+** and then click **Next**. The Restore V+ System dialog box opens.



*Restore V+ System Dialog*

2. In the Directory text box, specify the directory on the PC where the V+ system is stored. You can use the browse button (...) next to the text box to locate the directory on your PC.
3. Click **Restore**. The entire process will take several minutes.

**NOTE:** If any user tasks are running, you will see the following notification dialog. Click **Yes** to stop the tasks. After all V+ user tasks are stopped, the wizard will proceed.



*Abort V+User Tasks Dialog*

## DeviceNet Configuration

**NOTE:** When emulation mode is enabled, this item is not available. For more details on emulation mode, see Emulation Mode on page 1089.

The DeviceNet nodes are scanned using the DeviceNet scanner. For details, see DeviceNet Scan on page 193.

After you have scanned the DeviceNet nodes, you can use the information from the scanning process to configure the components in V+.

**NOTE:** The examples presented here are based on the WAGO I/O System 750-306 as a slave on the DeviceNet bus. If you use other DeviceNet equipment, you must ask your DeviceNet vendor for the correct process image of your components.

The following figure shows a typical configuration for the mapping of three digital inputs and one output on the DeviceNet bus to V+ input signals 1033, 1034, and 1035 and V+ output signal 33. The configuration is specific to the WAGO I/O System 750-306 block. The configuration statements are described in detail below.

**NOTE:** The information in the following figure was displayed using the V+ System Configuration editor (see V+ System Configuration Editor on page 221).

Section	Statements
HEADER	LOCAL = "/MACID 0 /BAUD 125K"
VPLUS	MACID 1 = "/INPUT 6 /OUTPUT 5 /VENDOR_ID 40 /DEVICE_TYPE 12 /PRODUCT_CODE 306 /DISABLE_POWER NO"
ANALOG_INPUT	MACID 2 = "/INPUT 6 /OUTPUT 5 /VENDOR_ID 40 /DEVICE_TYPE 12 /PRODUCT_CODE 306 /DISABLE_POWER YES"
ANALOG_OUTPUT	MAPPING 1 = "/MACID 1 /BYTE 2 /BIT 2 /BIT_LENGTH 1 /SIGNAL 33"
DIGITAL_INPUT	MAPPING 2 = "/MACID 1 /BYTE 2 /BIT 2 /BIT_LENGTH 3 /SIGNAL 1033"
DIGITAL_OUTPUT	
DEVICENET	
NETWORK	

### Sample DeviceNet Configuration for Digital Input and Output

To modify the DeviceNet configuration data:

1. Open the Controller object editor
2. Click Configure
3. Select the Configure DeviceNet option
4. Click Next
5. Click Scan to scan for DeviceNet nodes
6. Select the desired nodes
7. Click Add to add those nodes to the DeviceNet configuration.

If the DeviceNet statements require changes, you can edit them through the Add/Edit Statement tool. For details, see Adding, Editing, or Removing a Statement on page 222.

The following sections provide details on the DeviceNet statements and their parameters.

## LOCAL Statement

The LOCAL statement in the DeviceNet configuration specifies the MAC ID of the Adept controller on the DeviceNet bus. The default setting is 0. Set the MAC ID so that all the nodes on the bus have different MAC IDs.

This statement also defines the baud rate of the DeviceNet scanner. The baud rate depends on multiple factors, such as the length of the DeviceNet cable, the DeviceNet components on the bus, etc.

Syntax of the LOCAL statement:

```
LOCAL = "/MACID local_id  
/BAUD baud_rate"
```

Parameter	Description	Range
local_id	MACID for the Adept controller on the bus	0 - 63
baud_rate	The baud rate to be used on the DeviceNet	125K, 250K, or 500K

## MACID Statement

The MACID statement defines the parameters for a node on the bus. The DeviceNet scanner uses this information to communicate with each node. There must be a MACID statement for every node (for example, each I/O coupler) that the V+ system is to access.

Except for the /DISABLE\_POWER parameter, which you can set as you wish, the values for all the parameters should be taken directly from the results of scanning the DeviceNet.

Syntax of the MACID statement:

```
MACID node_macid = "/INPUT in_bytes
                      /OUTPUT out_bytes
                      /VENDOR_ID vend_num
                      /DEVICE_TYPE dev_num
                      /PRODUCT_CODE prod_num
                      /DISABLE_POWER pow_switch"
```

Parameter	Description
node_macid	MACID number of the node on the bus (0 to 63). 0 is the default; -1 will cancel. <b>NOTE:</b> Adept recommends that you not use 63 as a MACID, because some manufacturers use this as a general broadcast node.
in_bytes	Number of data bytes of input from the node (0 to 256)
out_bytes	Number of data bytes of output to the node (0 to 256)
vend_num	Vendor identification number, assigned by ODVA. Use the number that the scanning process found.
dev_num	Device number of the node. Specifies the type of the node, for instance an I/O block.
prod_num	Specified by the DeviceNet vendor.
pow_switch	Either YES or NO. The default value is NO. If set to YES, robot power cannot be enabled if the specified node is offline, and robot power is turned off if the node ever goes offline.

**NOTE:** The values of the /VENDOR\_ID, /DEVICE\_TYPE, and /PRODUCT\_CODE parameters are not currently used by the V+ system.

## MAPPING Statement

The MAPPING statement is used to map the digital inputs and outputs of a node to the appropriate V+ signals. This part of the DeviceNet setup requires that you define the way the connected nodes organize their

I/O data in messages to and from V+. Refer to the documentation of your DeviceNet vendor for this information.

Syntax of the MAPPING statement:

```
MAPPING map_index = "/MACID node_macid  
/BYTE byte_num  
/BIT bit_num  
/BIT_LENGTH bit_len  
/SIGNAL sig_num"
```

Parameter	Description
map_index	This is an integer that is used as an index for mapping DeviceNet information to V+ (1 to 100).
node_macid	MACID number of a selected node on the bus (0 to 63).
byte_num	Start reading/writing data at the specified byte number (1 to 32; for V+ version 16.0 and later running on SmartController systems, using CONFIG_C version 16.4 edit B4 or later, the allowed values are 1 to 64).
bit_num	Start reading/writing information at the specified bit number in the specified byte (1 to 256; for V+ version 16.0 and later running on SmartController systems, using CONFIG_C version 16.4 edit B4 or later, the allowed values are 1 to 512).
bit_len	Number of bits to read/write (1 to 256; for V+ version 16.0 and later running on SmartController systems, using CONFIG_C version 16.4 edit B4 or later, the allowed values are 1 to 512).
sig_num	Starting number of a range of V+ signals that receive/transmit status information from/to the selected node. The range is specified by the value of bit_len.  This number establishes whether the bits being mapped are for input or output. If the number is for a V+ input signal (1001 through 1512), the bits are used for inputs. If the number is for a V+ output signal (1 through 512), the bits are used for outputs.

1. First, select the DeviceNet component to which to write information with the parameter /MACID in the MAPPING statement.

The following example shows that V+ writes information to /MACID 1, which is the WAGO I/O block that was previously configured with the MACID 1 statement.

2. The /BYTE parameter defines the start byte of the DeviceNet Output stream to the I/O block.
3. After defining the start byte of the data stream, the /BIT parameter specifies the first bit that is

mapped to a V+ signal.

The /BIT\_LENGTH parameter defines how many bits to map to V+ signal numbers. In the MAPPING 1 statement the parameter /BIT\_LENGTH has the value 1. This means that exactly one bit is mapped to one signal, starting with the signal number given in the /SIGNAL parameter.

4. The /SIGNAL parameter specifies the starting number of a series of V+ signals.

The following example shows the mapping of V+ signals 33, 1033, 1034, and 1035 to digital I/Os on the DeviceNet.

- The MAPPING 1 statement maps one digital output on the WAGO I/O block to one digital I/O signal in V+.
- In the MAPPING 2 statement the parameter /BIT\_LENGTH has the value 3, which means that three signals, starting with signal 1033, are mapped.

Current statements in the DEVICENET section:

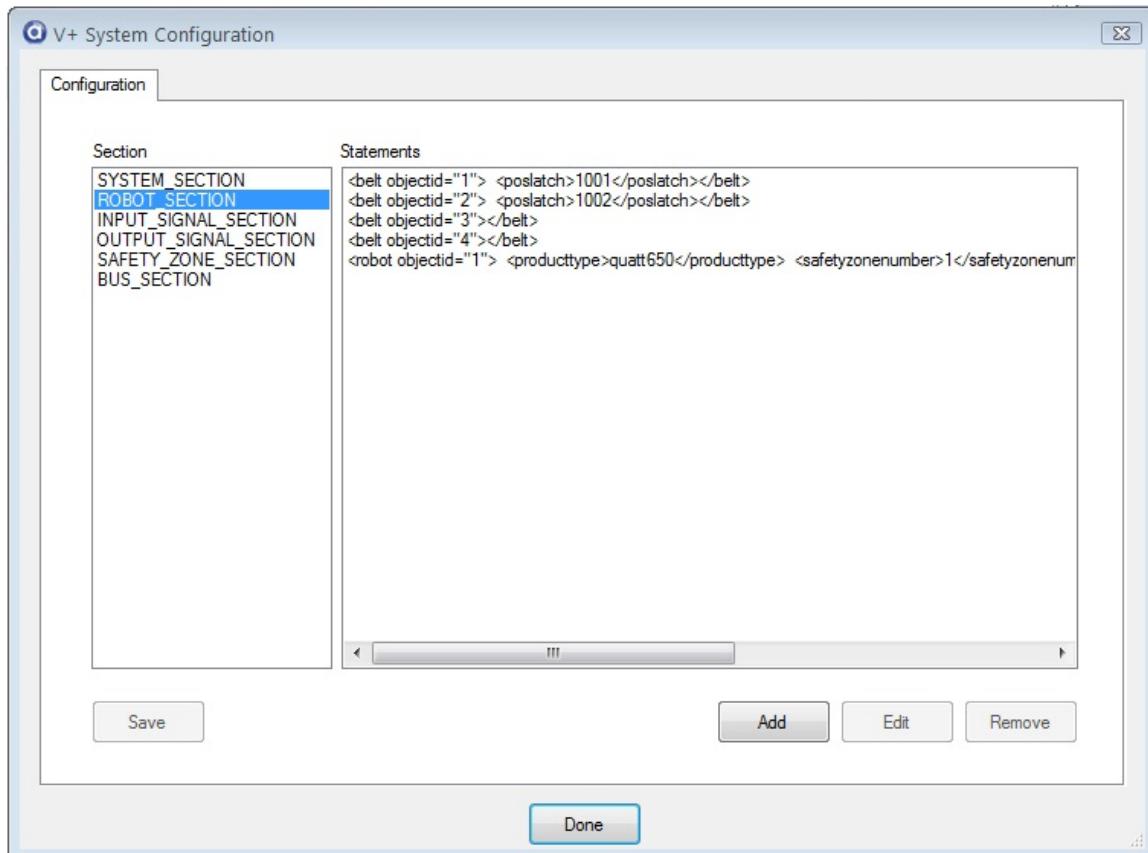
```
MAPPING 1 = "/MACID 1 /BYTE 2 /BIT 2 /BIT_LENGTH 1 /SIGNAL 33"  
MAPPING 2 = "/MACID 1 /BYTE 2 /BIT 2 /BIT_LENGTH 3 /SIGNAL 1033"
```

The information about how many bytes V+ reads from the DeviceNet bus and how many it writes is specified in the MACID statement. These values are typically found while scanning the DeviceNet bus with the DeviceNet scanner.

For more information on DeviceNet, see the DeviceNet topic in the *V+ Language User's Guide*.

## V+ System Configuration Editor

The Configuration tab opens the V+ System Configuration editor (see the following figure), which provides a user-friendly interface for editing the contents of the V+ configuration file.



*V+ System Configuration Editor - Section/Statement Selection*

The following table describes the items on the main editor window.

Item	Description
Section	Lists the sections of the V+ configuration file. Select a section to view the statements within that section. For a description of the sections, see V+ System Configuration Editor on page 221.
Statements	Lists the statements for the selected section of the V+ configuration file. Use the Add, Edit, and Remove buttons to add, edit, or remove (delete) statements. For a description of the statements, see V+ System Configuration Editor on page 221.

Item	Description
Add	Opens a dialog for adding a new statement to the selected section. For details, see the section Adding, Editing, or Removing a Statement on page 222.
Edit	Opens a dialog for editing the selected statement. For details, see the section Adding, Editing, or Removing a Statement on page 222.
Remove	Opens a dialog for deleting the selected statement. For details, see the section Adding, Editing, or Removing a Statement on page 222.
Save	Saves the current changes. At least one item must be changed to activate the Save button. For details, see the section Saving the Configuration File on page 226.
Done	Closes the V+ System Configuration window.

## Adding, Editing, or Removing a Statement

**NOTE:** After any changes are made (add, edit, or delete statements) to the configuration file, it must be saved. For details, see the section Saving the Configuration File on page 226.

To add a statement:

1. Select an item in the Section group and then click **Add**. The Add Statement dialog opens, as shown in the following figure.
2. Select the Statement Type
3. Set the item values. As the item values are changed, the complete statement is shown in the Composed Statement area. If an invalid value is entered, an error message will be displayed when **Accept** is clicked.
4. Click **Accept** to save the new statement; or, click **Cancel** to close the editor without saving.

To edit a statement:

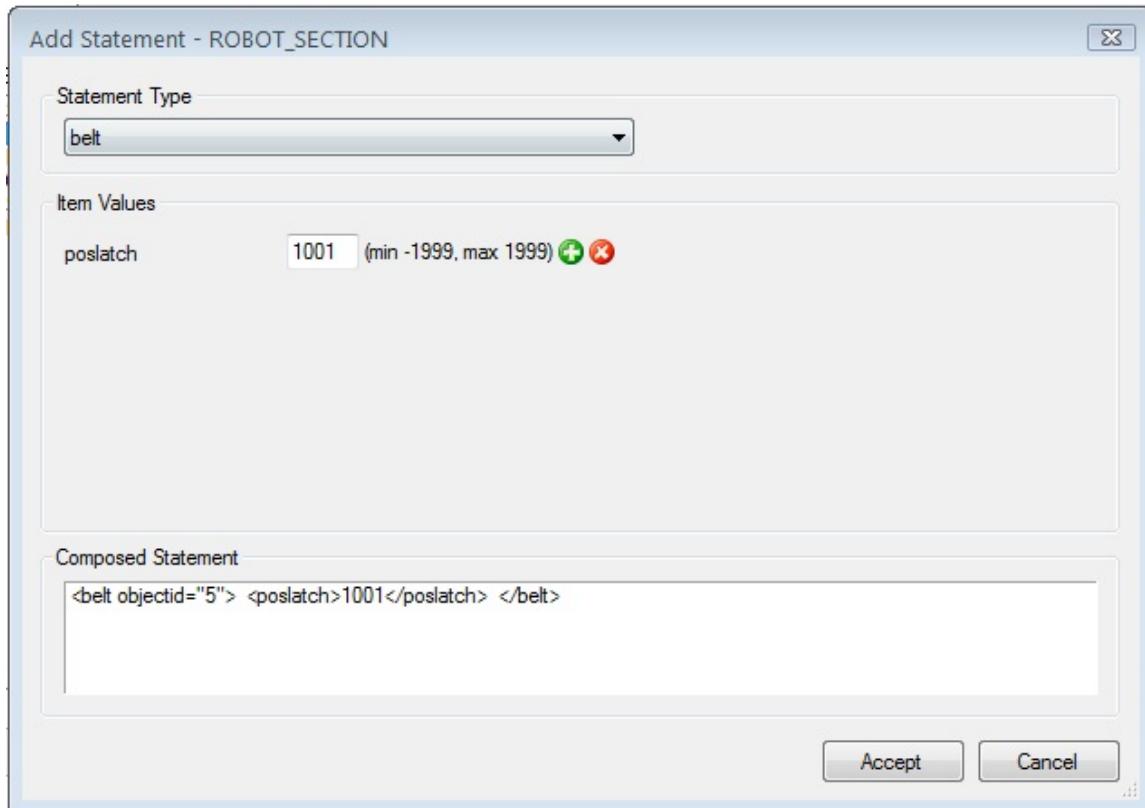
1. Select an item in the Section group, select an item in the Statements group, and then click **Edit**. The Edit Statement dialog opens (it is similar to the dialog shown in the following figure; however, the Statement Type selector will be dimmed).
2. Edit the item values. As the item values are changed, the complete statement is shown in the Composed Statement area. If an invalid value is entered, an error message will be displayed when **Accept** is clicked.
3. Click **Accept** to save the new statement; or, click **Cancel** to close the editor without saving.

To remove (delete) a statement:

## Adding, Editing, or Removing a Statement

---

1. Select an item in the Section group
2. Select an item in the Statements group
3. Click **Remove**. The selected Statement is removed (deleted).



### V+ Configuration File — Add/Edit Statement

The Add/Edit Statement window presents configuration items in the statement along with their editing rules such as their value ranges and valid enumeration options.

The following table describes the items on the Add/Edit Statement editor window.

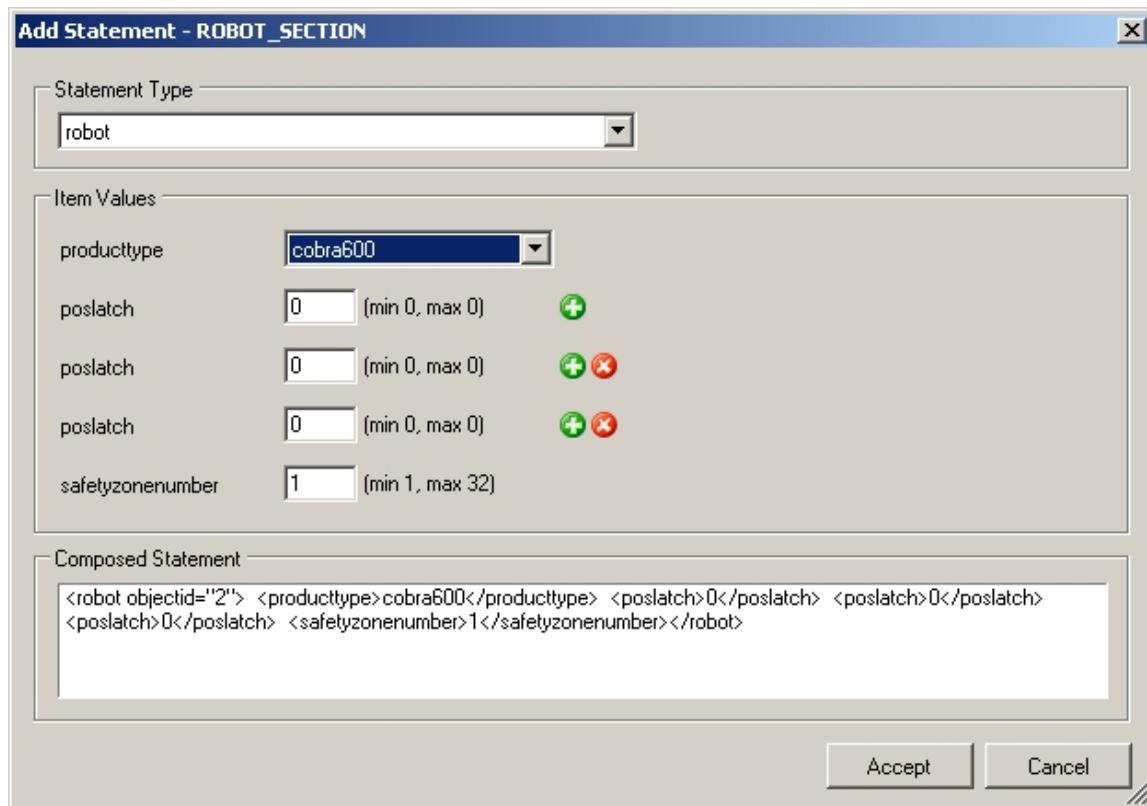
Item	Description
Statement Type	Selects the statement type.
Item Values	Contains all the available values for the selected statement type. Each item contains a selector with pre-set values, or a field with the allowed values/ranges shown. Invalid inputs (for fields) are flagged with a red exclamation icon. For details, see Invalid Inputs on page 225.

Item	Description
Composed Statement	Provides a preview of the statement, based on the current settings in the Item Values group.
Accept	Stores the changes. Note that the configuration file is not saved to the controller until the Save button (on the main screen) is clicked. For details, see Saving the Configuration File on page 226.
Cancel	Cancels the Add/Edit operation.

## Adding Multiple Item Values Within a Statement

**NOTE:** This feature only applies to eV+ systems and emulated systems.

For certain statement types (for example, the ROBOT statement), the Add/Edit statement window allows you to add multiple item values, as shown in the following figure (note the multiple "poslatch" items).



V+ Configuration File — Add Multiple Item Values

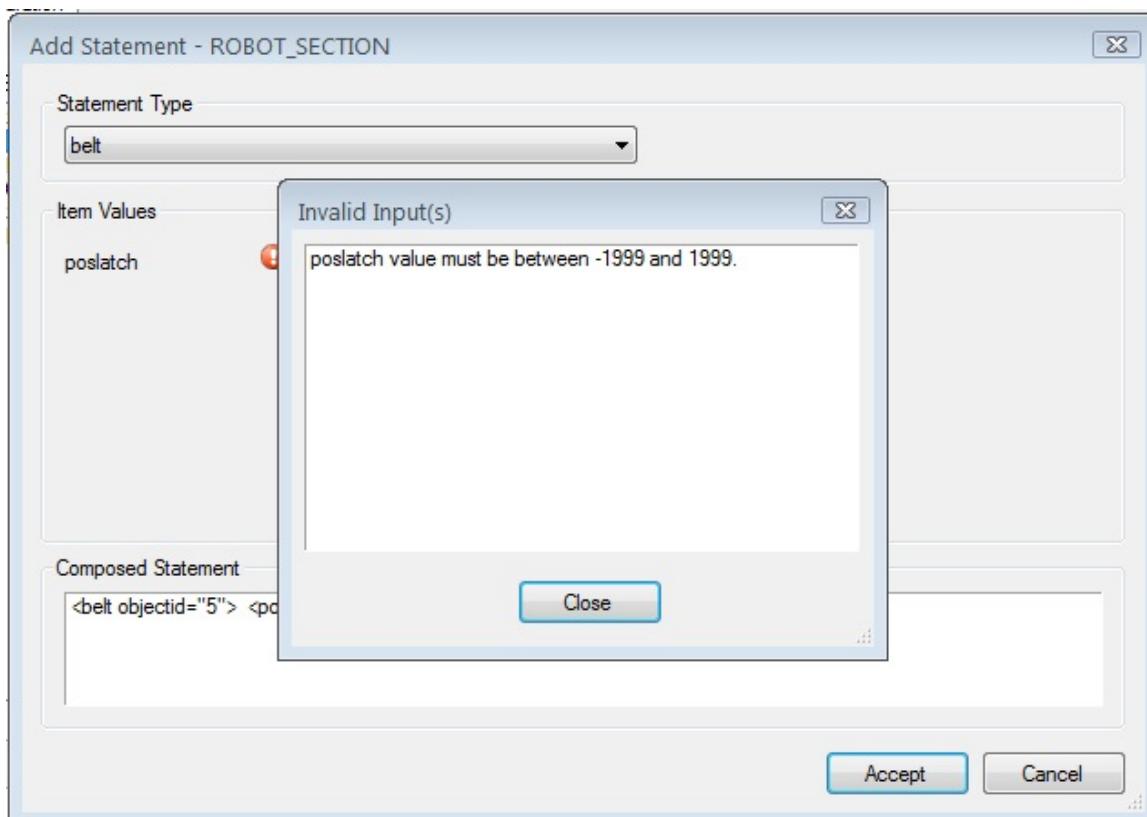
- To add another value of an item, click the Add (+) icon.
- To remove a value, click the Delete (X) icon.

The Composed Statement area automatically updates as values are added or deleted.

### Invalid Inputs

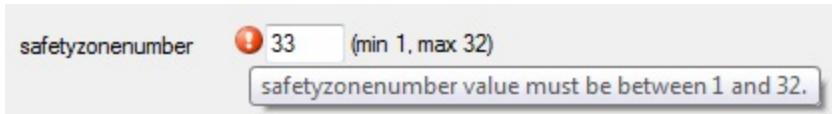
Invalid inputs are identified through two methods:

- When you click **Accept**, any invalid inputs are listed in the Invalid Input(s) window, as shown in the following figure.



V+ Configuration File — Add/Edit Statement

- A red exclamation icon (!) is placed next to each invalid input.  
After the Invalid Input(s) window is closed, hover messages can be displayed, by placing the mouse over the red exclamation icon, as a reminder about why a particular input is invalid, as shown in the following figure.



*V+ Configuration File — Hover message showing error*

## Saving the Configuration File

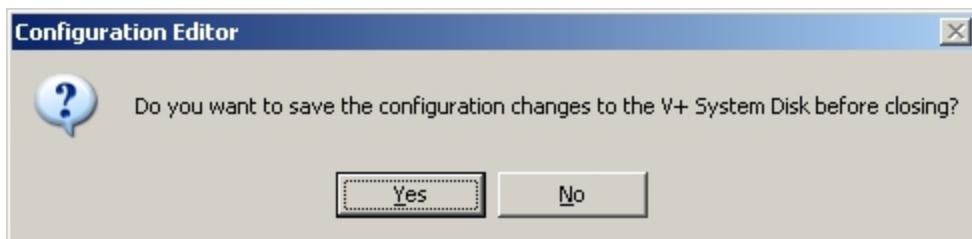
After any changes are made to the configuration, the Save button is enabled, as shown in the following figure. To save the configuration file, click **Save**. The new configuration file is saved to the current controller.

**NOTE:** The Save button is not enabled until there is at least one change in the configuration.



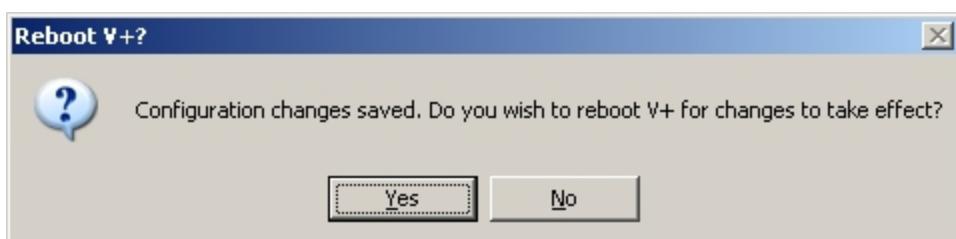
*Save button enabled*

If you attempt to exit without saving, a warning prompt is displayed. Click **Yes** to save the configuration changes; click **No** to exit without saving.



After the configuration is saved to the controller, the controller must be rebooted to apply the new configuration. A prompt is displayed that allows you to reboot the controller directly from the ACE software interface (without powering off the controller).

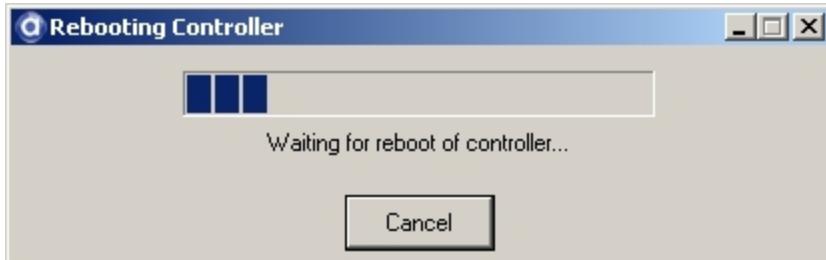
**NOTE:** When emulation mode is enabled, the emulated controller is not restarted after making configuration changes, it is simply reset.



## Saving the Configuration File

---

Click **Yes** to proceed. The controller reboots. A progress indicator shows the status of the reboot process.



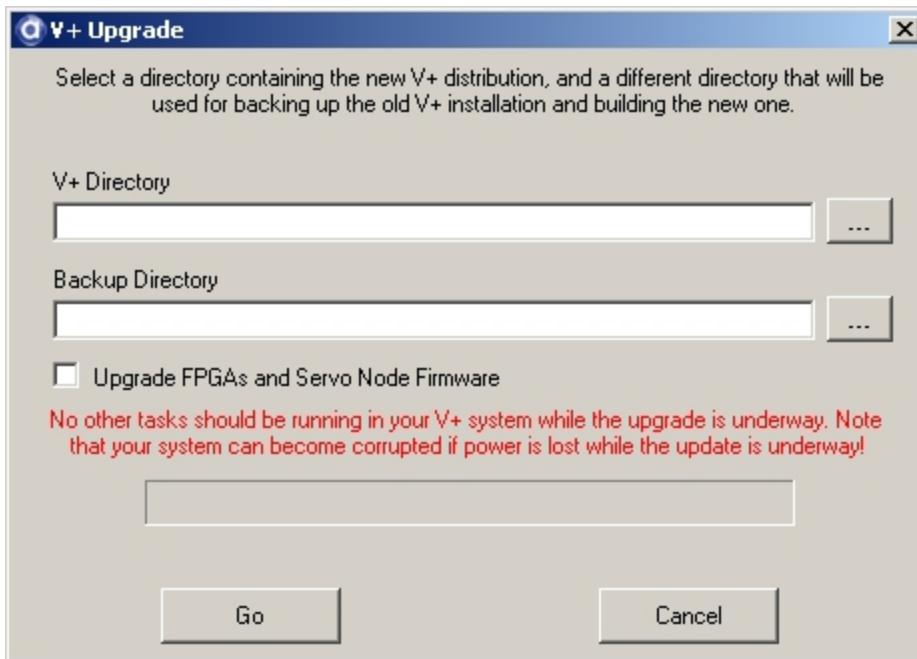
## V+ Upgrade

The V+ Upgrade dialog box, shown in the following figure, is used to upgrade the V+ operating system and, optionally, to upgrade the FPGA and Servo Node firmware.

**NOTE:** The FireWire and FPGA firmware can also be upgraded through a separate operation. For details, see [Upgrading FireWire Node Firmware](#).

To upgrade the V+ operating system:

1. Copy the new V+ directory from the distribution media to the PC that is connected to the controller you wish to upgrade.
2. On the [Controller Editor](#), click **Upgrade > Upgrade V+** and then click **Next**. The V+ Upgrade dialog box opens.



*V+ Upgrade Dialog*

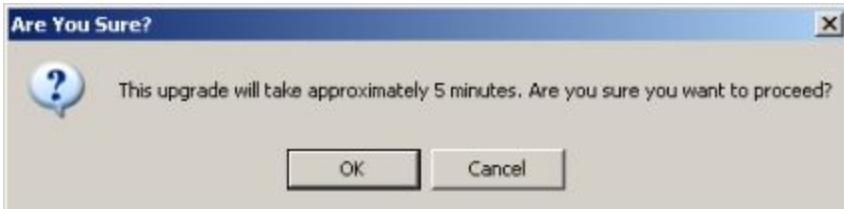
3. In the V+ Directory text box, specify the directory that you copied from the distribution media. You can use the browse button (...) next to the text box to locate the directory on your PC.
4. In the Backup Directory text box, specify the directory into which the existing V+ system will be backed up before the upgrade. You can use the browse button (...) next to the text box to locate the directory on your PC.

**NOTE:** If V+ Backup operation fails, an error is displayed and the upgrade process is automatically stopped. This prevents overwriting the new V+ system with corrupt data.

5. If you wish to upgrade your FPGA and Servo Node firmware, select the Upgrade FPGAs and Servo Node Firmware check box. This option upgrades all programmable devices on each node on the distributed network.

**NOTE:** The firmware upgrade will require several minutes for each node in the distributed network.

6. Click **Start**. A confirmation dialog box opens, which displays the estimated time to complete the upgrade operation.



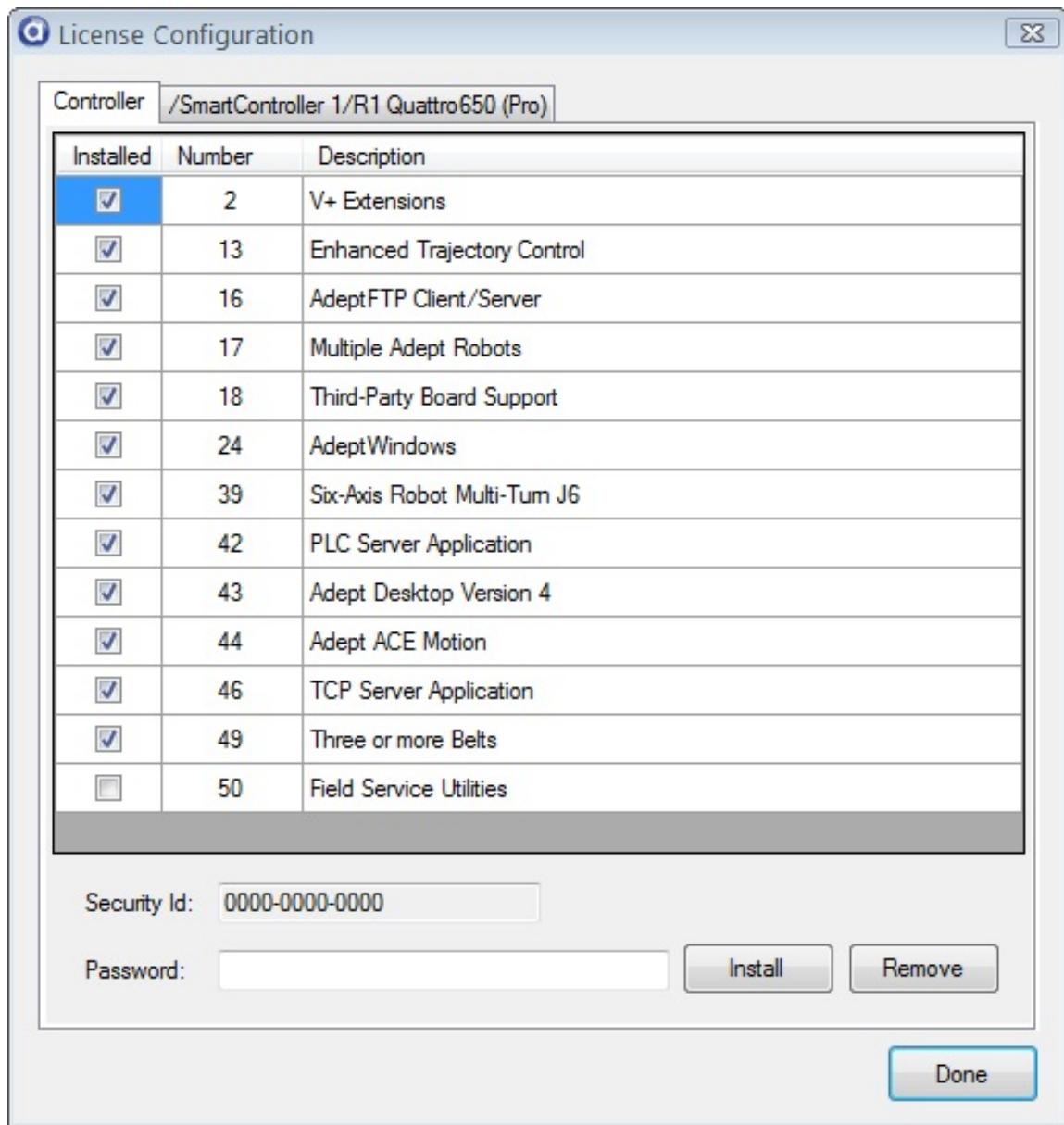
The upgrade will take several minutes. Additional time will be required if the Upgrade FPGAs and Servo Node Firmware option was selected, as described above.

## V+ License Configuration

### **Licenses**

The V+ licenses can be accessed through the **Configure > Configure Licenses** option in the controller editor. This tool allows you to view all available licenses, view the installed licenses (checked items), and add or remove licenses.

**NOTE:** For license management on the Cobra i600/i800 (Cobra i-Series) robot, see the topic License Manager on page 296.



### License Manager

When using the license manager, you can:

- Select a controller or robot you want to view the licenses for. All licenses will be valid for a specific security Id.
- Click **Install** to install a license. A password is required for this operation, which must be obtained

from Adept Customer Service.

- Click **Remove** to remove a license. A password is required for this operation, which must be obtained from Adept Customer Service.

## Robot and Motor Configuration

The topics in this chapter describe the tools used to configure robots and motors. If you are configuring the Adept Cobra i600 or i800 robot, please see [Cobra i600/i800 Robot Configuration](#) on page 265.

There are thousands of parameters that must be defined for proper robot operation. If you are putting together a custom robot systems using Adept components (such as Adept SmartMotion), the tools described in this section will be used to configure the robots and motors. For standard Adept robots, the required parameters have already been set at the factory.

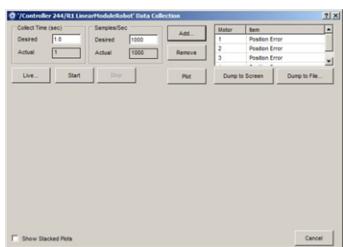
Collecting Data .....	234
Configuring Axes, Options, and Kinematic Parameters .....	243
Configuring S-Curve Profiles .....	246
Editing Obstacles .....	249
Loading and Saving Robot/Motor Specification Files .....	251
Safety Settings .....	254
Tuning the Motors .....	256
Using a Tool Offset .....	261

## Collecting Data

**NOTE:** When emulation mode is enabled, this item is not available. For more details on emulation mode, see Emulation Mode on page 1089.

The Data Collection dialog box, shown in the following figure, is used to collect values on the servo node at up to the servo rate of 8 KHz. You can collect data on up to 8 items (at rates of up to 8000 samples per second), up to the memory limit of the robot.

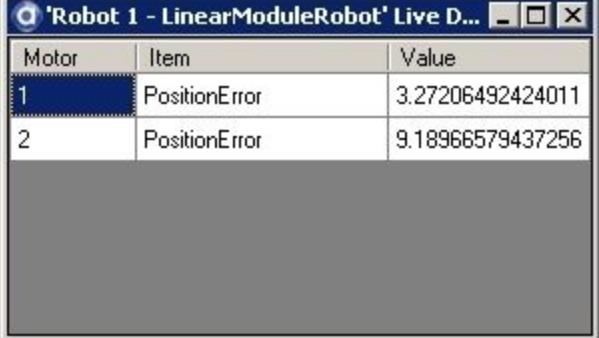
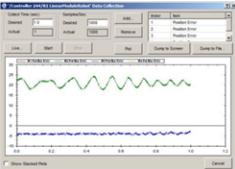
To open the Data Collection dialog box, on the [Robot Editor](#) menu, click **Control > Data Collection**. The Data Collection dialog box opens.



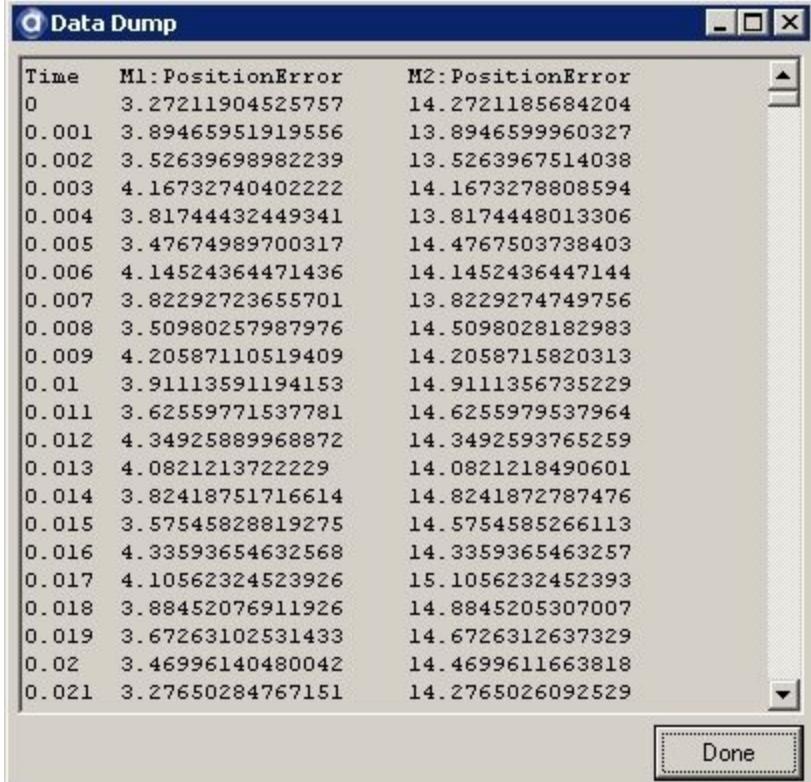
*Data Collection Dialog*

This section describes the items in the Data Collection dialog box:

Item	Description
Collect Time (sec)	Specifies the data collection time in seconds. The default value is 1. For example, to collect data for five seconds, enter the value 5.
Samples/Sec	Specifies the data collection rate in samples per second. The default value is 1000. For example, to collect data at a sample rate of 1600 samples/sec, enter the value 1600.
Live...	Displays a window that shows the "live" data being collected, as shown in the following figure.

Item	Description
	 <p><i>Live Data Display</i></p>
Start	Click to start the data collection. The data collection will continue until either: Stop is clicked, or the specified collect time is reached.
Stop	Click to stop the data collection. If the specified collect time has already expired, this button is disabled.
Plot	<p>Click to plot the collected data. A progress bar displays while the data is processed. After the processing has completed, the data is plotted on the graph located at the lower portion of the Data Collection dialog box, as shown in the following figure.</p>  <p><i>Data Collection Plot</i></p> <p>If Plot is clicked but no data has been collected, an error message is displayed. You must click Start and then run the robot to generate data, before attempting to plot the data.</p> <p>You can zoom (magnify) an area of the graph by dragging the mouse pointer to define the zoom area.</p> <p>You can pan the graph by holding the Ctrl key while dragging the mouse pointer.</p> <p>You can right-click on the graph to display a menu of options, which are described in the following table.</p>

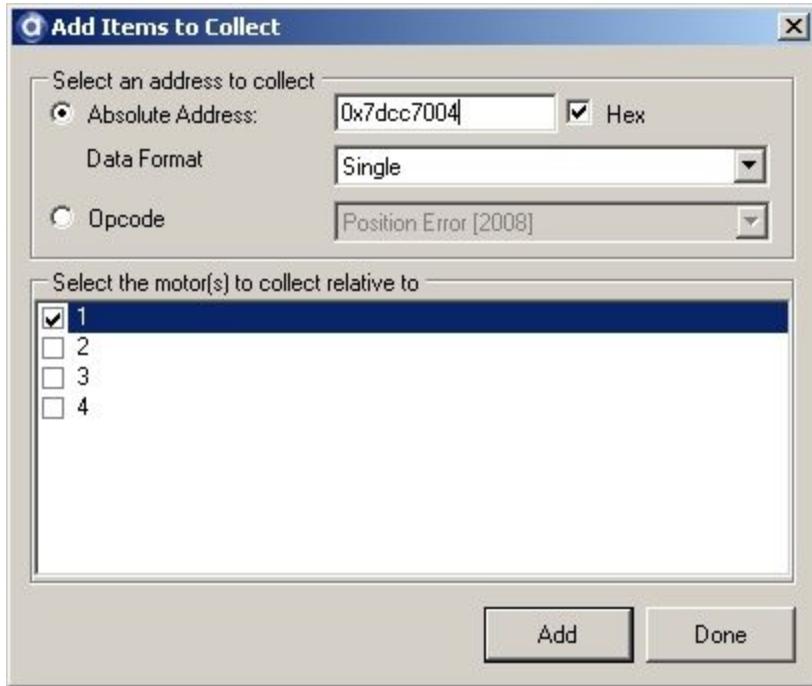
<b>Item</b>	<b>Description</b>	
	<b>Item</b>	<b>Description</b>
	Copy	Copy the image in the graph area.
	Save Image As...	Save the graph image as a file.
	Page Setup...	Opens the Page Setup dialog box for setting the print page size, orientation, and margins.
	Print...	Opens the Print dialog box for printing the graph.
	Show Point Values	When selected, place the cursor at a point on the graph to show the values of that point.
	Un-Zoom	Returns the graph to its original magnification.
	Undo All Zoom/Pan	Returns the graph to its original magnification and origin.
	Set Scale to Default	Returns the graph scale to its default setting.
Dump to Screen	Displays the collected data in a Data Dump window in text-file format, as shown in the following figure.	

Item	Description																																																																					
	 <p>The screenshot shows a window titled "Data Dump" with a table of data. The columns are labeled "Time", "M1:PositionError", and "M2:PositionError". The data consists of approximately 40 rows of timestamped position error values.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Time</th> <th>M1:PositionError</th> <th>M2:PositionError</th> </tr> </thead> <tbody> <tr><td>0</td><td>3.27211904525757</td><td>14.2721185684204</td></tr> <tr><td>0.001</td><td>3.89465951919556</td><td>13.8946599960327</td></tr> <tr><td>0.002</td><td>3.52639698982239</td><td>13.5263967514038</td></tr> <tr><td>0.003</td><td>4.16732740402222</td><td>14.1673278808594</td></tr> <tr><td>0.004</td><td>3.81744432449341</td><td>13.8174448013306</td></tr> <tr><td>0.005</td><td>3.47674989700317</td><td>14.4767503738403</td></tr> <tr><td>0.006</td><td>4.14524364471436</td><td>14.1452436447144</td></tr> <tr><td>0.007</td><td>3.82292723655701</td><td>13.8229274749756</td></tr> <tr><td>0.008</td><td>3.50980257987976</td><td>14.5098028182983</td></tr> <tr><td>0.009</td><td>4.20587110519409</td><td>14.2058715820313</td></tr> <tr><td>0.01</td><td>3.91113591194153</td><td>14.9111356735229</td></tr> <tr><td>0.011</td><td>3.62559771537781</td><td>14.6255979537964</td></tr> <tr><td>0.012</td><td>4.34925889968872</td><td>14.3492593765259</td></tr> <tr><td>0.013</td><td>4.0821213722229</td><td>14.0821218490601</td></tr> <tr><td>0.014</td><td>3.82418751716614</td><td>14.8241872787476</td></tr> <tr><td>0.015</td><td>3.57545828819275</td><td>14.5754585266113</td></tr> <tr><td>0.016</td><td>4.33593654632568</td><td>14.3359365463257</td></tr> <tr><td>0.017</td><td>4.10562324523926</td><td>15.1056232452393</td></tr> <tr><td>0.018</td><td>3.88452076911926</td><td>14.8845205307007</td></tr> <tr><td>0.019</td><td>3.67263102531433</td><td>14.6726312637329</td></tr> <tr><td>0.02</td><td>3.46996140480042</td><td>14.4699611663818</td></tr> <tr><td>0.021</td><td>3.27650284767151</td><td>14.2765026092529</td></tr> </tbody> </table> <p style="text-align: right;"><b>Done</b></p> <p><i>Data Dump Window</i></p>	Time	M1:PositionError	M2:PositionError	0	3.27211904525757	14.2721185684204	0.001	3.89465951919556	13.8946599960327	0.002	3.52639698982239	13.5263967514038	0.003	4.16732740402222	14.1673278808594	0.004	3.81744432449341	13.8174448013306	0.005	3.47674989700317	14.4767503738403	0.006	4.14524364471436	14.1452436447144	0.007	3.82292723655701	13.8229274749756	0.008	3.50980257987976	14.5098028182983	0.009	4.20587110519409	14.2058715820313	0.01	3.91113591194153	14.9111356735229	0.011	3.62559771537781	14.6255979537964	0.012	4.34925889968872	14.3492593765259	0.013	4.0821213722229	14.0821218490601	0.014	3.82418751716614	14.8241872787476	0.015	3.57545828819275	14.5754585266113	0.016	4.33593654632568	14.3359365463257	0.017	4.10562324523926	15.1056232452393	0.018	3.88452076911926	14.8845205307007	0.019	3.67263102531433	14.6726312637329	0.02	3.46996140480042	14.4699611663818	0.021	3.27650284767151	14.2765026092529
Time	M1:PositionError	M2:PositionError																																																																				
0	3.27211904525757	14.2721185684204																																																																				
0.001	3.89465951919556	13.8946599960327																																																																				
0.002	3.52639698982239	13.5263967514038																																																																				
0.003	4.16732740402222	14.1673278808594																																																																				
0.004	3.81744432449341	13.8174448013306																																																																				
0.005	3.47674989700317	14.4767503738403																																																																				
0.006	4.14524364471436	14.1452436447144																																																																				
0.007	3.82292723655701	13.8229274749756																																																																				
0.008	3.50980257987976	14.5098028182983																																																																				
0.009	4.20587110519409	14.2058715820313																																																																				
0.01	3.91113591194153	14.9111356735229																																																																				
0.011	3.62559771537781	14.6255979537964																																																																				
0.012	4.34925889968872	14.3492593765259																																																																				
0.013	4.0821213722229	14.0821218490601																																																																				
0.014	3.82418751716614	14.8241872787476																																																																				
0.015	3.57545828819275	14.5754585266113																																																																				
0.016	4.33593654632568	14.3359365463257																																																																				
0.017	4.10562324523926	15.1056232452393																																																																				
0.018	3.88452076911926	14.8845205307007																																																																				
0.019	3.67263102531433	14.6726312637329																																																																				
0.02	3.46996140480042	14.4699611663818																																																																				
0.021	3.27650284767151	14.2765026092529																																																																				
Dump to File...	Displays a Save As dialog box used for saving the collected data to a text file, which can be viewed or processed at a later time.																																																																					
Add	Click to add monitored items to the list. See the section Adding and Removing Monitored Items, below, for more details.																																																																					
Remove	Click to remove monitored items from the list. See the section Adding and Removing Monitored Items, below, for more details.																																																																					

### Adding and Removing Monitored Items

The Add Items to Collect dialog box is used to add items to the data collection list. Typically, for motor tuning, you will want to collect position error data on the motor receiving the square wave positioning command.

**NOTE:** You can monitor up to seven data items.



*Add Items to Collect Dialog*

To add a new item:

1. On the Motor Tuning dialog box, click **Add**. The Add Items to Collect dialog box opens (see the figure above).
2. Specify the value to be monitored:
  - Enter an Absolute Address (select the Hex check box, if you wish to enter the value in hex format) and select the Data Format from the drop-down list.  
OR
  - Select an Opcode from the drop-down list. For a description of each of the items, see the topic [Data Collection Parameters](#).
3. Select (check) the box next to the motor(s) to be added.
4. Click **Add** to add the items and close the dialog box. The items are listed in the Motor Tuning dialog box.

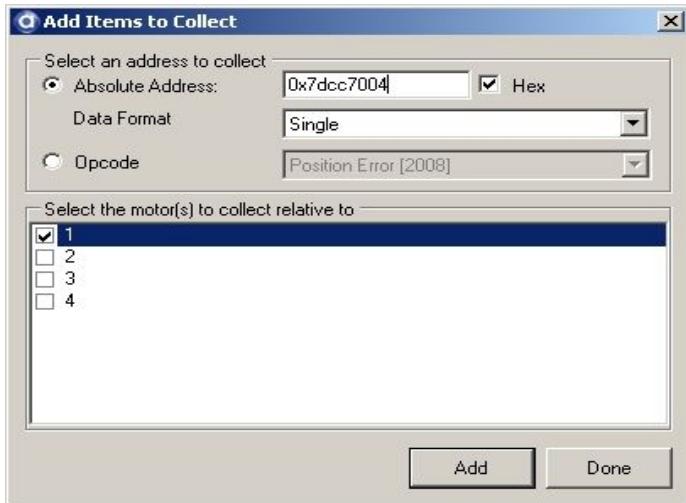
To remove an item:

1. On the Motor Tuning dialog box, select the item to be removed from the list.
2. Click **Remove**. The selected item is removed from the list.

## Data Collection Parameters

The data collection and motor tuning processes require you to select a motor and corresponding data parameter to monitor. This is done through the Add Items to Collect dialog box, shown in the following figure.

To open the Add Items to Collect dialog box, click **Add...** on the [Data Collection](#) or [Motor Tuning](#) dialog box.



### *Data Collection Parameters*

This following table describes the parameters available from the Opcode drop-down list.

Parameter	Description
Amp AC Input RMS Voltage	Monitors the RMS input voltage to amplifier (in volts) for the selected motor.
Amp Bus Voltage	Monitors the high-voltage DC bus for the selected motor.
Amp Temperature	Monitors the amplifier temperature (in degrees C) for the selected motor.
Base Board Temperature	Monitors the amplifier base-board temperature (in degrees C), per-amp not per-motor, for the selected motor.
Bus Energy Filter	Monitors the Bus Energy Filter for the selected motor.
Commanded	Monitors the commanded acceleration (in counts/ms <sup>2</sup> ) for the selected motor.

## Data Collection Parameters

---

<b>Parameter</b>	<b>Description</b>
Acceleration	
Commanded Position	Monitors the commanded position (in counts) for the selected motor.
Commanded Velocity	Monitors the commanded velocity (in counts/ms) for the selected motor.
Current Loop Output	Monitors the output of the 'PI' current loop for the selected motor.
Current Loop Peak-To-Peak Output	Monitors the peak output of the 'PI' current loop for the selected motor.
DC Input Voltage	Monitors the DC control voltage (in volts) for the selected motor.
Duty Cycle Level	Monitors the current value for the motor's duty cycle. See Duty Cycle Limit and Peak Duty Cycle.
Duty Cycle Limit	Specifies the maximum allowable value for the Duty Cycle. When Duty Cycle Level = Duty Cycle Limit, the robot is stopped to protect the motor. See Duty Cycle Level and Peak Duty Cycle.
Encoder Position	Monitors the actual position (in counts) for the selected motor.
Encoder Temperature	Monitors the encoder temperature (in degrees C) for the selected motor.
Encoder Velocity	Monitors the actual velocity, in counts/ms, for the selected motor.
E-Stop Status	Monitors the E-stop status register for the selected motor.
Index Delta	Monitors the index delta (last index position minus previous index position), in counts, for the selected motor.
Last Motion Settling Time	Monitors the motor settling time (in milliseconds) of the last motion completed for the selected motor.
Latched Errors	Monitors the latched error bit mask for the selected motor. For possible values

Parameter	Description
	and their descriptions, see the topic MotorLatchedErrorBits Enumeration in the ACE API Documentation Help file. You can access this file by selecting <b>Help &gt; ACE API Reference</b> from the ACE software menu.
Output Level	Monitors the output level, typically a torque command between -32768 and 32767, for the selected motor.
Peak Duty-Cycle	Monitors the largest Duty Cycle value that has been reached since the last controller boot. This allows you to see if the robot is approaching the Duty Cycle Limit. See Duty Cycle Level and Duty Cycle Limit.
Position Error	Monitors the position error (commanded position minus actual encoder position), in counts, for the selected motor.
Servo Status	Monitors the servo status bit mask for the selected motor. For possible values and their descriptions, see the topic MotorStatusBits Enumeration in the ACE API Documentation Help file. You can access this file by selecting <b>Help &gt; API Reference</b> from the ACE software menu.
Unlatched Errors	Monitors the unlatched error bit mask for the selected motor. For possible values and their descriptions, see the topic MotorUnlatchedErrorBits Enumeration in the ACE API Documentation Help file. You can access this file by selecting <b>Help &gt; API Reference</b> from the ACE software menu.
V+ Command	<p>V+ command code. Valid commands are:</p> <ul style="list-style-type: none"> <li>• 0: Current mode. Arg: DAC output.</li> <li>• 1: Free mode. Arg: DAC output.</li> <li>• 2: Position mode. Arg: Scaled position command.</li> <li>• 3: Set position. Arg: Scaled position.</li> <li>• 4: Amp command. Arg: 1 to enable, 0 to disable.</li> <li>• 5: Cal mode. Arg: None.</li> <li>• 7: Adjust position. Arg: Scaled position offset.</li> <li>• 8: Clear latched errors. Arg: Bit mask to clear.</li> <li>• 9: Drive motor. Arg: Scaled position.</li> <li>• 0xF: NOP. Arg: None.</li> <li>• 0x10: High power. Arg: 1 to turn on, 0 to turn off.</li> <li>• 0x11: Brake rel. Arg: 1 to release, 0 to engage.</li> </ul>

## Data Collection Parameters

---

Parameter	Description
	<ul style="list-style-type: none"><li>• 0x12: Velocity mode Arg: Target velocity.</li></ul>
V+ Command Argument	V+ command argument.
Velocity Error	Monitors the velocity error (commanded velocity minus encoder velocity), in counts/ms, for the selected motor.

## Configuring Axes, Options, and Kinematic Parameters

Some robots have a variable number of joints, option bits that control the presence of special features, and kinematic parameters used in position calculations. The Axes, Options, and Kinematic Parameters dialog box allows you to edit these parameters. For background information on robot axes, options, and kinematic parameters, see the [Adept SmartMotion Developer's Guide](#). Note that the ACE interface, described below, can be used in place of the SPEC utility program to edit these items.



**CAUTION:** Improper editing of robot joints, option bits, and kinematic parameters can cause the robot to malfunction or become inoperable. Therefore, editing must be performed by qualified personnel.

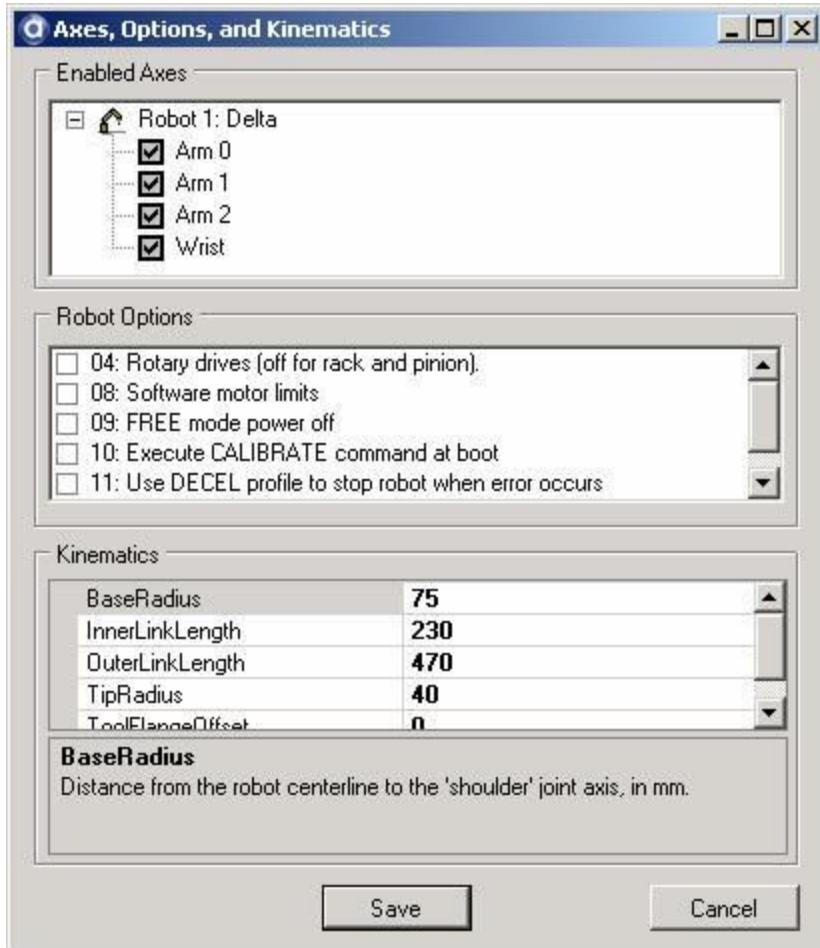
There are two ways to access the dialog box:

- From the Robot Installation Wizard. For details on the Robot Installation Wizard, see the topic [Installing Robots](#).
- From the [Robot Editor](#) menu, click **Configure > Axes, Options, and Kinematics**.

The Axes, Options, and Kinematics dialog box opens.

## Enabled Axes

---



Axes, Options, and Kinematics Dialog

### Enabled Axes

The Enabled Axes area is used to enable/disable the joints (axes) of the robot. If the robot doesn't have joints that can be enabled/disabled, the Enabled Axes check boxes will appear dimmed.

### Robot Options

The Robot Options area is used to select the robot option bits for your robot. See your robot kinematic module documentation for the robot option bits that apply to your robot. Some of the common option bits include:

Item	Description
Free mode power off	Robot power is turned off rather than disabling the individual amplifier.
Execute CALIBRATE command at boot	Calibrate the robot after the V+ operating system boots. Starting with V+ version 16.3D1, this is set by default on all Adept Viper and Adept Cobra s350 robots. This only works if the robot can calibrate with power off. It does not work on other Cobra robots because they have to move J4 during calibration.
Check joint interpolated collisions	While moving, check for obstacle collisions even for joint-interpolated moves. This causes slightly more CPU usage, if set, because it requires the robot to perform a kinematic solution that is not part of the normal operation.
Z-up during J4 calibration	On Adept Cobra robots, J4 must rotate slightly during calibration. This causes J3 to retract before moving J4.
J6 multi-turn	If your system has the multi-turn license installed, this bit allows infinite rotation of J6. Note that individual moves must be no more than 360 degrees.
Software motor limits	In robot models with multiple motors coupled to move a single joint, the standard joint motion limits may not be adequate to prevent the motors from hitting physical limits. In such cases, you may use software motor limits to restrict motor motion.
Split X-axis	Uses an extra motor to run a split X-axis.

## Kinematics

The Kinematics area is used to configure the kinematic parameters for your robot. See your robot kinematic module documentation for details.

The bottom section of the Kinematics area displays online help for the selected parameter.

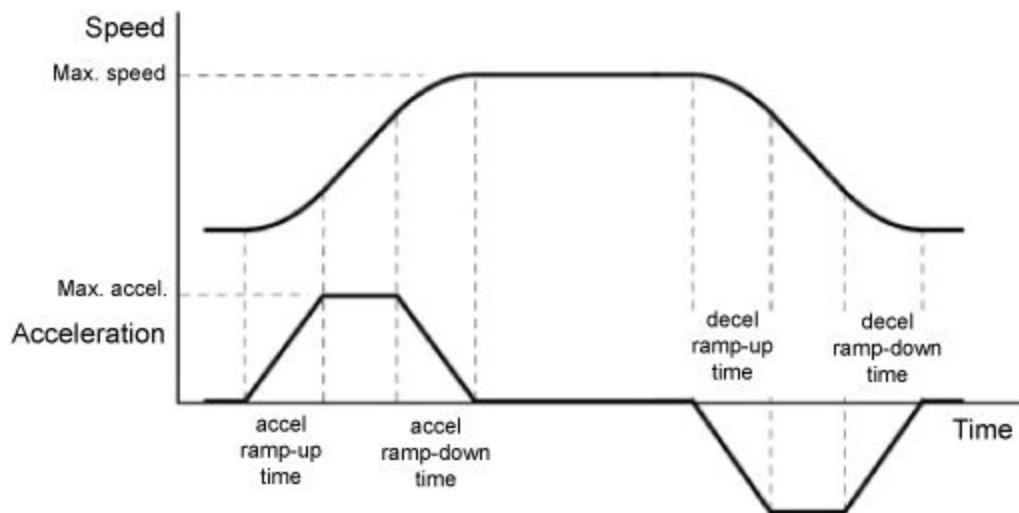
**NOTE:** Because online help for each parameter is provided directly in the editor, it is not duplicated in this documentation.

## Configuring S-Curve Profiles

While there are many different motion profiles that can be characterized as "S-curves", in the V+ operating system, an S-curve is a trajectory that has a trapezoidal acceleration profile, giving an S-shaped velocity profile.

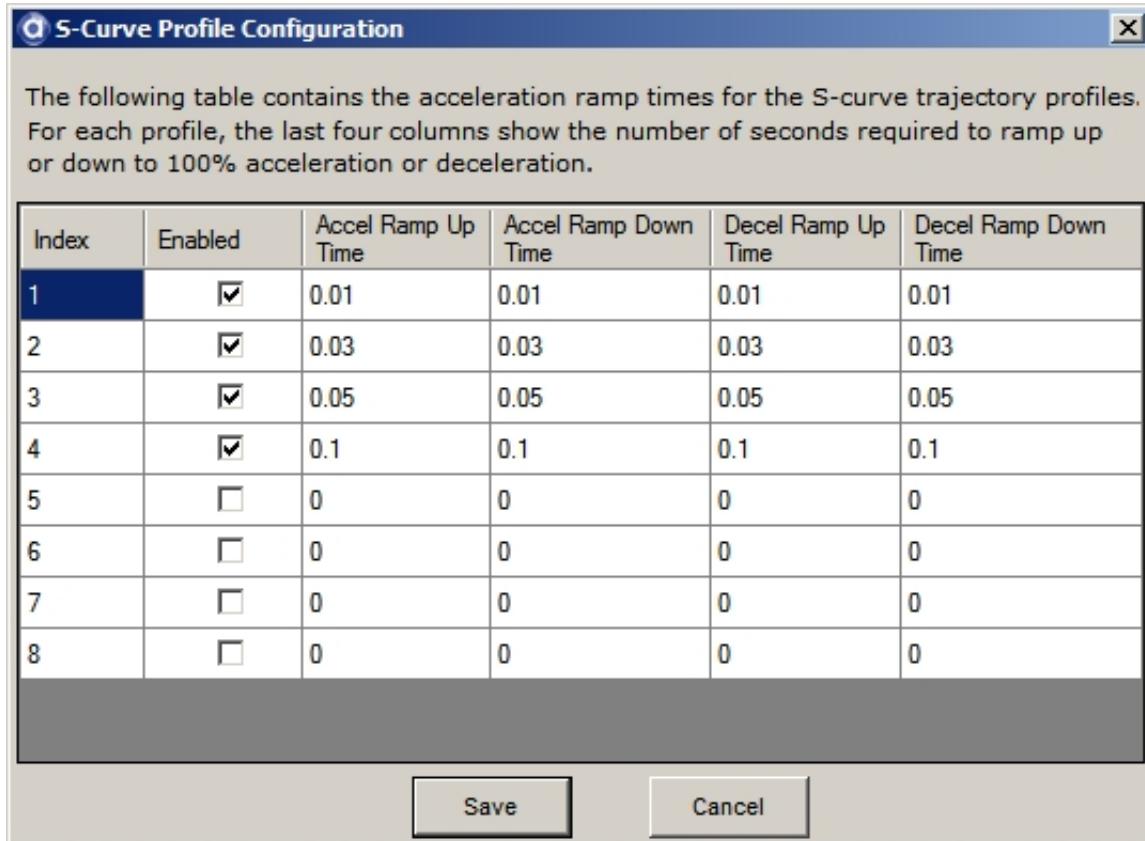
The benefit of a trapezoidal acceleration profile is that the rate of change of acceleration (the "jerk") can be controlled. (By comparison, the magnitude of the jerk for a square-wave acceleration profile is always infinite.) For many mechanisms, controlling the jerk is significant because high jerk values can cause the mechanical structure of the robot to vibrate. Minimizing structural vibrations is especially important at the end of a motion, since such oscillations can adversely affect the settling time of the robot, which can affect the cycle time. However, for stiff, strong mechanisms, a square-wave profile may result in shorter cycle times.

For a general trapezoidal profile, there are four acceleration values that can be specified, as shown in S-Curve (Trapezoidal Acceleration) Profile: the ramp up to maximum acceleration, the ramp down from maximum acceleration, the ramp up to maximum deceleration, and the ramp down to zero acceleration. In V+ each of these four acceleration values can be individually specified, and a set of the four values defines a specific acceleration "profile".



*S-Curve (Trapezoidal Acceleration) Profile*

The S-Curve Profile Configuration dialog box, shown in the figure below, is used to configure the four acceleration values. To open the dialog box, on the [Robot Editor](#) menu, click **Configure > S-Curve Profiles**.

*S-Curve Profile Configuration Dialog*

For each profile, the last four columns show the number of seconds required to ramp up or down to 100% acceleration or deceleration.

The elements of the S-Curve Profile Configuration dialog box are described below:

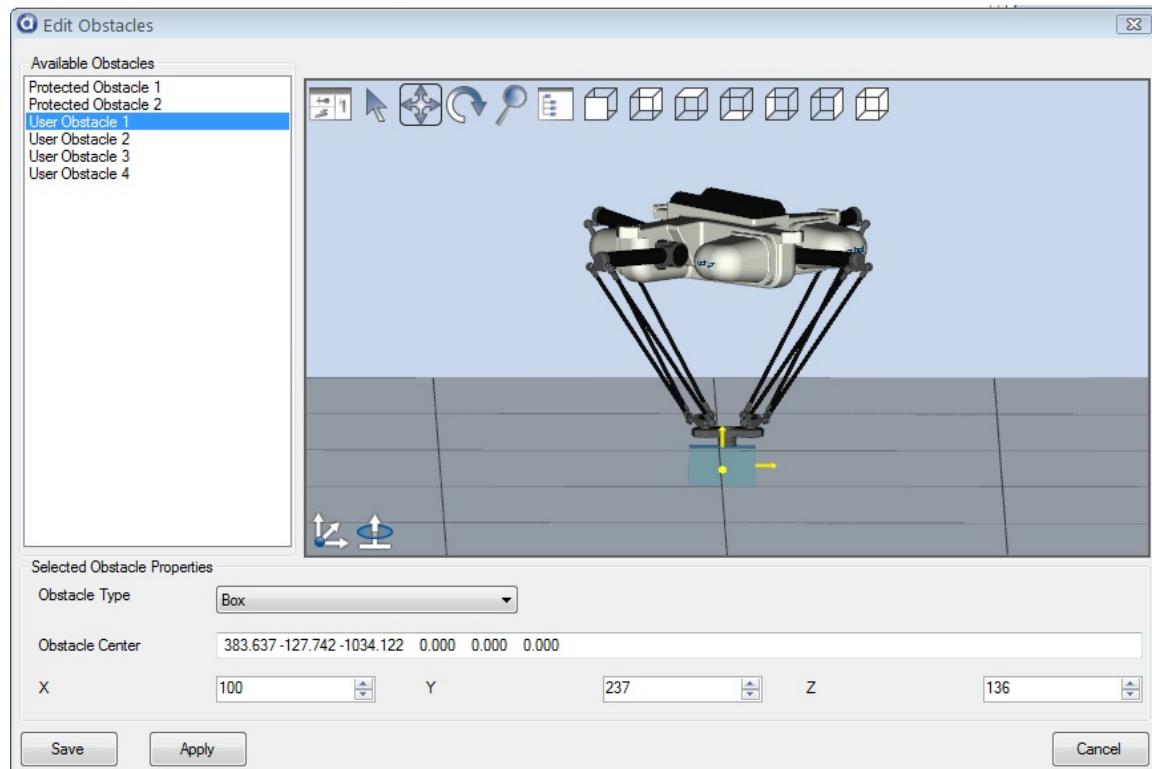
Item	Description
Index	The index position to which the corresponding profile will be stored.
Enabled	If checked, the corresponding profile is available for use in your programs.
Accel Ramp Up Time	The ramp-up time for the acceleration phase of the profile.
Accel Ramp Down Time	The ramp-down time for the acceleration phase of the profile.
Decel Ramp Up Time	The ramp-up time for the deceleration phase of the profile.
Decel Ramp Down Time	The ramp-down time for the deceleration phase of the profile.

Item	Description
Save	<p>Stores the profile information and closes the dialog box. You are prompted to save the changes to the controller (so they will be available after the next reboot).</p>  <p>The changes will be made in memory only. Do you want to save the modified profiles on the controller so they are used after the next reboot?</p> <p style="text-align: center;"><input type="button" value="Yes"/> <input type="button" value="No"/> <input type="button" value="Cancel"/></p> <ul style="list-style-type: none"> <li>Click <b>Yes</b> to store the profile changes to the controller. The changes will be permanently saved, and will still be available after the controller is rebooted.</li> <li>Click <b>No</b> to store the changes in system memory only (they are not saved to the controller). The profile changes will be lost when the controller is rebooted.</li> <li>Click <b>Cancel</b> to close the prompt and return to the S-Curve Profile Configuration dialog box.</li> </ul>
Cancel	Cancels all edits and closes the dialog box.

## Editing Obstacles

Some robot workcells contain "obstacles", such as part fixtures, camera supports, or other hardware, which you do not want the robot to contact during operation. An obstacle can also define a space (or shape) that you want the robot to work in; these are called containment obstacles. The Edit Obstacles dialog box allows you to add/edit the location, type, and size of workcell obstacles.

To open the Edit Obstacles dialog box, on the [Robot Editor](#) menu, click **Configure > Obstacles**. The Edit Obstacles dialog box opens.



*Edit Obstacles Dialog*

The elements of the Edit Obstacles dialog box are described below:

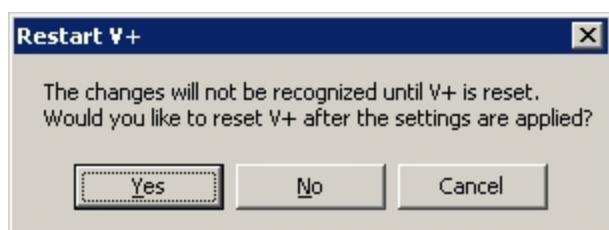
Item	Description
Protected Obstacle	For Adept robots, these are predefined obstacles that cannot be edited by the user. For non-Adept robots, these provide two additional obstacle entries.
User Obstacle	Each row provides an entry for a workcell obstacle.
Obstacle Type	This drop-down list is used to select the type of obstacle: box, cylinder, sphere or frustum. These types are also offered as containment obstacles for applic-

Item	Description
	ations where you want to keep the robot working within a defined area.
Obstacle Center	This text box is used to enter the coordinates of the center of the obstacle.
X, Y, Z Diameter , Height Base Diameter, Z Height, Top Dia- meter	Used to define the dimensions (in millimeters) of the obstacle
Apply	Stores the obstacle information but does not close the dialog box.
Save	Stores the obstacle information and closes the dialog box.
Cancel	Cancels all edits and closes the dialog box.

## Applying or Saving Your Changes

When Applying or Saving your changes, a prompt is displayed, which asks if you would like to restart V+ so that your changes are immediately recognized on the controller.

**NOTE:** If you select No, your changes will not take effect until the controller is rebooted.



*Restart V+ Prompt*

- Click **Yes** to store the changes, restart V+, and recognize the changes.
- Click **No** to store the changes but wait to recognize the changes until the next time the controller is rebooted.
- Click **Cancel** to close the prompt and close the Obstacle editor without storing any changes.

## Loading and Saving Robot/Motor Specification Files

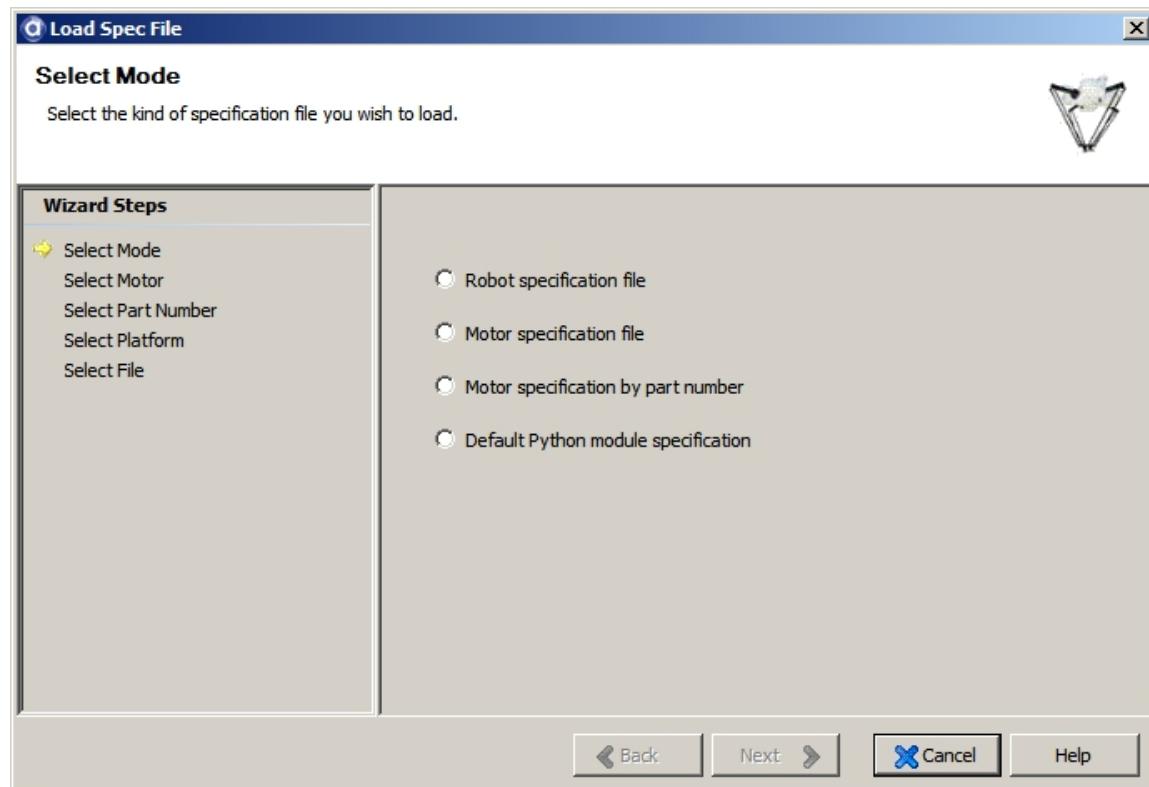
Robot and motor specification files can be used to back up and restore robot and motor specifications. The following sections describe how to load and save the specification files.

### Loading a Robot/Motor Specification File

The Load Robot/Motor Specification File dialog box is used to load a robot or motor specification file. There are two ways to access the dialog box:

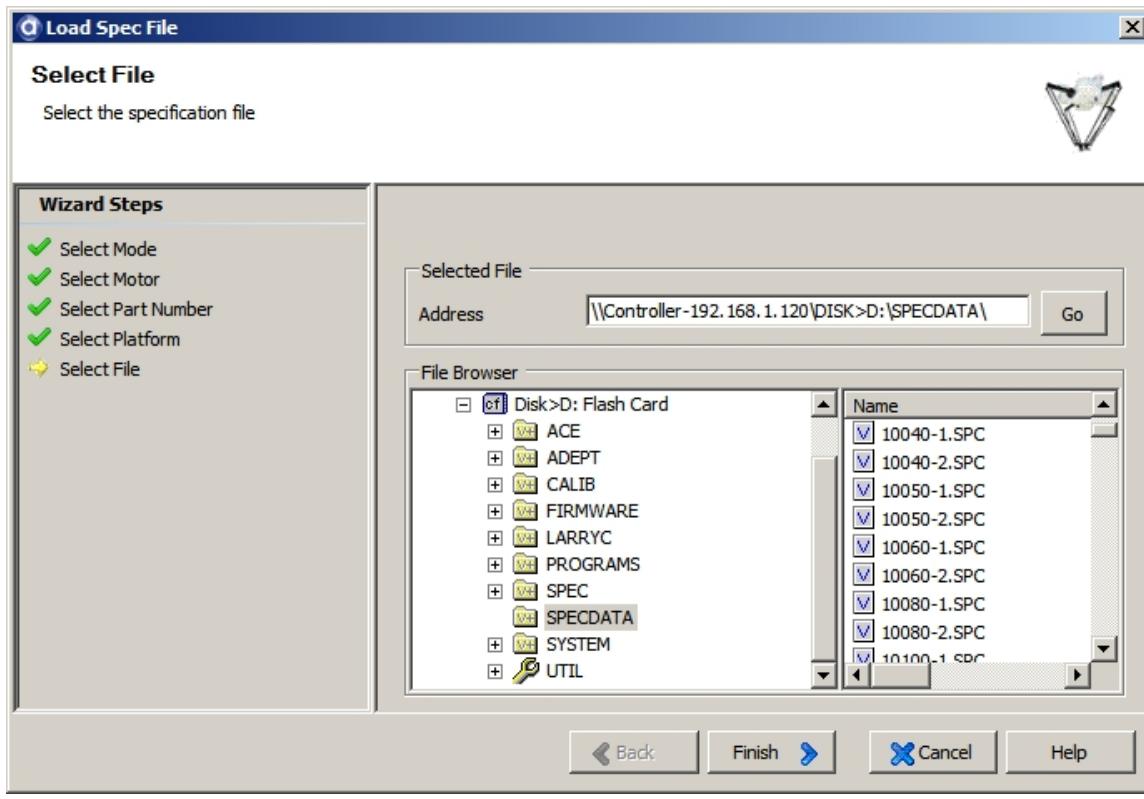
- From the Robot Installation wizard. For details on the Robot Installation Wizard, see the topic [Installing Robots](#).
- From the [Robot Editor](#) menu, select **Configure > Load Spec File...**

The Load Spec File wizard opens.



*Load Spec File Wizard*

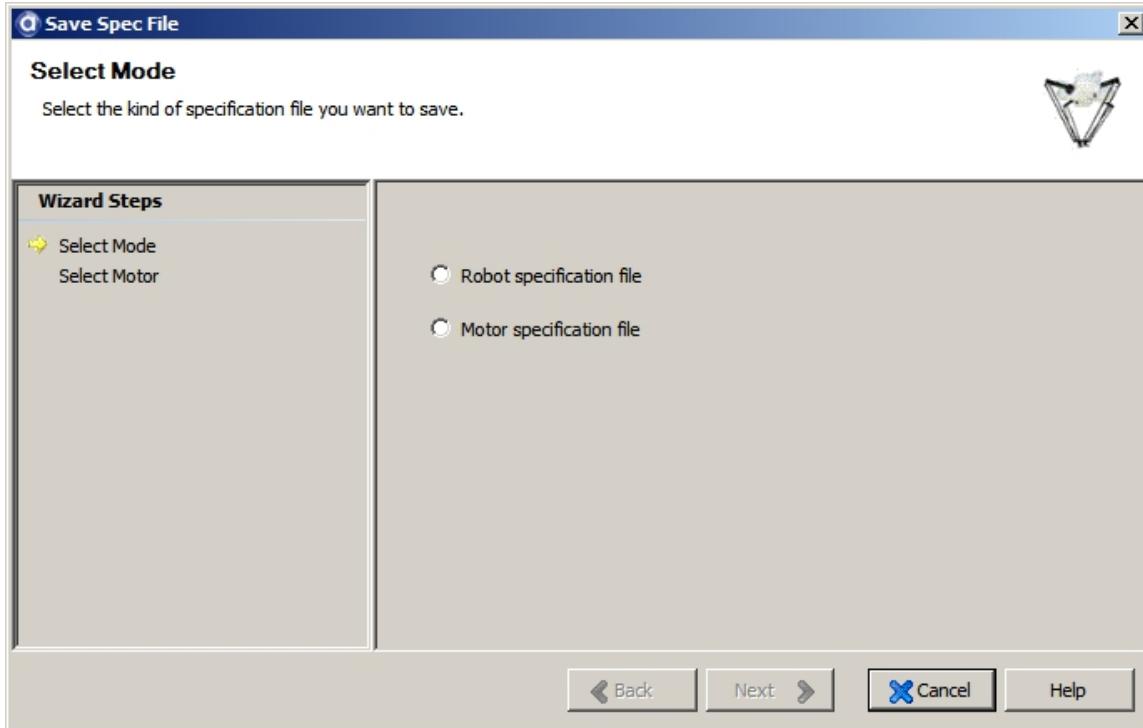
1. Select the method for loading the specification file. There are multiple ways of loading specifications:
  - Load default specification file by part number. On the next page of the wizard, the part number text box may be prepopulated with a part number, if a part number file has been previously loaded. The part number can also be entered directly into the text box.
  - Load default Python Module specification file (only used for Python linear modules). This method automatically obtains the part number from the motor itself.
  - Load motor specifications from a specification file on the PC or controller. You can use the browse button (...) next to the text box to locate the directory on your PC or controller.
  - Load robot specifications from a specification file on the PC or controller. You can use the browse button (...) next to the text box to locate the directory on your PC or controller.
  - Load platform file (only used when the Quattro robot is loaded in the workspace).
2. Click Next to continue loading the specification file. Follow the instructions in the wizard to complete the loading operation. The remaining wizard pages will vary, depending on the selection made in the previous step. For example, in the following figure, you are asked to browse for the location of the specification file.



Select File

## Saving a Robot/Motor Specification File

To save robot /motor specifications, on the [Robot Editor](#) menu, click **Configure > Save Spec File...** The Save Spec File wizard opens.



### Save Spec File Wizard

1. Select either the Robot or Motor radio button.
2. If the Motor radio button is selected, use the next wizard page to select the motor number.
3. Use the next wizard page to specify the file name and location.
4. Click Save to save the file.

## Safety Settings

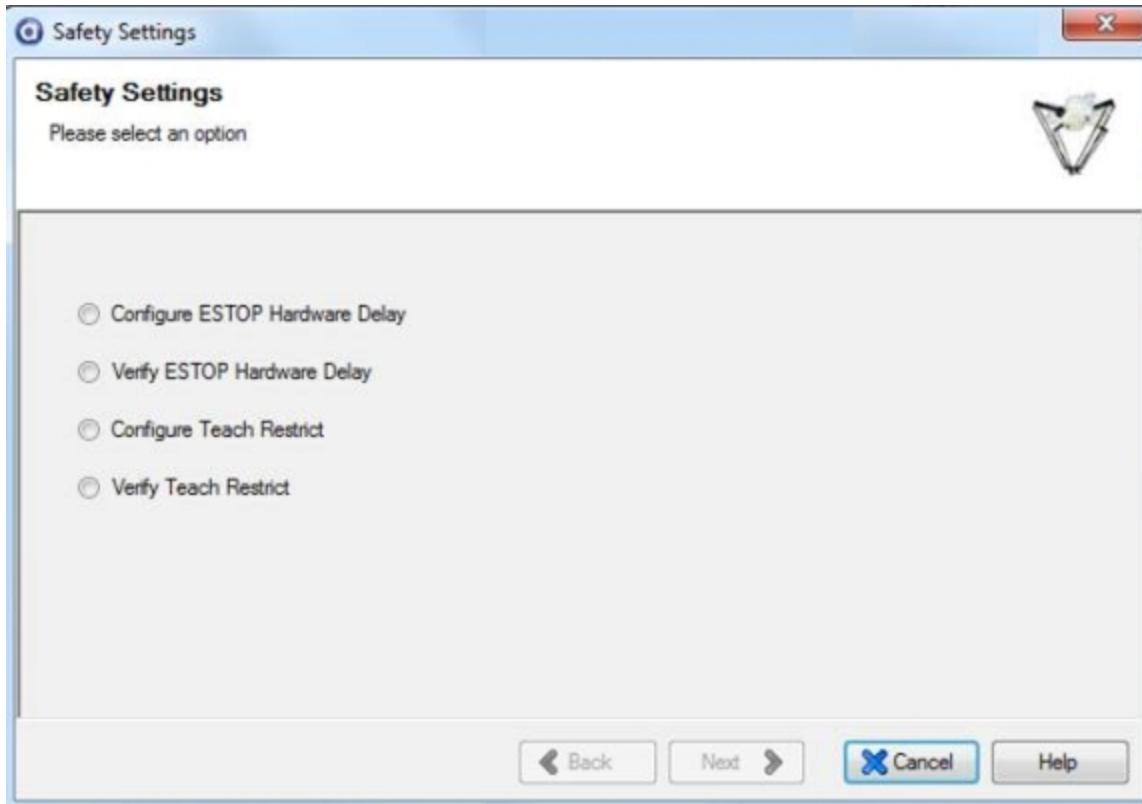
**NOTE:** This feature operates differently in emulation mode. For details on emulation mode wizards, see Emulation Mode Wizards on page 185.

For robots using Adept eAIB (enhanced Amp-in-Base) or eMB-40R/60R (enhanced MB-40R/60R) amplifiers, the Configure menu offers a Safety Settings option, which is used to restore the E-Stop hardware delay and the teach-mode restricted speed to the factory settings. This process is also known as "commissioning the system".



**CAUTION:** The eAIB or eMB-60R amplifiers are configured on the Adept robot at the factory. Therefore, the Safety Settings configuration is only required when the eAIB or eMB-60R amplifiers have been replaced in the field.

To open the Safety Settings interview wizard, on the [Robot Editor](#) menu, click **Configure > Safety Settings**. The Safety Settings interview wizard opens.



*Safety Settings Interview Wizard*

## Safety Settings

---

To use the interview wizard, select the desired option and click Next. Read and follow the instructions on each interview wizard screen. The interview wizard guides you through the configuration process.

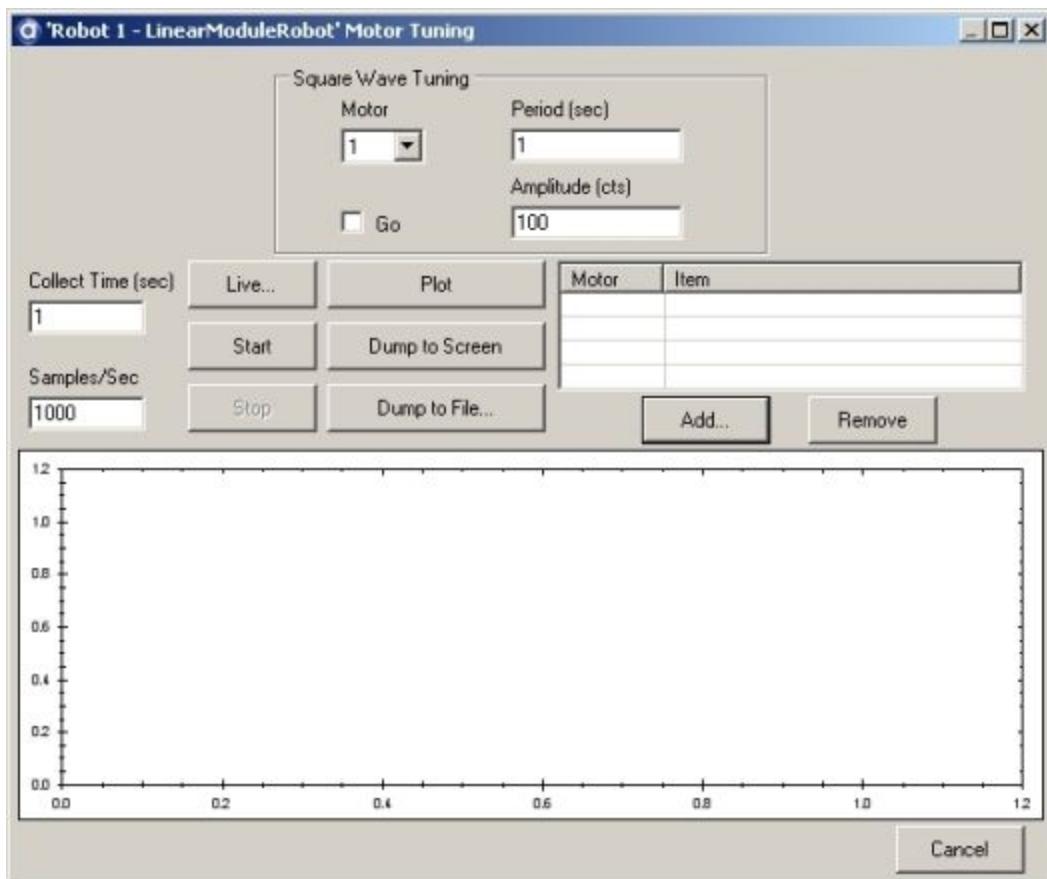
For more details on the replacing the eAIB or eMB-60R amplifiers, and on the Safety Settings wizard, see the Maintenance chapter of your Adept robot user's guide.

For details on the buttons and indicators on the wizard pages, see Wizards on page 181.

## Tuning the Motors

The Motor Tuning dialog box, shown in the following figure, is used to send a square wave positioning command to the specified motor, and observe the response. This dialog box allows you to collect values on the specified motor(s) at up to the servo rate of 8KHz. You can collect up to 8 data items, at rates of up to 8000 samples per second, up to the memory limit of the robot.

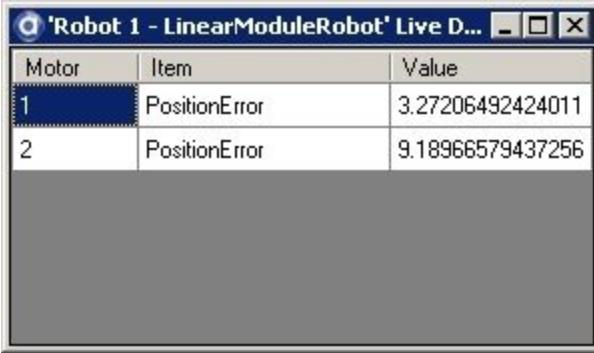
To open the Motor Tuning dialog box, on the [Robot Editor](#) menu, click **Control > Motor Tuning**. The Motor Tuning dialog box opens.



*Motor Tuning Dialog*

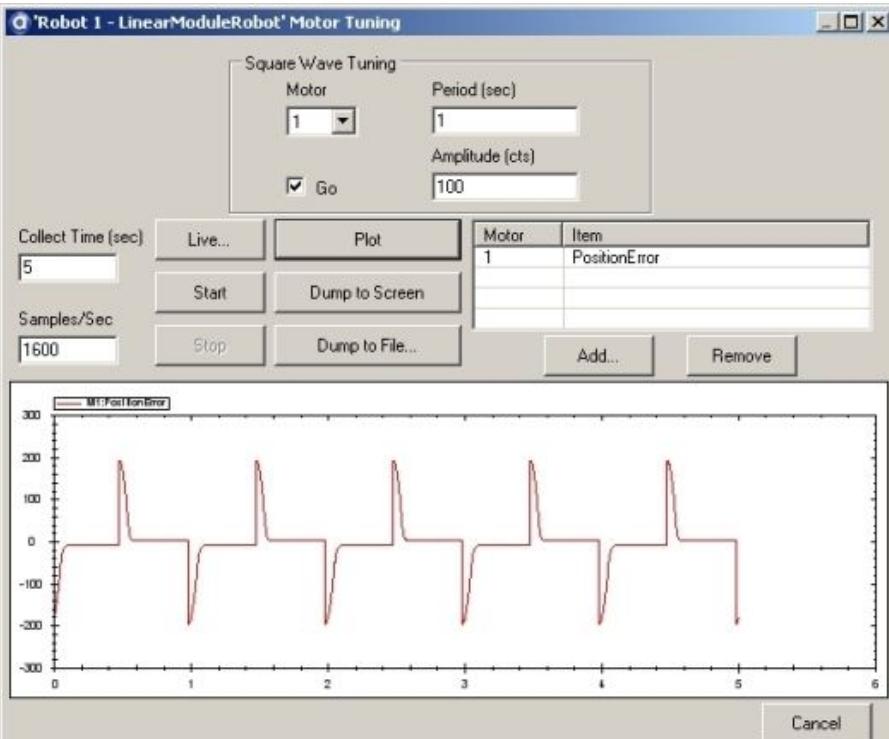
This section describes the items in the Motor Tuning dialog box:

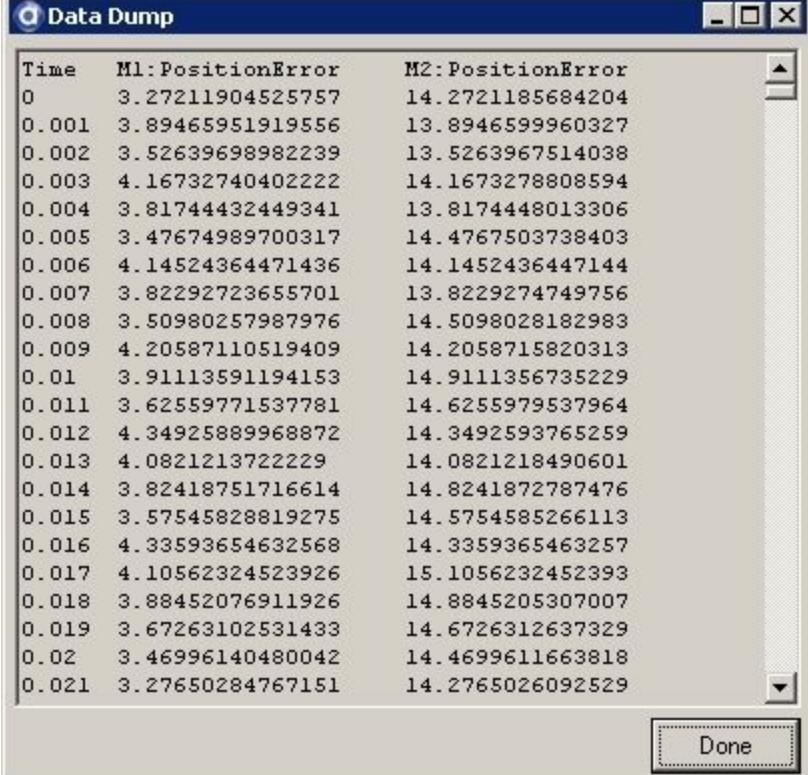
Item	Description
Square Wave Tuning	This area contains items for specifying and initiating a square wave positioning command to one of the robot motors.

Item	Description									
	<ul style="list-style-type: none"> <li>• Motor: Specifies the motor that will receive the square wave positioning command.</li> <li>• Period (sec): Specifies the length of the square wave (in seconds).</li> <li>• Amplitude (cts): Specifies the amplitude of the square wave (in servo counts)</li> <li>• Go: Toggles (turns on/off) the square wave positioning command to the specified motor.</li> </ul>									
Collect Time (sec)	Specifies the data collection time in seconds, the default value is 1. For example, to collect data for five seconds, enter the value 5.									
Samples/Sec	Specifies the data collection rate in samples per second. The default value is 1000. For example, to collect data at a sample rate of 1600 samples/sec, enter the value 1600.									
Live...	<p>Displays a window that shows the "live" data being collected, as shown in the following figure.</p>  <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th style="background-color: #cccccc;">Motor</th> <th style="background-color: #cccccc;">Item</th> <th style="background-color: #cccccc;">Value</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>PositionError</td> <td>3.27206492424011</td> </tr> <tr> <td>2</td> <td>PositionError</td> <td>9.18966579437256</td> </tr> </tbody> </table> <p><i>Live Data Display</i></p>	Motor	Item	Value	1	PositionError	3.27206492424011	2	PositionError	9.18966579437256
Motor	Item	Value								
1	PositionError	3.27206492424011								
2	PositionError	9.18966579437256								
Start	Click to start the data collection. The data collection will continue until either: Stop is clicked, or the specified collect time is reached.									
Stop	Click to stop the data collection. If the specified collect time has already expired, this button is disabled.									
Plot	Click to plot the collected data. A progress bar displays while the data is processed. After the processing has completed, the data is plotted on the graph located at the lower portion of the Motor Tuning dialog box, as shown in the figure below.									

## Tuning the Motors

---

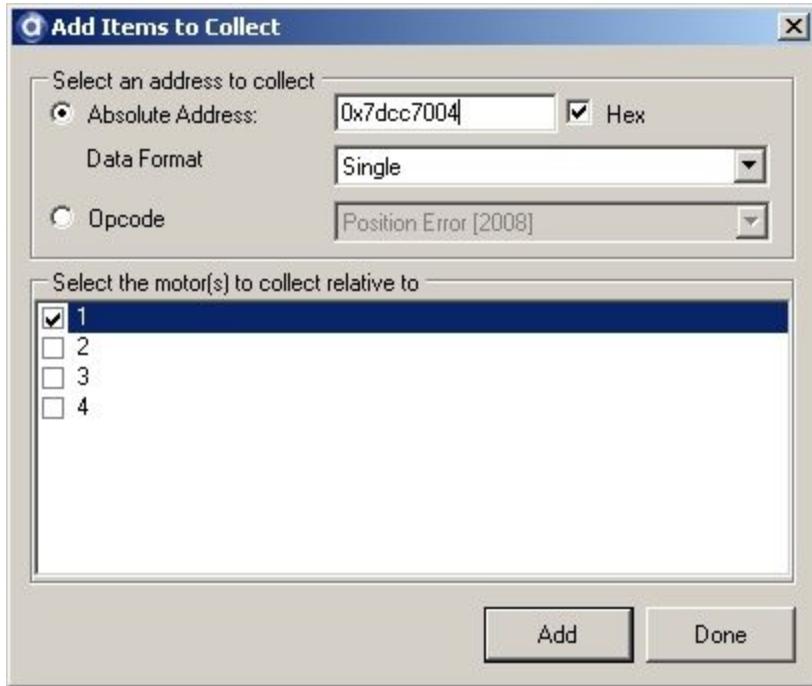
Item	Description
	 <p>The screenshot shows the 'Robot 1 - LinearModuleRobot' Motor Tuning dialog box. At the top, there's a 'Square Wave Tuning' section with fields for 'Motor' (set to 1), 'Period (sec)' (set to 1), and 'Amplitude (cts)' (set to 100). A checked checkbox labeled 'Go' is present. Below this are buttons for 'Live...', 'Plot', 'Start', 'Dump to Screen', 'Stop', and 'Dump to File...'. To the right is a table titled 'Motor' with columns 'Motor' and 'Item', showing a single entry for Motor 1 and Item PositionError. Buttons for 'Add...' and 'Remove' are also visible. At the bottom is a plot titled 'Motor Tuning Plot' showing a square wave signal over time, ranging from -300 to 300 units. The x-axis is labeled from 0 to 6. A legend indicates the red line represents 'M1 PositionError'. A 'Cancel' button is at the bottom right.</p>
Dump to Screen	Displays the collected data in a Data Dump window in text-file format, as shown in the following figure.

Item	Description																																																																					
	 <p>The screenshot shows a window titled "Data Dump" with two columns: "Time", "M1:PositionError", and "M2:PositionError". The data is as follows:</p> <table border="1"> <thead> <tr> <th>Time</th> <th>M1:PositionError</th> <th>M2:PositionError</th> </tr> </thead> <tbody> <tr><td>0</td><td>3.27211904525757</td><td>14.2721185684204</td></tr> <tr><td>0.001</td><td>3.89465951919556</td><td>13.8946599960327</td></tr> <tr><td>0.002</td><td>3.52639698982239</td><td>13.5263967514038</td></tr> <tr><td>0.003</td><td>4.16732740402222</td><td>14.1673278808594</td></tr> <tr><td>0.004</td><td>3.81744432449341</td><td>13.8174448013306</td></tr> <tr><td>0.005</td><td>3.47674989700317</td><td>14.4767503738403</td></tr> <tr><td>0.006</td><td>4.14524364471436</td><td>14.1452436447144</td></tr> <tr><td>0.007</td><td>3.82292723655701</td><td>13.8229274749756</td></tr> <tr><td>0.008</td><td>3.50980257987976</td><td>14.5098028182983</td></tr> <tr><td>0.009</td><td>4.20587110519409</td><td>14.2058715820313</td></tr> <tr><td>0.01</td><td>3.91113591194153</td><td>14.9111356735229</td></tr> <tr><td>0.011</td><td>3.62559771537781</td><td>14.6255979537964</td></tr> <tr><td>0.012</td><td>4.34925889968872</td><td>14.3492593765259</td></tr> <tr><td>0.013</td><td>4.0821213722229</td><td>14.0821218490601</td></tr> <tr><td>0.014</td><td>3.82418751716614</td><td>14.8241872787476</td></tr> <tr><td>0.015</td><td>3.57545828819275</td><td>14.5754585266113</td></tr> <tr><td>0.016</td><td>4.33593654632568</td><td>14.3359365463257</td></tr> <tr><td>0.017</td><td>4.10562324523926</td><td>15.1056232452393</td></tr> <tr><td>0.018</td><td>3.88452076911926</td><td>14.8845205307007</td></tr> <tr><td>0.019</td><td>3.67263102531433</td><td>14.6726312637329</td></tr> <tr><td>0.02</td><td>3.46996140480042</td><td>14.4699611663818</td></tr> <tr><td>0.021</td><td>3.27650284767151</td><td>14.2765026092529</td></tr> </tbody> </table> <p style="text-align: right;"><input type="button" value="Done"/></p> <p><i>Data Dump Window</i></p>	Time	M1:PositionError	M2:PositionError	0	3.27211904525757	14.2721185684204	0.001	3.89465951919556	13.8946599960327	0.002	3.52639698982239	13.5263967514038	0.003	4.16732740402222	14.1673278808594	0.004	3.81744432449341	13.8174448013306	0.005	3.47674989700317	14.4767503738403	0.006	4.14524364471436	14.1452436447144	0.007	3.82292723655701	13.8229274749756	0.008	3.50980257987976	14.5098028182983	0.009	4.20587110519409	14.2058715820313	0.01	3.91113591194153	14.9111356735229	0.011	3.62559771537781	14.6255979537964	0.012	4.34925889968872	14.3492593765259	0.013	4.0821213722229	14.0821218490601	0.014	3.82418751716614	14.8241872787476	0.015	3.57545828819275	14.5754585266113	0.016	4.33593654632568	14.3359365463257	0.017	4.10562324523926	15.1056232452393	0.018	3.88452076911926	14.8845205307007	0.019	3.67263102531433	14.6726312637329	0.02	3.46996140480042	14.4699611663818	0.021	3.27650284767151	14.2765026092529
Time	M1:PositionError	M2:PositionError																																																																				
0	3.27211904525757	14.2721185684204																																																																				
0.001	3.89465951919556	13.8946599960327																																																																				
0.002	3.52639698982239	13.5263967514038																																																																				
0.003	4.16732740402222	14.1673278808594																																																																				
0.004	3.81744432449341	13.8174448013306																																																																				
0.005	3.47674989700317	14.4767503738403																																																																				
0.006	4.14524364471436	14.1452436447144																																																																				
0.007	3.82292723655701	13.8229274749756																																																																				
0.008	3.50980257987976	14.5098028182983																																																																				
0.009	4.20587110519409	14.2058715820313																																																																				
0.01	3.91113591194153	14.9111356735229																																																																				
0.011	3.62559771537781	14.6255979537964																																																																				
0.012	4.34925889968872	14.3492593765259																																																																				
0.013	4.0821213722229	14.0821218490601																																																																				
0.014	3.82418751716614	14.8241872787476																																																																				
0.015	3.57545828819275	14.5754585266113																																																																				
0.016	4.33593654632568	14.3359365463257																																																																				
0.017	4.10562324523926	15.1056232452393																																																																				
0.018	3.88452076911926	14.8845205307007																																																																				
0.019	3.67263102531433	14.6726312637329																																																																				
0.02	3.46996140480042	14.4699611663818																																																																				
0.021	3.27650284767151	14.2765026092529																																																																				
Dump to File...	Displays a Save As dialog box used for saving the collected data to a text file, which can be viewed or processed at a later time.																																																																					
Add	Click to add monitored items to the list. See the next section, <a href="#">Adding and Removing Monitored Items</a> , for more details.																																																																					
Remove	Click to remove monitored items from the list. See the next section, <a href="#">Adding and Removing Monitored Items</a> , for more details.																																																																					

## Adding and Removing Monitored Items

The Add Items to Collect dialog box is used to add items to the data collection list. Typically, for motor tuning, you will want to collect position error data on the motor receiving the square wave positioning command.

**NOTE:** You can monitor up to seven data items.



*Add Items to Collect Dialog*

To add a new item:

1. On the Motor Tuning dialog box, click **Add**. The Add Items to Collect dialog box opens (see the figure above).
2. Specify the value to be monitored:
  - Enter an Absolute Address (select the Hex check box, if you wish to enter the value in hex format) and select the Data Format from the drop-down list.
  - OR
  - Select an Opcode from the drop-down list. For a description of each of the items, see the topic [Data Collection Parameters](#).
3. Select (check) the box next to the motor(s) to be added.
4. Click **Add** to add the items and close the dialog box. The items are listed in the Motor Tuning dialog box.

To remove an item:

1. On the Motor Tuning dialog box, select the item to be removed from the list.
2. Click **Remove**. The selected item is removed from the list.

## Using a Tool Offset

In certain robotics applications, it is necessary to use an end-effector (also referred to as a "tool" or "gripper") that has an offset from the origin of the robot end-effector mount (also referred to as the "tool flange"). The ACE software provides a Tool Offset Wizard, which simplifies the task of teaching a tool offset.

When teaching the tool offset, you must rotate the end-effector around the tool tip and teach at least two points. If more points are taught, the tool offset will be slightly more accurate and repeatable.

With many robots (for example, SCARA robots), you can only rotate in the X-Y plane, in which case only the X-Y tool offset will be computed by the tool offset wizard. For these robots, you must manually edit the Z-extension of the tool offset. This will ensure that the heights of taught locations are accurate. It will also allow the 3D Visualization window to display the proper gripper dimensions.

When using six-axis robots, you can rotate around the tool tip in three dimensions. With these robots, either the X-Y or full X-Y-Z tool offset will be computed depending on the amount of rotation. However, the tool offset wizard does not calculate the orientation of the tool tip. Therefore, you must enter that value manually.

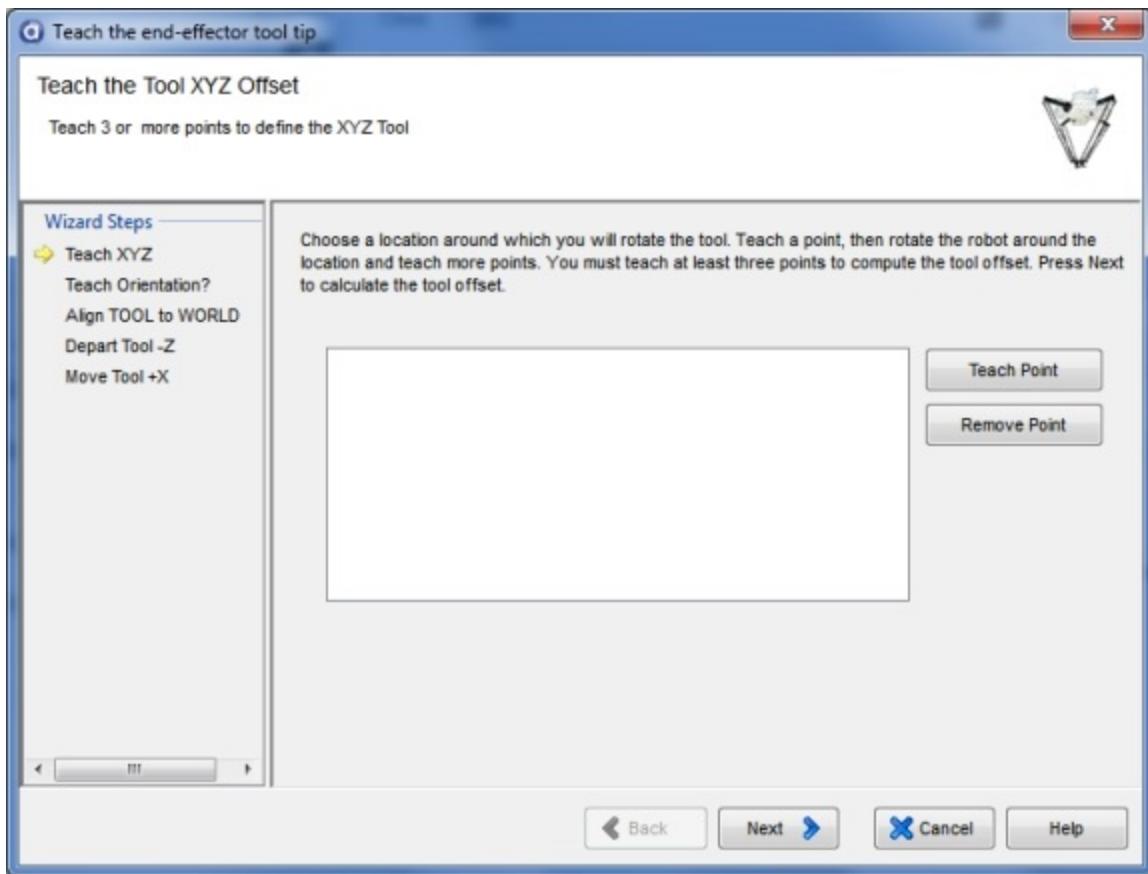
**NOTE:** When a workspace loads or the connection state of a controller changes, the TOOL (tool offset) is not updated. However, if you do something that requires a gripper (end-effector) to be set, the tool offset will be updated. For example, if you perform a robot-to-belt calibration, it will apply the end-effector and tool offset used in the calibration settings. For details on the robot-to-belt calibration, see Belt Calibrations on page 848.

## Teaching a Tool Offset

To teach a tool offset:

**NOTE:** You must teach at least points. However, teaching more points will improve the accuracy of the computed tool offset.

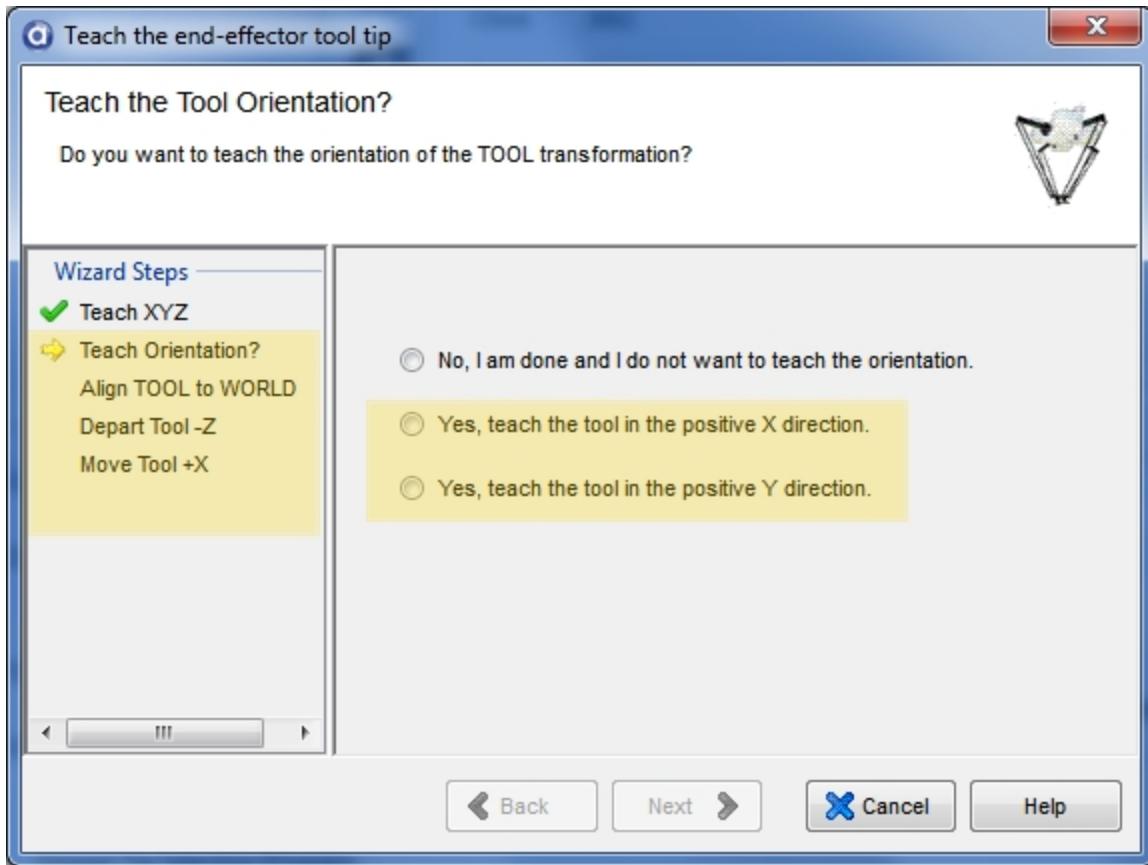
1. From the Gripper Editor click the tool offset icon (). For details, see Gripper Editor on page 108. The Tool Offset Wizard opens.



### *Tool Offset Wizard, Introduction*

2. Follow the instructions on the wizard pages to complete the steps. For more details on the wizard features, see Wizards on page 181.

After you teach the points for calculating the tool offset, you will have the option to continue with additional steps to teach the tool orientation, align the tool, and teach a depart tool and move tool, as shown in the following figure:



3. When you get to the last page, click Finish to accept the computed tool offset and exit the wizard.

The computed tool offset is entered as the value for the parameter on the Gripper Editor.



## Cobra i600/i800 Robot Configuration

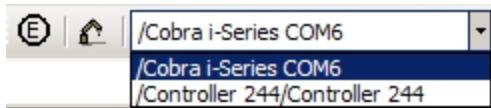
The topics in this chapter describe how to use the ACE software configuration tools for the Cobra i600/i800 robots.

Configuration Overview .....	266
Working with Configuration Files .....	268
Configuration Manager .....	269
Controller View .....	273
Password-Protect Configuration Data .....	275
Robot Configuration Pages .....	276
Apply Changes to the Controller .....	287
Load Configuration Data from a File .....	290
Save Configuration Data to a File .....	292
Restore Configuration .....	294
License Manager .....	296
Switches and Parameters .....	299

## Configuration Overview

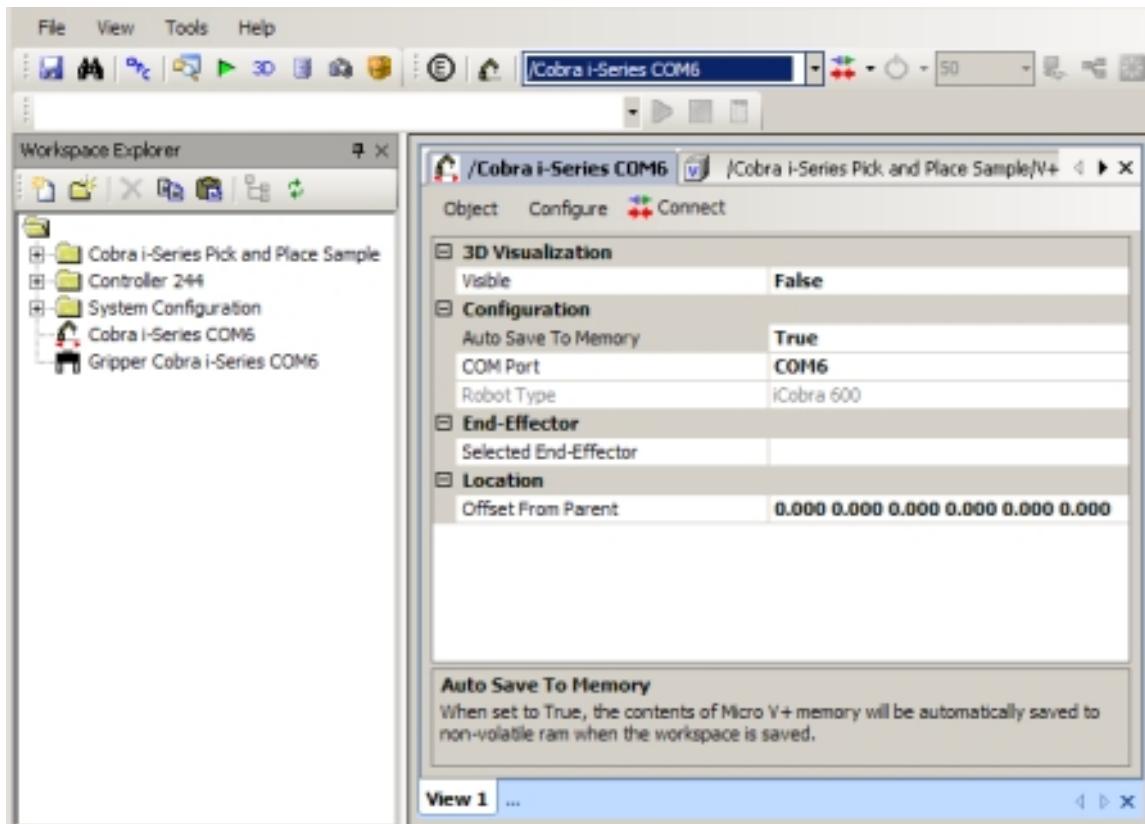
The Configuration Manager tool is used to configure the Cobra i600/i800 (Cobra i-Series) robots. If you are configuring other robots or motors, see Robot and Motor Configuration on page 233. In addition, a diagnostics wizard is available for the Cobra i600/i800 (Cobra i-Series) robots. For details, see Cobra i600/i800 Robot Diagnostics on page 1130.

When an Cobra i-Series robot is created, it is added as an object in the ACE workspace. The Cobra i-Series robot also appears on the controller drop-down list, just like a standard SmartController. From here you can connect/disconnect, change monitor speed, etc.



*Cobra i-Series Shown on Controller List*

Multiple Cobra i-Series robots can belong to the same workspace along with SmartController CX controllers and SmartController CX-based robots. Each Cobra i-Series robot object includes an editor that is used to define, for example, the COM port used for communications with the robot.



*Cobra i-Series Object in the Workspace with Editor*

## Working with Configuration Files

The Configuration Manager tool allows you to: Modify the Cobra i-Series robot configuration, save your system configuration (all of the parameters) to a disk file, and to load a configuration from a disk file. These disk files are called "configuration" files and use the .cfg extension. You can launch the Configuration Manager from the Configure item on the object menu bar.

Configuration files allow you to:

- Make a backup copy of your configuration. After you have configured your robot with the correct parameter settings, save the configuration to a configuration file so you have a backup copy. The configuration file can then be used to restore the configuration in the event that the parameters or configuration are inadvertently changed, or if the hardware needs to be re-installed because of technical issues.
- Configure a robot once, and then reuse that information for other robots that require the same settings. In this case, you configure the first robot and save the configuration to a file. After that, you connect to the other robots and load the configuration file you just created.

**NOTE:** It is important that you use configuration files that are configured for the type of robot you are connected to. If you load a configuration file created for one type of robot to a different type, the operation will fail.

## Configuration Manager

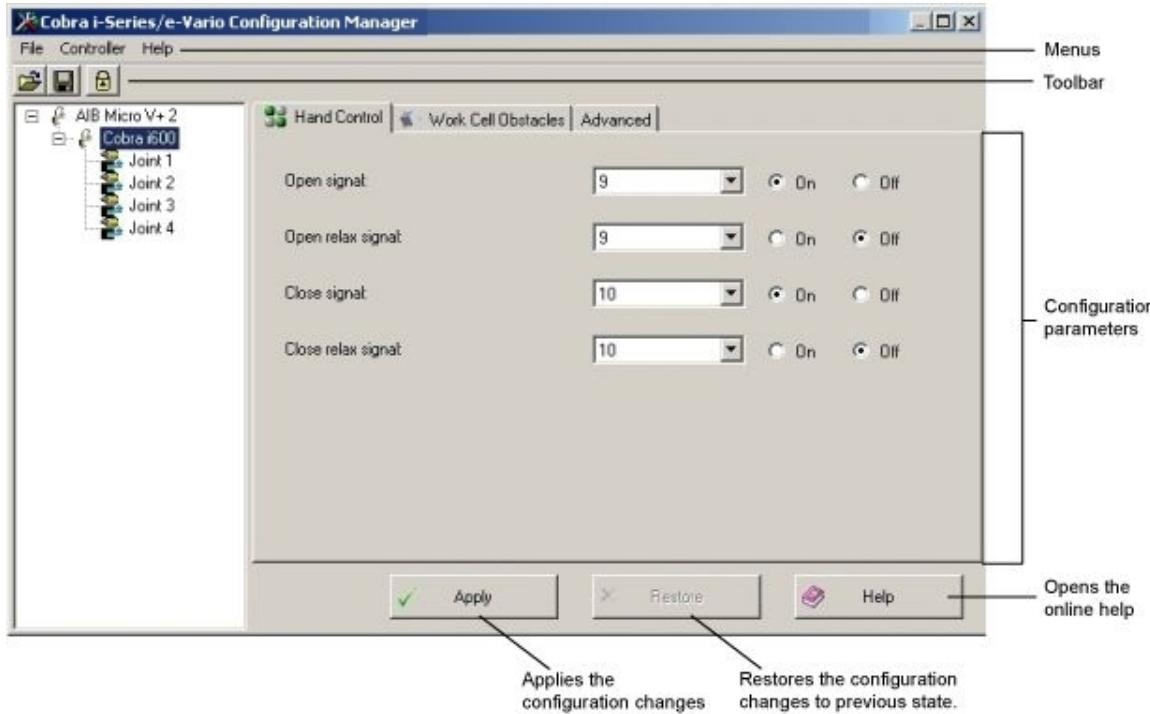
The Configuration Manager allows you to configure the Cobra i600/i800 (Cobra i-Series) or e-Vario robot running on the MicroV+ operating system. You can configure the following items:

Item	Description
Travel Limits	<p>Change the individual robot joint travel limits from their default settings to limit robot joint travel. These software-defined joint limits (soft stops) are intended to protect the robot joints from colliding with custom work-cell hardware and to provide a safety buffer so that the joint does not impact a physical hardstop.</p> <p>To access the travel limits configuration page, select a joint node in the system tree view. For details, see <a href="#">Edit Joint Limits on page 284</a>.</p>
Hand Control Signals	<p>View and/or modify the digital output signals used to control the opening and closing of the robot hand. The available signals include both the digital output signals and software signals.</p> <p>To access the hand control signals configuration page, select a robot node in the system tree view. Then, choose the Hand Control tab in the Configuration view. For details, see <a href="#">Hand Control Signals on page 277</a>.</p>
Workcell Obstacles	<p>Define workcell obstacles to avoid, or areas in which the robot tool tip must work. You can define up to six of these obstacles or areas.</p> <p>To access the workcell obstacles configuration page, select a robot node in the system tree view. Then, choose the Workcell Obstacles tab in the Configuration view. For details, see <a href="#">Workcell Obstacles on page 279</a>.</p>

To open the Configuration Manager, double-click the Cobra robot object in the Workspace Explorer, and then select Configuration Manager from the Configure menu.

## Menu Items

---

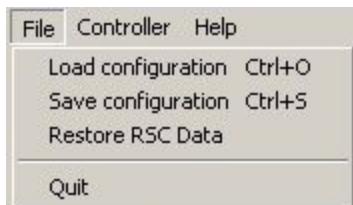


### Configuration Manager

## Menu Items

This section describes the selections available from the Configuration Manager menu.

### File Menu



#### Load Configuration

Displays a dialog to load a configuration data file. These files are created using the Save Configuration option. For details, see Load Configuration Data from a File on page 290.

#### Save Configuration

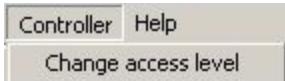
Displays a dialog to save the configuration parameters currently stored in the Configuration Manager. For details, see Save Configuration Data to a File on page 292.

#### Restore RSC Data

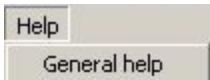
Restores the robot signature card (RSC) data from a disk file.

#### Quit

Closes the Configuration Manager. If you have not saved your changes, you will be prompted to do so.

**Controller Menu****Change Access Level**

Changes the user access level. By default, access to some parameters is restricted. Only authorized users can modify these restricted parameters. Use Change Access Level to allow full access to all parameters. For details, see Change Access Level on page 275.

**Help Menu****General Help**

Displays the online help (the document you are currently reading).

**Toolbar Items**

This section describes the selections available from the Configuration Manager toolbar.

Item	Description
(Load)	Loads a previously-saved robot configuration from a disk file. For details, see Load Configuration Data from a File on page 290.
(Save)	Saves the current configuration to a disk file. For details, see Save Configuration Data to a File on page 292.
(Access level)	Changes the user access level. For details, see Change Access Level on page 275.

**Editor Parameters**

The middle portion of the Configuration Manager contains the editor parameters. These are used to configure various settings on the selected robot. The items in this area change, based on the selected tab.

**Tabs**

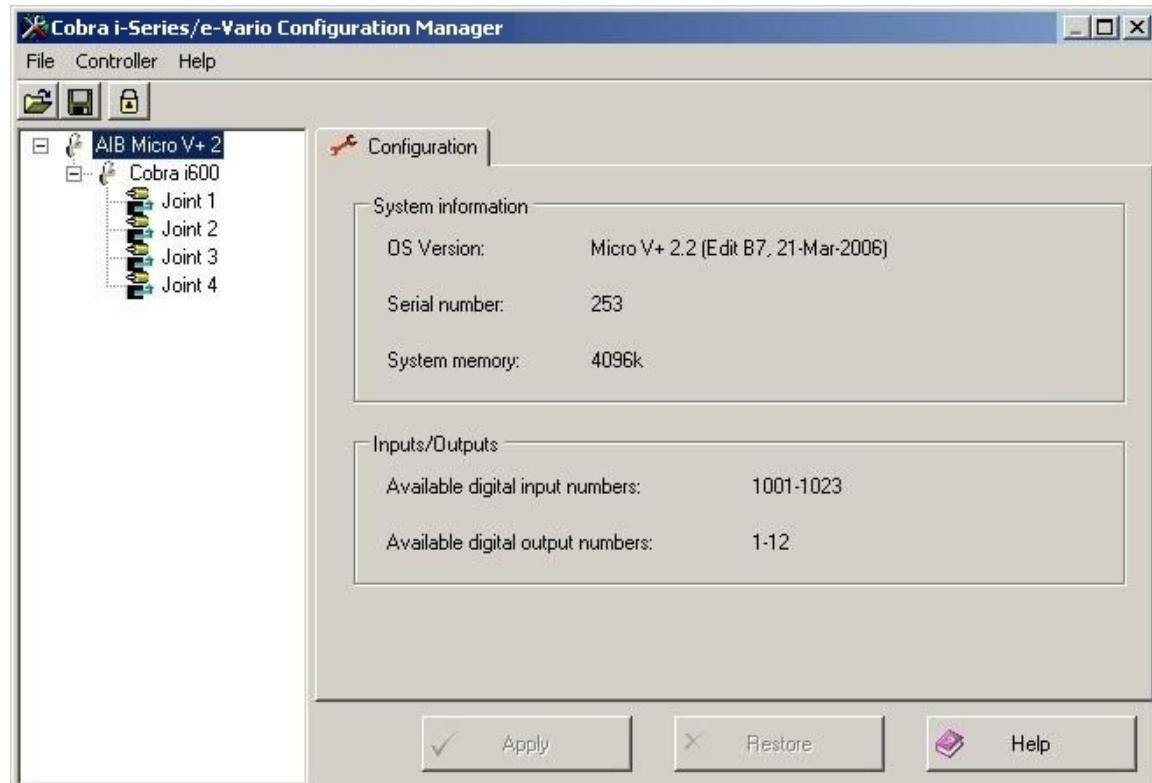
Item	Description
Hand Control	Displays the configuration page for hand control open and close signals. For details, see Specify Hand Control Signals on page 278.
Workcell Obstacles	Displays the configuration page to specify parameters for obstacles in the Workcell. For details, see Manage Obstacles in the Workcell on page 281.
Advanced	Displays advanced configuration options. For details, see Advanced Robot Configuration on page 283.

**Buttons**

Item	Description
Apply	Configuration data changes remain local to the Configuration Manager tool until they are saved to the controller using the Apply button. The Apply button opens the Apply Changes dialog. For details, see <a href="#">Apply Changes to the Controller</a> on page 287.
Restore	In the event of an error, the Restore button is available to reset the configuration data. The Restore button is available only after you have edited the data. The Restore button opens the Restore Configuration dialog. For details, see <a href="#">Restore Configuration</a> on page 294.
Help	Launches the online help documentation.

## Controller View

When the Controller node is selected in the system tree view, the configuration page for the controller displays. This page provides information about the system and the available digital inputs and outputs.



*Controller Node Configuration Information*

## System Information

Item	Description
OS Version	Version and revision number of the MicroV+ software in use.
Serial number	Serial number of the system controller.
System memory	Size of the system program memory in kilobytes (1K = 1024 8-bit bytes)

## Inputs/Outputs

Item	Description
Available digital input numbers	Lists the available digital input signals for the robot. For additional details on I/O signals, see the user's guide for your robot.
Available digital output numbers	Lists the available digital output signals for the robot. For additional details on I/O signals, see the user's guide for your robot.

## Password-Protect Configuration Data

Password protection prevents end users from modifying the default configuration parameters such as the minimum joint travel. By default, the access level is restricted, providing view-only access to the protected parameters.

Depending on the version you have, restricted parameters may be editable only by Adept personnel. Contact Adept Technical Support for details and for a password, if applicable.

### Change Access Level

To change the access level:

1. Contact Adept Technical Support to verify if your version allows end-user access to restricted parameters. If so, Adept can provide you with the required password.
2. From the Controller menu, select Change Access Level to display the following dialog:

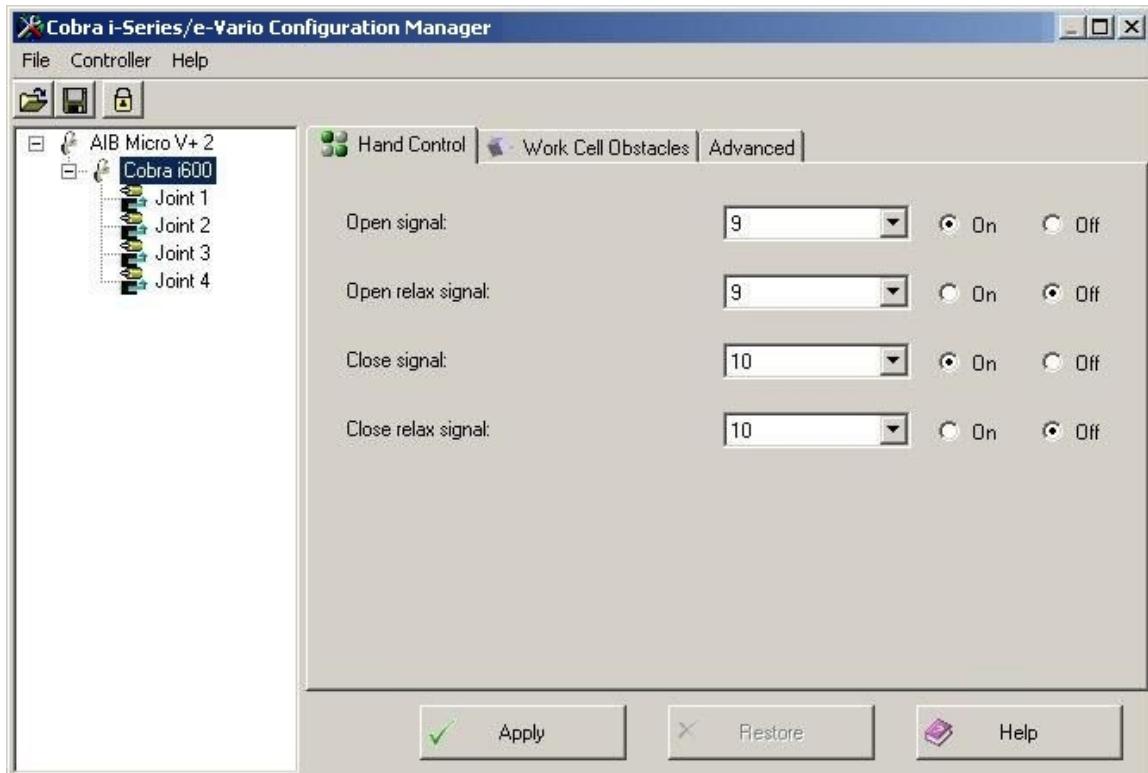


*Access Level Change Dialog*

3. Enter the correct password to change the access level.
4. Click **OK** to save the changes.

## Robot Configuration Pages

When the robot node is selected in the system tree view, the Robot Configuration pages display in the Configuration View. The view includes three pages: Hand Control, Workcell Obstacles and Advanced configuration.



*Robot Configuration Pages*

### Tabs

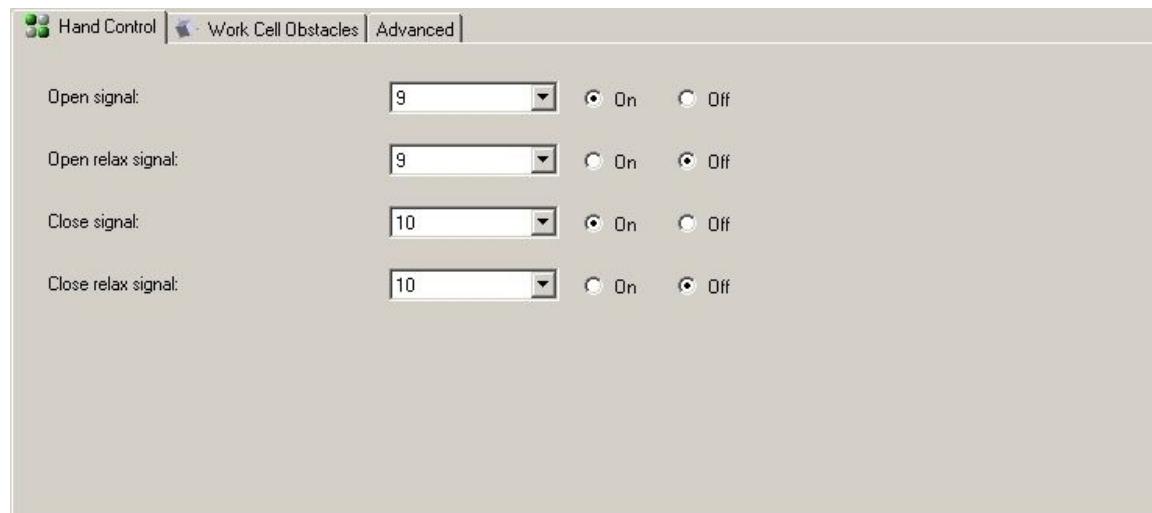
Item	Description
Hand Control	Displays the configuration page for hand control open and close signals. For details, see <a href="#">Specify Hand Control Signals</a> on page 278.
Workcell Obstacles	Displays the configuration page to specify parameters for obstacles in the Workcell. For details, see <a href="#">Manage Obstacles in the Workcell</a> on page 281.
Advanced	Displays advanced configuration options. For details, see <a href="#">Advanced Robot Configuration</a> on page 283.

## Buttons

Item	Description
Apply	Configuration data changes remain local to the Configuration Manager tool until they are saved to the controller using the Apply button. The Apply button opens the Apply Changes dialog. For details, see <a href="#">Apply Changes to the Controller</a> on page 287.
Restore	In the event of an error, the Restore button is available to reset the configuration data. The Restore button is available only after you have edited the data. The Restore button opens the Restore Configuration dialog. For details, see <a href="#">Restore Configuration</a> on page 294.
Help	Launches the online help documentation.

## Hand Control Signals

When the robot node is selected in the system tree view, click the Hand Control tab to display the Hand Control parameters for the selected robot. This page allows you to view and edit the digital signals to open and close the gripper hand. For details on setting the signals, see [Specify Hand Control Signals](#) on page 278.



*The Hand Control Signals tab*

**Signal Fields**

Item	Description
On/Off	Each signal field has an On or Off option which determines how the hand control responds to the specified signal. <b>On</b> indicates that the hand will perform the specified operation when the signal is turned on. <b>Off</b> indicates that the hand will perform the specified operation when the signal is turned off.
	The signals listed in the selection list for each signal field include both the hard and soft signals.
Open Signal	Specifies the signal used to Open the hand control.
Open Relax Signal	Specifies the signal to relax the pneumatic hand after it opens.
Close Signal	Specifies the signal used to Close the hand control.
Close Relax Signal	Specifies the signal used to relax the pneumatic hand after it closes.

**Specify Hand Control Signals**

The numbers of the digital signals accessed by MicroV+ for controlling the end-effector are stored on the MicroV+ system disk. The Hand Control Configuration page provides fields to change these specifications. Choosing the Off option for a specified signal has the effect of setting a negative signal number which indicates that the gripper will be activated when the signal is deasserted.

For additional information on the digital signal configuration, see the MicroV+ help topic Understanding Digital I/O Signal Mapping.

When the robot is running, you can monitor the digital signals using the Digital I/O Tool available in ACE. For details, see Digital I/O Window on page 166.

**To specify the Hand Control Signals:**

1. Click on the robot node in the tree view:

*Robot Node in Tree View*

2. Click the Hand Control tab to view the Hand Control Configuration page.
3. For each signal field, click the arrow to view a drop-down list of the available signals:

Open signal:	9	<input type="button" value="▼"/>	<input type="radio"/> On	<input checked="" type="radio"/> Off
Open relax signal:	10	<input type="button" value="▼"/>	<input type="radio"/> On	<input checked="" type="radio"/> Off
Close signal:	9	<input type="button" value="▼"/>	<input checked="" type="radio"/> On	<input type="radio"/> Off
Close relax signal:	11	<input type="button" value="▼"/>	<input checked="" type="radio"/> On	<input type="radio"/> Off

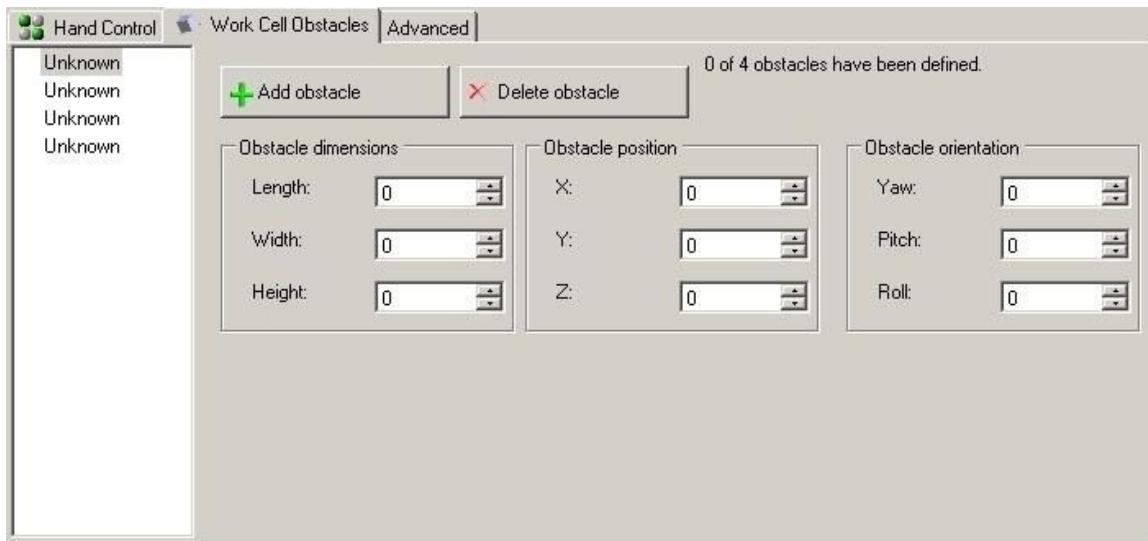
*Hand Control Signals*

4. Select the desired signal. For example, for Open signal: choose On to indicate that the hand will open when the specified signal is turned on; choose Off to indicate that the hand will open when the specified signal is turned off.
5. After your changes are complete, click **Apply** to save the changes.

**Workcell Obstacles**

The Workcell Obstacle page, shown in the following figure, allows you to define and edit workcell obstacles. As a new obstacle is created, it is displayed in the list on the left side of the page. For details on user interface elements see the descriptions below.

---



*Robot Configuration: Workcell Obstacles Tab*

### **Obstacle Management**

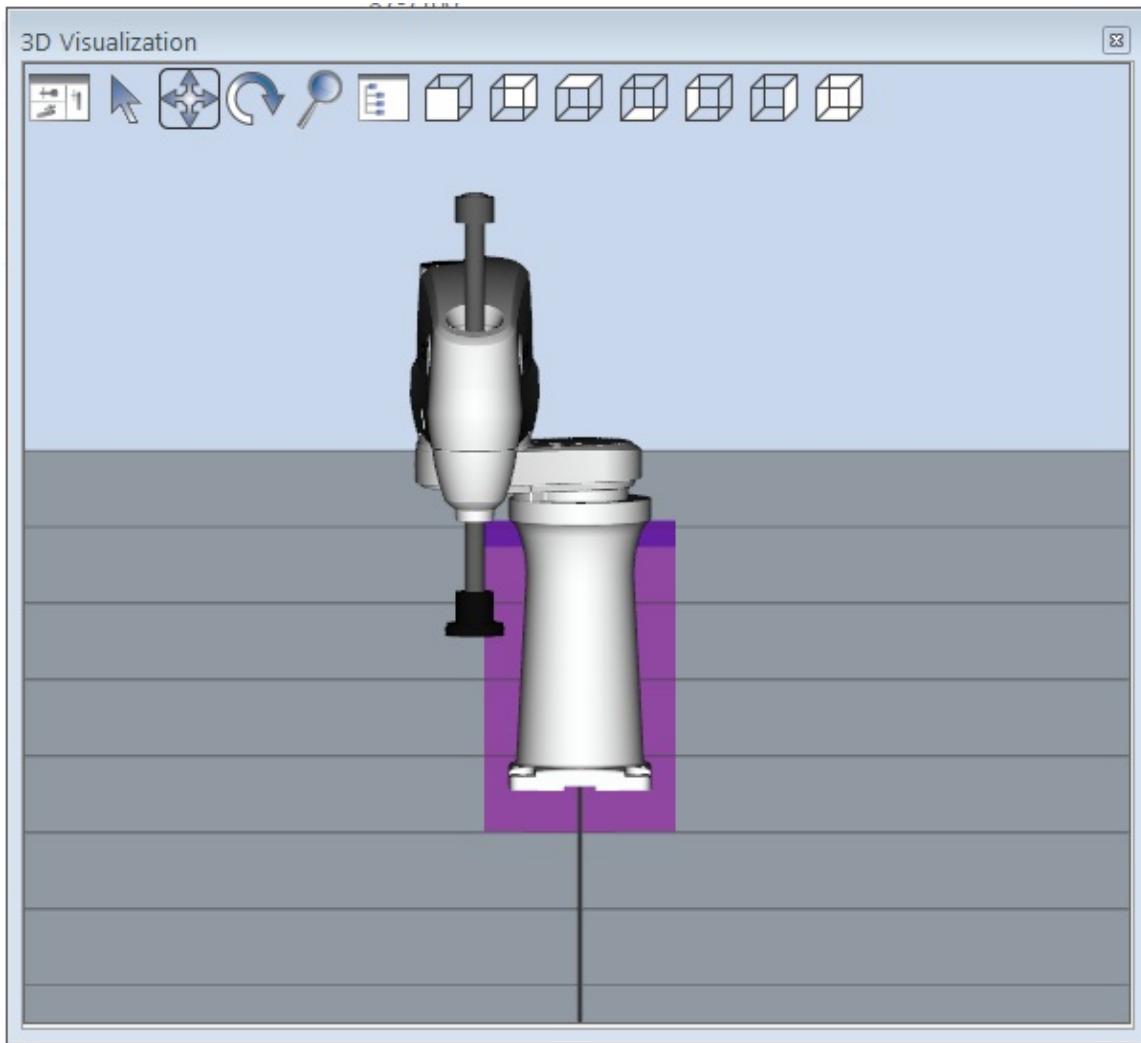
The following items are used to create the workcell obstacles. After you define an obstacle, you can change its parameters by clicking the obstacle name on the list. If you want to change the obstacle type (for example, from Box to Sphere), you must delete the current obstacle and then create (add) a new one to replace it.

Item	Description
Obstacle List	Lists the currently defined obstacles for the workcell. To edit the obstacle parameters, click on the obstacle name. The obstacle parameters fields will update to show the settings for that obstacle.
Add Obstacle	Click this button to display a drop-down list of available obstacles. Click the desired obstacle to add it to the Obstacle List and to specify the parameters.
Delete Obstacle	Select an obstacle in the Obstacle List, then click the Delete Obstacle button to remove it.
Obstacle parameters	The obstacle parameters include the dimension, position, and orientation of the obstacle in the workcell with respect to the base frame of the robot.

**NOTE:** There is no internal checking to detect conflicts between obstacles.

### **Viewing Workcell Obstacles**

As obstacles are created, and the configuration is applied, the obstacles can be viewed in the 3D Visualization window.



*Obstacle in the 3D Visualization Window*

You can view the obstacles, and other items in the workcell, from different perspectives and magnification. For details, see [3D Visualization](#) on page 176.

### **Manage Obstacles in the Workcell**

You can define travel constraints for a robot within the workcell from the Workcell Obstacles page. The constraints may be obstacles to avoid, or areas in which the robot tool tip must work. You can define up to six of these obstacles or areas.

During operation, the path of the robot tool tip is automatically tested to ensure that it does not collide with these objects under the following circumstances: when the robot is being moved in WORLD or TOOL

manual control mode; when the destination of each motion is being planned; and while straight-line motions are being performed.

From the Workcell Obstacles page, you can perform the following tasks:

- Add Obstacles
- Delete Obstacles
- Edit Obstacles

### **Add Obstacles**

To add an obstacle

1. In the tree view, select the robot you want to configure.
2. If the Workcell Obstacles page is not visible, click the Workcell Obstacles tab in the Configuration View.
3. Click **Add Obstacle** to display a list of available obstacles.
4. Select the type of obstacle or area that most closely matches the motion limitations you need to accommodate in your workcell. For example, if you want the robot to avoid a box-shaped obstacle, select Avoid Box. If you want the robot to work within a box-shaped area, select Inside Box.

After you select the obstacle, the Defined Obstacle list is updated. You can define up to six obstacles for the current workcell.

5. Specify the parameters for the obstacle dimensions, position, and orientation in the corresponding fields. The location of the obstacle is defined with respect to the base reference frame of the robot.
6. Click **Apply** to save your changes.

### **Delete Obstacles**

To delete an obstacle:

1. Select the obstacle name in the Defined Obstacle list.
2. Click **Delete**.
3. Click **Apply** to save your changes.

### **Edit Obstacles**

To edit the an obstacle:

1. Select the obstacle name in the Defined Obstacle list.
2. Enter the new values in the dimension, position, and orientation fields, as needed.
3. Click **Apply** to save your changes.

## Advanced Robot Configuration

When the robot node is selected in the system tree view, click the Advanced tab to display the advanced configuration options. This page allows you to select options that control E-STOP and calibration behavior. For details on selecting the options, see Select Advanced Configuration Options on page 284.



*Robot Configuration: Advanced Tab*

### Option Fields

Item	Description
Retract Z axis on CALIBRATE	During calibration of Joint 4, the Z-axis moves slightly. To protect the tool flange or gripper, select this option to retract the Z-axis before Joint 4 moves during the execution of the CALIBRATE instruction.
Break-away E-STOP	The Break-away E-STOP function is provided to enable a high power shutdown from the outer link area. For example, it would be used if you want a break-away gripper to shut down robot high power. It essentially lets you disable high power through a user relay circuit inside the robot. This feature is set to OFF at the factory. Check this box to enable the Break-away E-STOP if required.
E-STOP reflected on output 1	Select this option to enable the indicator signal for the state of the emergency stop circuits. The indicator-- Digital Output 1-- is ON when the E-Stop circuits

Item	Description
	are closed (E-Stop buttons are not pushed, muted safety gates are closed, etc..) and OFF when an E-Stop circuit is open (e.g. an E-Stop button is pushed).

## Select Advanced Configuration Options

The Advanced Configuration page provides options to control Emergency Stop (ESTOP) and calibration behavior.

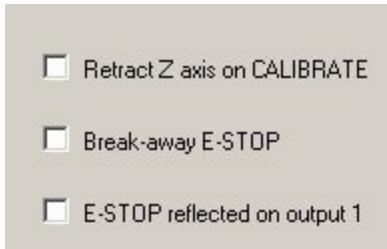
To specify the Advanced Options:

1. Click on the robot node in the tree view:



*Robot Node in Tree View*

2. Click the Advanced tab to view the advanced configuration options.
3. Use the check box for each option to enable or disable the option.



*Advanced Configuration Options*

4. After your changes are complete, click **Apply** to save the changes.

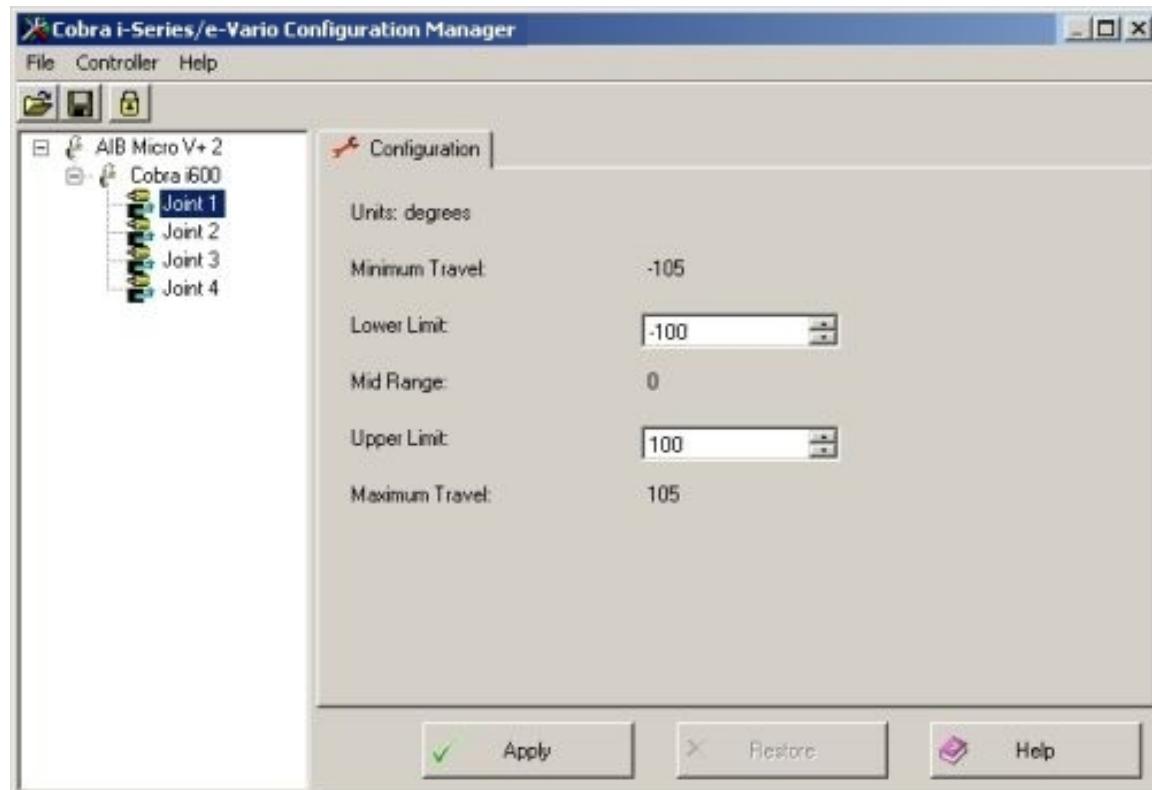
## Edit Joint Limits

The motion limits for the individual robot joints can be changed from their standard settings to limit robot joint travel. These software-defined joint limits (soft stops) are intended to protect the robot joints from colliding with custom workcell hardware. After the robot is calibrated, it is not possible to exceed the limits under program or pendant control. These limits are often set just inside of the hardware limit switches. Although MicroV+ should prevent the robot from moving beyond the soft stops during normal operation, safety considerations require that hardware limit switches also be used.

On a password-protected robot, the maximum and minimum joint limits can only be changed after entering the correct password. For details, see Password-Protect Configuration Data on page 275.



**WARNING:** Before attempting to move the robot with a MicroV+ program, check that the software joint limits are set to the correct values. Otherwise, the robot could move to an unintended location, which may damage the robot or other equipment in the workcell.



### **Joint View**

When the Joint node is selected in the system tree view, the joint configuration page is displayed. This page allows you to view and edit the default travel limits for the selected joint to limit robot joint travel. Lower and Upper limits are always editable. Minimum and Maximum travel and Mid range are not editable in restricted-access mode.

- For details on setting joint limits, see Specify Joint Limits on page 286.
- For details on user interface elements see the descriptions below.
- For details on changing restricted access to parameters see Change Access Level on page 275.

***Travel Limits***

Item	Description
Minimum Travel	The minimum physical location to which the joint may travel before hitting a physical hardstop. (This item is read-only in restricted-access mode.)
Lower Limit	The minimum physical location to which the joint may travel before disabling power. This value must be greater than the Minimum Travel limit to provide a safety buffer so that the joint does not travel to a physical hardstop. If you try to move the robot beyond the limit of the joint, you will receive an "out of range" message or error. The Lower Limit is also referred to as a "soft stop".
Mid Range	Indicates the approximate middle of travel for the joint. This value can be used to specify the safe position for the joint. (This item is read-only in restricted-access mode.)
Upper Limit	The maximum physical location that the joint may travel before disabling power. This value must be smaller than the Maximum Travel limit to provide a safety buffer so that the joint does not travel to a physical hardstop. If you try to move the robot beyond the limit of the joint, you will receive an "out of range" message or error. The Upper Limit is also referred to as a "soft stop".
Maximum Travel	Specifies the maximum physical location that the joint may travel before hitting a physical hardstop. (This item is read-only in restricted-access mode.)

***Specify Joint Limits***

1. Click the joint to configure in the tree view. The fields on the configuration page will update to display the current settings for the joint.
2. To change a limit, position the cursor in the field and enter a value, or use the arrow buttons on the left of the field to adjust the value up or down.
3. After making changes, click **Apply** to save your changes.

## Apply Changes to the Controller

Changes to the configuration data are stored in the Configuration Manager until they are written to the controller using the Apply button. When you apply the changes, you have two options:

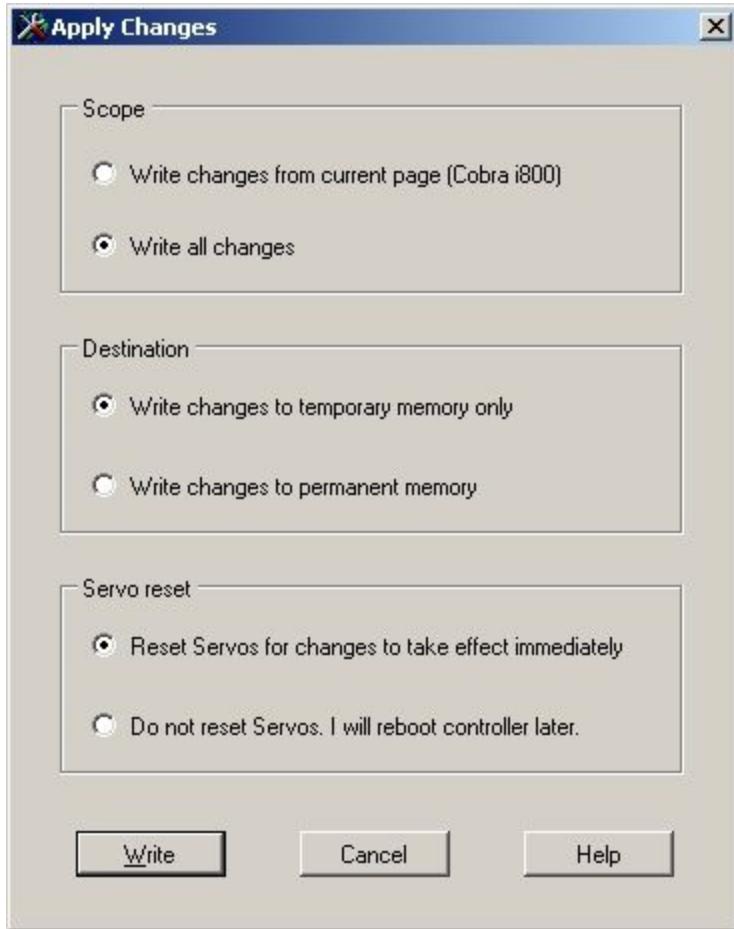
- Save the changes temporarily on the controller by writing them to the RAM.

Choose this option to test the new settings before saving them permanently. When you select this option, the changes are lost when the controller is rebooted.

- Save the changes permanently to the controller NVRAM (non-volatile) memory.

Choose this option to write the changes permanently. The updated settings will be loaded when the controller is rebooted.

**NOTE:** The Apply Changes to Controller function is different than saving data to a local file (on your PC). Data is saved to a file to back up the configuration or to move the same configuration from one system to another. For details, see Cobra i600/i800 Robot Configuration on page 265.



Apply Changes Dialog

## Option Descriptions

Item	Description	
Scope	Scope options determine whether to save all configuration changes made or only made on the current configuration page.	
Write changes from current page	Select this option to save only changes from the current configuration page. The name of the current page is displayed in parentheses.	Select this option to save all changes on the current configuration page along with any other parameter changes you have made on other pages.

<b>Item</b>		<b>Description</b>
Destination		Destination options determine whether the configuration data is committed to the non-volatile memory on the controller.
	Write changes to temporary memory only	Writes the changes to the RAM on the controller. When the controller is rebooted, the changes will be lost.
	Write changes to permanent memory	Saves the configuration data to the non-volatile memory (NVRAM). When the controller is rebooted the updated configuration data will be loaded.
Servo Reset		Servo Reset options determine whether the servos are reset when the configuration data is written to the controller.
	Reset Servos for changes to take effect immediately	When you select this option, the Apply Changes operation issues a SRV.RESET command to restart the servos. This ensures that all configuration changes are immediately in effect.
	Do not reset Servos. I will reboot the controller later	When you select this option, the Apply Changes operation will execute without restarting the servos.

## Writing Data to the Controller

To write data to the controller:

1. After making changes to the configuration parameters, click **Apply** to display the Apply Changes dialog.
2. Choose the appropriate options to write the changes. For details on user interface elements, see Option Descriptions in the previous section.
3. If you close the Configuration Manager Utility program without saving the configuration changes to the controller, a message box will prompt you to save your configuration.
4. Click **Write** to write the changes using the selected options, or click **Cancel** to close the dialog without writing the changes.

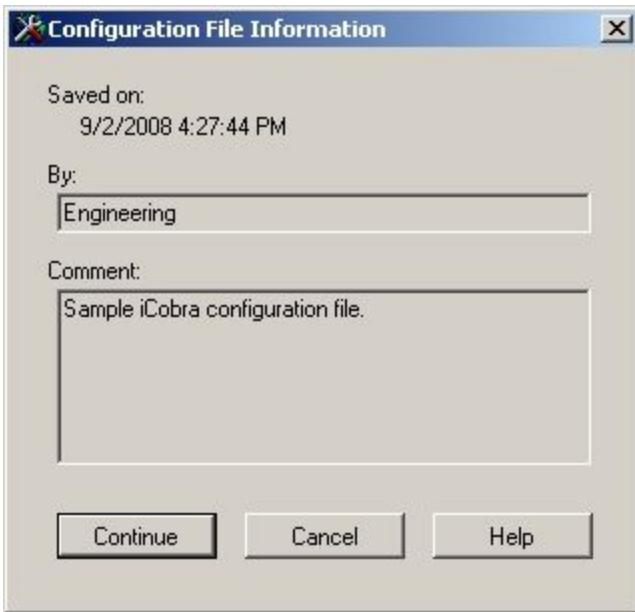
## Load Configuration Data from a File

Use this option to:

- Restore configuration data for the current robot from a backup file created using the Save Configuration option.
- Copy the configuration data from another robot to the currently selected robot.

After you have loaded a configuration file, use the Apply button to write the changes to the controller. To load configuration parameters from a file:

1. From the Configuration Manager menu, select **File > Load configuration** (or use shortcut: Ctrl+O) to display the file selection dialog.
2. Choose the desired .cfg file, then click **OK**. The configuration file information dialog is displayed:



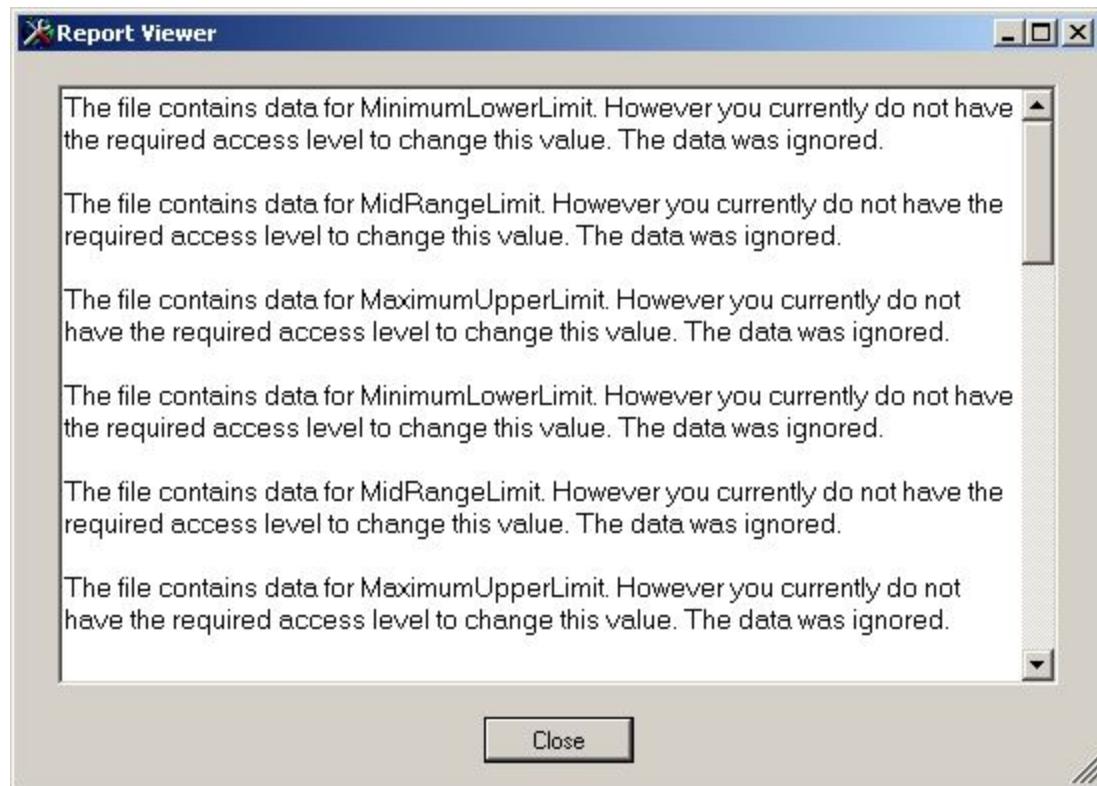
*Configuration File information Dialog*

3. Click **Continue** to load the file. The values in the configuration file will be loaded into the Configuration Manager.
4. After the file is loaded, use the Apply option to save the changes to the controller. If you don't have full access to modify the configuration parameters, the Configuration Manager Utility generates a Load-File Report listing the values that were not loaded. For details on access levels, see Change Access Level on page 275.

## Load-File Report

If the Configuration Manager Utility program encounters problems during the load process, a load-file report is generated with a list of warnings detailing the problems. Warnings may occur because you do not have full access to change the configuration data values, or if the file has been manually edited and has missing or extra information.

The following figure shows the Report Viewer, which contains a load-file report generated when the user did not have the access level required to edit the configuration data.



*Report Viewer with Load-File Report*

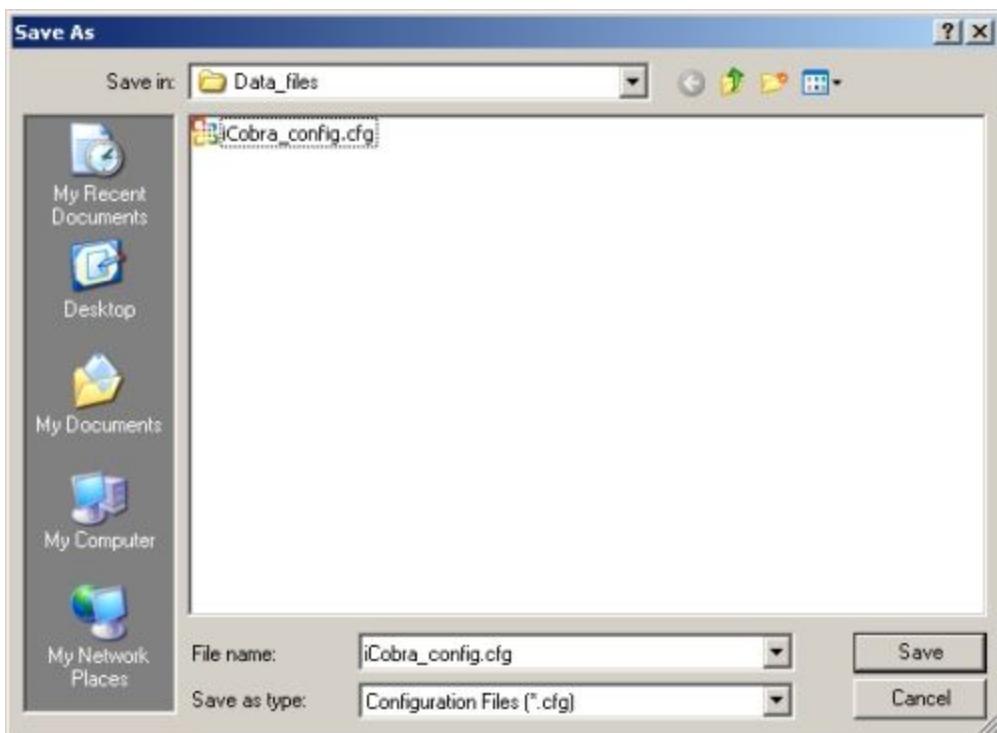
## Save Configuration Data to a File

Use this option to save the configuration data for the current device to a disk file. The configuration file can be used later to load the configuration data to another system or to restore the configuration in the event of a system problem.

When the Save is performed, the configuration data currently stored in the Configuration Manager Utility program is written to a file.

To save configuration parameters:

1. Before saving, make sure that you have written all of your current configuration changes to the controller.
2. From the Configuration Manager menu, select **File > Save configuration** (or use shortcut Ctrl+S) to display the File Save dialog.

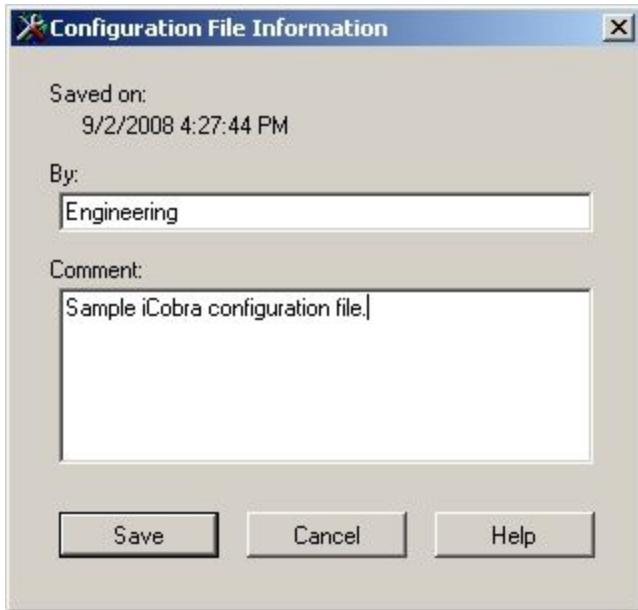


*Save As Dialog*

If you have not applied the changes to the controller, you will receive a warning message and an opportunity to apply the changes to the controller before saving the configuration data to a file.

3. Choose the directory for the file and enter the file name.

4. Click **Save**. The Configuration File Information dialog opens.



*Configuration File Information Dialog*

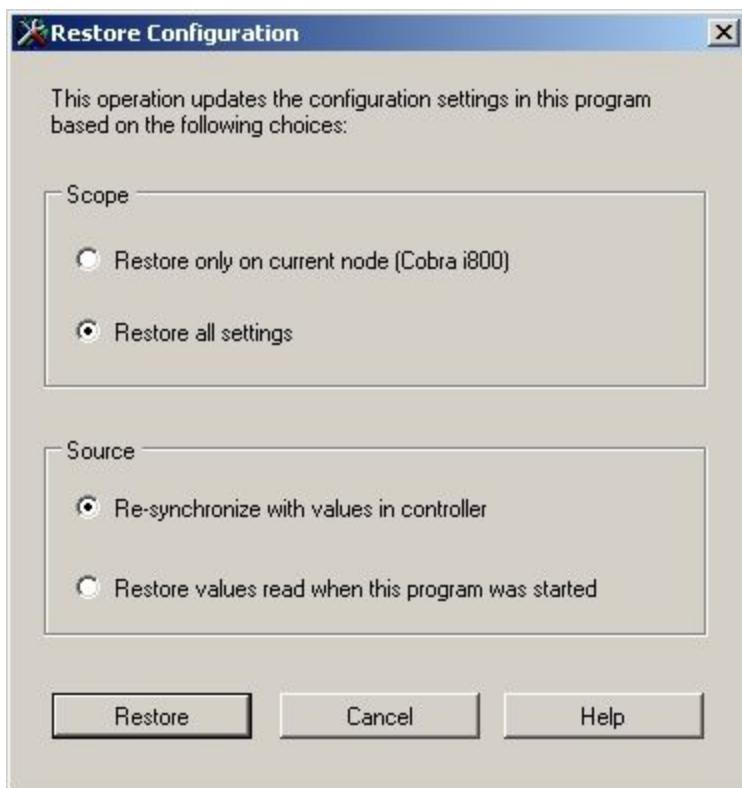
5. On the Configuration File Information dialog, enter a comment to describe the contents of the configuration file.
6. Click **Save**.

The values in the current configuration will be saved to the specified configuration file, with a .cfg extension.

## Restore Configuration

Restore allows you to reset the configuration parameters to either the values currently stored on the controller, or to the values read when the Configuration Manager utility program was launched.

This option is only available after you have edited the configuration parameters. For details on user interface elements, see the descriptions below.



*Restore Configuration Dialog*

### Option Descriptions

Item	Description
Restore only on current node	Restores the values on the currently displayed configuration page using the selected data source. The current page name is displayed in parentheses.
Restore all settings	Restores all values in the configuration data using the selected data source.
Re-synchronize with	Specifies the controller RAM as the data source to be used when restoring

<b>Item</b>	<b>Description</b>
values in controller	the data specified in the Scope group.
Restore values read when this program was started	<p>Specifies the configuration data loaded when the Configuration Manager Utility was started as the data source to be used when restoring the data specified in the Scope group.</p> <p>Choosing this option allows you to reverse any Apply actions that you may have performed during your current editing session.</p>

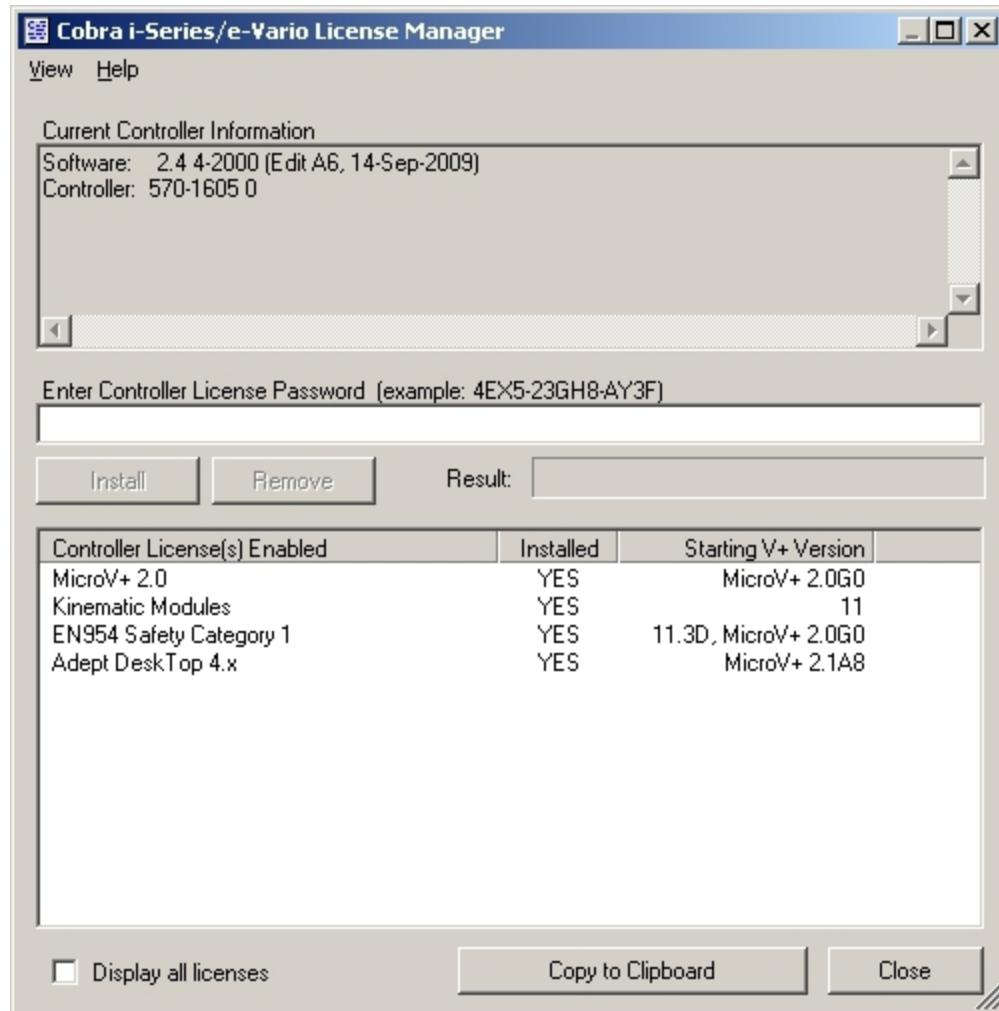
## Buttons

<b>Item</b>	<b>Description</b>
Restore	Completes the restore (reset) process, based on the selected options, and closes the dialog.
Cancel	Cancels the restore process and closes the dialog.
Help	Opens the online help documentation.

## License Manager

The License Manager allows you to configure the licenses for the Cobra i600/i800 (Cobra i-Series) robot running on the MicroV+ operating system. You can use the License Manager to track, manage, and activate Adept licenses, passwords, and security levels on the connected robot.

To open the License Manager, double-click the Cobra i-Series robot object in the Workspace Explorer, and then select License Manager from the Configure menu.



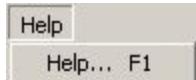
*License Manager*

## Menu Items

This section describes the selections available from the License Manager menu.

***View Menu*****All licenses**

Displays all known licenses. This is the same as checking the Display all licenses check box.

***Help Menu*****Help**

Displays this user guide.

**Editor Parameters**

The middle portion of the License Manager contains the editor parameters. These are used to configure various settings on the selected robot. The items in this area change, based on the selected tab.

Item	Description
Current Controller Information	Displays the controller ID and information in a scrolling panel. This allows you to verify the information that is provided on the license page.
Enter Controller License Password	To install or remove a license, enter the password provided by Adept Customer Service for the license, then click <b>Install</b> or <b>Remove</b> . The result of the installation process appears in the Results field.
Install	Click to install a license. This operation requires a valid password in the Enter Controller License Password field, which must be obtained from Adept Customer Service.
Remove	Click to remove the controller license. This operation requires a valid password in the Enter Controller License Password field, which must be obtained from Adept Customer Service.
Results	Displays the result of the installation process.
Display area	This area provides a list of licenses that are currently installed. Optionally, you can enable the <b>Display all licenses</b> check box to display all known licenses.
Display all licenses	Select this check box to display all known licenses. Otherwise, the display area will display the licenses that are currently installed.
Copy to Clipboard	Click to copy the contents of the display area to the Windows clipboard. The clipboard contents can be pasted into an email when corresponding with service personnel.
Close	Closes the License Manager.

## **Installing and Removing a License**

To install a license

1. Enter the License password in the Password field.
2. Click **Install**.
3. The result of the installation appears in the Results field.

To remove a license

1. Enter the License password in the Password field.
2. Click **Remove**.
3. The result of the removal process appears in the Results field.

## Switches and Parameters

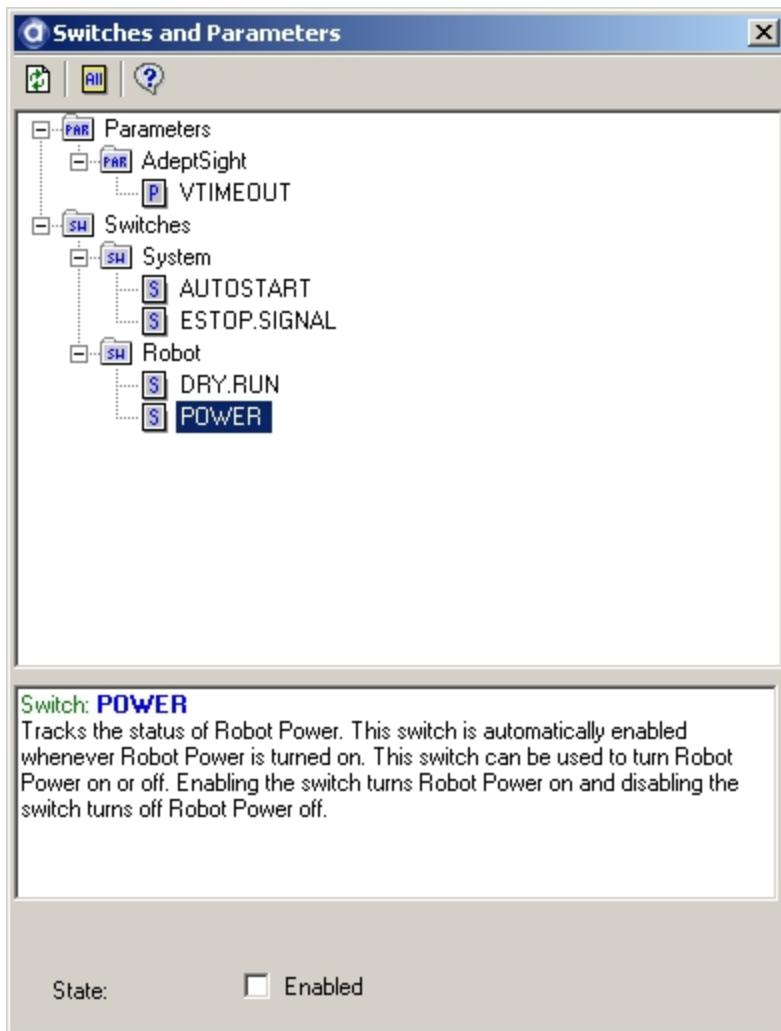
The Cobra i600/i800 Switches and Parameters dialog is used to access all available switches and parameters for the current robot connection. You can use the Switches and Parameters dialog to set parameter values or enable and disable switches.

**NOTE:** Only available/public switches and parameters are displayed.

### Opening the Switches and Parameters Editor

To open the Switches and Parameters editor, from the Cobra i-Series Robot editor menu, select:

**Configuration > Switches and Parameters**



*Switches and Parameters Dialog - Robot POWER Switch*

## Toolbar Items

This section describes the selections available from the Switches and Parameters editor toolbar.

Item	Description
 (Refresh)	Refreshes the switches and parameters list.
 (Show All)	Shows all switches and parameters.
 (Help)	When active, indicates that help information is available, in the bottom area of the editor, for the selected item.

## Value

The Value selector shows the current value of the selected parameter.



To modify the value of the selected parameter:

- Enter a new value in the text box, or
- Select a value using the up/down arrows.

**NOTE:** The upper and lower limits for the value are shown to the right of the text box.

## State

The State field shows the state of the selected switch.



To change the state of the selected switch:

- Enable the switch by selecting the Enabled check box.
- Disable the switch by clearing the Enabled check box.

## Vision

The topics in this chapter briefly describe the vision features in the ACE software, such as vision calibrations and vision programs. For detailed information on the vision features, see the *ACE Sight User's Guide*.

Vision Overview .....	301
-----------------------	-----

### Vision Overview

A vision system consists of some means of obtaining an image, and some means of processing that image.

- ACE Sight systems typically uses one or more cameras to capture images for processing. These images can be stored for later processing, if desired.
- Images are processed with ACE Sight vision tools. See Vision Tools on page 345.

This topic introduces some of the concepts and terms used in the other vision topics.

If you're new to vision guidance/inspection, you should read the section on Vision Basics, starting with [Vision Basics - Resolution on page 303](#).

### Vision Devices

A vision device can be either a physical camera or an emulated camera. Vision devices are added to the workspace as needed. You can have multiple vision devices in the workspace. See [Vision Devices on page 315](#).

#### Physical Camera

ACE Sight supports the following physical cameras:

- Basler 1394a
- Basler 1394b
- Basler GigE
- Basler Line Scan

A physical camera needs to be calibrated, to establish the relationship between physical distance in the camera's field of view and the pixels returned from the camera.

This type of calibration also helps correct for a certain amount of distortion in the lens. This calibration is covered in the section [Standalone Camera Calibration on page 320](#).

If the camera will be used with a robot and/or conveyor belt, the camera also needs to be calibrated with respect to those devices. Those are separate calibrations, distinct from the relatively simple distance-pixel calibration. These calibrations are added as objects in the workspace. See [ACE Sight Camera Calibration Interview Wizard on page 320](#).

[642.](#)

The area that a camera can "see" is called its field of view (FOV).

### ***Emulated Camera (emulation device)***

An emulated camera is a collection of stored images that can be used as input, via a virtual camera, to vision tools. Using an existing image, rather than taking an image with a camera, can be useful when setting up a new system, or when troubleshooting. Images taken at a system that is having trouble can be sent to a remote facility for analysis.

### ***Virtual Camera***

The interface between ACE Sight and Vision Devices is a virtual camera. Both physical and emulated cameras require the definition of an associated virtual camera. This allows ACE Sight to access either type of camera using the same interface.

## **Vision Tools**

A vision tool is software that performs one or more vision-related operations on its input. Each tool performs a specific function, such as inspection or some other processing. The input can be an image from a virtual camera or it can be the output of another vision tool.

**NOTE:** A new workspace doesn't have any tools in it. You add (create) them in the workspace as needed. See [Adding Vision Tools on page 352](#).

### **Sequences**

The output of one vision tool can be used as the input to another. In this way, the order of tool execution is implicitly defined. You can add a Sequence object to the workspace, to help you see the order in which tools will be executed. See [ACE Sight Sequence on page 608](#).

### **Related Topics**

[Adding Vision Devices](#)

[Adding Vision Tools](#)

[Calibration Overview](#)

## **Vision Basics**

The Vision Basics topics present a brief background of concepts that are needed when obtaining an image for use in a Adept system.

### ***Obtaining an Image***

A very brief overview of how digital cameras obtain an image, in three topics.

### **Resolution**

[Vision Basics - Resolution](#)

### **Mode**

[Vision Basics - Mode](#)

### **Camera Calibration and Lighting**

[Vision Basics - Calibration](#)

### ***Using an Image - Computer Image Analysis***

The computer system uses the information extracted from the matrix of pixels to perform vision tasks. The tools ACE Sight provides analyze the matrix and return information about object size, relative location, feature recognition, and object identification.

[Overview of ACE Sight Vision Tools](#)

### ***Blobs and Prototypes***

When ACE Sight examines the matrix of pixels returned by the camera, it looks for bounded regions. A bounded region is any contiguous section of pixels that is the opposite color from the background (and is within the minimum and maximum pixel limits). When a blob finder tool locates such a region, it reports that a Blob has been found and returns the information it discovered about the blob.

[Blob Analyzer](#)

## **Vision Basics - Resolution**

### ***Pixel***

The basic unit of an image is a pixel (picture element). It is the smallest unit of information that a vision system can return. The number of pixels a system can process determines the system's resolution and affects the computer processing time needed to analyze an image.

---

A pixel can be thought of as a single cell in a matrix that the camera overlays on the field of view (FOV). For a black-and-white camera, the value that is placed in a cell (pixel) is a shade of grey that represents the intensity of the light reflected from the corresponding area in the field of view.

Note that, in a line scan camera, one dimension of this matrix is retrieved by a row of sensors, each of which returns a pixel. The difference between matrix cameras and line scan is in the other dimension of the matrix. A line scan camera takes repeated images of a moving target to build the second dimension. The first dimension's resolution is just like a matrix camera, and dependent on the number of pixels that the camera returns. The second dimension's resolution is dependent on how often an image is acquired relative to the amount of movement of the target, typically moving on a conveyor belt.

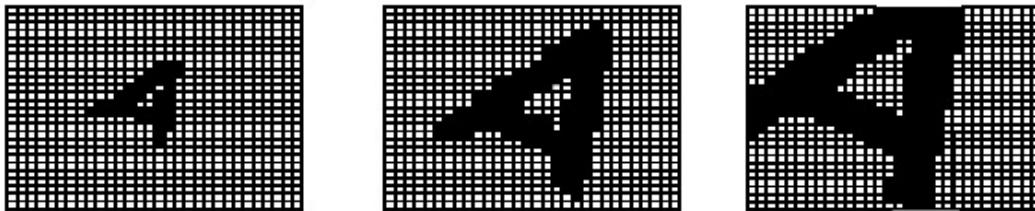
### ***Resolution***

The camera representation of our hypothetical  $22 \times 16$  camera shows very poor resolution due to the low density of pixels.

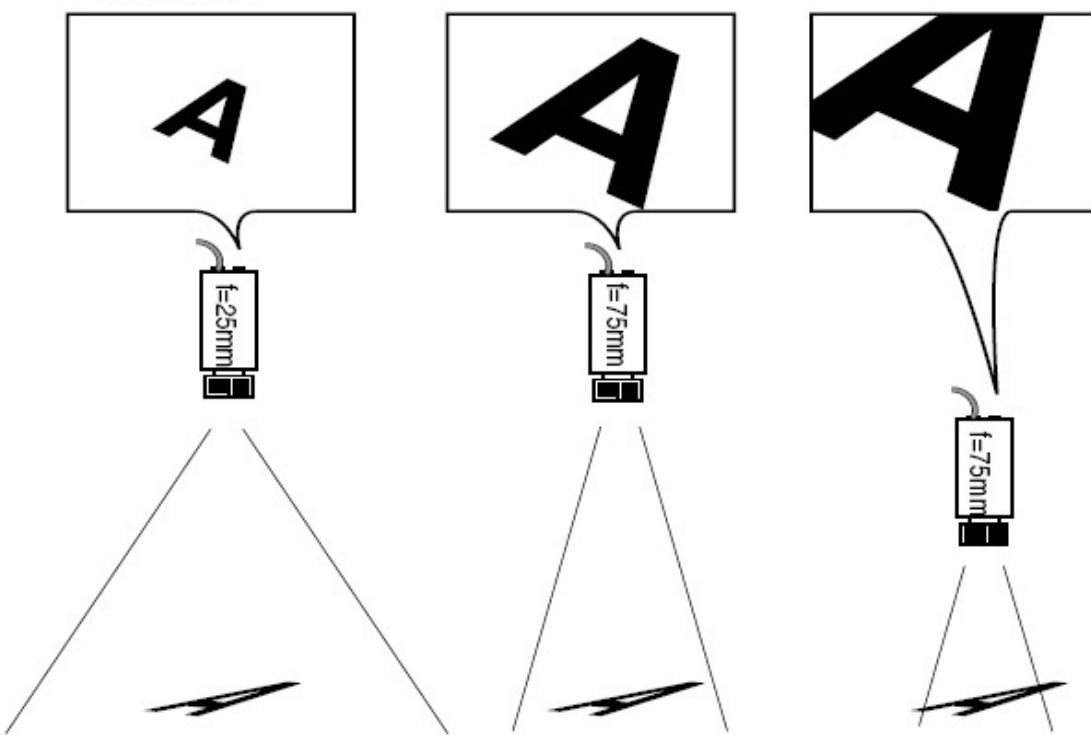
In addition to the number of pixels in the camera array, the lens focal length combined with the distance of an object from the camera has the greatest effect on the final resolution of whatever you are viewing. The following figures show the relationship between focal length and viewing distance. In general, the best resolution comes when the object of interest fills as much of the field of view as possible while still being in focus.

An important concept that is illustrated by these figures is the relationship between a pixel and the physical size of an object. A pixel always has a relative relationship to the size of an object. It has an absolute relationship only when you fix your viewing distance and focal length of the lens, and then calibrate the vision system. The camera calibration process (Standalone Camera Calibration on page 320) establishes an absolute relationship between a pixel and the actual dimensions of the field of view.

Vision System Representation



Field of View



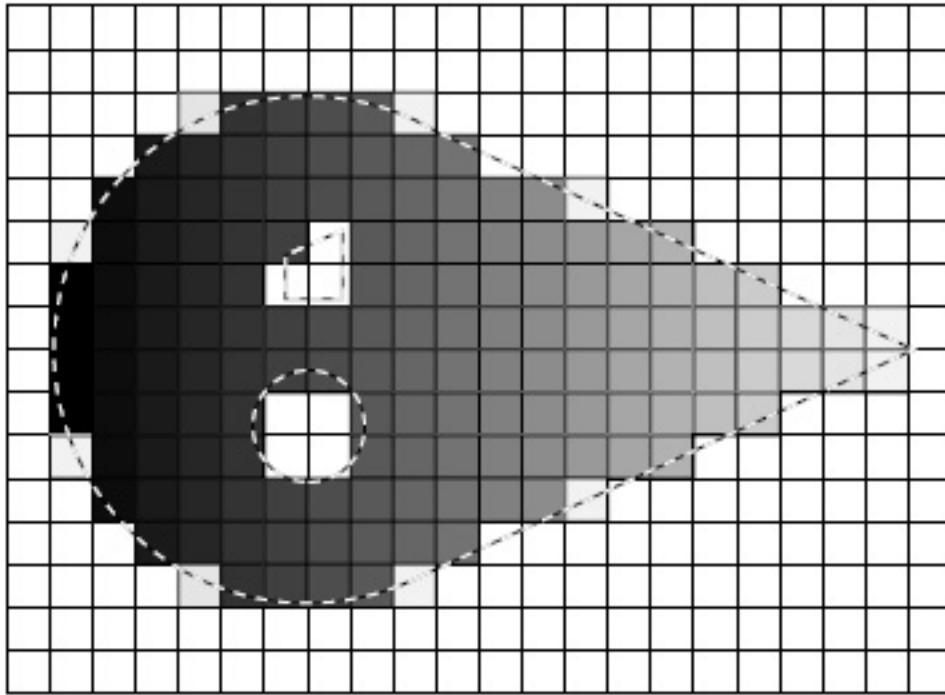
Resolution Factors

## Vision Basics - Mode

### **Binary vs. Greyscale**

The pixel value that a camera detects is a shade of grey that corresponds to the intensity of the light reflected from the corresponding area in the camera's field of view. The ability to analyze an image based on intensity values is referred to as greyscale vision. The following figure

shows how a 22 x 16 pixel camera sees the sample object. (The dashed lines show the original object; they are not actually “seen” by the system.)



*A Greyscale Image*

In addition to greyscale processing, ACE Sight can process image data in binary mode. In binary mode, all the cells with a value above a certain threshold are seen as white and those below that value are seen as black.

The following figure shows how the sample object is seen in binary mode.



*A Binary Image*

Depending on the tool and operation you select, ACE Sight processes either in binary mode or greyscale mode. The processing mode is selected by ACE Sight and, in most cases, you do not have to be concerned with which type of processing is taking place. In many cases, however, you have to set threshold values for binary processing or the edge strength value for grey-scale processing. This discussion gives you an idea of how to set those values.

[Sample Vision Matrix](#) shows a magnified section of the array of pixels that might be returned by a camera. In each pixel of the matrix is the greyscale value the camera has registered from the field of view.

88	82	84	88	85	83	80	93	102
88	80	78	80	80	78	73	94	100
85	79	80	78	77	74	65	91	99
38	35	40	35	39	74	77	70	65
20	25	23	28	37	69	64	60	57
22	26	22	28	40	65	64	59	34
24	28	24	30	37	60	58	56	66
21	22	23	27	38	60	67	65	67
23	22	22	25	38	59	64	67	66

*Sample Vision Matrix*

### **Binary Processing**

When the software processes image data in binary mode, each value in the matrix is compared with the threshold value. All the pixels with a value above the threshold value are considered white and all the pixels below this value are considered black. In binary mode, edges of objects within the field of view are found by looking for a change from a white section of pixels to a dark section of pixels. [Binary Representation of Sample Matrix](#) shows how the system would process the data from [a greyscale image](#) into a binary image if the threshold were set to 34. The threshold, when used, is set from within a vision tool.

There are several vision tools that allow you to set a binary threshold specifically for that tool. These tools create a new thresholded image within the area of the tool and perform their work based on this new data.

88	82	84	88	85	83	80	93	102
88	80	78	80	80	78	73	94	100
85	79	80	78	77	74	65	91	99
38	35	40	35	39	74	77	70	65
<b>20</b>	<b>25</b>	<b>23</b>	<b>28</b>	37	69	64	60	57
<b>22</b>	<b>26</b>	<b>22</b>	<b>28</b>	40	65	64	59	61
<b>24</b>	<b>28</b>	<b>24</b>	<b>30</b>	37	60	58	56	66
<b>21</b>	<b>22</b>	<b>23</b>	<b>27</b>	38	60	67	65	67
<b>23</b>	<b>22</b>	<b>22</b>	<b>25</b>	38	59	64	67	66

*Binary Representation of Sample Matrix*

There are actually two threshold values that can be used together to isolate a range of intensity values. When the 2<sup>nd</sup> threshold is given a value other than 0, all pixels with values between the 1<sup>st</sup> and 2<sup>nd</sup> threshold are white. [Setting Both Binary Thresholds](#) shows the sample array with the threshold set to 34 and 2<sup>nd</sup> threshold set to 89.

88	82	84	88	85	83	80	<b>93</b>	<b>102</b>
88	80	78	80	80	78	73	<b>94</b>	<b>100</b>
85	79	80	78	77	74	65	<b>91</b>	<b>99</b>
38	35	40	35	39	74	77	70	65
<b>20</b>	<b>25</b>	<b>23</b>	<b>28</b>	37	69	64	60	57
<b>22</b>	<b>26</b>	<b>22</b>	<b>28</b>	40	65	64	59	61
<b>24</b>	<b>28</b>	<b>24</b>	<b>30</b>	37	60	58	56	66
<b>21</b>	<b>22</b>	<b>23</b>	<b>27</b>	38	60	67	65	67
<b>23</b>	<b>22</b>	<b>22</b>	<b>25</b>	38	59	64	67	66

*Setting Both Binary Thresholds*

## Greyscale Processing

When the software processes the data from [a greyscale image](#) in greyscale mode, it looks at a three-by-three section of pixels and compares the difference in intensity values found in the neighboring pixels to the value of the variable Edge Strength. If the difference found exceeds Edge Strength, the system considers the three-by-three area to be part of an edge. [Grey-scale Representation of Sample Matrix](#) shows the edge the system finds in the data from the array in [Sample Vision Matrix](#) if Edge Strength is set to 20. (This figure is somewhat idealized to help illustrate the point.)

88	82	84	88	85	83	80	93	102
88	80	78	80	80	78	73	94	100
85	79	80	78	77	74	65	91	99
38	35	40	35	39	74	77	70	65
20	25	23	28	37	69	64	60	57
22	26	22	28	40	65	64	59	61
24	28	24	30	37	60	58	56	66
21	22	23	27	38	60	67	65	67
23	22	22	25	38	59	64	67	66

*Greyscale Representation of Sample Matrix*

The algorithm that ACE Sight uses when looking at the three-by-three neighborhood of pixels allows edges to be located with sub-pixel accuracy. Thus, greyscale processing is generally more accurate than binary processing.

## Contrast

There are two settings that allow you to control the level of contrast in an image.

### Gain

The sensors in the camera report an absolute intensity value. A camera operating in a low light environment returns values in a limited range at the low end of possible values (0-127) and the image would be very dark (similar to under-exposing a piece of photographic film). If all the objects in the field of view reflect light of a uniform intensity, the values returned are in a very narrow range. If the intensity values were multiplied by a given value, the relative differences between intensity values would increase and edges would be easier to detect. Gain is the ACE Sight parameter that performs this function.

### **Black Level**

Similarly, if a given value, or offset, were added to all the pixels, the absolute brightness of the image would increase, making it easier to work with. Black Level is the ACE Sight parameter that performs this function. If the value of Black Level is set too high, all the intensity values are bunched up around the high end of the scale and the images are washed out (similar to overexposing a piece of photographic film).

Using Gain and Black Level together allows you to maximize the brightness and contrast within an image. The binary threshold values are calculated after the effects of Gain and Black Level have been included in the image, so the difference between light and dark areas of the image is maximized and the binary image is most useful. Exposure, Gain, and Black Level can be set from the Edit window (within Acquisition Settings) of the Virtual Camera window.

Gain and Black Level should be adjusted after the camera Exposure has been set for maximum contrast.

## **Vision Basics - Camera Calibration and Lighting**

### ***Camera Calibration***

Camera calibration is the process of relating the image size of the camera to the actual size of the field of view. Two primary pieces of information are established when camera calibration is run:

- The number of pixels needed to see a millimeter of distance in the field of view  
This relationship is used by any tools that make distance measurements. It is also used by vision-guided motion devices.
- The ratio of the height of a pixel to its width

See Standalone Camera Calibration on page 320.

**NOTE:** The offset of the camera from the robot or other equipment is not part of this calibration. That information is obtained after the standalone camera calibration. See ACE Sight Calibrations on page 625.

Some important points to remember about camera calibration:

- The millimeter-to-pixel ratio is different for different camera-to-object distances. If you are making distance measurements of objects that are different distances from the camera, you have to create a calibration data set (virtual camera) for each distance.
- When you change the focus of a camera lens, the amount of the viewing area in the field of view changes. This means that the millimeter-to-pixel ratio changes, and cam-

era calibration created before the lens was refocused are not valid. The camera must be recalibrated.

### ***Lighting***

Type	Advantages	Disadvantages
Incandescent	Inexpensive, can be cycled	Short life (for AC lamps), heat
Fluorescent	Efficient, cool	Cannot be cycled, flickers
Tungsten-Halogen	High output	Heat, can't be cycled
Gas Strobes	High power, freezes motion	Expensive, poor CCD sensitivity
IR Laser	Good CCD sensitivity	Highly divergent, federally regulated
He-Ne Laser	Highly collimated, infinite depth of field	Fragile, expensive, bulky
IR & Laser Diodes	Not regulated	Low power, highly divergent

### *Types of Lighting*

#### **Diffuse**

Light is reflected from a shiny surface at the opposite angle from which it strikes the surface. If most of the light that is reflected from a surface comes from the same angle, the surface has sharp shadows and a great deal of surface glare. Diffuse lighting illuminates a surface with light that strikes the surface from as many different angles as possible, thus minimizing shadows, reflections, and the need for critically placed light sources. Fluorescent lighting is the most diffuse of the lighting types listed in [Types of Lighting](#). Diffuser plates and reflecting panels produce a more diffuse light. True diffuse lighting requires a parabolic shaped reflector. Applications with high-contrast, complicated objects, spherical objects, shiny objects, or objects that require multiple inspections of interior features are candidates for diffuse lighting.

#### **Back**

In backlighting, the light source (usually a diffuse source) is placed behind the object to be inspected. Backlighting effectively lights objects whose silhouettes are the critical feature. This is particularly effective if the objects are shiny or have highly variable surfaces.

## **Directional**

Incandescent floods, ring lights, and fiber lights mounted above or to the side of an object provide directional lighting. This lighting is the simplest to install, but effective vision operations depend on this type of light source remaining constant. If the light source dims, the object appears different to the camera. If the angle changes, shadows may be added that are interpreted as features of the object. This type of lighting is most effective with simple objects or objects where specific, highly-identifiable features are being inspected. Shiny surfaces or objects with variable surface brightness are difficult to inspect with directional lighting.

## **Structured**

In structured lighting, a highly collimated light source is applied to the object. Lasers are sources of collimated light. This type of lighting allows the vision system to detect three-dimensional features, such as depth changes in the surface plane or holes in the object. Shiny surfaces are not amenable to structured lighting.

## **Strobe**

Strobe lighting is required when the speed of moving objects exceeds one pixel every 17 milliseconds. Strobes cast harsh shadows, but can be diffused (see above).

### ***Filtering and Special Effects***

In many cases, specific lighting problems can be solved by placing an optical filter on the camera lens. The three most common filters used for black and white cameras are polarizing filters, color filters, and infrared filters.

## **Polarizing Filters**

Reflected light is highly polarized (the light waves have a predominant orientation about the wave axis). A polarizing filter can be adjusted so that light waves with a predominant orientation are filtered. If reflected glare from an object is a problem, a polarizing filter may be able to minimize the problem. A polarizing filter reduces the overall scene brightness so more intense lighting sources are needed with this type of filter. By adjusting the orientation of polarizing filters on both the light source and lens, you can significantly reduce ambient light and reduce shiny reflections.

## **Color Filters**

Color filters allow you to reduce or eliminate different colors of light that reach the camera. Color filters may enable the system to ignore irrelevant object features that are a given color, or ignore non-significant differences in an object that develop due to differences in the colors of the feature. Color filtration is difficult and should be attempted only when other avenues have been exhausted.

## **Infrared Filters**

CCD cameras usually have a relatively high sensitivity to infrared light. Warm or hot objects emit infrared light (which is invisible to the human eye). If you suspect your camera data may be inaccurate due to infrared emissions, try placing an infrared filter over the lens.

### ***Related Topics***

[Adding Vision Devices](#)

[Calibration Overview](#)

## Vision Devices

### ***Cameras***

The ACE Sight software supports the following cameras:

- Basler 1394a
- Basler 1394b
- Basler Gigabit Ethernet
- Emulation devices

### ***Calibration***

You should calibrate the camera before you create any vision tools. The basic camera calibration is a "spatial" calibration that corrects for perspective distortion, lens distortion, and defines the relationship between the camera pixels and real-world dimensions.

You can perform this calibration with just a camera and a calibration grid (or fixed-pixel calibration). See See "Standalone Camera Calibration".

**NOTE:** The offset of the camera from the robot or other equipment is not part of this calibration. That information is obtained during the robot-to-camera calibration. See ACE Sight Calibrations on page 625.

### ***Emulation Devices***

An Emulation Device is a stored collection of images, which the ACE Sight software can treat as if they were coming from a "live" (physical) camera. This is mostly used for working offline: when a camera is not available, or when viewing images from an application at a remote facility (you need a copy of the workspace and the sample images).

### ***Virtual Cameras***

The ACE Sight software uses a Virtual Camera to interface with either a physical camera or an emulation device. This allows the vision tools to interface with any image input in the same way, through the virtual camera.

A virtual camera can be added to the workspace at the same time that you add a camera or emulation device by checking the "Add Virtual Camera" box.

### ***Related Topics***

[Adding a Camera](#)

[Standalone Camera Calibration](#)

[Emulation Devices](#)

[Adding a Virtual Camera](#)

## **Adding a Camera**

All of the supported physical cameras use a Basler Pylon device driver.

### ***Adding a Camera to the Workspace***

To add a camera:

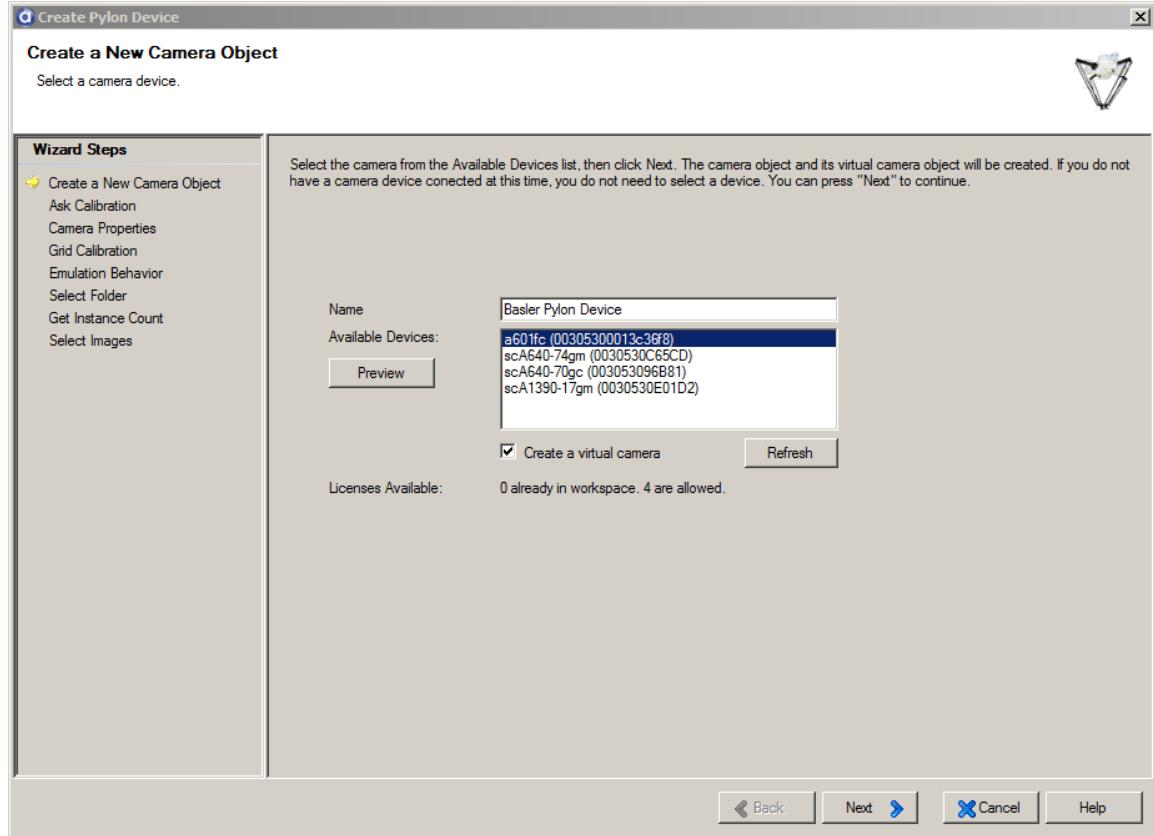
1. Right-click in the Tree structure pane of the ACE Workspace Explorer and select:

#### **New > Vision > Device > Basler Pylon Device**

- Basler Pylon Device will be displayed in a pop-up window, along with a list of available devices.  
If there is only one camera attached to the system, that will be the only camera displayed.
- For most devices, you are offered the option of creating a virtual camera when you add the device, by checking a box in the camera-creation window.  
This box is checked by default. Click it if you want to un-check it.
- In Emulation Mode, you do not have to select a camera.

## Adding a Camera

---

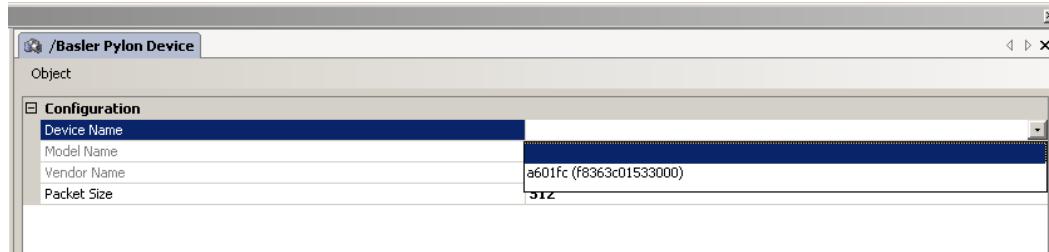


### Adding a Basler Camera

2. Click FinishNext.

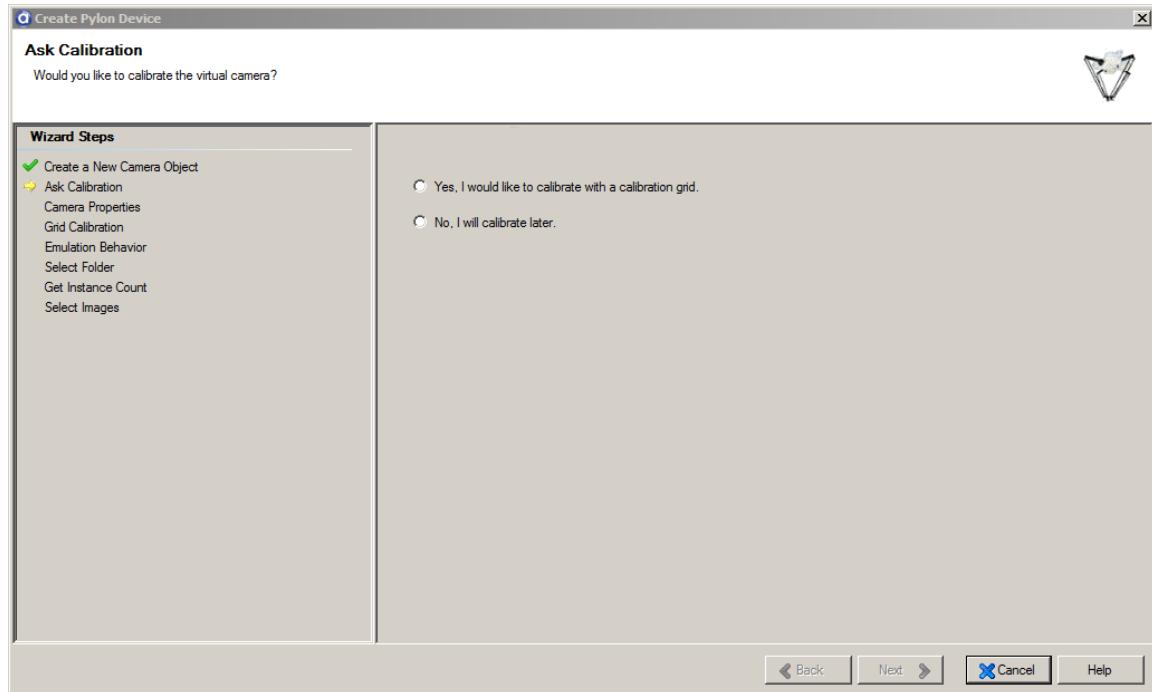
A camera is added to the ACE workspace and the corresponding object editor opens:

**NOTE:** The ACE Sight software opens the object editor for the tab that is highlighted.



### *Object Editor for Basler Pylon Device*

3. You will be asked if you want to calibrate the camera now or later.

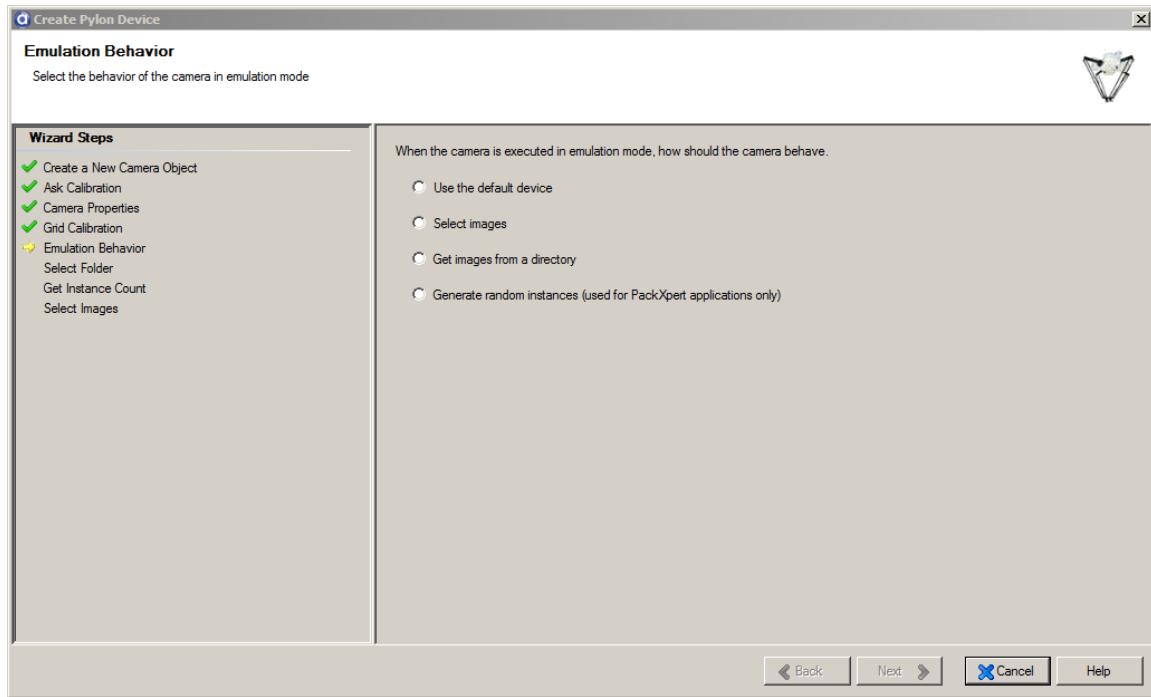


### *Camera Calibration Now or Later*

4. In Emulation Mode, you will be asked where you want the images to come from.

## Adding a Camera

---



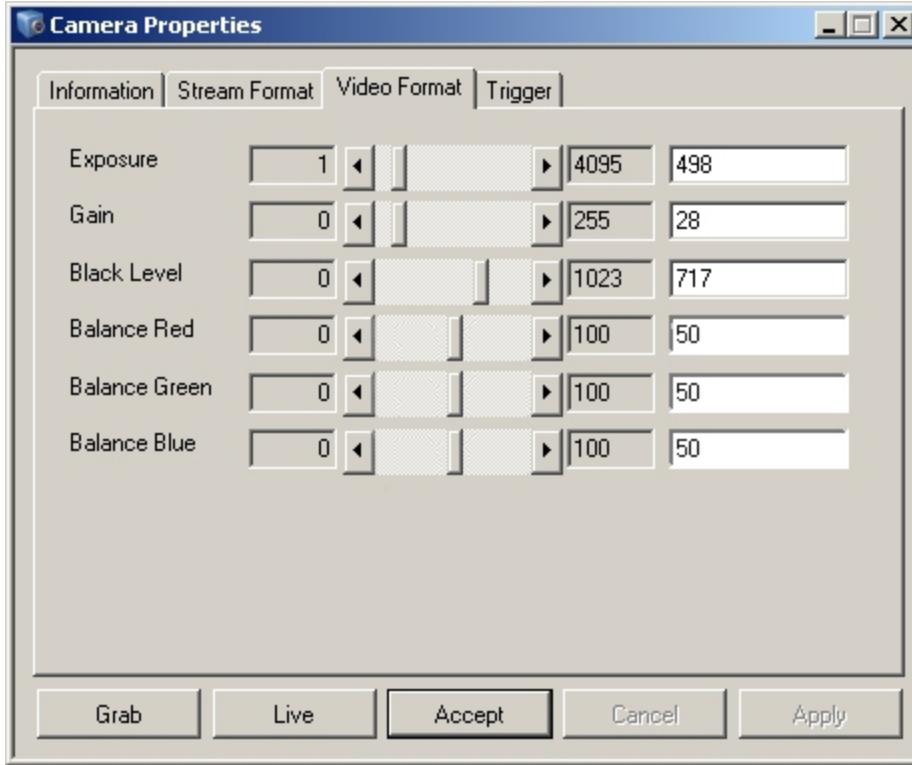
### *Emulation Mode Image Source Selection*

If you check "Get images from a directory", and supply a directory that has no valid images, the ACE Sight software will generate an error message.

5. Click the Device Name field, then click the down-arrow to display a list of available cameras.
6. Click on the physical camera you are adding to select it.

You can right-click the object name in the Tree structure to rename the object.

Click Edit in the Acquisition Settings pane to open the Camera Properties dialog. This is used to view information about the camera and make changes to shutter, gain, exposure, and other settings. See the following figure and the Camera Properties on page 335 topic.



*Camera Properties Screen*

In the Acquisition Settings group, the current acquisition settings for the selected physical camera are automatically added.

#### ***Related Topics***

[Adding Vision Tools](#)

[Calibration Overview](#)

[Camera Properties](#)

### **Standalone Camera Calibration**

You should calibrate the camera before you create any vision tools. The basic camera calibration is a "spatial" calibration that corrects for perspective distortion, lens distortion, and defines the relationship between the camera pixels and real-world dimensions.

You can perform this calibration with just a camera and a calibration grid or fixed-pixel calibration.

**NOTE:** The offset of the camera from the robot or other equipment is not part of this calibration. That information is obtained during the robot-to-camera calibration. Standalone calibration should be performed first. See ACE Sight Calibrations on page 625.

### **Calibration with a Grid of Dots**

The camera calibration usually uses a *grid of dots* target. A sample dot target file is provided with the ACE Sight installation. See See "Creating a Dot Target".

**NOTE:** The sample target is intended for teaching purposes only; it is not a genuine, accurate vision target.

Because any error introduced into the vision system at this calibration will be carried into all following calibrations, the target should be as accurate as possible.

Adept recommends using a commercial-grade target, available from suppliers such as:

- Edmund Optics ([edmundoptics.com](http://edmundoptics.com), Fixed Frequency Grid Distortion Targets)
- or
- Applied Image Inc ([aig-imaging.com](http://aig-imaging.com), Accu-Place™ Dot Pattern)

### **Before Calibrating**

Correct lighting on the target is important to obtain accurate calibration.

Before starting this calibration, make sure that the entire area covered by the camera field of view is within the robot work range, if required by your application.

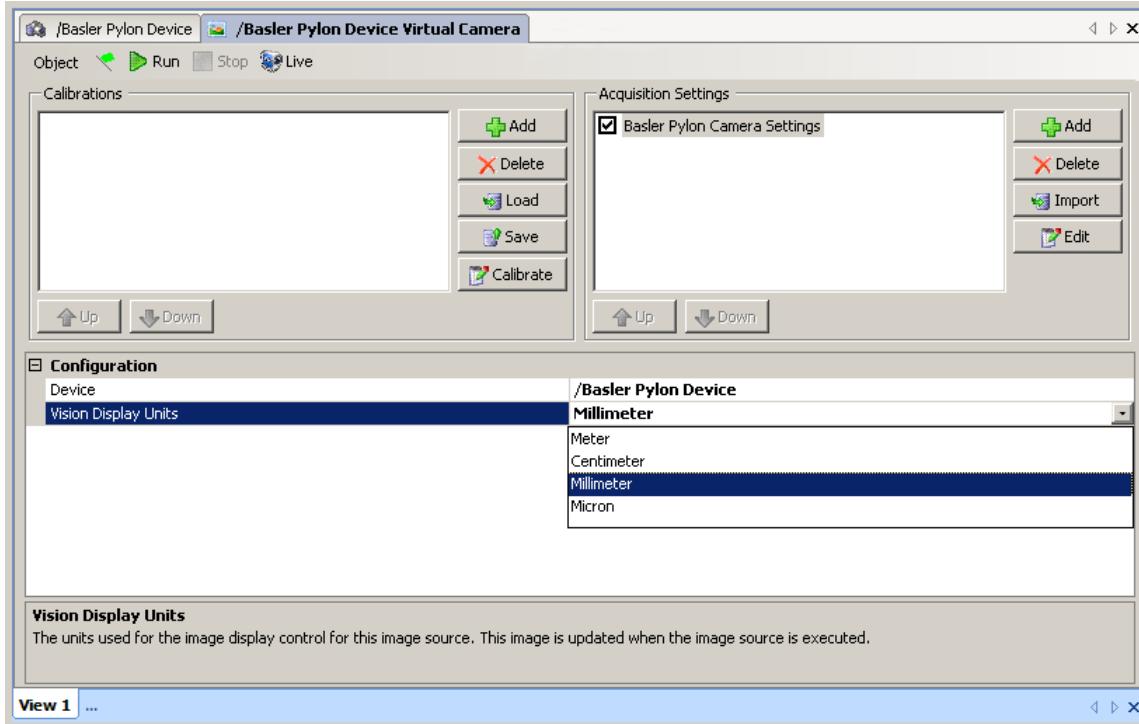
If you have more than one camera installed, select the camera you want to calibrate:

1. Click Device, to highlight that field.
2. Click the drop-down arrow, to display the available cameras.
3. Highlight the camera you want to calibrate.
4. Click OK.

The Vision Display Units field controls the image display from this source. The default is Millimeter. For most applications, this is the setting you will want.

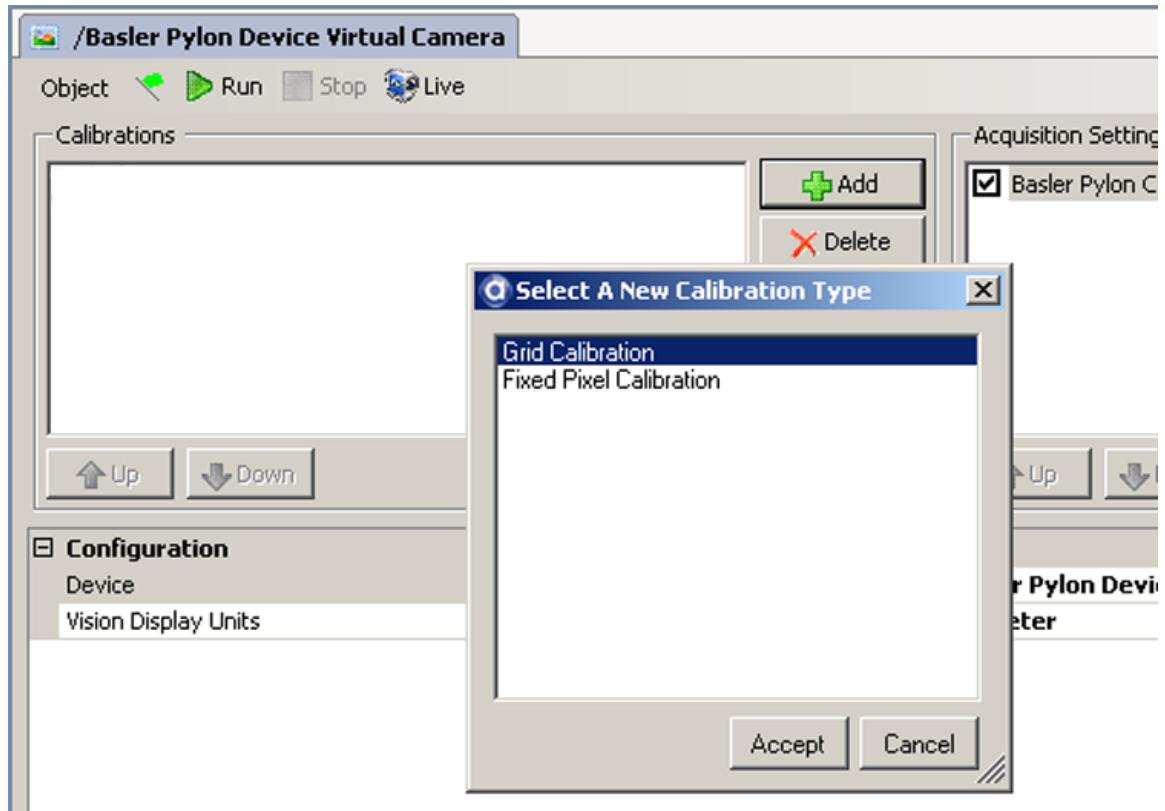
## Standalone Camera Calibration

---



### Performing a Calibration

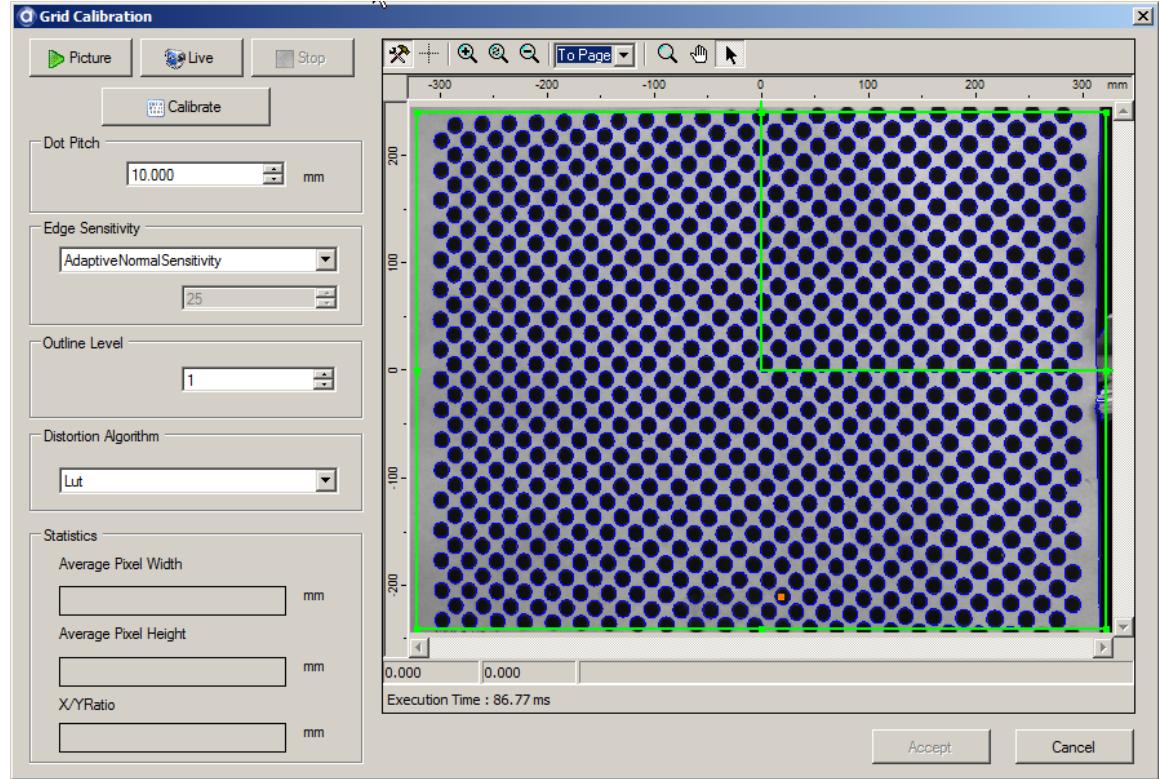
1. Click Add in the Calibrations group (within the Virtual Camera object editor window).
  - a. Select Grid Calibration from the displayed list.
  - b. Click Accept.



2. Click Live to view a live image from the camera input. The live image is displayed in the Vision Window.
  - a. The grid should cover as much of the camera field of view as possible, and the camera should be properly focused.
  - b. Click Stop to cancel the live input.
3. Enter the dot pitch (distance between dot centers) of the target in the Dot Pitch field. The background of this field will be yellow if the dot pitch value is not valid (such as 0).
4. Click Picture to acquire an image. The grid pattern appears in the Grid Calibration window.

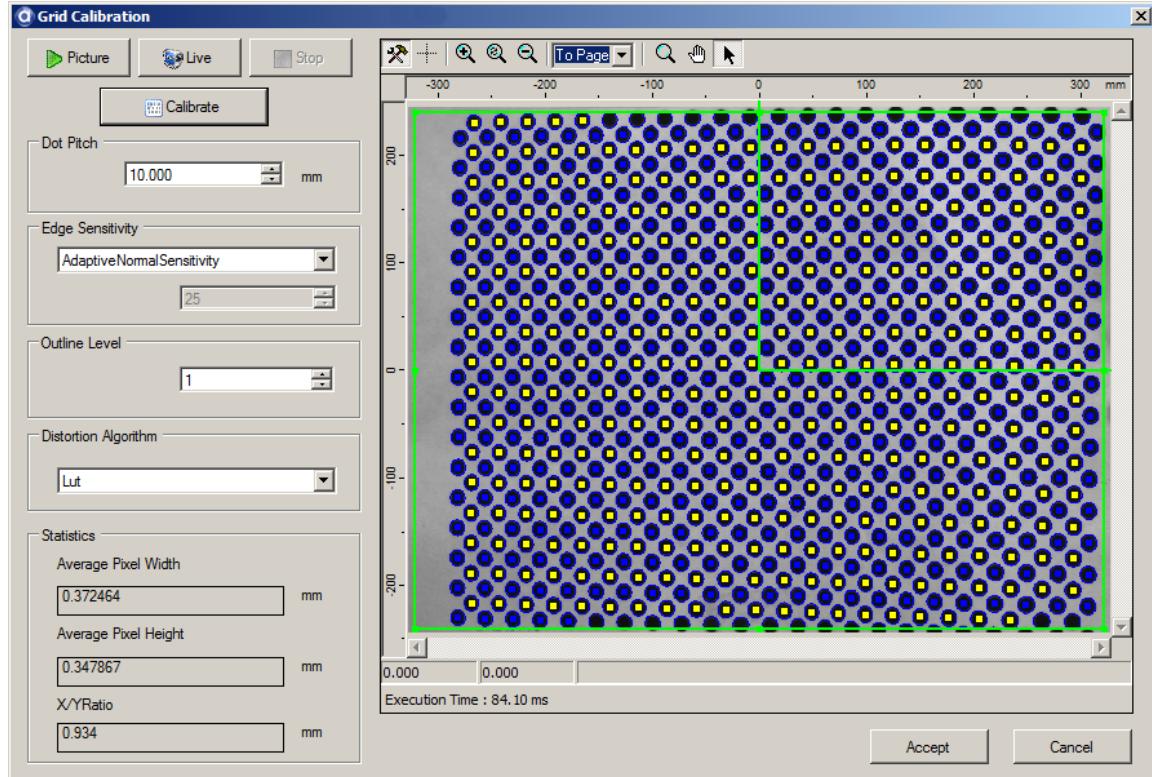
## Standalone Camera Calibration

---



*Grid Pattern, Before Calibration*

5. Click Calibrate to begin the calibration.  
The system scans the grid and places an overlay (yellow and blue squares) on the grid pattern.



*Grid Pattern, After Calibration*

**NOTE:** If you get an error message that there were an insufficient number of calibration dots found, your dot target is probably not sharp enough. It takes only 9 dots to calibrate. Try adjusting the Edge Sensitivity, so the calibration will find more dots.

For very small fields-of-view, it may be difficult to calibrate a camera if the dot grid is not high quality. Modifying the Outline Level may allow calibration in such cases, even with less-than-perfect dot grids.

If a dot is found that is less than one-half the average dot size, or greater than one and one-half the average dot size, the dot will be displayed in orange, to indicate that the calibration may be picking up artifacts within a dot, or outside of a dot. If this occurs, you need to adjust the Edge Sensitivity, lighting, or quality of the dot grid you are using.

6. Click Accept to save the calibration.

### ***Calibration with Fixed-Pixels***

Fixed-pixel calibration allows you to specify what physical distance is represented by each camera pixel. All camera pixels will be given the same dimension, which is not necessarily the case with a grid of dots. This method of camera calibration will not correct for lens distortion or perspective.

### **Related Topics**

[Vision Devices Overview](#)

[Adding a Physical Camera](#)

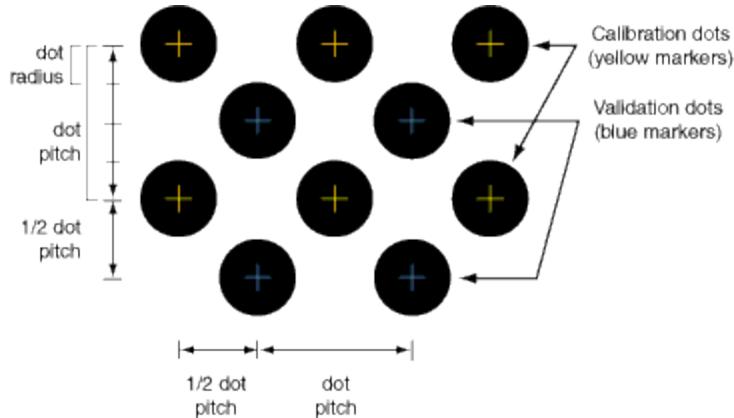
[Adding Vision Tools](#)

[Calibration Overview](#)

## **Creating a Dot Target**

The quality and precision of a grid of dots target have a direct impact on the overall precision of your application. Dot targets are commercially available, but you can also create your own targets by following the guidelines provided below.

- A dot target is made up of a matrix of evenly-spaced, identical calibration dots.
- A secondary matrix of validation dots can be added, offset to the matrix of calibration dots, to validate the calibration process. Although they are not absolutely required, these dots are useful for error calculations.
- Dots in both matrices should be identical in size and have the same dot pitch (the distance between the centers of two dots in the same matrix). Dot pitch must be the same in both X and Y directions.
- The offset between the calibration dot matrix and the validation dot matrix must be 1/2 dot pitch in the X and Y axes.



### *Specifications for a Grid of Dots Target*

- Dots should be round and well-contrasted.
- The recommended pitch range is 4 to 12 mm.  
Dot pitch should be four times the dot radius.
- Dot pitch must be the same in both X and Y axes.
- The target should cover the entire field of view.
- For best results, targets should be of high photo quality on a stable medium, not printed.

**NOTE:** You should measure the pitch of your dots after printing a grid, to confirm that your printer did not scale the grid. If the dots are not exactly the pitch you think, your camera calibration will be inaccurate.

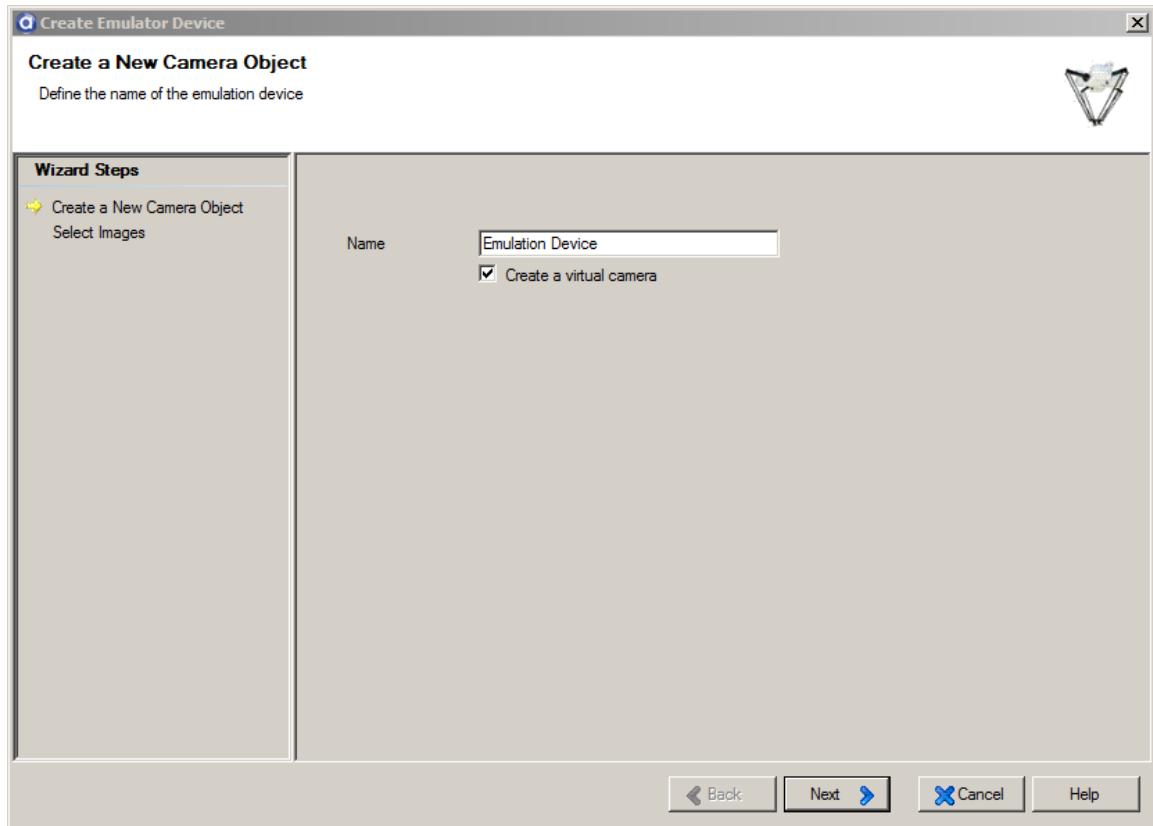
## **Adding an Emulation Device**

An Emulation Device is a stored collection of images, which the ACE Sight software can treat as if they were coming from a "live" (physical) camera. This is mostly used for working offline, when a camera is not available, or when viewing images from an application at a remote facility (you need a copy of the workspace and the sample images).

To add an Emulation Device:

1. Right-click in the Tree structure of the Workspace Explorer and select:

**New > Vision > Device > Emulation Device**



### *Creating an Emulation Device*

For Emulation Devices, you are offered the option of creating a virtual camera when the Emulation Device is added:

- a. Check "Create a virtual camera" (by default, this box is checked).
- b. Give the emulation device a name.

The name is just to help you remember which emulation device this is, because you can add multiple emulation devices to a workspace.

- c. Click FinishNext.

An Emulation Device and an Emulation Device Virtual Camera are added to the ACE Sight workspace. By default, the virtual camera for the emulation device will be named <emulation device name> Virtual Camera.

You will be asked to add images to this emulation device.

2. Click Add to add an image.

3. Select an image. You can browse to find your image.
4. Either click Add, to add another image, or click Finish.

You can right-click an object name in the Tree structure to rename the object.

You can Import an image to work with:

1. Double-click the Emulation Device in the Tree structure.

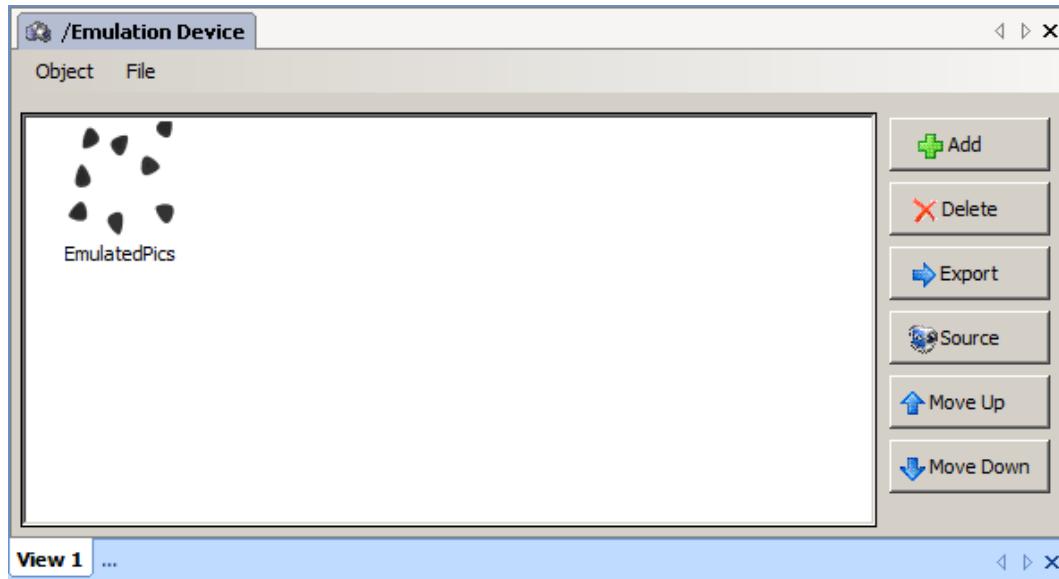
To use an image database (stored as .hdb):

- a. Select **File > Load from Database.**
- b. Select the desired database.  
You may need to set the "files of type" filter.
- c. Click Open.

The images in the database will be displayed in the Emulation Device object editor.

To select a single image:

- a. Click Add.
- b. Select an image to use. You may need to browse to find the image.  
Select the image type (if "all valid images" is not displayed).
- c. Click Open.  
The image will be displayed in the Emulation Device object editor.
- d. Either repeat, to add another single image, or click Finish.



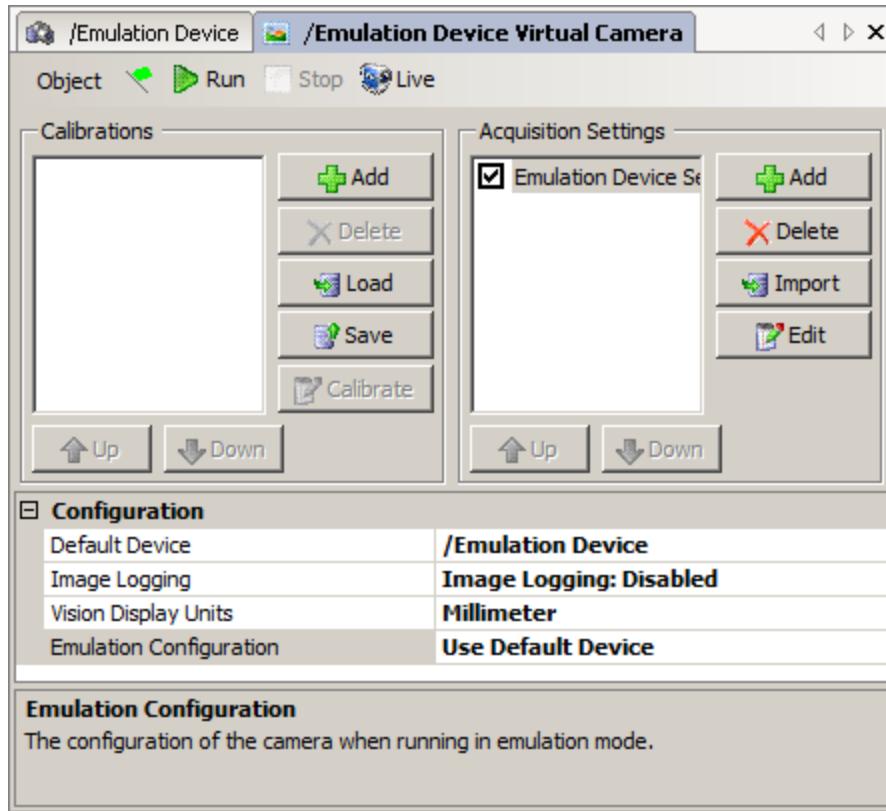
*Image Added to the Emulation Device*

When you have multiple images for an Emulation Device, you can arrange them with the Move Up and Move Down buttons. You can arrange all of them with **File > Sort**, which can be useful if the images are time-stamped.

**NOTE:** This File dropdown is in the Object editor, not the main ACE screen.

### **Configuring the Emulation Device Virtual Camera**

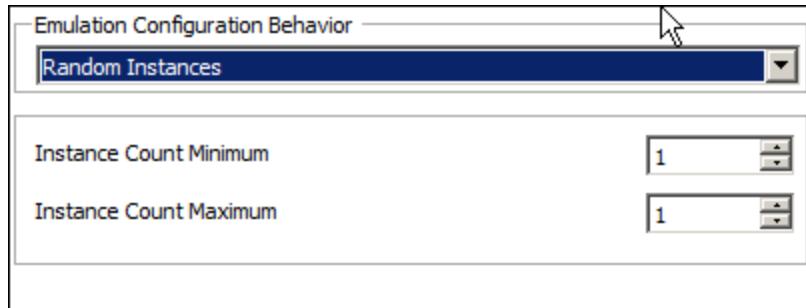
1. Double-click the Emulation Device Virtual Camera in the Folders pane. The Emulation Device Virtual Camera editor will open.



*Emulation Device Virtual Camera Screen*

Edit, in the Acquisition Settings pane, can be used to modify the Emulation Device properties.

2. Click the Device field in the Configuration group.  
If there is no Emulation Device associated with this virtual camera, you need to choose one. (Click the down-arrow, then select a device.)
3. Click the Emulation Configuration field to choose how to obtain the image(s).
  - Default - use the camera device set in the Default Device field.
  - Random Instances - select a minimum and maximum instance count, and it will provide a random number of images according to your choice. This is used only for ACE PackXpert applications.



### Selecting a Random Instances Range

The right portion of the Emulation Configuration field will say what the setting or range is.

- Use Alternate Device - use a device other than the default.
- Image Replay - get images from a directory (hig files only).

#### 4. If there are multiple images in the Emulation Device:

- a. Click Add in Acquisition Settings.
- b. Check the box for each image you want selected.
- c. Click Ok. A new Emulation Device Setting will be created (and check-marked) to specify that the selected image(s) will be used for this virtual camera.

You can choose the active Emulation Device Settings as appropriate.

The name of an Emulation Device Setting can be changed by clicking on an already-selected Setting.

#### 5. Click Add in the Calibrations group.

A Fixed Pixel Calibration will automatically be created.

6. You can specify the mm/pixel ratio for both x and y. For most uses, you can leave this as 1 mm/pixel.
7. Click OK. The selected image(s) can now be used by other vision tools through the Emulation Device Virtual Camera. The current image will be displayed in the Vision Window.

### **Related Topics**

[Vision Devices Overview](#)

## **Adding a Virtual Camera**

You can add a virtual camera to the workspace when adding a physical camera, or you can choose to add a virtual camera when you add a physical or emulated camera, by checking the "Create a virtual camera" checkbox.

To add a virtual camera:

1. Right-click in the Tree structure of the Workspace Explorer and select:

**New > Vision > Device > Virtual Camera**

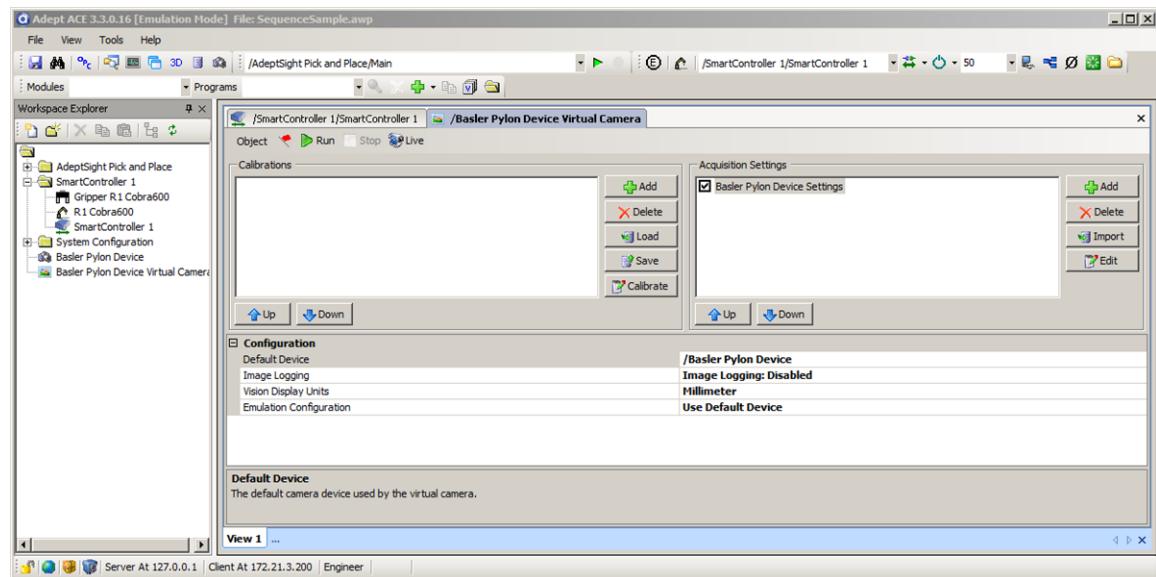
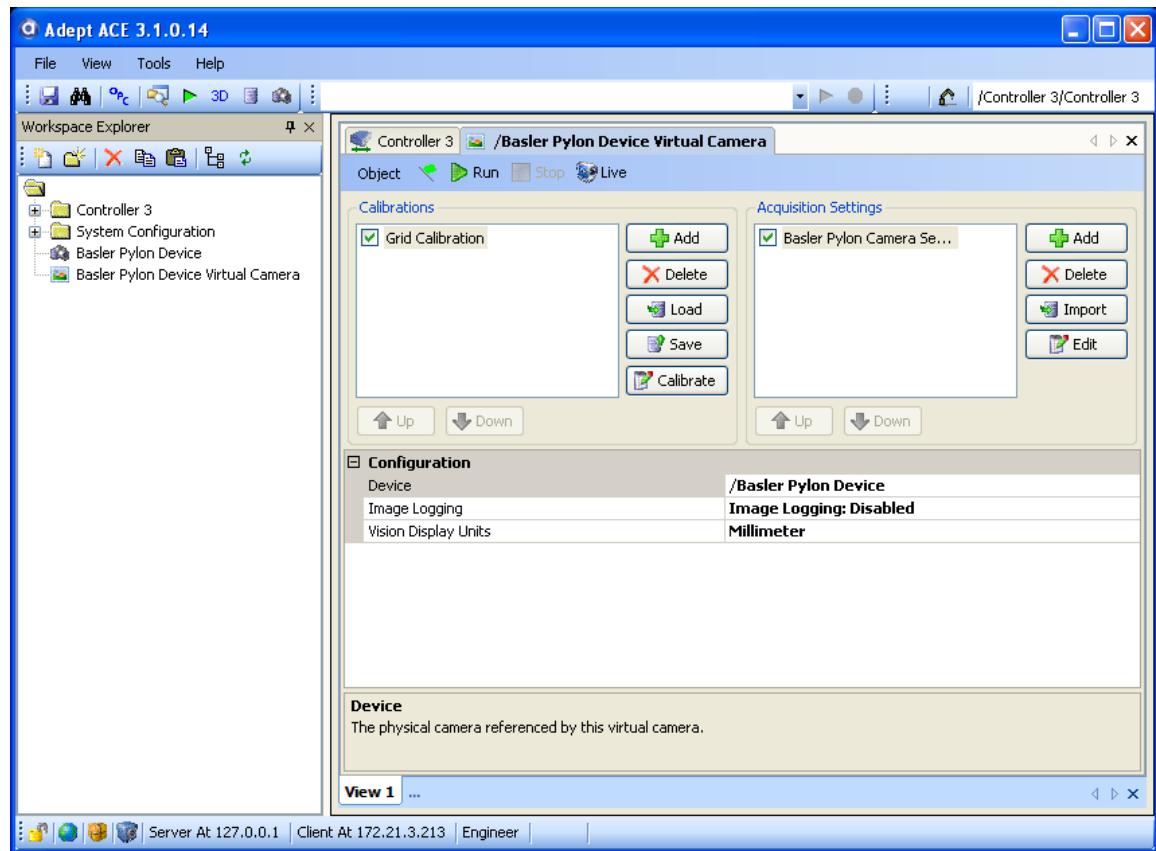
A virtual camera is added to the ACE workspace and the corresponding object editor opens.

You can rename the virtual camera immediately after creating it, or later, by right-clicking the object name in the Tree structure.

**NOTE:** If a virtual camera has already been created, and you just need to configure it, double-click on the virtual camera object in the Tree structure of the Workspace Explorer. The virtual camera object editor will open.

## Adding a Virtual Camera

---



### *Virtual Camera Object Editor*

Use Default Device  
Random Instances  
Use Alternate Device  
**Images Replay**

### *Emulation Configuration Selection*

2. Use the Device list to select the physical or emulated camera to associate with this virtual camera.

If you checked the "Create a virtual camera" box when adding the camera, this step is not necessary.

- a. Click Device in the Configuration group.
  - b. Click the drop-down arrow to display the available options to associate with the virtual camera.  
If only one camera is installed in the workspace, it will be the only camera displayed in the list.
  - c. Click the camera you want to associate with this virtual camera.
  - d. Click OK.
3. Click Edit to open the Camera Properties dialog. This is used to view information about the camera and make changes to shutter, gain, brightness, and other settings. See See "Camera Properties".
  4. Click and Enable Image Logging to specify a directory and the number of images you want to save. It will delete old images as needed when that number of images is exceeded.
  5. Set the Vision Display Units, if a different display scale is needed. The options are: Meter, Centimeter, Millimeter, or Micron. Millimeter is the default, and will not usually need to be changed.

**NOTE:** The Calibration for a camera can be Saved, and then Loaded into another workspace, if desired.

### **Related Topics**

[Adding Vision Tools](#)  
[Calibration Overview](#)

## **Camera Properties**

Use the Camera Properties window to get information about your camera and set various parameters to control the camera's behavior.

---

## Camera Properties

---

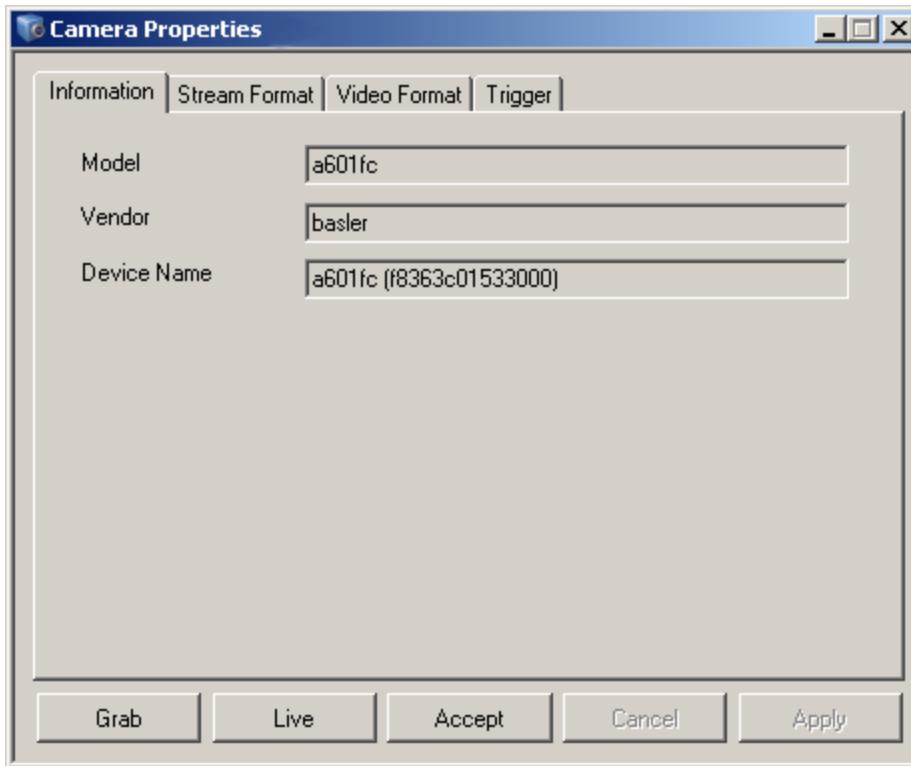
You can use the Grab and Live buttons at the bottom of the window to view the effects of the changes as you make them. The Live button turns into a Stop button after you click it, and then turns back into a Live button when Stop is clicked.

The Belt Control button, available only for line scan cameras, is covered at the end of this topic.

### ***Information***

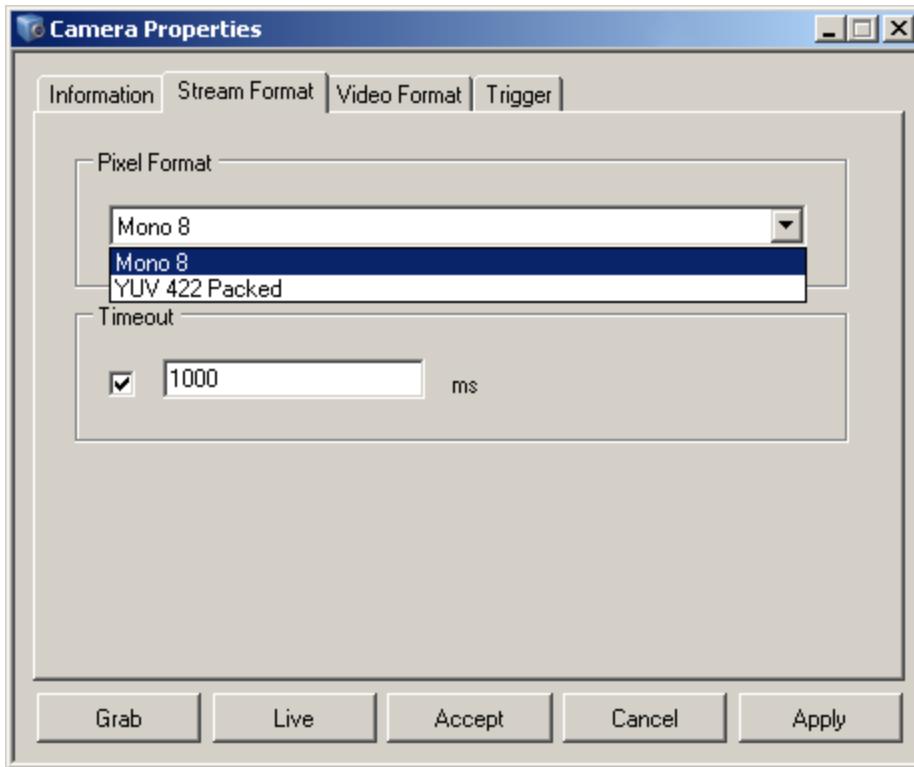
The Information tab displays the Model, Vendor, and Serial Number of the attached camera.

These fields are read-only.



### ***Stream Format***

The Stream Format tab lets you set the Pixel Format and Timeout value for the data being sent from the camera.



The available pixel formats will be displayed in the drop-down box when you click the down-arrow. In the preceding example, this is either Mono 8 or YUV 422 Packed.

Adept recommends that you accept the default format.

The Timeout value sets a time limit, in milliseconds, after which the vision tool terminates the processing of an image. If the vision tool has not finished processing an image within the allotted time, the tool returns all the instances it has located up to the timeout.

Although Timeout can be disabled, Adept recommends that you use a Timeout value.

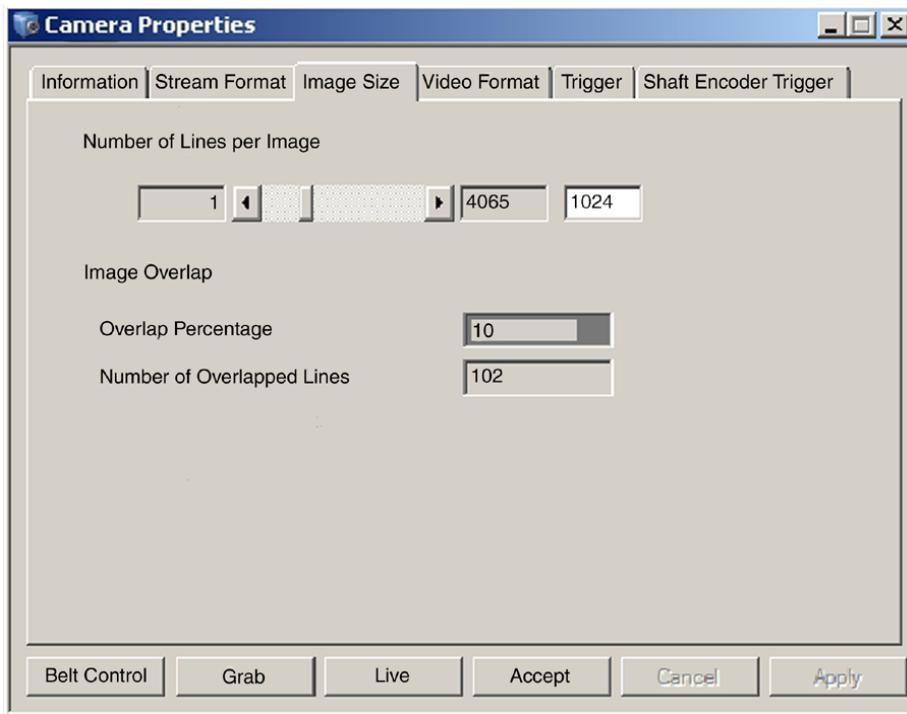
- This is useful for time-critical applications in which fast operation is more important than the occasional occurrence of undetected object instances.
- This value is only approximate; the actual processing time may be slightly greater.

### ***Image Size***

This tab is only displayed if a line scan camera is connected.

## Camera Properties

---



- Number of Lines per Image

This is the number of scan lines that are used to create one image, or frame.

- Image Overlap

- Overlap Percentage

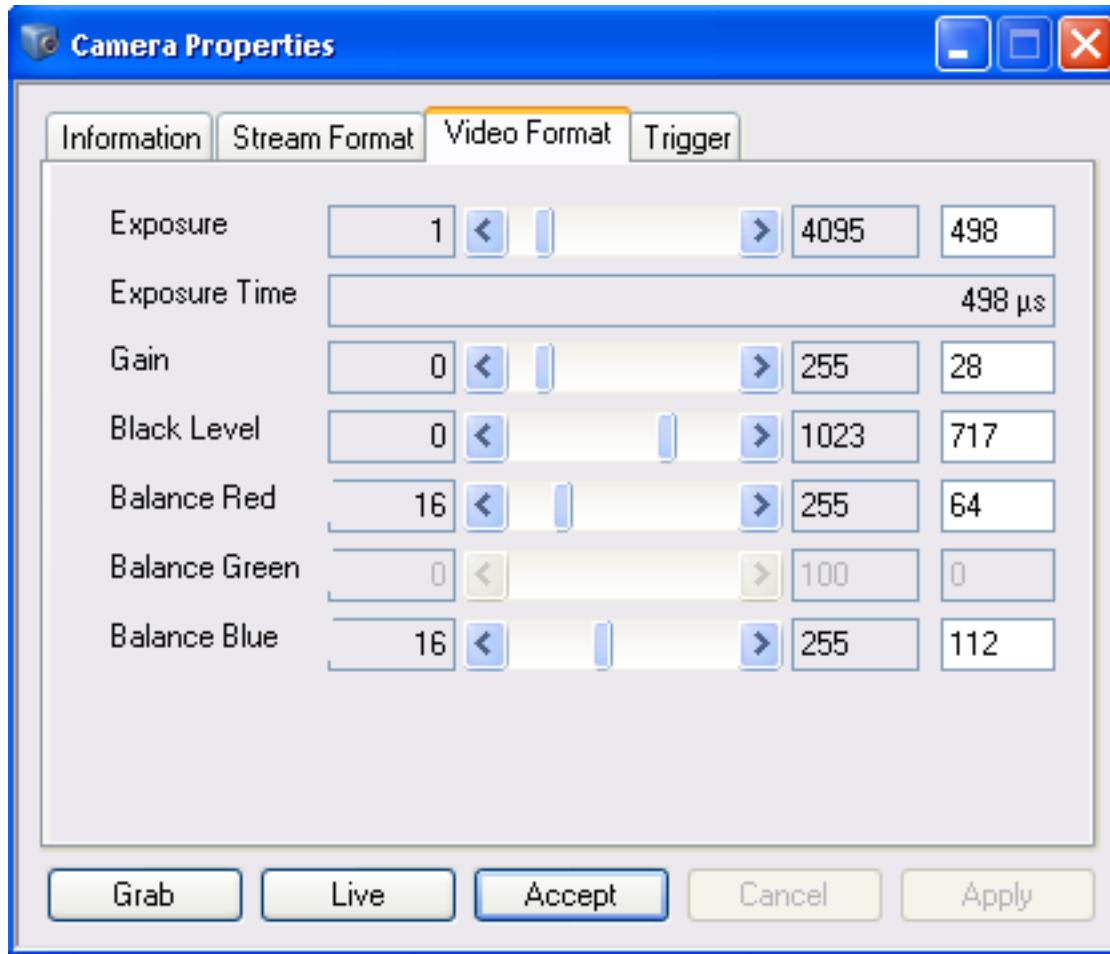
This value can be from 0 to 50%.

- Number of Overlapped Lines

This is a read-only field.

### **Video Format**

The Video Format tab lets you set Exposure, Gain, Black Level, and color balance.



Each line displays the minimum allowable value for that property, a bar indicating the current value, the maximum allowable value , and the numeric value of the current level. You can change the current value of a property by clicking the arrows at either side of the bar, by sliding the bar, or by typing in a new value.

Note that some of the minimum and maximum values, particularly for Gain, will differ depending on the camera being used.

Adjust Exposure, Gain, and Black Level (in that order) to improve the quality of acquired images.

- The Exposure time setting determines the time interval during which the sensor is exposed to light.

Choose an exposure time setting that takes into account whether you want to acquire images of still or moving objects:

- If the object is not moving, you can choose a high exposure time setting (i.e., a long exposure interval).
- Note that high exposure time settings may reduce the camera's maximum allowed

acquisition frame rate and may cause artifacts to appear in the image.

- If the object is moving, choose a low exposure time setting to prevent motion blur.

As a general rule, choose a short enough exposure time to make sure that the image of the object does not move by more than one pixel during exposure.

A shorter exposure time setting may require a higher illumination level.

Acquisition parameters are validated before being sent to the camera. If you enter an exposure time that your camera does not support, the time will be adjusted to be valid. If you haven't typed in an invalid exposure time, the left and right arrows will provide valid times.

- Gain is the amplification of the signal being sent from the camera.

The readout from each pixel is amplified by the Gain, so both signal and noise are amplified. This means that it is not possible to improve the signal-to-noise ratio by increasing gain.

You can increase the contrast in the image by increasing the camera's gain setting. Unless your application requires extreme contrast, make sure that detail remains visible in the brightest portions of the image when increasing gain. Note also that noise is increased by increasing gain.

Increasing gain will increase the image brightness.

Set the gain only as high as is necessary.

- Black Level is an offset, which is used to establish which parts of an image should appear black.

High black level settings will prevent high contrast. Make fine adjustments to the Black Level to ensure that detail is still visible in the darkest parts of the acquired images.

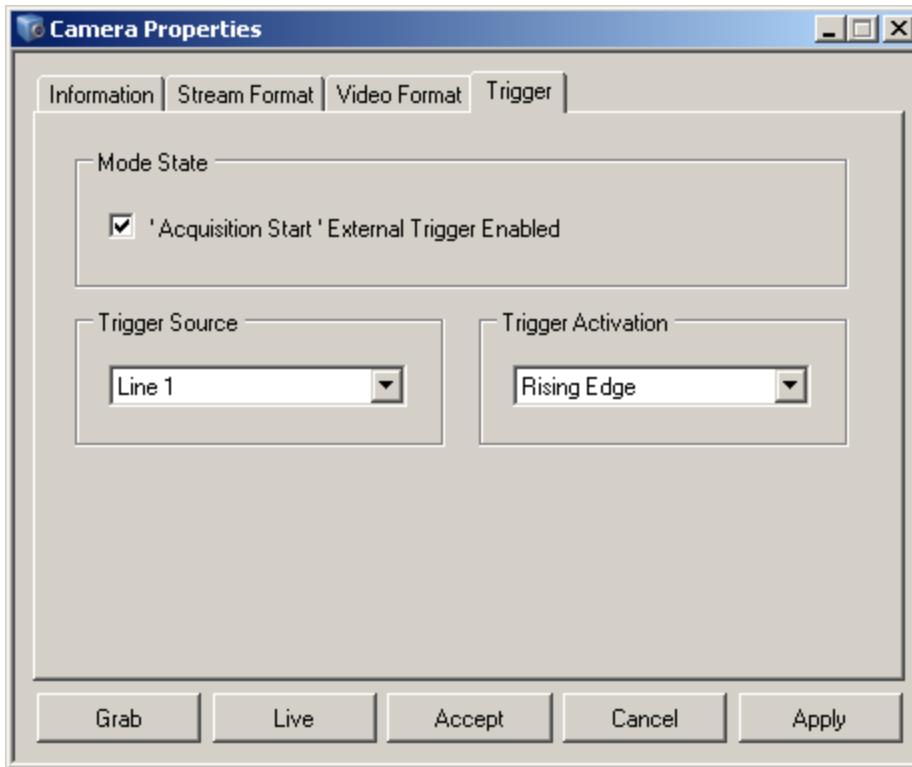
Balance Red, Balance Green, and Balance Blue are only available if your ACE dongle has the color license enabled and you have a color camera connected. On some Basler color cameras, such as the A601fc-2, the green balance is a fixed value that cannot be adjusted. In such cases, only the balance for blue and red will be enabled in this window- Balance Green will be greyed out.

### ***Trigger***

The Trigger tab lets you enable an external trigger for taking a picture, and set parameters that pertain to that trigger.

## Camera Properties

---



Mode State is a checkbox that determines whether or not to use an external trigger. The default is off.

Trigger Source and Trigger Activation have no effect unless the trigger is enabled.

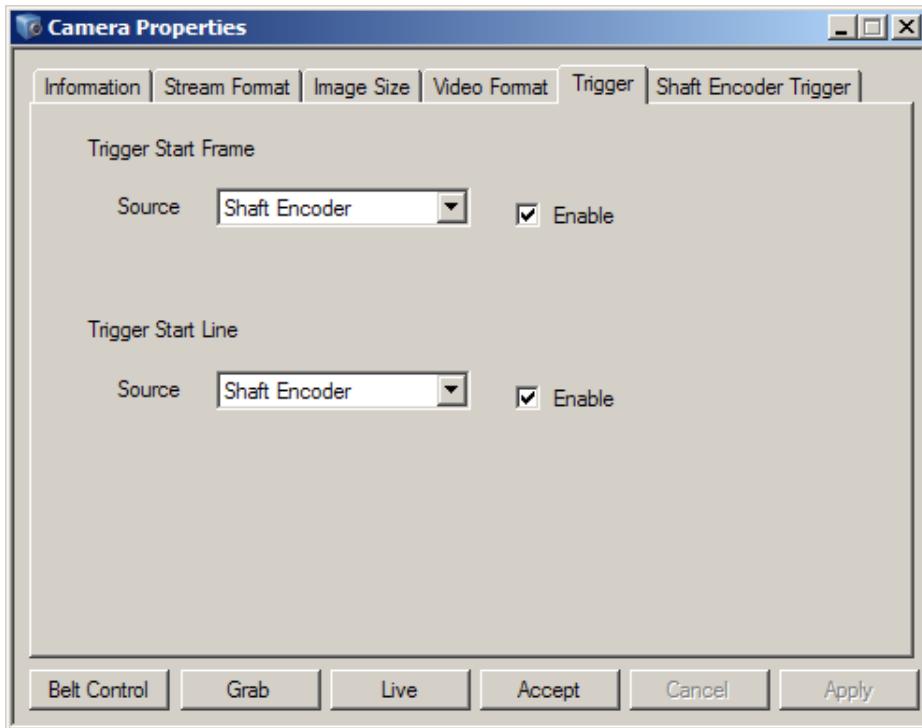
Trigger Source is for cameras that support multiple trigger lines. Specify which line will be used to cause an image to be taken.

The Trigger Activation field specifies which part of the trigger signal to react to.

For line scan cameras, you must specify what signals trigger the start of a frame and the start of a line.  
Refer to the following figure:

## Camera Properties

---



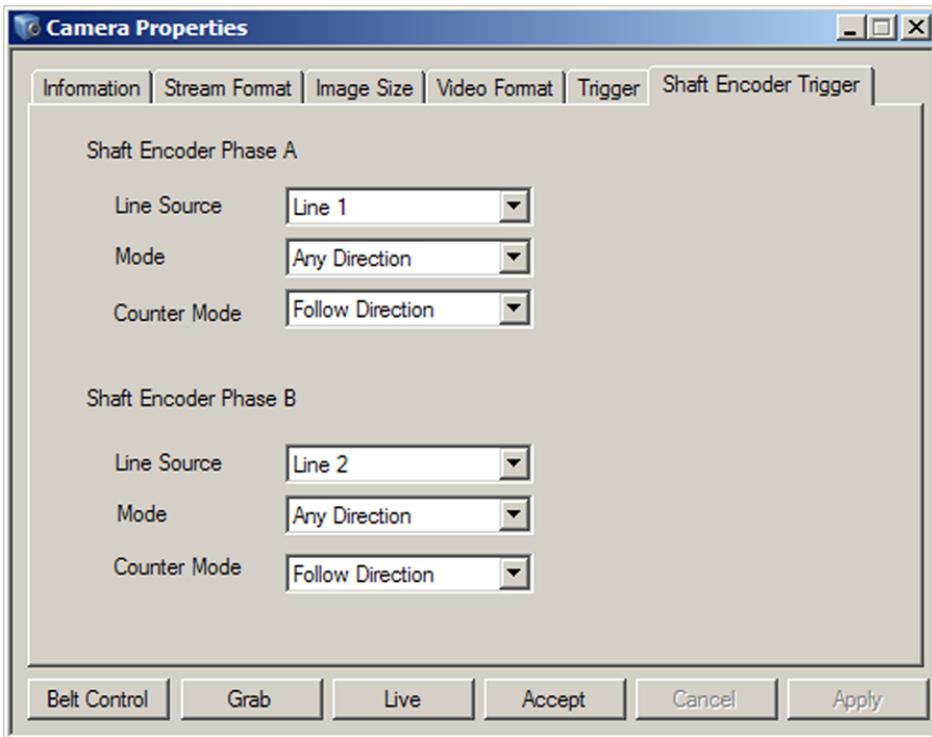
If you are going to use Shaft Encoder as the trigger(s), you have to set it up with the Shaft Encoder Trigger tab.

### **Shaft Encoder Trigger**

This tab is only displayed if a line scan camera is connected. The Shaft Encoder Trigger tab lets you set the encoder signals to be used to trigger the frame and line start signals.

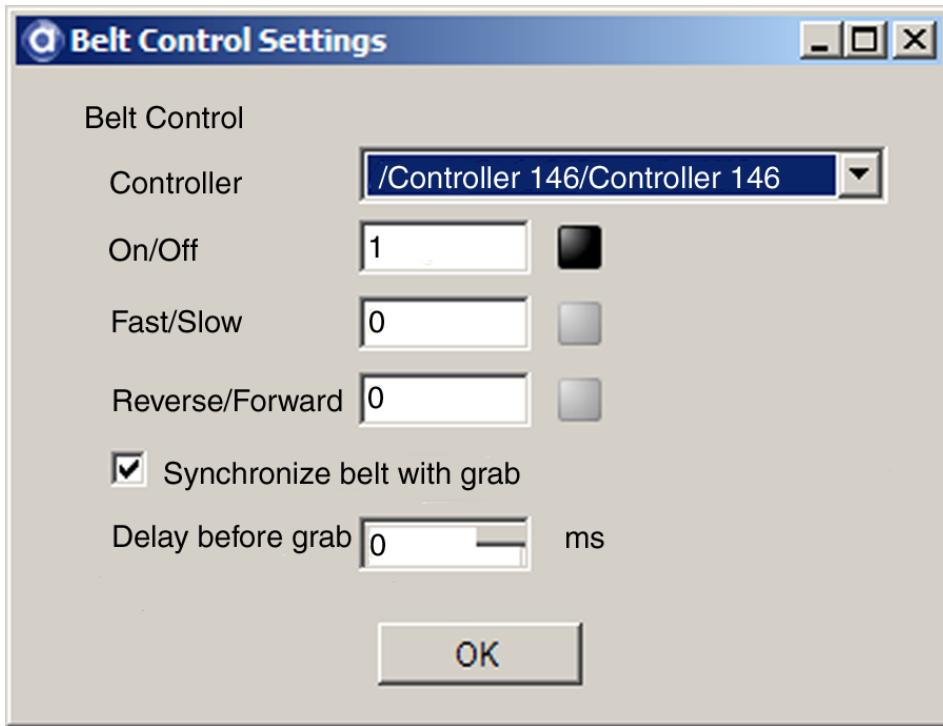
## Camera Properties

---



### **Belt Control**

The Belt Control button is available on the Camera Properties forms and on all calibration wizard pages that need to acquire images when a line scan camera is attached. Clicking Belt Control opens the following window:



- Controller

Specify the controller attached to the conveyor belt.

- In the following three fields, a green square to the right of the field indicates that the signal is On, a black square indicates that the signal is Off, and a grey square indicates that the signal has not been set (=0).

- On/Off

Specify the signal used to turn the conveyor belt on and off.

- Fast/Slow

Specify the signal used to switch the conveyor belt between fast and slow speeds.

- Reverse/Forward

Specify the signal used to switch the conveyor belt between reverse and forward directions.

- Synchronize belt with grab

If checked, the belt is started when an image acquisition is requested, and stopped after the image has been acquired.

Delay before grab sets the milliseconds, before the scan starts, to start the conveyor belt movement.

## Vision Tools

ACE Sight software provides an extensive set of vision tools for basic and complex applications.

These tools include rulers for measuring distances, windows for calculating the intensity ranges in an image, and finder tools for locating lines, circles, and other features of objects. Each of these tools requires several pieces of information to know what data to collect, how to interpret the data, and where in the camera field of view to look for that data. ACE Sight software allows you to specify all this information, either by making menu selections, or by using the mouse to manipulate the tools directly.

### ***Region of Interest***

The area of the image in which the tool carries out its process or operation is called the region of interest (ROI).

- A vision tool can be positioned to operate on the entire image.
- Inspection tools are usually applied to a specific part of the image, or to a specific area relative to an object.

Tools can be positioned from the Region of Interest dialogs and in the Display window.

**NOTE:** There are several related tools, not covered in this topic. See "Communication Tool", Overlap Tool on page 618, and [ACE Sight Sequence on page 608](#).

### ***Locator and Finder Tools***

The Locator and Finder Tools create a description of objects or object features using vectors. These tools are typically faster, more reliable, and more accurate than greyscale (rasterized) inspection tools.

#### **Locator, Locator Model**

The Locator identifies instances of model-defined objects. Models characterize object types and are created and edited through the Locator's Model Editor. The Locator is the ideal frame-provider tool for positioning inspection tools. See "Locator" and See "Locator Model".

#### **Arc Finder**

The Arc Finder identifies circular features on objects and returns the coordinates of the center of the arc, the angle of separation between the two ends, and the radius. See "Arc Finder".

## **Line Finder**

The Line Finder identifies linear features on objects and returns the line angle and point coordinates. See "Line Finder".

## **Point Finder**

The Point Finder identifies point features on objects and returns the coordinates of the found point features. See "Point Finder".

### ***Image Processing Tools***

Image processing tools provide various operations and functions for the analysis and processing of images.

## **Image Processing**

The Image Processing Tool processes greyscale images by applying arithmetic, assignment, logical, filtering, morphological, or histogram operators. Users can define custom filtering operators. See "Image Processing".

## **Image Sharpness**

The Image Sharpness Tool computes the sharpness of major edges in a user-defined region of interest. See "Image Sharpness".

## **Image Histogram**

The Image Histogram tool computes grey-level statistics within a user-defined region of interest. See "Image Histogram".

## **Image Sampling**

The sampling tool is used to extract an area of an image and output it as a separate Image. See "Image Sampling".

### ***Inspection Tools***

Inspection tools are commonly used in vision applications to inspect objects and parts, which are typically found by a Locator or Finder tool. Inspection tools rely on the analysis of pixel information, and do not create vector descriptions of objects.

## **Blob Analyzer**

The Blob Analyzer identifies blobs, and returns various results for each blob. See "Blob Analyzer".

---

## **Caliper**

The Caliper identifies one or more edge pairs and measures distances between the two edges within each pair. See "Caliper".

## **Arc Caliper**

The Arc Caliper identifies one or more edge pairs on an arc-shaped or circular area and measures distances between the two edges within each pair. See "Arc Caliper".

## **Edge Locator**

The Edge Locator identifies an edge or a set of edges that meet user-defined criteria. See "Edge Locator".

## **Arc Edge Locator**

The Arc Edge Locator identifies an edge or a set of edges in an arc-shaped or circular area. See "Arc Edge Locator".

## **Pattern Locator**

The Pattern Locator identifies instances of a greyscale pattern. See "Pattern Locator".

## **Filter Tools**

Filter tools let you specify criteria for limiting which instances will be processed. These criteria include physical appearance (including color) of the instances and proximity of the instances to each other.

## **Inspection Tool**

The Inspection Tool analyzes the results of other tools based on a configuration of inspection filters. See "Inspection".

## **Color Matching Tool**

The Color Matching tool filters and analyzes areas of specified color, or color ranges in color images. See "Color Matching".

## **Gripper Clearance**

The Gripper Clearance tool analyzes the results of histograms to determine which parts can safely be picked. [See "Gripper Clearance".](#)

---

### ***Calculation Tools***

Calculation Tools let you calculate various geometric features based on the results of other tools. For instance, given the edges of a square part, you can calculate the center of the part.

Calculated Frames are used to create a vision frame from other features, allowing you to place vision tools on objects that are not always in the same location or orientation.

#### **Calculated Point**

The Calculated Point tool generates a point based on input from other tools. See "Calculated Point".

#### **Calculated Line**

The Calculated Line tool generates a line based on input from other tools. See "Calculated Line".

#### **Calculated Arc**

The Calculated Arc tool generates a circle based on input from other tools. See "Calculated Arc".

#### **Calculated Frame**

The Calculated Frame tool generates a frame based on input from other tools. See "Calculated Frame".

### ***Other Tools***

#### **Custom Vision Tool**

The Custom Vision tool allows you to specify a program that will be called when the tool is executed. From within a Custom Vision tool, other tools can be executed, and return a set of results, which are used as the output of the tool. See "Custom Vision Tool".

#### **Remote Vision Tool**

The Remote Vision tool lets an application, such as an ACE PackXpert packaging application, run a vision tool on a remote PC.

Because image processing is computationally intense, this can offload a significant portion of the processing load from the main ACE PC to a remote PC. See "Remote Vision Tool".

## Calibration Grid Locator

The Calibration Grid Locator tool identifies a collection of dots in the field of view. It is used by the grid calibration procedure. See "Calibration Grid Locator".

### **Related Topics**

[Adding Vision Devices](#)

[Adding Vision Tools](#)

[ACE Sight Sequences](#)

## Vision Tools Summary Table

Function	Tool	Use/function/description	See also
Locating and Finding Tools		The Locator and Finder Tools create a vectorized description of objects or object features. These tools are typically faster, more reliable, and more accurate than greyscale inspection tools.	
	Locator	Identifies instances of model-defined objects.	Locator Model
	Locator Model	Models describe the geometry of an object used by the Locator tool.	Locator
	Arc Finder	Identifies circular features on objects and returns the coordinates of the center of the arc, the angle of separation between the two ends, and the radius.	
	Line Finder	Identifies linear features on objects and returns the line angle and point coordinates.	
	Point Finder	Identifies point features on objects and returns the coordinates of the found point.	
Image Processing Tools		Image processing tools provide various operations and functions for the analysis and processing of images.	

## Vision Tools Summary Table

---

<b>Function</b>	<b>Tool</b>	<b>Use/function/description</b>	<b>See also</b>
	Image Processing	Analyzes greyscale images by applying arithmetic, assignment, logical, filtering, morphological, or histogram operators. Users can define custom filtering operators.	
	Image Histogram	Computes grey-level statistics within a user-defined region of interest.	
	Image Sampling	Extracts an area of an image and outputs it as a separate Image.	
	Image Sharpness	Computes the sharpness of major edges in a user-defined region of interest.	
Inspection Tools		Inspection tools are commonly used in vision applications to inspect objects and parts, typically found by a Locator tool. Inspection tools rely on the analysis of pixel information.	
	Blob Analyzer	Identifies blobs, and returns various results for each blob.	
	Caliper	Identifies one or more edge pairs and measures distances between the two edges within each pair.	
	Arc Caliper	Identifies one or more edge pairs on an arc-shaped or circular area and measures distances between the two edges within each pair.	
	Edge Locator	Identifies an edge or a set of edges that meet user-defined criteria.	
	Arc Edge Locator	Identifies an edge or a set of edges in an arc-shaped or circular area.	
	Pattern Locator	Identifies instances of a greyscale pattern.	
Filter Tools		Filter tools specify criteria for limiting which instances will be processed. These criteria include physical appearance (including color) and proximity of the instances to each other.	

## Vision Tools Summary Table

---

<b>Function</b>	<b>Tool</b>	<b>Use/function/description</b>	<b>See also</b>
	Inspection Tool	Analyzes the results of other tools based on a configuration of inspection filters.	
	Color Matching	Filters and analyzes areas of specified color, or color ranges in color images.	
	Gripper Clearance	Analyzes results of histograms to determine which parts can safely be picked.	
Calculation Tools			
	Calculated Frame	Calculates a frame based on a specified calculation mode.	
	Calculated Arc	Calculates the circle that encompasses an arc, based on a specified calculation mode.	
	Calculated Line	Calculates a line based on a specified calculation mode.	
	Calculated Point	Calculates a point based on the calculation mode.	
Other Tools			
	Custom Vision	Allows user specify a program to be called when the tool is executed. From within a custom tool, other tools can be executed, and return a set of results which are used as the output of the tool.	
	Remote Vision	Adds the ability to run a vision tool on a remote PC, to offload that portion of the processor load to the remote PC. This allows a single instance of ACE to manage a larger ACE application.	
	Calibration Grid Locator	Identifies a collection of dots in the field of view. It is used by the grid calibration procedure.	

### ***Related Topics***

[Overview of ACE Sight Vision Tools](#)

[Adding Vision Tools](#)

## **Adding Vision Tools**

Adding a vision tool is also referred to as Creating the tool (you are creating an instance of the tool).

Each vision tool is configured using its corresponding object editor.

Tools can be linked together using the Relative To parameter in the object editor, which allows the results of one tool to become the input for another tool. For example, you could use one tool to do a preliminary coarse search, and then use other tools to "zoom in" on a region of interest.

1. To add a new vision tool, right-click in the Tree structure of the Workspace Explorer and select:

**New > Vision > Tool > [Vision Tool Name]**

The selected vision tool is added to the ACE workspace and the corresponding object editor opens.

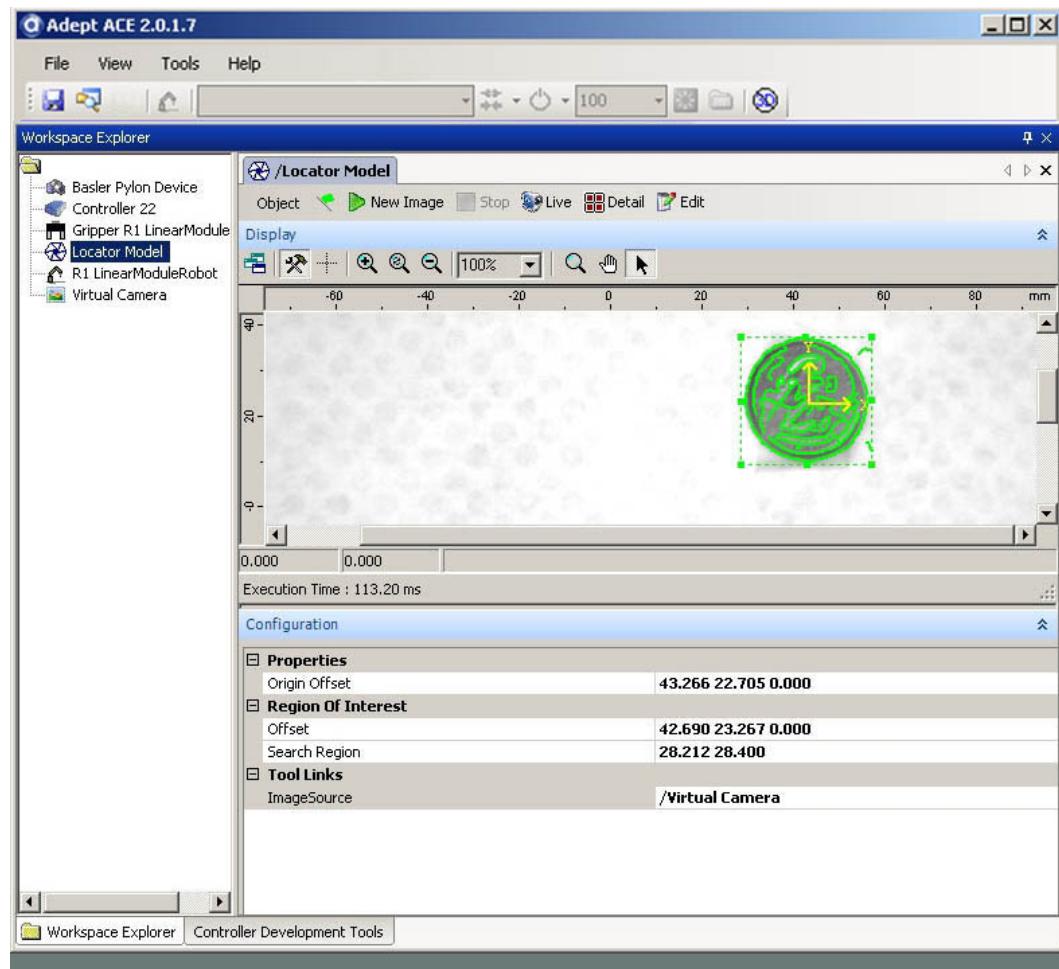
For example,

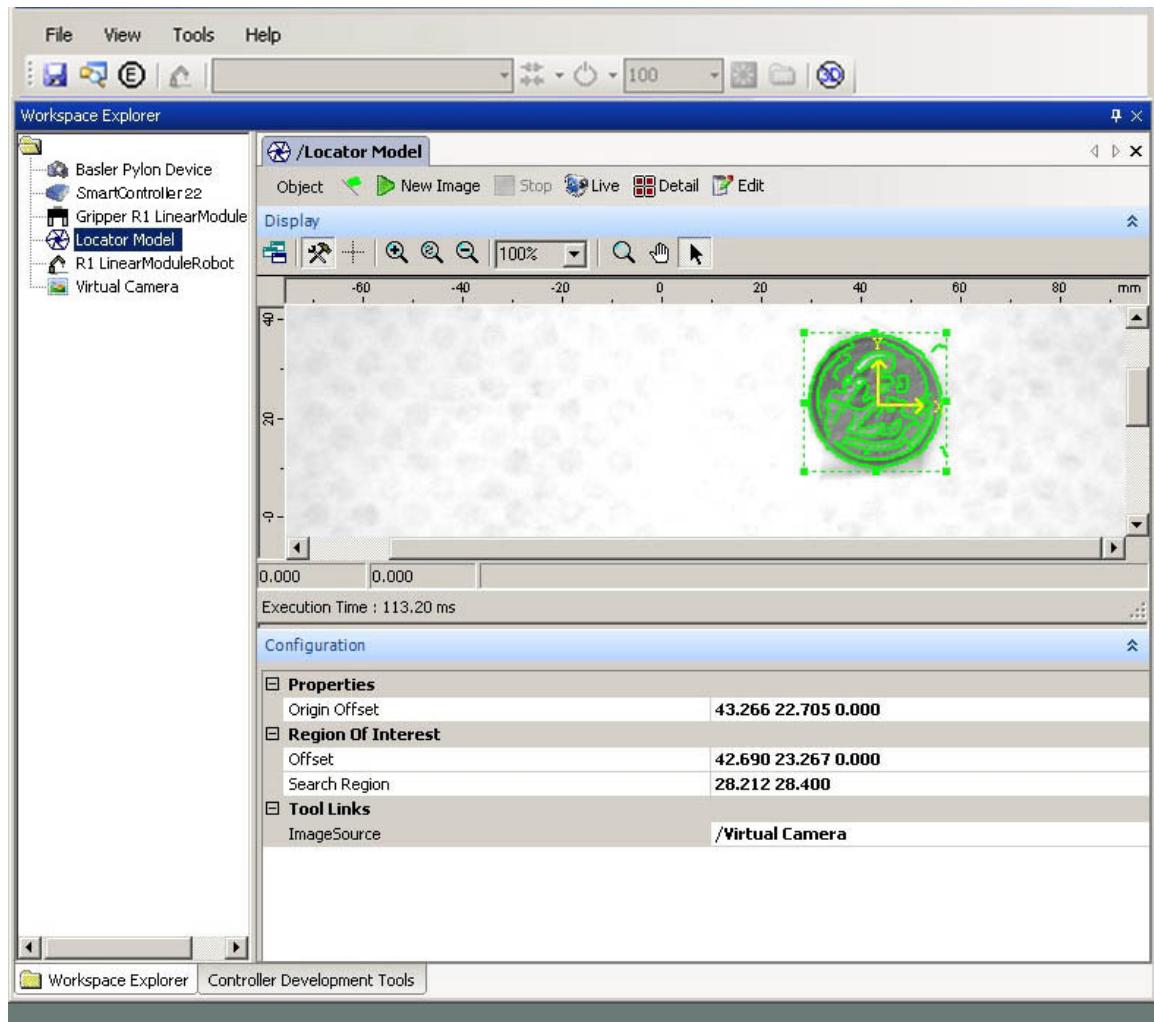
- a. Right-click in the Tree structure.
- b. Select **New > Vision > Tool > Locator Model** from the menu.

A Locator Model tool is added to the workspace and opens the corresponding object editor, as shown in the following figure:

## Adding Vision Tools

---





### *Adding a Locator Model Tool*

2. Use the controls within the object editor to configure the tool for your application. See "Workspace Explorer".

#### **Related Topics**

[Adding Vision Devices](#)

[Calibration Overview](#)

## Arc Caliper

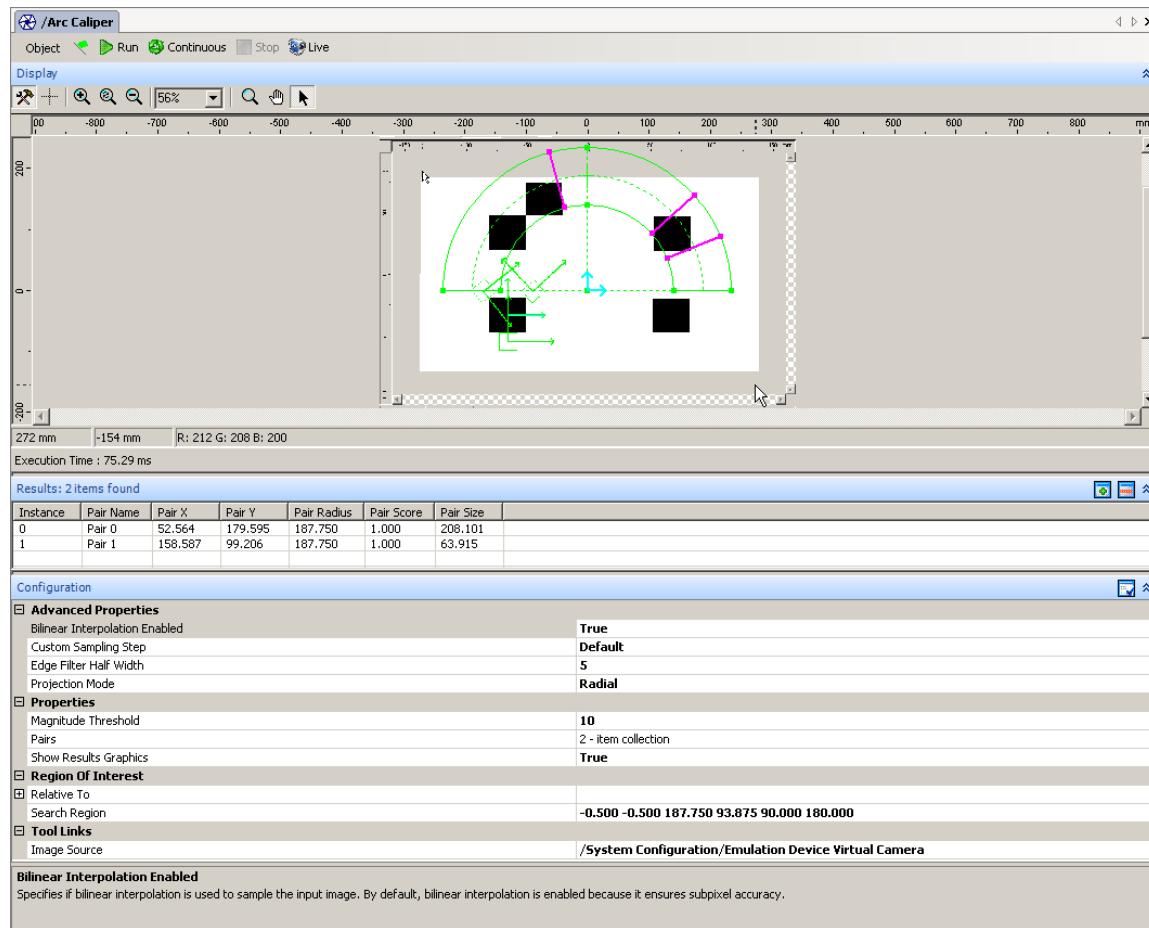
The Arc Caliper tool identifies and measures the gap between one or more edge pairs of arc-shaped objects. Edges can be disposed in a radial or an annular position.

The Arc Caliper uses pixel grey-level values within the region of interest to build projections needed for edge detection.

After detecting potential edges, the Arc Caliper determines which edge pairs are valid by applying the constraints that are configured for each edge pair. Finally, the Arc Caliper scores and measures each valid edge pair.

To create an Arc Caliper tool, right-click in the Tree structure, then select:

**New > Vision > Tool > Arc Caliper**



*Arc Caliper Object Editor*

### Configuration

Basic steps for configuring an Arc Caliper

1. Select the tool that will provide input images.
2. Position the Arc Caliper tool.
3. Configure Pair Settings for each edge pair. See Configuring Arc Caliper Properties on page 359
4. Test and verify results. See Arc Caliper ResultsResults for edges found by the Arc Caliper tool are shown in the grid of results, below the Display window, as shown in the following figure. Representation of Arc Caliper Results in Display and Results GridDisplay WindowThe Display window shows the image being processed, the region of interest selected, and the locations of found instances. The elapsed time is displayed at the bottom of this window. Grid of ResultsThe grid of results presents the result values for all caliper measures found by the Arc Caliper tool. ObjectDefinitionPair NameThe name of the edge pair, as it appears in the Pairs list. Pair ScoreScore is the calculated score, between 1 and 0, for the edge pair. The score is calculated according to the defined constraint functions. If both Position and Magnitude constraints are enabled, each constraint accounts for 50% of the score. Each edge of the pair is also scored individually, in a similar manner. See Edge1/Edge2 Score. Pair SizeSize is the Caliper measure, which is the calculated distance between the pair of edges. Pair RadiusMean of Edge 1 Radius and Edge 2 Radius. Pair XThe X coordinate of the center point of the caliper measure, at the midpoint of the edge pair. Pair YThe Y coordinate of the center point of the caliper measure, at the midpoint of the edge pair. Frame/GroupThe frame or group to which the result belongs. Edge 1/Edge 2 ScoreThe score of the individual edge, calculated according to the defined constraints. Edge 1/Edge 2 XThe X coordinate of the edge, at the midpoint of the edge segment. Edge 1/Edge 2 YThe Y coordinate of the edge, at the midpoint of the edge segment. Edge 1/Edge 2 RotationThe angle of rotation for the edge. Edge 1/Edge 2 RadiusRadius of detected arc. Edge 1/Edge 2 Position ScorePosition score for the edge, calculated according to the Position constraint function. Edge 1/Edge 2 MagnitudeThe calculated Magnitude value for the edge. Edge 1/Edge 2 Magnitude ScoreMagnitude score for the edge, calculated according to the Magnitude constraint function. Related TopicsArc Caliper ToolConfiguring Arc Caliper PropertiesConfiguring Arc Caliper Properties - Advanced. on page 1
5. Configure Advanced properties if required. See Configuring Arc Caliper Properties - Advanced on page 367

Object	Definition
<b>Properties</b>	
Pairs	Opens Edge Pair Collection Editor to set criteria for valid edges.
Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.

Object	Definition
<b>Tool Links</b>	
Image Source	Defines the image source used for processing by this vision tool.
<b>Region of Interest (ROI)</b>	
Relative To	Specifies the tool that will provide the input to this tool. The tool can be dragged from the Folder pane and dropped into this field.
Search Region	Defines the Center, Radius, Rotation, Thickness, and Opening of the ROI.
<b>Advanced Properties</b>	
Bilinear Interpolation Enabled	Specifies if bilinear interpolation is used to sample the input image. By default, bilinear interpolation is enabled because it ensures subpixel accuracy.
Custom Sampling Step	When enabled, uses the user-defined sampling step instead of the default optimal sampling step to sample the region of interest from the input image.
Edge Filter Half Width	The half width of the filter should be set to the width of the edge as it appears in the greyscale image .
Projection Mode	Projection mode used by the tool to detect edges (radial or annular).

### Results

Object	Definition
Instance	The index of the located instance
Pair Name	Name of the edge pair
Pair X	The X coordinate of the center point of the caliper measure, at the midpoint of the edge pair
Pair Y	The Y coordinate of the center point of the caliper measure, at the midpoint of the edge pair
Pair Radius	Mean value of Edge 1 Radius and Edge 2 Radius

Object	Definition
Pair Score	Score of the selected pair. The score of the pair is equal to the mean score of the two edges (Edge 1 Score and Edge 2 Score) of the pair.
Pair Size	Distance between the midpoints of the two arcs of the selected pair in the currently selected coordinate system. Read-only.
Edge 1 X	X coordinate of the center of the first edge of the selected pair
Edge 1 Y	Y coordinate of the center of the first edge of the selected pair
Edge 1 Radius	The radius of the first edge point (same as Pair Radius for Radial)
Edge 1 Score	Score of the first edge of the selected pair. The score is computed according to the constraints set by the Edge Constraints property.
Edge 1 Magnitude	Magnitude of the first edge of the selected pair
Edge 1 Magnitude Score	Magnitude score of the first edge of the selected pair
Edge 1 Position Score	Position score of the first edge of the selected pair
Edge 2 X	X coordinate of the center of the second edge of the selected pair
Edge 2 Y	Y coordinate of the center of the second edge of the selected pair
Edge 2 Radius	The radius of the second edge point (same as Pair Radius for Radial)
Edge 2 Score	Score of the second edge of the selected pair. The score is computed according to the constraints set by the Edge Constraints property.
Edge 2 Magnitude	Magnitude of the second edge of the selected pair
Edge 2 Magnitude Score	Magnitude score of the second edge of the selected pair
Edge 2 Position Score	Position score of the second edge of the selected pair

### ***Edge Pair Collection Editor***

The Arc Caliper tool uses a form for configuring the edge pairs. See [Configuring Arc Caliper Properties](#)

### ***Related Topics***

[Configuring Arc Caliper Properties](#)

[Configuring Arc Caliper Properties - Advanced](#)

[Arc Caliper Results](#)

### ***Configuring Arc Caliper Properties***

#### **Edge Detection**

Edge Detection settings are used to configure the parameters that the Arc Caliper will use to find potential edges in the region of interest. The display represents the Arc Caliper region of interest and provides information to assist in configuring Edge Detection parameters.

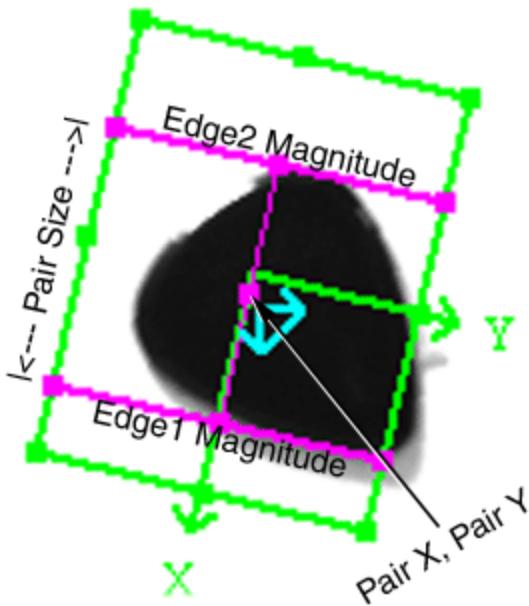
#### **Configuring Edge Pair Properties**

When a caliper is executed, the caliper first applies edge detection constraints to the entire region of interest. Then, the tool applies edge scoring constraints to determine which edges are valid for the caliper measure. If only one valid edge is found, no caliper measure is output.

Pair Settings parameters determine how the tool detects edges and which edge pair are valid.

**NOTE:** Before configuring the caliper, execute the tool at least once and verify that the tool is being positioned correctly in the image.

The display represents the caliper as green, with found edges and caliper measure represented in magenta:



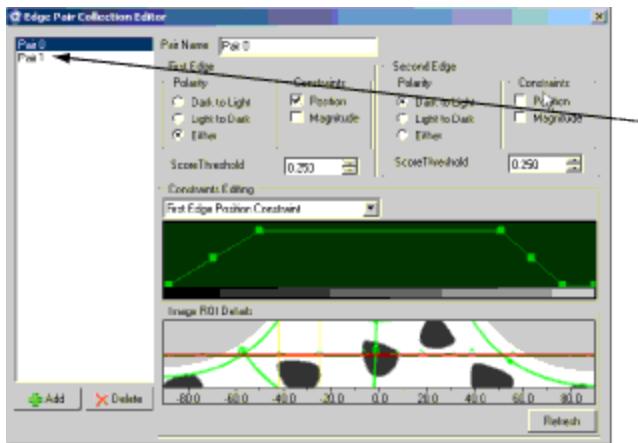
#### *Found Edges and Caliper*

If the display in the Pair Settings window is blank, or the edges are not properly placed, close the window and verify the following:

- The tool was executed after positioning the tool.
- The tool was executed at least once before opening the Pair Settings window.
- The Location parameters are correct.
- The Y-axis of the tool is parallel to the edges you want to detect.

Constraints are set with the Edge Pair Collection Editor window. From the Object editor window, select

**Configuration > Properties >Pairs >**



Edge Pairs  
Configurations  
List

### Edge Pair Collection Editor

The caliper tool can measure any number of pairs. When the caliper is executed, it first applies edge detection parameters to the entire region of interest. Then, the tool applies pair settings constraints to determine which edges are valid. Results are then calculated for each valid edge pair as well as for individual edges in each edge pair.

### Pairs Configuration List

The Pairs list of the Edge Pair Collection Editor contains a list of all the pair configurations for the current caliper tool. This list always contains at least one pair configuration, which is named Pair 0 by default.

Each pair configuration has a name, with polarity and constraints for the first edge and second edge.

From the Edge Pair Collection Editor, you can:

- Access the parameters for each pair configuration
- Add and remove edge pair configurations
- Rename edge pair configurations



#### *Pairs List in the Caliper Interface*

To access the parameters for an edge pair configuration, click on that pair configuration in the Pairs Configurations List.

To add an edge pair configuration:

1. Under the Pairs Configuration list, click add (+).

A pair configuration is added with the default name: "Pair n", where n is the next (unused) integer.

2. Edit the parameters for the new pair configuration.

To remove an edge pair configuration:

1. In the Pairs configuration list, select the pair to be removed.
2. Click delete (X).

To rename an edge pair configuration:

1. Click the pair configuration to be renamed in the pairs list.
2. Highlight the Pair Name field for the pair to be renamed.
3. Type a new name for the edge pair.

**NOTE:** This will not affect the parameters of the pair configuration.

To configure edge pair configuration parameters:

1. In the Edge Pair Collection Editor, select a pair configuration from the list. The default name for a first pair is Pair0.
2. The remainder of the window provides parameters for each edge of the caliper edge pair configuration, which are named First Edge and Second Edge.
3. Configure parameters for each edge. Refer to the following sections for help on configuring Pair Settings, and using the display and Function Editor.

## Edge Pair Parameters

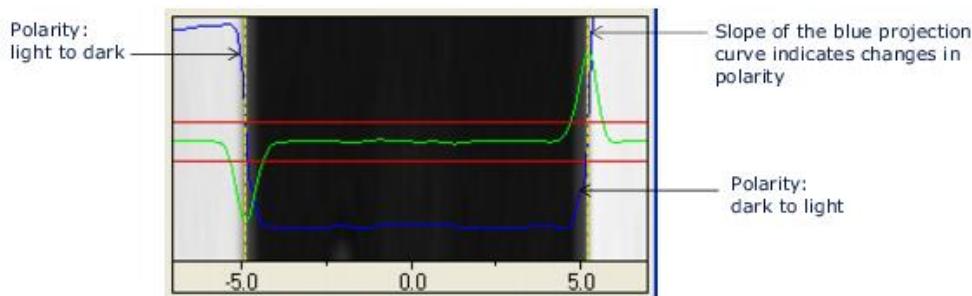
There are two criteria that affect the choice of valid edges: Polarity and Edge Score Constraints.

### Polarity

Polarity corresponds to the change in light values, moving from left to right in the display, along the X-Axis in the region of interest. The caliper applies the Polarity constraint before applying edge-score Constraints.

Polarity does not affect the edge score, but only edges that meet the selected Polarity are retained as valid edges, regardless of their scores.

- Dark to Light will only accept edges occurring at transitions from a dark area to a light area.
- Light to Dark will only accept edges occurring at transitions from a light area to a dark area.
- Either will accept any edge, regardless of its polarity.



*Edge Polarity*

### Edge Score Constraints

There are two types of constraints: Position and Magnitude. You can set the caliper to use only one constraint or both. The graphical Function Editor is provided for viewing and setting each type of constraint.

- If only one constraint is selected, edges are scored based on the selected constraint.
- If both constraints are selected, each constraint accounts for 50% of the edge score.

#### Magnitude Constraint

The Magnitude constraint is based on edge values relative to the Magnitude

Threshold, which is represented in the display by two red lines.

Edges having a magnitude equal to or exceeding the Magnitude Threshold receive a score of 1. Edges with values below the Magnitude Threshold receive a score ranging from 0 to 0.999, according to a manually set magnitude constraint function.

The Magnitude Threshold value can be modified in the Advanced Parameters section of the tool interface.

A Magnitude Constraint must be defined individually for each edge of an edge pair configuration.

The following figure shows examples of two different setups for a magnitude constraint function.

To set a Magnitude Constraint:

1. In the drop-down list above the Function Editor, select First Edge Magnitude Constraint or Second Edge Magnitude Constraint.  
If either edge does not have the Magnitude box checked, it will not be shown in the drop-down list.
  2. In the Function Editor, use the mouse to drag handles and set the magnitude limits. See examples in the following figure.



## *Setting the Magnitude Constraint in the Function Editor*

## Position Constraint

Position constraints restrict the caliper's search for edges to a specific zone of the region of interest.

- It is possible to graphically set a position constraint function when the approximate position of an edge is known beforehand. This is useful for

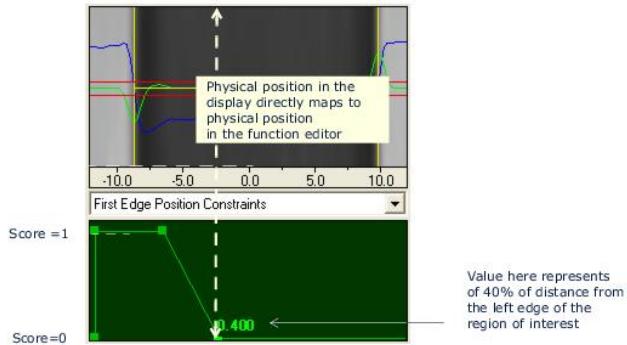
scoring an edge based on its offset from the expected position.

- Values in the Constraint Function Editor indicate relative distance in the region of interest where 0.0 is the leftmost position and 1.0 is the right-most position.

To set a Position Constraint:

1. In the drop-down list above the Function Editor, select First Edge Position Constraint or Second Edge Position Constraint.  
If either edge does not have the Position box checked, it will not be shown in the drop-down list.
2. In the Function Editor, use the mouse to drag handles and set the position limits. See the following figure.

The physical position in the Function Editor corresponds to the same physical position in the display.



### *Setting the Position Constraint Function Editor*

## Score Threshold

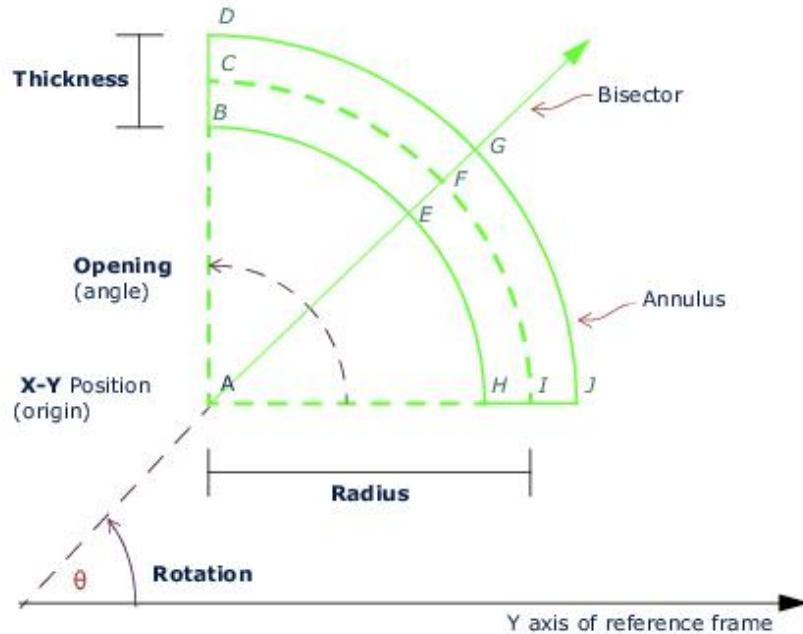
The score threshold sets the minimum acceptable score for a valid edge. The caliper will disregard edges that obtain a score lower than the Score Threshold.

- Caliper constraint scores range from 0 to 1.
- If both Position and Magnitude constraints are enabled, each constraint accounts for 50% of the total edge score.

## Region of Interest (ROI)

Most tool position parameters can be set through the Region of Interest section of the tool interface. The following are the parameters that define the tool region of interest:

Object	Definition
Opening	Angle between the two bounding radii of the tool sector.
Radius	The radius of the tool corresponds to the radius of the median annulus of the tool sector.
Thickness	Distance between the two bounding annuli of the tool sector. Note that the thickness cannot be more than twice the Radius, to prevent the ROI from overlapping itself.
Rotation	Angle of rotation of the region of interest.
Center X	X coordinate of the origin of the tool.
Center Y	Y coordinate of the origin of the tool.



Example of Tool Position for a Sector-type Region of Interest

#### Related Topics

[Arc Caliper Tool](#)

[Configuring Arc Caliper Properties - Advanced](#)[Arc Caliper Results](#)***Configuring Arc Caliper Properties - Advanced***

The Advanced Properties section of the Arc Caliper tool interface provides access to advanced Arc Caliper parameters and properties.

***Tool Sampling***

Sampling refers to the procedure used by the tool for gathering values within the portion of the input image that is bounded by the tool region of interest. Two sampling parameters, the Sampling Step and Bilinear Interpolation, can be used to create the desired balance between speed and accuracy.

**Bilinear Interpolation**

Bilinear Interpolation uses pixel averaging to improve the quality of the image. It is applied before the image is analyzed.

To ensure subpixel accuracy in inspection applications, Bilinear Interpolation should always be set to True (enabled). Non-interpolated sampling (Bilinear Interpolation disabled) should only be used in applications where the speed requirements are more critical than accuracy.

**Custom Sampling Step (1 - 100)**

The Sampling Step is the step used by the tool to sample the area of the input image that is bounded by the tool region of interest. The sampling step represents the height and the width of a sampled pixel.

For applications where a more specific tradeoff between speed and accuracy must be established, the sampling step can be modified by enabling the Custom Sampling Step and setting the desired value.

Object	Definition
Default	Default is the best sampling step computed by the tool. It is based on the average size, in calibrated units, of a pixel in the image. This default sampling step is usually recommended. Default is automatically used by the tool if no other value is specified.

---

Object	Definition
Custom	<p>Custom Sampling Step lets you set a sampling step value other than the default. To select a custom sampling step, Enable must be checked.</p> <ul style="list-style-type: none"><li>Increasing the sampling step value reduces the tool accuracy and decreases the execution time.</li><li>Reducing the sampling step can increase the tool accuracy but will also increase the execution time.</li></ul>

### Edge Filter Half-Width

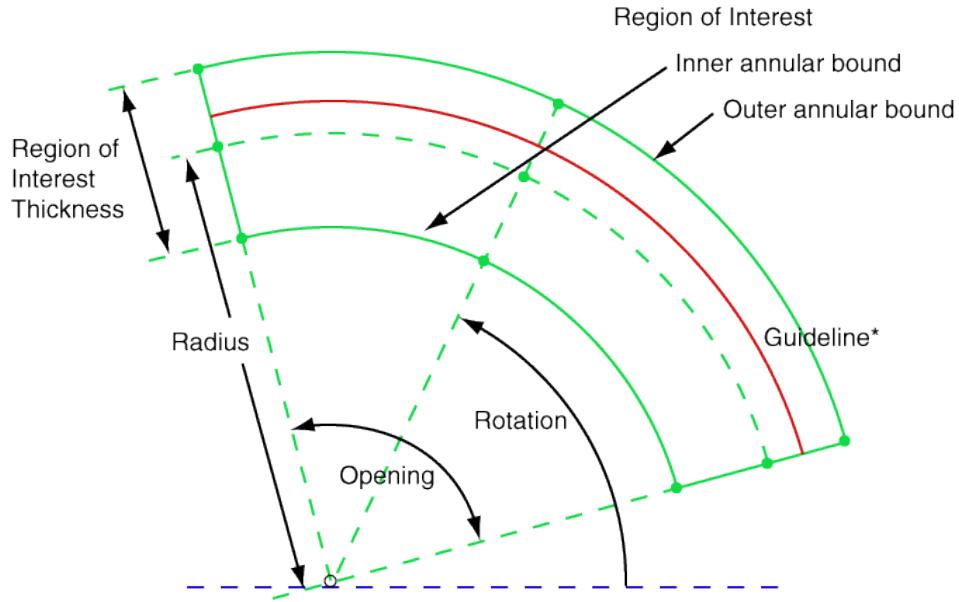
The filtering process smoothes out peaks in the magnitude curve that are caused by noise. Filter Half-Width should be set to approximately the width of the edge, in pixels. An incorrect value can cause edges to be incorrectly detected.

### Projection Mode

The two available projection modes are:

- Annular - Annular projection is used to find edges that are aligned with the median annulus, such as arcs on concentric circles.

The dotted arc in the following figure is the median annulus.



\*Only used for “Arc Closest To Guideline” Search Mode

- Radial - Radial projection is used to find edges aligned along radial projections, much like the spokes of a wheel.

The default Projection Mode is Radial.

### Results Logging

If enabled, the results of the tool will be logged to a file.

If the extension of the destination file is csv, then the log is written in csv format, which opens with Excel. If a different extension is selected, the log is written in a plain text format (similar to the ACE Sight 2 format).

The size of the log file can be set (default size is 5 Mb).

### Related Topics

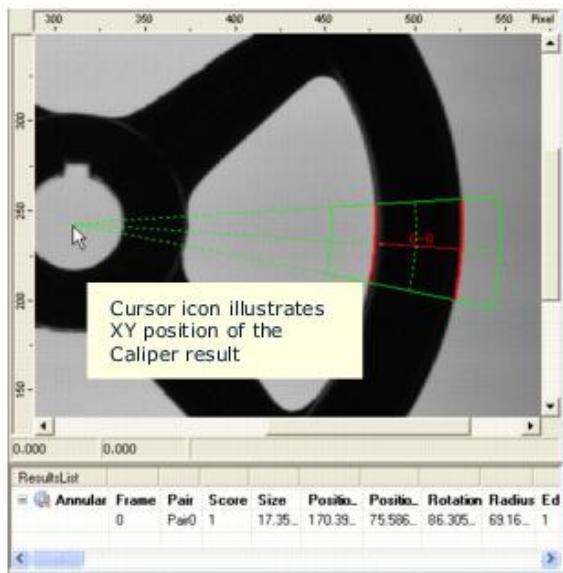
[Arc Caliper Tool](#)

[Configuring Caliper Properties](#)

[Arc Caliper Results](#)

### Arc Caliper Results

Results for edges found by the Arc Caliper tool are shown in the grid of results, below the Display window, as shown in the following figure.



*Representation of Arc Caliper Results in Display and Results Grid*

## Display Window

The Display window shows the image being processed, the region of interest selected, and the locations of found instances. The elapsed time is displayed at the bottom of this window.

### Grid of Results

The grid of results presents the result values for all caliper measures found by the Arc Caliper tool.

Object	Definition
Pair Name	The name of the edge pair, as it appears in the Pairs list.
Pair Score	Score is the calculated score, between 1 and 0, for the edge pair. The score is calculated according to the defined constraint functions. If both Position and Magnitude constraints are enabled, each constraint accounts for 50% of the score.  Each edge of the pair is also scored individually, in a similar manner. See Edge1/Edge2 Score.
Pair Size	Size is the Caliper measure, which is the calculated distance between the pair of edges.

Object	Definition
Pair Radius	Mean of Edge 1 Radius and Edge 2 Radius.
Pair X	The X coordinate of the center point of the caliper measure, at the midpoint of the edge pair.
Pair Y	The Y coordinate of the center point of the caliper measure, at the midpoint of the edge pair.
Frame/Group	The frame or group to which the result belongs.
Edge 1/Edge 2 Score	The score of the individual edge, calculated according to the defined constraints.
Edge 1/Edge 2 X	The X coordinate of the edge, at the midpoint of the edge segment.
Edge 1/Edge 2 Y	The Y coordinate of the edge, at the midpoint of the edge segment.
Edge 1/Edge 2 Rotation	The angle of rotation for the edge.
Edge 1/Edge 2 Radius	Radius of detected arc.
Edge 1/Edge 2 Position Score	Position score for the edge, calculated according to the Position constraint function.
Edge 1/Edge 2 Magnitude	The calculated Magnitude value for the edge.
Edge 1/Edge 2 Magnitude Score	Magnitude score for the edge, calculated according to the Magnitude constraint function.

#### **Related Topics**

[Arc Caliper Tool](#)

[Configuring Arc Caliper Properties](#)

[Configuring Arc Caliper Properties - Advanced](#)

---

## **Arc Edge Locator**

The Arc Edge Locator tool identifies and measures the position of one or more edges on a circular object. Edges can occur in a radial or an annular position.

The Arc Edge Locator uses pixel grey-level values within the region of interest to build projections needed for edge detection.

---

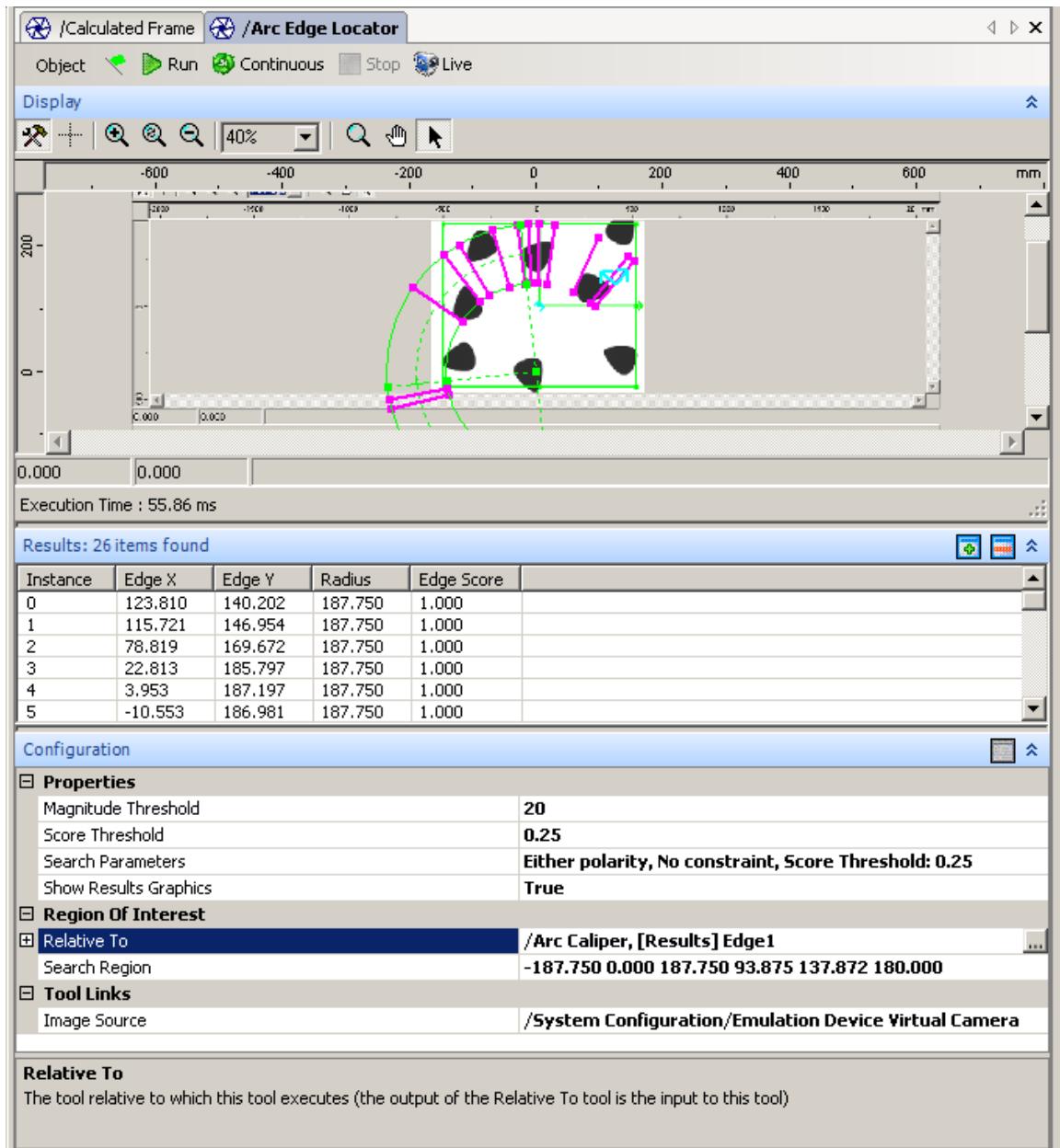
After the Arc Edge Locator detects potential edges, it determines which edge pairs are valid by applying the constraints that are configured for each edge pair.

The Arc Edge Locator determines the position of one or more edges, but it does not measure the length of lines detected in the region of interest. To extrapolate and measure a line on an object, use the Edge Finder tool.

To create an Arc Edge Locator tool, right-click in the Tree structure, then select:

**New > Vision > Tool > Arc Edge Locator**

## Arc Edge Locator



*Arc Edge Locator Object Editor*

***Configuration***

Object	Definition
<b>Properties</b>	
Magnitude Threshold	The Magnitude Threshold sets the acceptable magnitude value for potential edges.
Edge Score Threshold	Minimum score to accept an edge. The score of an edge is returned by the Edge Score result.
Search Parameters	Show all edges found by the tool.
Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
<b>Region of Interest (ROI)</b>	
Relative To	Specifies the tool that will provide the input to this tool. The tool can be dragged from the Folder pane and dropped into this field.
Search Region	Defines the Center, Radius, Rotation, Thickness, and Opening of the ROI.
<b>Tool Links</b>	
Image Source	Defines the image source used for processing by this vision tool.
<b>Advanced Properties</b>	
Bilinear Interpolation Enabled	Specifies if bilinear interpolation is used to sample the input image. By default, bilinear interpolation is enabled because it improves accuracy.
Custom Sampling Step	When enabled, the tool uses the user-defined sampling step instead of the default optimal sampling step to sample the region of interest from the input image.
Projection Mode	Projection mode used by the tool to detect edges.
Edge Filter Half-Width	Width of the edge, in pixels, as it appears in the image.
Sort Results Enabled	Specifies if the results should be sorted.

***Results***

Object	Definition
Instance	The result instance (numbered from 0).

Object	Definition
Edge X	The X coordinate of the located edge (midpoint of located arc).
Edge Y	The Y coordinate of the located edge (midpoint of located arc).
Radius	Radial Projection: radius of region of interest. Annular Projection: radius of the found edge.
Edge Score	Score of the selected edge.
Magnitude	Magnitude indicates how well the ROI arc matches the found arc.
Magnitude Score	Magnitude score of the selected edge.
Position Score	Position score of the selected edge.
Projection Average	Average Projection.
Projection Magnitude	Projection Magnitude.

### ***Edge Constraint Editor***

The Arc Edge Locator tool uses a form for configuring the edge constraints - [See "Configuring Arc Edge Locator Settings".](#)

### ***Related Topics***

[Configuring Arc Edge Locator Properties](#)

[Configuring Arc Edge Locator Properties - Advanced](#)

[Arc Edge Locator Results](#)

### ***Configuring Arc Edge Locator Properties***

When the Arc Edge Locator is executed, the Arc Edge Locator first applies edge detection parameters to the entire region of interest. Then, the tool applies edge scoring constraints to determine which edges are valid edges.

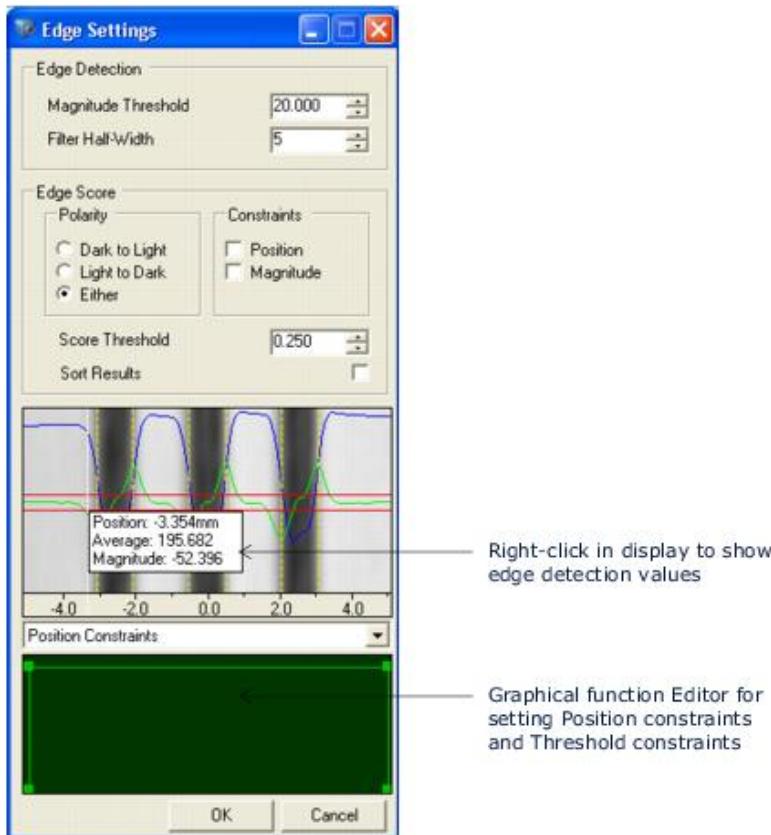
Edge Settings parameters determines how the tool detects edges and which edges are valid.

Before configuring the Arc Edge Locator, execute the tool at least once and verify that the tool is being positioned correctly in the image.

1. Under the Edges section of the interface, click Configure.
2. The Edge Settings window opens, as shown in the following figure. This window provides edge detection settings and constraints, as well as visual aids for configuring

edge location settings.

- Refer to sections below for help on configuring edge settings, and using the display and function editor.



*The Edge Settings Window*

**NOTE:** If the display in the Edge Settings window is blank, or the edges are not properly placed, close the window and verify that the correct Projection Mode is enabled in the Advanced Parameters section. Choose between Annular and Radial. Also verify that the tool was executed after positioning the tool. Execute the tool at least once before opening the Edge Settings window.

### Edge Detection

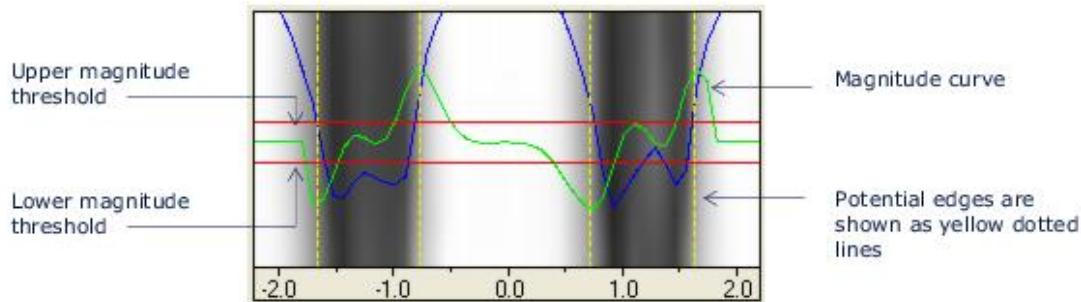
Edge Detection settings configure the parameters that the Arc Edge Locator will use to find potential edges in the region of interest. The display represents the Arc Edge Locator region of interest and provides information to assist in configuring Edge Detection parameters.

### Magnitude Threshold

Sets the acceptable magnitude value for potential edges. This value is expressed as an absolute value. There are two magnitude lines: an upper (positive) threshold and lower (negative) threshold.

Possible values are from 0 - 255.

Edge Magnitude expresses the strength of a potential edge. The (green) magnitude curve, represents magnitude values across the region of interest. Potential edges must have a magnitude greater than the upper threshold, or lower than the lower threshold. See the following figure.



*Interpreting the Magnitude Threshold in the display area*

### Edge Constraint Editor

This tool uses a form for defining the constraints.

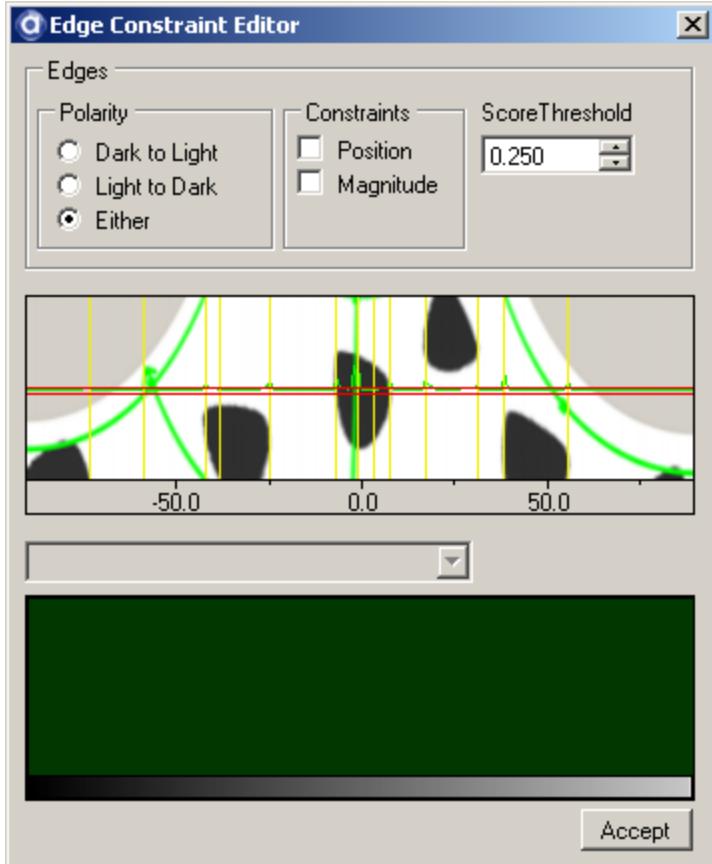
To set constraints, from the object editor:

1. Select **Configuration > Properties > Search Parameters**

Click Search Parameters to select it.

2. Click the browse icon (...

The following window opens.



*Edge Constraint Editor Window*

### **Edge Score**

The tool scores potential edges according to the specified edge parameters. The scoring method restricts the search so that only results for valid edge pairs are returned.

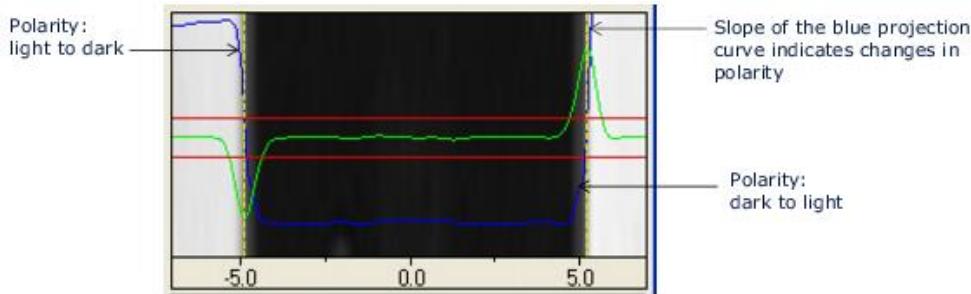
There are two types of parameters that affect the choice of valid edges: Polarity and Edge Score Constraints.

#### **Polarity**

Polarity corresponds to the change in light values, moving from left to right in the display, along the X-Axis in the region of interest. The tool applies the Polarity constraint before applying edge-score Constraints.

Polarity does not affect the Edge Score, but only edges that meet the selected Polarity are output as valid edges, regardless of their scores.

- Dark to Light will only accept edges occurring at transitions from a dark area to a light area.
- Light to Dark will only accept edges occurring at transitions from a light area to a dark area.
- Either will accept any edge, regardless of polarity.



*Edge Polarity*

### Edge Score Constraints

There are two types of constraints: Position and Magnitude. You can set the tool to use only one constraint or both. A graphical function editor is provided for viewing and setting each type of constraint.

- If only one constraint is selected, edges are scored based on the selected constraint.
- If both constraints are selected, each constraint accounts for 50% of the edge score.

### Magnitude Constraint

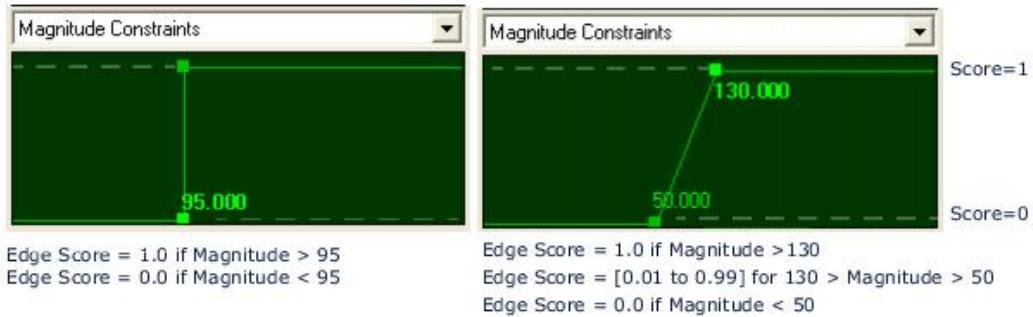
The Magnitude constraint is based on edge values relative to the Magnitude Threshold. Edges having a magnitude equal to or exceeding the Magnitude Threshold are given a score of 1. Edges with values below the Magnitude Threshold receive a score ranging from 0 to 0.999, according to a manually set magnitude constraint function.

The Magnitude Constraint is applied globally to all edges detected.

The following figure shows two different setups for a magnitude constraint function.

To set the Magnitude Constraint:

1. In the drop-down list above the function editor, select Magnitude Constraints.
2. In the Function Editor, use the mouse to drag handles and set the Magnitude limits. See examples in the following figure.



#### *Setting the Magnitude Constraint in the Function Editor*

### **Position Constraint**

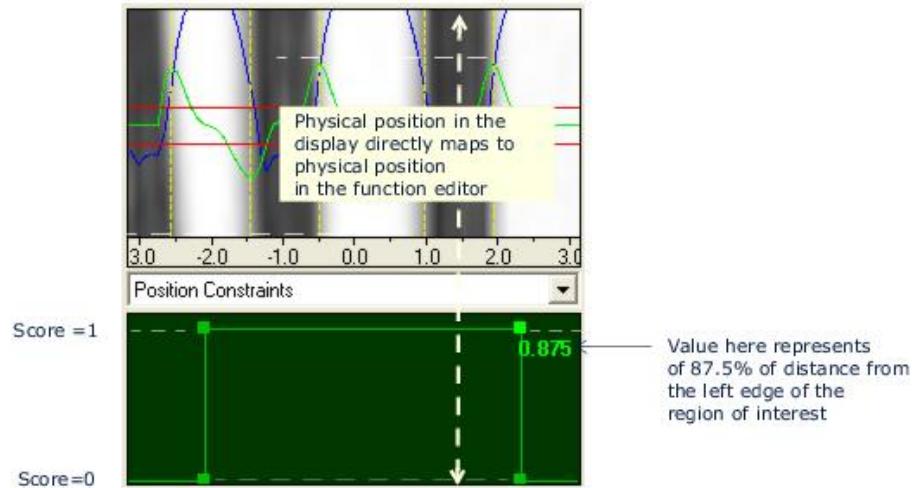
The Position constraint restricts the tool's search for edges to a specific zone of the region of interest.

- It is possible to graphically set a position constraint function when the approximate position of an edge is known beforehand. This is useful for scoring an edge based on its offset from the expected position.
- Values in the Constraint Function Editor indicate relative distance in the region of interest where 0.0 is the left-most position and 1.0 is the right-most position.

To set the Position Constraint:

1. In the drop-down list above the function editor, select Position Constraints.
2. In the Function Editor, use the mouse to drag handles and set the Position limits. See examples in the following figure.

The position in the function editor corresponds to the same position in the display.



### *Setting the Position Constraint Function Editor*

#### **Score Threshold**

The score threshold sets the minimum acceptable score for a valid edge. The tool will disregard edges that obtain a score lower than the Score Threshold.

- Scores attributed for constraints range from 0 to 1.
- If both Position and Magnitude constraints are enabled, each constraint accounts for 50% of the total edge score.

#### **Sort Results**

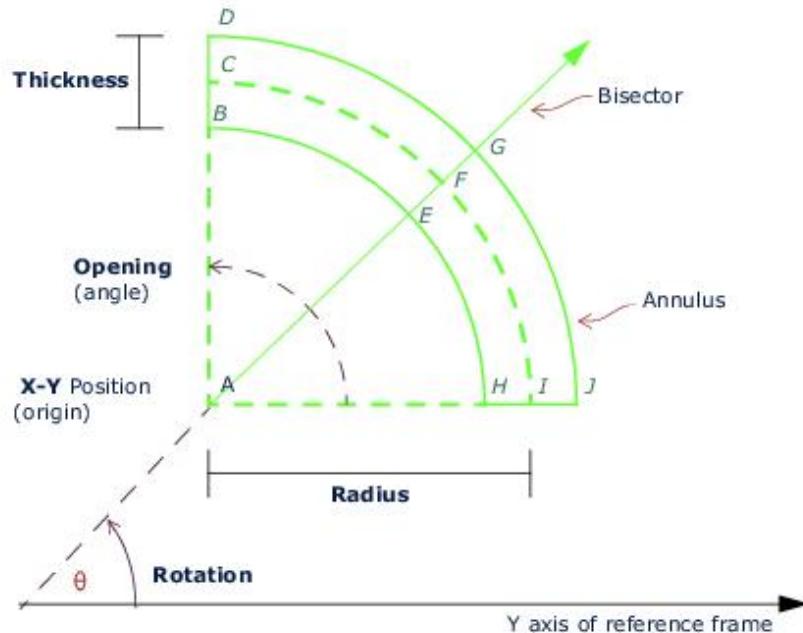
You can enable the Sort Results checkbox to sort the located edges in descending order based on values. By default, Sort Results is not enabled and edges are output in the same left to right order as they appear on the projection curve.

#### **Region of Interest (ROI)**

Most tool position parameters can be set through the Region of Interest section of the tool interface. The following are the parameters that define the tool region of interest:

Object	Definition
Opening	Angle between the two bounding radii of the tool sector.

Object	Definition
Radius	The radius of the tool corresponds to the radius of the median annulus of the tool sector.
Thickness	Distance between the two bounding annuli of the tool sector. Note that the thickness cannot be more than twice the Radius, to prevent the ROI from overlapping itself.
Rotation	Angle of rotation of the region of interest.
Center X	X coordinate of the origin of the tool.
Center Y	Y coordinate of the origin of the tool.



*Example of Tool Position for a Sector-type Region of Interest*

### Related Topics

[Arc Edge Locator](#)

[Configuring Arc Edge Locator Properties - Advanced](#)

## Arc Edge Locator Results

### **Configuring Arc Edge Locator Properties - Advanced**

The Advanced Properties section of the Arc Edge Locator tool interface provides access to advanced Arc Edge Locator parameters and properties.

#### **Tool Sampling Parameters**

Sampling refers to the procedure used by the tool for gathering values within the portion of the input image that is bounded by the tool region of interest. Two sampling parameters, the Sampling Step and Bilinear Interpolation, can be used to create the desired balance between speed and accuracy.

#### **Bilinear Interpolation**

Bilinear Interpolation uses pixel averaging to improve the quality of the image. It is applied before the image is analyzed.

To ensure subpixel accuracy in inspection applications, Bilinear Interpolation should always be set to True (enabled). Non-interpolated sampling (Bilinear Interpolation disabled) should only be used in applications where the speed requirements are more critical than accuracy.

#### **Custom Sampling Step (1 - 100)**

The Sampling Step is the step used by the tool to sample the area of the input image that is bounded by the tool region of interest. The sampling step represents the height and the width of a sampled pixel.

For applications where a more specific tradeoff between speed and accuracy must be established, the sampling step can be modified by enabling the Custom Sampling Step and setting the desired value.

Object	Definition
Default	Default is the best sampling step computed by the tool. It is based on the average size, in calibrated units, of a pixel in the image. This default sampling step is usually recommended. Default is automatically used by the tool if no other value is specified.

---

Object	Definition
Custom	<p>Custom Sampling Step lets you set a sampling step value other than the default. To select a custom sampling step, Enable must be checked.</p> <ul style="list-style-type: none"><li>• Increasing the sampling step value reduces the tool accuracy and decreases the execution time.</li><li>• Reducing the sampling step can increase the tool accuracy but will also increase the execution time.</li></ul>

### **Edge Filter Half-Width**

The filtering process attenuates peaks in the magnitude curve that are caused by noise. Filter Half-Width should be set to a value approximately equivalent to the width of the edge, in pixels, as it appears in the image. An incorrect value can cause edges to be incorrectly detected.

### **Projection Mode**

Projection mode used by the tool to detect edges. Projection mode is either radial or annular.

### **Sort Results Enabled**

When Sort Results Enabled is set to false (default) edges are sorted in order of their location within the region of interest. When true, edges are sorted in the order of their score, from highest to lowest.

### **Results Logging**

If enabled, the results of the tool will be logged to a file.

If the extension of the destination file is csv, then the log is written in csv format, which opens with Excel. If a different extension is selected, the log is written in a plain text format (similar to the ACE Sight 2 format).

The size of the log file can be set (default size is 5 Mb).

### ***Related Topics***

[Arc Edge Locator](#)

[Configuring Arc Edge Locator Properties](#)

[Arc Edge Locator Results](#)

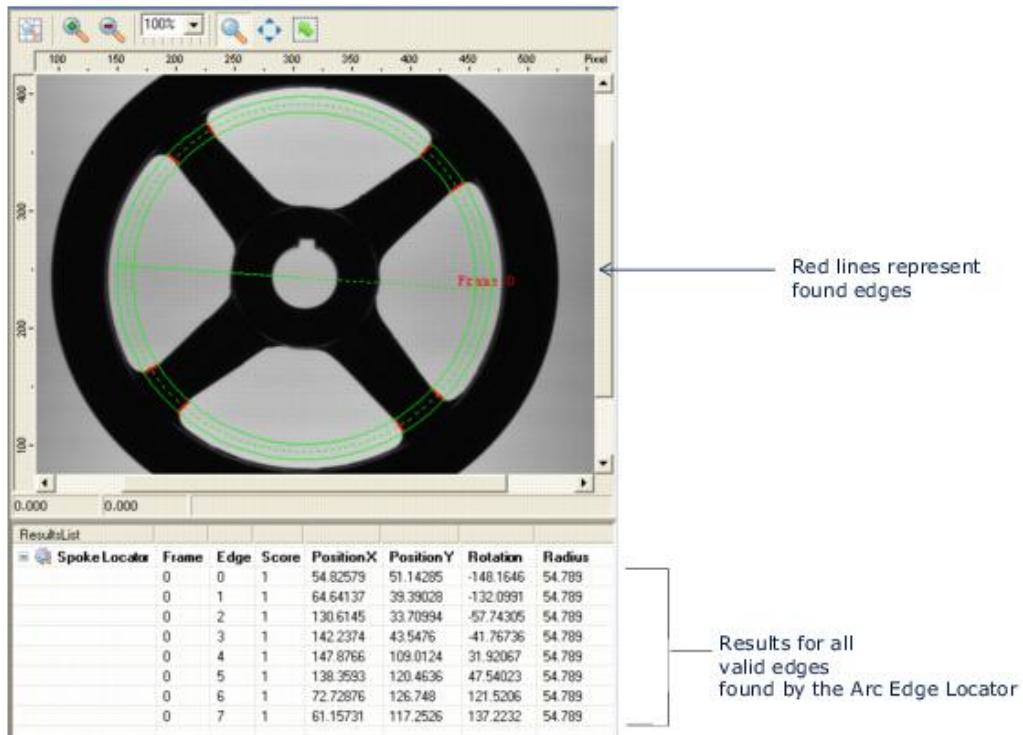
---

### Arc Edge Locator Results

The Arc Edge Locator outputs an image showing the located arc edges and a grid of results, with data on the edges.

#### Display Window

The Display window shows the image being processed, the region of interest selected, and the locations of found instances. The elapsed time is displayed at the bottom of this window.



Representation of Arc Edge Locator Results in Display and Results Grid

### Grid of Results

The grid of results presents the result values for all results found by the Arc Edge Locator tool.

The Arc Edge Locator outputs the following results:

Object	Definition
Elapsed Time	The Elapsed Time is the total execution time of the Arc Edge Locator. Elapsed Time is not visible in the results grid, but it is displayed at the bottom left of the Display window after each iteration of the Arc Edge Locator.
Instance	The identification number of the edge. Enabling Edge Sort affects the order of edge numbering.
Frame/Group	The frame or group to which the result belongs.
Edge X	The X coordinate of the center point for each edge segment.
Edge Y	The Y coordinate of the center point for each edge segment.
Edge Score	Score is the calculated score, between 0 and 1, for each edge. The score is calculated according to the defined constraint functions. If both Position and Magnitude constraints are enabled, each constraint accounts for 50% of the score.
Radius	Radius of the edge.
Position Score	Score, between 0 and 1, for the edge, calculated according to the Position Constraint function.
Magnitude	Magnitude indicates how well the region of interest arc matches the found arc. This will be negative if the found arc is a reflection of the region of interest arc. The Magnitude of the edge indicates its peak value in the magnitude curve.
Magnitude Score	Score, between 0 and 1, for the edge, calculated according to the Magnitude Constraint function.

Object	Definition
Projection Magnitude	<p>The Edge Locator processes data within the region of interest along pixel-wide lines, called projection paths, parallel to the Rectangle's Y-axis. The average grey-level value for each projection path is stored in a one-dimensional signal that can be used in the edge location process.</p> <p>The Projection Magnitude can range between -255 and 255. Positive and negative peaks in the value indicate potential edges. Sharp peaks indicate strong, well-defined edges whereas dull peaks may indicate noise or poorly-defined edges.</p>
Projection Average	The average grey-level value for all projection paths within the physical area bounded by the region of interest. This minimizes variations in pixel values caused by non-edge features or noise.

***Related Topics***

[Arc Edge Locator](#)

[Configuring Arc Edge Locator Properties](#)

[Configuring Arc Edge Locator Properties - Advanced](#)

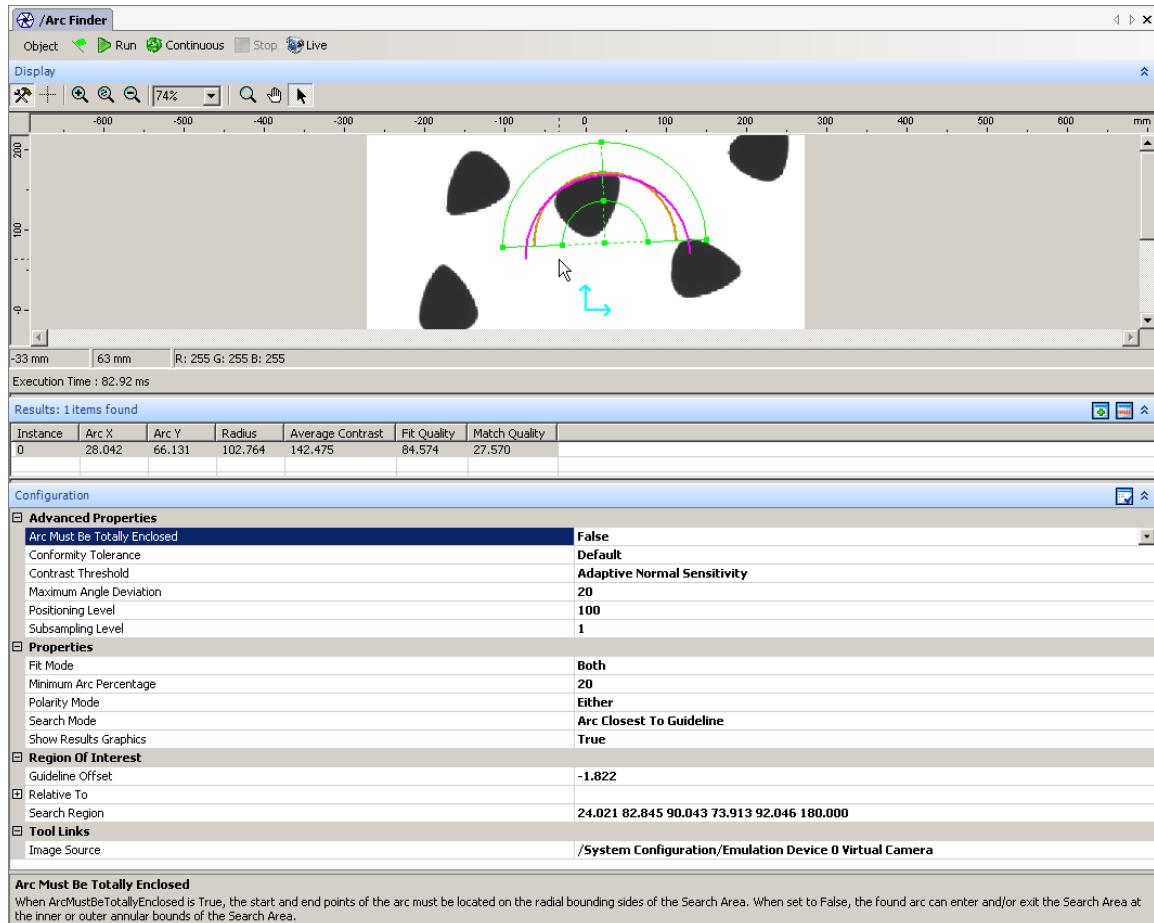
**Arc Finder**

The Arc Finder tool identifies circular features on objects and returns the coordinates of the center of the arc, the angle between the two ends, and the radius.

To create an Arc Finder tool, right-click in the Tree structure, then select:

**New > Vision > Tool > Arc Finder**

## Arc Finder



### Arc Finder Object Editor

#### Configuration

Object	Definition
<b>Properties</b>	
Fit Mode	Specifies the mode used to calculate and return values for the found arc.
Minimum Arc Percentage	Minimum percentage of arc contours that need to be matched for an arc hypothesis to be considered as valid.

Object	Definition
Polarity Mode	Selects the type of polarity accepted for finding an entity. Polarity identifies the change in grey-level values from the tool center (inside) towards the outside.
Search Mode	Specifies the method used to select a hypothesis.
Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
<b>Tool Links</b>	
Image Source	Defines the image source used for processing by this vision tool.
<b>Region of Interest (ROI)</b>	
Guideline Offset	Defines the offset, from the center of the region of interest, for the guideline.
Relative To	Specifies the tool that will provide the input to this tool. The tool can be dragged from the Folder pane and dropped into this field.
Search Region	Defines the Center, Radius, Rotation, Thickness, and Opening of the ROI.  The Thickness cannot be set to more than twice the radius, or the inner diameter would become negative. The tool prevents you from doing this.
<b>Advanced Properties</b>	
Arc Must Be Totally Enclosed	When true, the start and end points of the arc must be located on the radial bounding sides of the region of interest. When false, the found arc can enter and/or exit the region of interest at the inner or outer annular bounds of the region of interest.
Conformity Tolerance	Maximum local deviation between the expected model contours of an instance and the contours actually detected in the input image.
Contrast Threshold	Defines the minimum contrast needed for an edge to be detected in the input image and used for arc computation.

---

Object	Definition
Maximum Angle Deviation	Sets the maximum deviation in angle allowed for a detected edge to be used for generating an entity hypothesis.
Positioning Level	Sets the configurable effort level of the instance positioning process.
Subsampling Level	Sets the subsampling level used to detect edges that are used by the tool to generate hypotheses.

### **Results**

Object	Definition
Instance	Index of the result instance (numbered from 0)
Arc X	The X coordinate of the point midway between the ends of the found arc
Arc Y	The Y coordinate of the point midway between the ends of the found arc
Radius	The radius of the found arc, Arc X and Arc Y being the center of the circular arc described by the found arc
Opening	The angle of the found arc
Rotation	The rotation of the region of interest
Average Contrast	Average contrast between light and dark pixels on either side of the found arc, expressed in grey-level values.
Fit Quality	Normalized average error between the calculated arc or line entity and the actual edges matched to the found entity.
Match Quality	Percentage of edges matched to the found arc.

### **Related Topics**

[Configuring Arc Finder Properties](#)

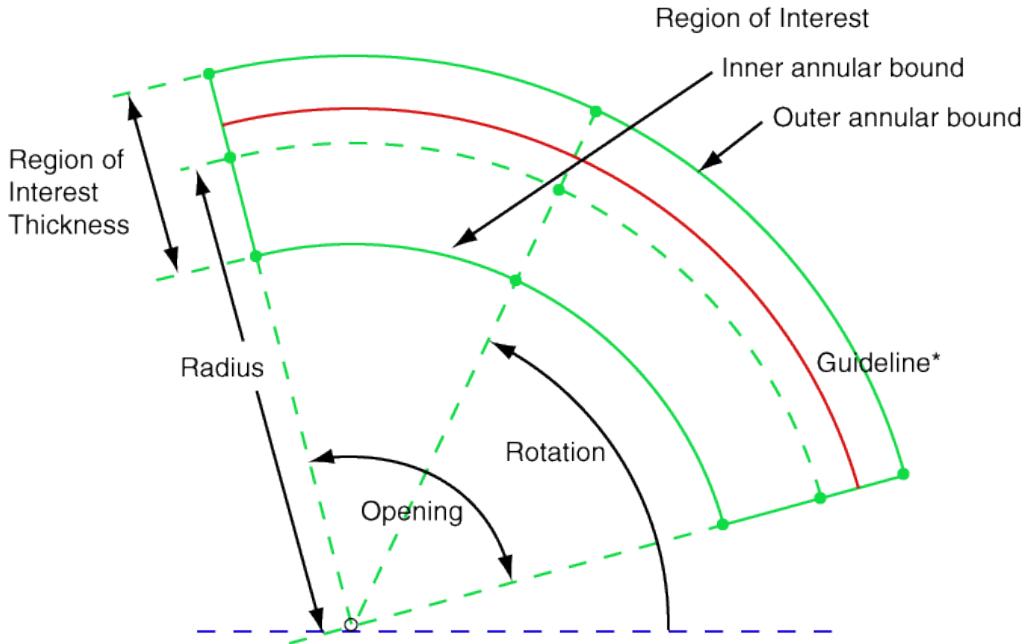
[Configuring Arc Finder Properties - Advanced](#)

[Arc Finder Results](#)

### **Configuring Arc Finder Properties**

#### **Results**

Results for each arc detected by the Arc Finder tool are shown in the grid of results, below the display.



\*Only used for “Arc Closest To Guideline” Search Mode

#### *Parts of the Arc Finder Region of Interest*

Object	Definition
Instance	Index of the result instance (numbered from 0)
Arc X	The X coordinate of the point midway between the ends of the found arc
Arc Y	The Y coordinate of the point midway between the ends of the found arc
Radius	The radius of the found arc, Arc X and Arc Y being the center of the circular arc described by the found arc

Object	Definition
Opening	The radius of the found arc, Arc X and Arc Y being the center of the circular arc described by the found arc
Rotation	The rotation of the region of interest
Average Contrast	Average contrast between light and dark pixels on either side of the found arc, expressed in grey-level values.
Fit Quality	Normalized average error between the calculated arc or line entity and the actual edges matched to the found entity. Fit quality ranges from 0 to 1, with 1 being the best quality. A value of 1 means that the average error is 0. Conversely, a value of 0 means that the average matched error is equal to Conformity Tolerance.
Match Quality	Percentage of edges actually matched to the found arc. Match Quality ranges from 0 to 1, with 1 being the best quality. A value of 1 means that edges were matched for every point along the found entity. Similarly, a value of 0.2 means edges were matched to 20% of the points along the found entity.

### Properties

#### Fit Mode

Fit Mode specifies the mode used by the tool to calculate and return values for the found arc.

There are three modes for fitting hypotheses to a valid arc entity.

- Both: The Arc Finder calculates and returns both the arc center and the arc radius. This is the default mode, which will typically provide the most accurate results.
- Center: The Arc Finder calculates the arc center. The returned center point values are the tool center (i.e. Arc X and Arc Y).
- Radius: The Arc Finder calculates the arc radius. The returned Radius value is the tool radius.

#### Minimum Arc Percentage

Minimum Arc Percentage sets the minimum percentage of arc contours that need to be matched for an arc hypothesis to be considered as valid.

## Polarity Mode

Polarity Mode sets the mode that will apply to the search for entities. Polarity identifies the change in grey-level values along the tool's X axis, in the positive direction.

The available modes are:

Mode	Description
Dark To Light	The Arc Finder searches only for arcs occurring at a dark to light transition in grey-level values.
Light To Dark	The Arc Finder searches only for arcs occurring at a light to dark transition in grey-level values.
Either	The Arc Finder searches only for arcs occurring either at a light to dark or dark to light transition in grey-level values. This mode will increase processing time.
Don't Care	The Arc Finder searches only for arcs occurring at any transition in grey-level values including reversals in contrast, for example on an unevenly colored background.

## Search Mode

Search Mode specifies the mode used by the tool to generate and select a hypothesis.

The available modes are:

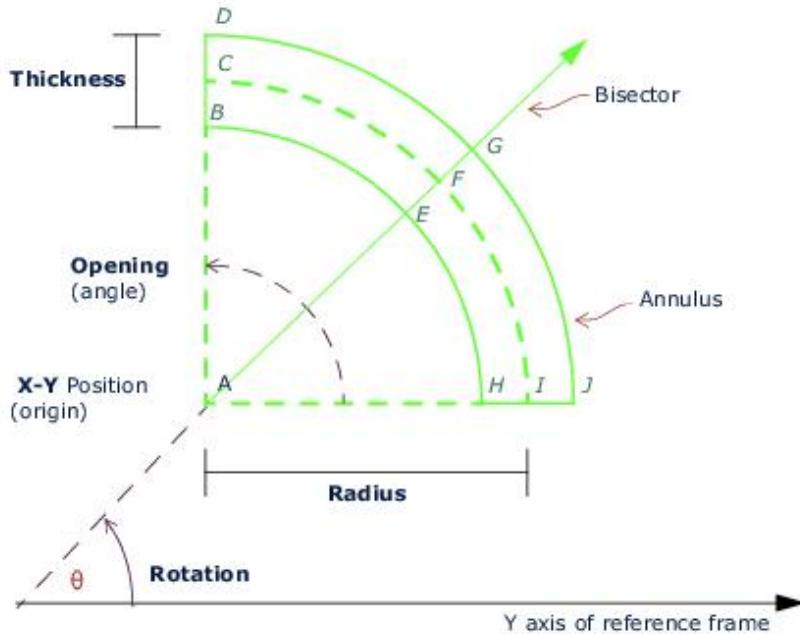
Mode	Description
Best Arc	Selects the best arc according to hypotheses strengths. This mode will increase processing time.
Arc Closest To Guideline	Selects the arc hypothesis closest to the Guideline.
Arc Closest To Inside	Selects the arc hypothesis closest to the inside of the tool Arc (closest to the tool center).
Arc Closest To Outside	Selects the arc hypothesis closest to the outside of the tool Arc (farthest to the tool center).

## Region of Interest (ROI)

Most tool position parameters can be set through the Region of Interest section of the tool interface. The following are the parameters that define the tool region of interest:

---

Object	Definition
Opening	Angle between the two bounding radii of the tool sector.
Radius	The radius of the tool corresponds to the radius of the median annulus of the tool sector.
Thickness	Distance between the two bounding annuli of the tool sector. Note that the thickness cannot be more than twice the Radius, to prevent the ROI from overlapping itself.
Rotation	Angle of rotation of the region of interest.
Center X	X coordinate of the origin of the tool.
Center Y	Y coordinate of the origin of the tool.



Example of Tool Position for a Sector-type Region of Interest

## Guideline Offset

The Guideline Offset is the offset from the tool X-axis. The Guideline marker can be displaced along the X-axis. This marker acts as both a visual guide for positioning the tool and as a constraint for the tool Search Mode.

### **Related Topics**

[Arc Finder](#)

[Configuring Arc Finder Properties - Advanced](#)

[Arc Finder Results](#)

### **Configuring Arc Finder Properties - Advanced**

The Advanced Properties section of the Arc Finder tool interface provides access to advanced Arc Finder parameters and properties.

### **Advanced Properties**

#### **Arc Must Be Totally Enclosed**

By default, Arc Must Be Totally Enclosed is true, which means that the tool will find an arc only if both its start and end points are located on the radial bounding sides of the Arc search area.

When false, the tool can find an arc that enters and/or exits the Arc at the inner or outer annular bounds of the Arc search area.

#### **Conformity Tolerance**

Conformity Tolerance corresponds to the maximum distance in calibrated units by which a matched edge can deviate from either side of its expected position on the arc.

Default Conformity Tolerance is a read-only value that is computed by the tool by analyzing the calibration, the edge detection parameters, and the search parameters.

To manually set the Conformity Tolerance, check the Enable box.

- If a value lower than the Minimum Conformity Tolerance value is set, the Conformity Tolerance value is automatically reset to the minimum valid value.
- If a value higher than the Maximum Conformity Tolerance value is set, the Conformity Tolerance value is automatically reset to the maximum valid value.

#### **Contrast Threshold**

Contrast Threshold sets the minimum contrast needed for an edge to be detected in the input image. The threshold value expresses the step in grey-level values required to detect

---

edges.

### **Contrast Threshold Modes**

Contrast Threshold Mode defines how contrast threshold is set. Contrast threshold is the level of sensitivity that is applied to the detection of edges in the input image. The contrast threshold can be either Adaptive, or Fixed.

Adaptive thresholds set a sensitivity level based on image content. This provides flexibility to variations in image lighting conditions and contrast during the Search process.

- Adaptive Low Sensitivity uses a low sensitivity, adaptive threshold for detecting edges. Adaptive Low Sensitivity detects strongly defined edges and eliminates noise, at the risk of losing significant edge segments.
- Adaptive Normal Sensitivity sets a default sensitivity threshold for detecting edges.
- Adaptive High Sensitivity detects weaker edges, at the risk of adding noise.
- Fixed Value sets an absolute value for sensitivity to contrast. A typical use of a fixed value is an application in which there is little variance in lighting conditions.
  - Higher values reduce sensitivity to contrast. This reduces noise and decreases the number of low-contrast edges detected.
  - Lower values increase sensitivity to contrast. This detects more edges, at the expense of adding more noise. This may generate false detections and/or slow the search process.

### **Maximum Angle Deviation**

Maximum Angle Deviation is the maximum allowable deviation in angle between the arc hypothesis and the arc found by the tool.

The deviation is calculated between the tangent angle of the arc at points where the edge is matched to the arc.

### **Positioning Level**

Positioning Level sets the effort level of the instance-positioning process. The range is from 10 (coarser positioning and lower execution time) to 100 (high-accuracy positioning of arcs).

### **Subsampling Level**

This property sets the subsampling level used to detect edges that are used by the tool to generate hypotheses.

---

- Higher values provide a coarser search, with a faster execution time.  
A higher subsampling value may help improve accuracy in blurry images.
- Lower values can provide a more refined search, with slower execution time.

## **Results Logging**

If enabled, the results of the tool will be logged to a file.

If the extension of the destination file is csv, then the log is written in csv format, which opens with Excel. If a different extension is selected, the log is written in a plain text format (similar to the ACE Sight 2 format).

The size of the log file can be set (default size is 5 Mb).

### ***Related Topics***

[Arc Finder](#)

[Configuring Arc Finder Properties](#)

[Arc Finder Results](#)

### ***Arc Finder Results***

#### ***Display Window***

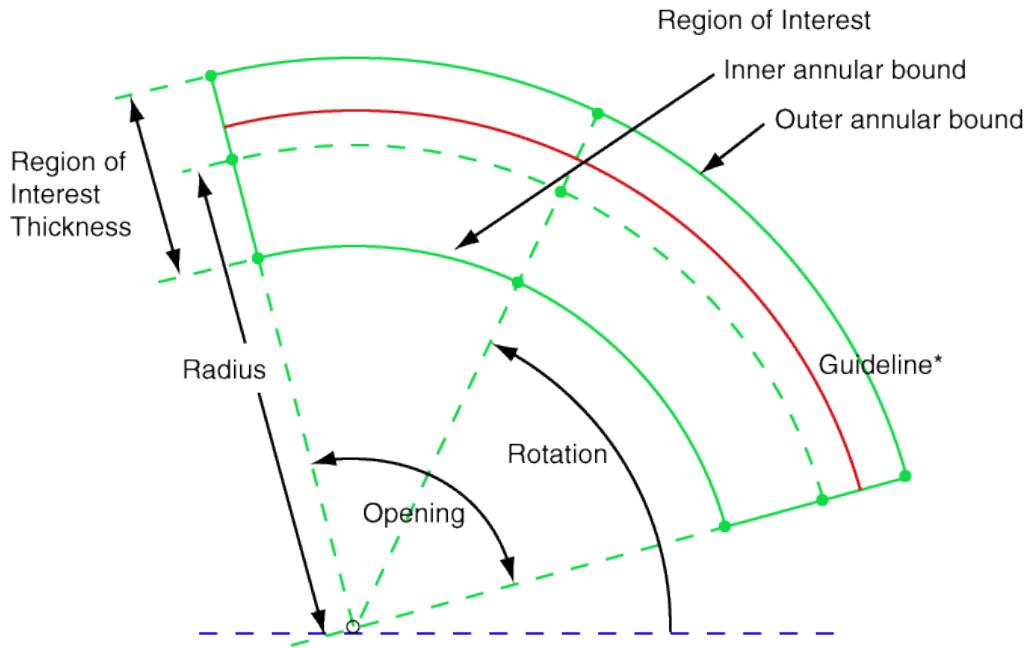
The Display window shows the image being processed, the region of interest selected, and the locations of found instances. The elapsed time is displayed at the bottom of this window.

#### ***Grid of Results***

The grid of results presents the result values for all caliper measurements found by the Arc Finder tool.

## Arc Finder

---



\*Only used for “Arc Closest To Guideline” Search Mode

### *Parts of the Arc Finder Region of Interest*

The Arc Finder outputs the following results:

Object	Definition
Instance	Index of the result instance (numbered from 0).
Frame/Group	The frame or group to which the result belongs.
Arc X	The X coordinate of the point midway between the ends of the found arc.
Arc Y	The Y coordinate of the point midway between the ends of the found arc.
Radius	The radius of the found arc, Arc X, Arc Y being the center of the circular arc described by the found arc.
Opening	The angle of the found arc.
Rotation	The rotation of the region of interest.

Object	Definition
Average Contrast	Average contrast between light and dark pixels on either side of the found arc, expressed in grey-level values.
Fit Quality	Normalized average error between the calculated arc or line entity, and the actual edges matched to the found entity. Fit quality ranges from 0 to 1, with 1 being the best quality. A value of 1 means that the average error is 0. Conversely, a value of 0 means that the average matched error is equal to Conformity Tolerance.
Match Quality	Percentage of edges matched to the found arc. Match Quality ranges from 0 to 100. A value of 100 means that edges were matched to 100% of the points along the found entity. Similarly, a value of 20 means edges were matched to 20% of the points along the found entity.

#### ***Related Topics***

[Arc Finder](#)

[Configuring Arc Finder Properties](#)

[Configuring Arc Finder Properties - Advanced](#)

#### **Blob Analyzer**

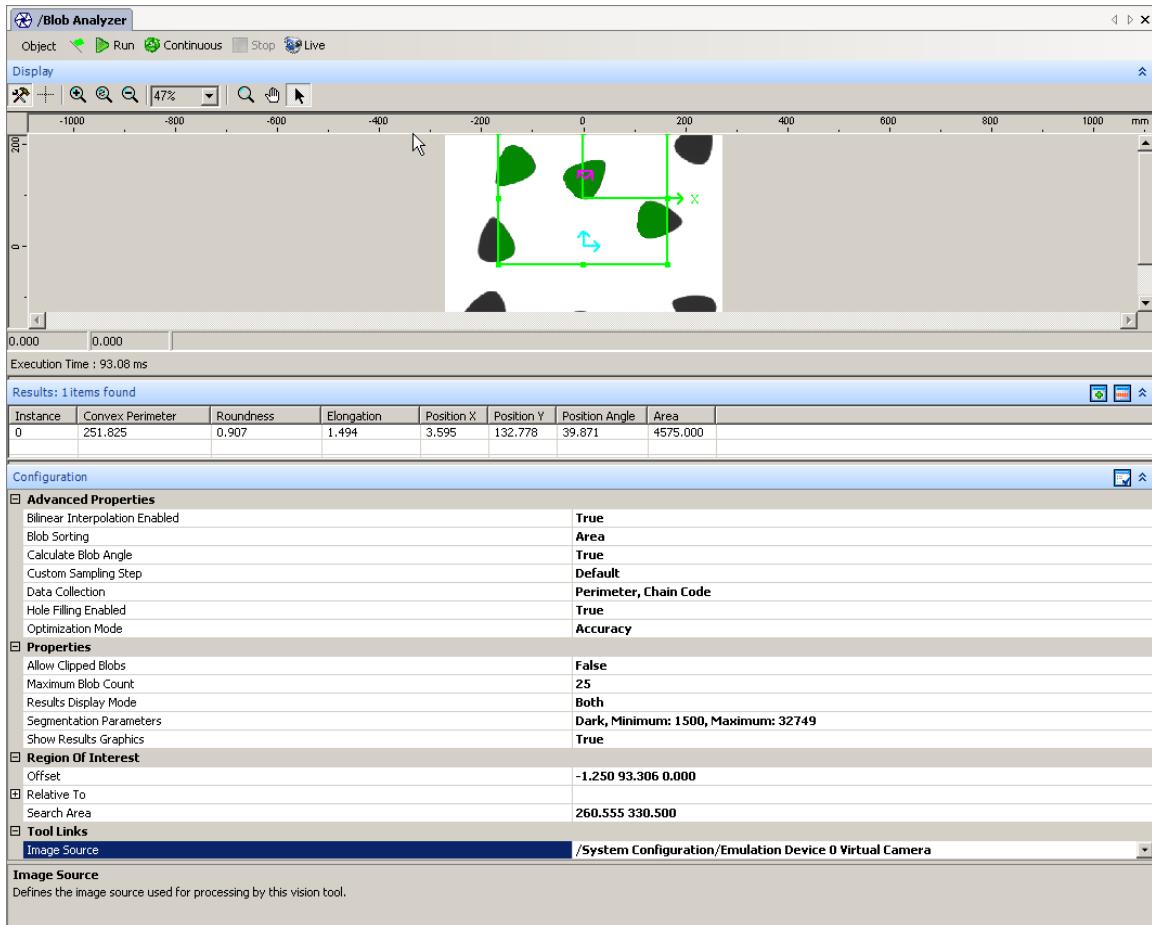
The Blob Analyzer processes pixel information within a rectangular region of interest. The tool uses these pixel values to apply image segmentation algorithms for blob detection.

A blob is any region within a greyscale image with a range of grey-level values that differs from the adjoining areas of the image.

To create a Blob Analyzer tool, right-click in the Tree structure, then select:

**New > Vision > Tool > Blob Analyzer**

## Blob Analyzer



*Blob Analyzer Object Editor*

### **Image Segmentation**

Image Segmentation algorithms provide thresholding functions that allow the Blob Analyzer to categorize regions as blobs or background. User-defined criteria modify the Blob Analyzer's classification of valid blobs.

The Blob Analyzer returns an array of numerical results for each valid blob that has been found and located. Blob results include geometric, topological, and grey-level properties.

### **Histogram**

For every new image, the Blob Analyzer generates a histogram representing the distribution of the pixel values in the region of interest. Pixel values in the histogram range from 0 (black) to 255 (white).

### ***Thresholding***

Thresholds are used to segment the image into two types of pixels: blob pixels and background pixels. Depending on the segmentation mode selected, either one or two thresholds can be used.

There are two types of thresholding functions:

#### **Hard Thresholding**

Hard Thresholding (also referred to as binary thresholding), segregates pixels into one of two states: 0 for background pixels, and 1 for blob pixels. The result is a binary image (for example, a white blob on a black background). Hard thresholding assumes that changes in data values occur at the boundary between pixels, without allowing for a variation in grey-level values across blob boundaries. Because this is rarely the case, soft thresholding (described in the next section) is more often used for applications.

#### **Soft Thresholding**

Soft Thresholding provides flexibility in treating pixels that border blob regions. A soft threshold is sloped and covers a range of pixel values. Processed pixels within the threshold range are output as weighted pixels. Weighted pixels are used to calculate blob results in a proportion corresponding to their value in the threshold range.

- The range of values within a soft threshold is user-defined and corresponds to the difference between a maximum and a minimum threshold value.
- All weighted pixels appear in the image. However, they contribute to the property results in proportion to their weight in the threshold range.
- The Pixel Weight Image displays the weighted pixels in corresponding levels of grey.

#### **Dynamic Thresholding**

Dynamic Thresholding provides an adaptive thresholding mode, which is useful when there are lighting variations from one instance to another. A dynamic threshold sets a percentage of the pixel distribution (histogram) to be considered as blob pixels.

A dynamic threshold can be either a soft or hard threshold. It can be set to apply light, dark, inside, or outside segmentation.

#### **Light Segmentation**

The Light Segmentation algorithm outputs all pixels with values higher than the threshold value as blob pixels. Pixels with values below the threshold become background pixels. This mode is used for light blobs on a dark background.

---

## Dark Segmentation

The Dark Segmentation algorithm is the inverse function of the Light segmentation algorithm. The Blob Analyzer outputs all pixels with values below the threshold value as blob pixels. Pixels with values above the threshold become background pixels. This mode is used for dark blobs on a light background.

## Inside Segmentation

The Inside Segmentation algorithm uses a dual-threshold function. Only pixels with grey-level values within the threshold range are output as blob pixels. This mode is used when the pixel values of blobs fall somewhere between the values of the object and the image background.

## Outside Segmentation

The Outside Segmentation operates as an inverse function of the Inside Segmentation mode. Pixel values that fall outside the threshold range are output as blob pixels. This is useful for situations where blobs can be either very dark or very light when compared to the background.

### ***Configuration***

Refer also to Configuring Blob Analyzer Properties on page 404.

Object	Definition
<b>Properties</b>	
Allow Clipped Blobs	Enables the inclusion of clipped blobs into the result set.
Maximum Blob Count	The maximum number of blobs that the tool returns.
Results Display Mode	Defines how the results are displayed in the image display. The following options are available: <ul style="list-style-type: none"><li>• Markers</li><li>• Blob Image</li><li>• Both</li></ul>
Segmentation Parameters	Segmentation parameters used by the tool to locate the blob. Refer to <a href="#">Image Segmentation</a> .

Object	Definition
Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
<b>Region of Interest (ROI)</b>	
Offset	Specifies the center and rotation of the ROI. X, Y, Degrees.
Relative To	Specifies the tool that will provide the input to this tool. The tool can be dragged from the Folder pane and dropped into this field.
Search Area	Defines the height and width of the region of interest.
<b>Tool Links</b>	
Image Source	Defines the image source used for processing by this vision tool.
<b>Advanced Properties</b>	
Bilinear Interpolation Enabled	Specifies if bilinear interpolation is used to sample the input image. By default, bilinear interpolation is enabled because it improves accuracy.
Blob Sorting	Specifies if the found blobs are sorted; specifies sort criteria.
Calculate Blob Angle	Enables or disables the calculation of the angle of each blob.
Custom Sampling Step	When enabled, the tool uses the user-defined sampling step instead of the default optimal sampling step to sample the region of interest from the input image.
Data Collection	The data collection configuration for the analysis.
Hole Filling Enabled	Enables the filling of the holes in each blob.

Object	Definition
Optimization Mode	Defines the mode of execution (Speed or Accuracy).

### **Results**

Refer to [Blob Analyzer Results](#) for details on the results generated by the Blob Analyzer.

### **Related Topics**

[Configuring Blob Analyzer Properties](#)

[Configuring Blob Analyzer Properties - Advanced](#)

[Blob Analyzer Results](#)

### **Configuring Blob Analyzer Properties**

The Blob Analyzer tool determines which areas of an image will be output as blobs by applying Image Segmentation and Area constraints.

- The Blob Analyzer can find any number of blobs in a single image.
- Because the Blob Analyzer relies on differences in pixel grey-level (or color) values to divide the region of interest into blob and background areas, efficient blob detection depends on the appropriate choice of a segmentation mode.

**NOTE:** Before configuring the Blob Analyzer, execute the tool at least once and verify that the tool is being positioned correctly in the image.

### **Segmentation Parameters**

To configure Blob Settings:

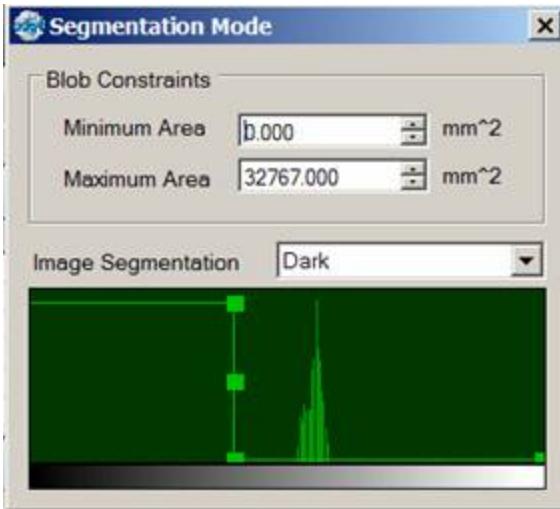
1. Under the Properties section of the interface, click Segmentation Parameters.
2. Click the browse icon .
3. The Segmentation Mode window opens, as shown in the following figure.

This window provides blob constraint parameters and a graphical editor for configuring image segmentation thresholds.

4. Set the Minimum Area and Maximum Area constraints.  
These specify the area of the smallest and largest areas of valid blobs.
5. Select an Image Segmentation mode. See [Selecting the Image Segmentation Mode](#)

for details.

6. Configure the Threshold function for the selected segmentation mode. See [Configuring Thresholds](#) for details.



The Blob Settings Window

### Setting Blob Area Constraints

Area constraints define the Minimum Area and Maximum Area of valid blobs.

Area constraints are useful for separating potential blobs from background regions, or from other blobs having similar pixel values.

All modes use Minimum Area and Maximum Area for identifying valid blobs.

### Selecting the Image Segmentation Mode

The Blob Analyzer applies image segmentation to categorize pixels within the region of interest as blob or background. The segmentation mode you choose depends on the nature of the images and the relationship between blob data and background data.

All modes except the two HSL modes provide a graph used to specify bounds for the segmentation. (See [Creating and Configuring Color Filters](#) for information on HSL.)

#### Dark Segmentation

The Dark Segmentation mode is used to extract dark blobs on a light background.

- All pixels with values to the left of the threshold function are potential blob regions.

- All pixels with values to the right of the threshold are background.
- Blobs include all pixels with a value equal to the threshold value.

### **Light Segmentation**

The Light Segmentation mode is used for extract light blobs on a dark background. This is the inverse function of the Dark Segmentation mode.

- All pixels values to the right of the threshold function are potential blob regions.
- All pixels with values to left of the threshold function are background.
- Blobs include all pixels with a value equal to the threshold value.

### **Inside Segmentation**

The Inside Segmentation mode applies a dual-threshold function. This mode is used to extract grey (neither dark nor light) blobs from a background containing dark and light areas.

Examples:

- The region of interest contains a grey blob on a light object or part, with a dark image background.
- The region of interest contains a grey blob on a dark object or part, with a light image background.

**NOTE:** Inside Segmentation should always be configured using [Soft Threshold Functions](#). In most cases the [Dynamic Inside](#) segmentation mode will provide better flexibility and better results than the Inside Segmentation mode.

### **Outside Segmentation**

The Outside Segmentation mode applies a dual-threshold function and is the inverse of the Inside mode. This mode is used for extracting dark and light blobs from a grey background.

Cases for using outside segmentation are not frequent because it is best to analyze dark and light blobs within an image by creating two Blob Analyzer tools: one for dark blobs and one for light blobs.

#### **Dynamic Dark**

The Dynamic Dark mode sets a percentage of the pixel distribution that is valid for the detection of dark blobs.

This mode is similar to the [Dark Segmentation](#) mode to which a dynamic threshold mode is applied.

- A dynamic threshold is an adaptive threshold that varies according to changes in lighting in the input images. See [Dynamic Threshold Functions](#) for more information.
- Either hard or soft thresholds can be applied to this mode.

### **Dynamic Light**

The Dynamic Light mode sets a percentage of the pixel distribution that is valid for the detection of light blobs.

This mode is similar to the [Light Segmentation](#) mode to which a dynamic threshold mode is applied.

- A dynamic threshold is an adaptive threshold mode varies according to changes in lighting in the input images. See [Dynamic Threshold Functions](#) for more information.
- Either hard or soft thresholds can be applied to this mode.

### **Dynamic Inside**

The Dynamic Inside mode sets a percentage of the pixel distribution that is valid for the detection of grey (neither dark nor light) blobs on a background containing dark and light areas.

This mode is similar to the [Inside Segmentation](#) mode to which a dynamic threshold mode is applied.

- A dynamic threshold is an adaptive threshold that varies according to changes in lighting in the input images. See [Dynamic Threshold Functions](#) for more information.
- Either hard or soft thresholds can be applied to this mode.

### **Dynamic Outside**

The Dynamic Inside mode sets a percentage of the pixel distribution that is valid for the detection of dark and light blobs on a grey (neither dark nor light) background.

This mode is similar to the [Outside Segmentation](#) mode to which a dynamic threshold mode is applied.

- A dynamic threshold is an adaptive threshold mode that varies according to changes in lighting in the input images. See [Dynamic Threshold Functions](#) for more information.
- Either hard or soft thresholds can be applied to this mode.

The two HSL modes let you specify H, S, and L values for both Color and Tolerance. These can be set numerically, using values from 0 to 255, or in a color or greyscale window, by adjusting the size and location of a selection box.

Color:

- Hue determines the colors to be considered. This is depicted and controlled by the horizontal location of the selection box within the color selection window.
- Saturation determines the intensity of the hues to be considered. This is depicted and controlled by the vertical location of the selection box within the color selection window.
- Luminance (or Lightness) sets the brightness of the colors that appear within the color selection window. Increasing luminance is like adding white to a color.

Tolerance:

- Tolerance values are distributed equally above and below the color range to which they apply. A larger tolerance value will include a larger number of colors.
- Hue tolerance determines how wide a range of colors will be considered. This is depicted and controlled by the width of the selection box.
- Saturation tolerance determines the range of intensity of colors to be considered. This is depicted and controlled by the height of the selection box.
- Luminance (or Lightness) tolerance determines the range of brightness to be considered for valid blobs. This is depicted and controlled by the width of the selection box in a grey scale selection window below the color selection window.

### **HSL Inside**

The HSL Inside mode sets an HSL range that specifies what colors can be contained in a valid blob.

### **HSL Outside**

The HSL Outside mode sets an HSL range that specifies what colors cannot be contained in a valid blob.

### **Configuring Thresholds**

Threshold functions set values at which image segmentation takes place. Depending on the segmentation mode selected, there may be either a single threshold or a double threshold.

---

There are three types of threshold functions: hard, soft and dynamic.

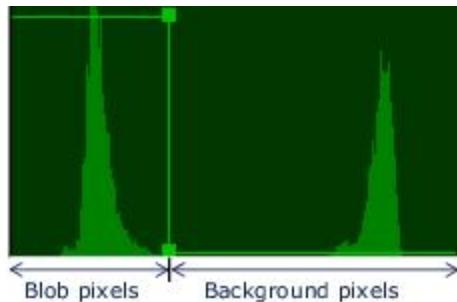
- Thresholds are modified in the threshold function Editor, using the mouse.
- The threshold function varies depending on the selected Image Segmentation mode.

### Hard Threshold Functions

Hard thresholds produce a blob image in which pixels have only two possible states: blob or background.

- Background pixels are each attributed a value of 0 in the Blob Image.
- Blob pixels are each attributed a value of 1 in the Blob Image.
- Hard thresholding is sometimes referred to as binary thresholding because all pixels can be considered as having one of two states: blob or background

Hard thresholding assumes that changes in data values occur at the boundary between pixels, without allowing a variation in pixels values across blob boundaries. Because this is rarely the case, soft thresholding is more often used for applications.



*Hard Thresholding Example*

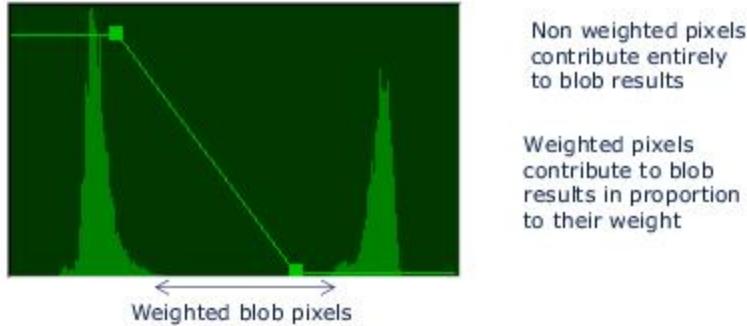
### Soft Threshold Functions

Soft thresholds let you use blob detection in cases where boundaries of a blob region span a few pixels in width, with varying grey-levels between the blob and the background.

A soft threshold is sloped and covers a range of pixel values that become weighted pixels once they are processed.

- Weighted pixels are used in the calculation of the blob's center of mass in proportion to their weighted value within the soft threshold range.

- Weighted pixels are shown as completely included within the Blob Image.

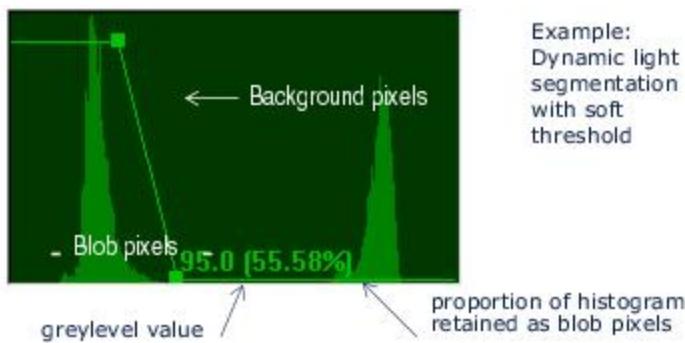


### *Soft Thresholding Example*

#### **Dynamic Threshold Functions**

Dynamic threshold modes provide the same functionality as other segmentation modes with the added advantage of adaptive threshold. A dynamic threshold is particularly useful when there are lighting variations from one image to another because the threshold is defined as a percentage of the pixel distribution, not a range of light values.

- A dynamic threshold is set as a percentage of the total pixels in the Image.
- To properly set a dynamic threshold, initially use an image that provides an "ideal blob" to determine what percentage of the image contains blob pixel values.



### *Dynamic Threshold Example*

### ***Blob Selection and Display***

Object	Definition
Allow Clipped Blobs	Enables or disables the inclusion of clipped blobs in the result set.
Maximum Blob Count	Sets the maximum number of blobs that the tool will return.
Results Display Mode	Defines how the results are displayed in the Display window. The options are: <ul style="list-style-type: none"><li>• Markers</li><li>• Blob Image</li><li>• Both</li></ul>

### ***Related Topics***

[Blob Analyzer](#)

[Configuring Blob Analyzer Properties - Advanced](#)

[Blob Analyzer Results](#)

### ***Configuring Blob Analyzer Properties - Advanced***

The Advanced Properties section of the Blob Analyzer tool interface provides access to advanced Blob Analyzer parameters and properties.

### ***Tool Sampling Parameters***

Sampling refers to the procedure used by the tool to gather values within the portion of the input image that is bounded by the tool region of interest. Two sampling parameters, the Custom Sampling Step and Bilinear Interpolation, are used to balance the tradeoff between speed and accuracy.

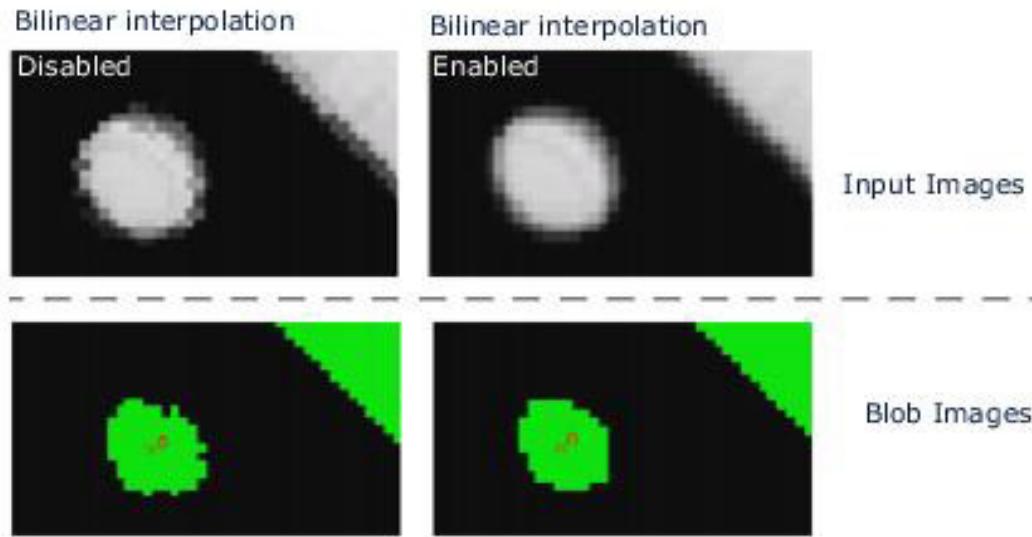
#### ***Bilinear Interpolation (True/False)***

Bilinear Interpolation specifies if bilinear interpolation is used to sample the image before it is analyzed.

Bilinear interpolation is crucial for obtaining accurate Blob Analyzer results. To ensure sub-pixel accuracy in blob results, Bilinear Interpolation should always be set to true (enabled).

If the Blob Analyzer is used in a frame-based mode, the tool region of interest, and the blobs found within it, are rarely aligned with the pixel grid. This results in jagged edges on blob borders. Therefore, interpolated pixel values provide a more true-to-life representation of blob

contours. As illustrated in the following figure, a detail from a non-interpolated image shows a blob's contour as being very jagged and irregular.



*Effect of Bilinear Interpolation on Blob Detection*

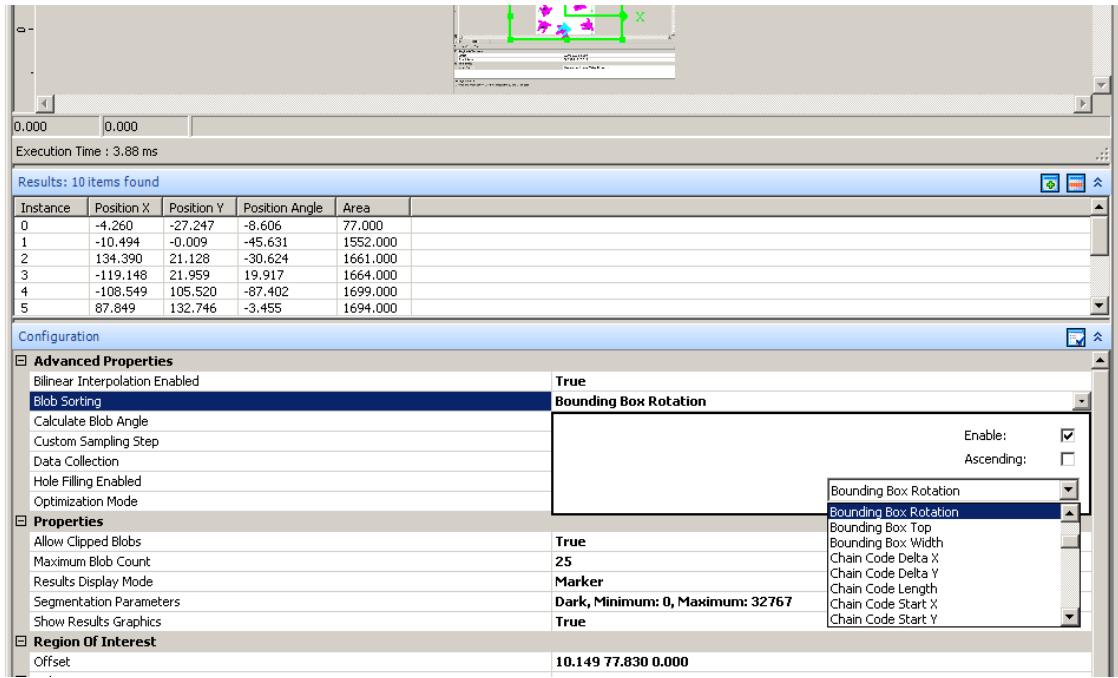
### Blob Sorting

Blob Sorting enables the sorting of blob instances, as displayed in the results grid.

When No Sorting is selected (default), blob instances are presented in the order in which they are found by the Blob Analyzer. Otherwise, blobs are sorted according to the sort criteria you set.

By default, results are displayed in descending order. You can override this by checking Ascending in the criteria drop-down box. See the following figure.

## Blob Analyzer



### Blob Sorting Drop-down Box

When Blob Sorting is highlighted, a drop-down arrow appears. This displays a criteria drop-down box when clicked. The criteria you choose will be used as the basis for sorting the blob instances, as displayed in the results grid.

To sort blobs:

1. Click on Blob Sorting to highlight the parameter.

In the criteria box:

2. Check the Enable box.
3. If you want the results sorted in ascending order, check the Ascending box.
4. Select the sort criteria that will serve as the basis for sorting the results.

These criteria are defined in [Blob Analyzer Results](#).

The sort criteria are:

Area	Hole Count
Bounding Box Bottom	Inertia Maximum
Bounding Box Center X	Inertia Minimum

Bounding Box Center Y	Inertia X Axis
Bounding Box Height	Inertia Y Axis
Bounding Box Left	Intrinsic Bounding Box Bottom
Bounding Box Right	Intrinsic Bounding Box Center X
Bounding Box Rotation	Intrinsic Bounding Box Center Y
Bounding Box Top	Intrinsic Bounding Box Left
Bounding Box Width	Intrinsic Bounding Box Right
Chain Code Delta X	Intrinsic Bounding Box Rotation
Chain Code Delta Y	Intrinsic Bounding Box Top
Chain Code Length	Intrinsic Bounding Box Width
Chain Code Start X	Intrinsic Bounding Box Height
Chain Code Start Y	Intrinsic Extent Bottom
Convex Perimeter	Intrinsic Extent Left
Elongation	Intrinsic Extent Right
Extent Bottom	Intrinsic Extent Top
Extent Left	Position X
Extent Right	Position Y
Extent Top	Principal Axes Rotation
Grey-Level Maximum	Raw Perimeter
Grey-Level Mean	Roundness
Grey-Level Minimum	
Grey-Level Range	
Grey-Level Standard Deviation	

### **Calculate Blob Angle (True/False)**

Enables the calculation of the angle of each blob. Enabling this option collects the following blob properties:

These properties are defined in [Blob Analyzer Results](#).

- Inertia Minimum
- Inertial Maximum
- Elongation
- Position Angle

### **Custom Sampling Step (1 - 100)**

The sampling step is the step used by the tool to sample the area of the input image that is bounded by the tool region of interest. The sampling step represents the height and width of a sampled pixel.

#### **Default**

Default is the best sampling step computed by the tool. This is based on the average size of a pixel in the image. This default sampling step is usually recommended. Default is automatically used by the tool if no other value is specified.

#### **Custom**

To select a custom sampling step, Enable must be checked.

- Increasing the sampling step value reduces the tool accuracy and decreases the execution time.
- Reducing the sampling step value increases the tool accuracy and increases the execution time.

### **Data Collection**

Selecting this property allows you to choose which results you want collected. You can select as many as you want from the following:

These properties are defined in Blob Analyzer Results on page 417.

- Perimeter Results
- Grey-Level Results
- Extrinsic Inertial Results
- Chain Code Results
- Intrinsic Box Results
- Topological Results

## Hole Filling Enabled (True/False)

**NOTE:** Hole Filling affects the Display window, but not the Vision window.

When Hole Filling Enabled is true, all background pixels inside within the perimeter of a given blob become included in the blob. All smaller blobs within a larger blob are also included in the "filled" larger blob. Both the background and smaller blobs are then considered as part of the filled blob.



*Example of the Hole Filling Parameter*

## Optimization Mode (Speed/Accuracy)

This parameter lets you choose between either Speed or Accuracy for execution mode.

## Results Logging

If enabled, the results of the tool will be logged to a file.

If the extension of the destination file is csv, then the log is written in csv format, which opens with Excel. If a different extension is selected, the log is written in a plain text format (similar to the ACE Sight 2 format).

The size of the log file can be set (default size is 5 Mb).

### Related Topics

[Blob Analyzer](#)

[Blob Analyzer Results](#)

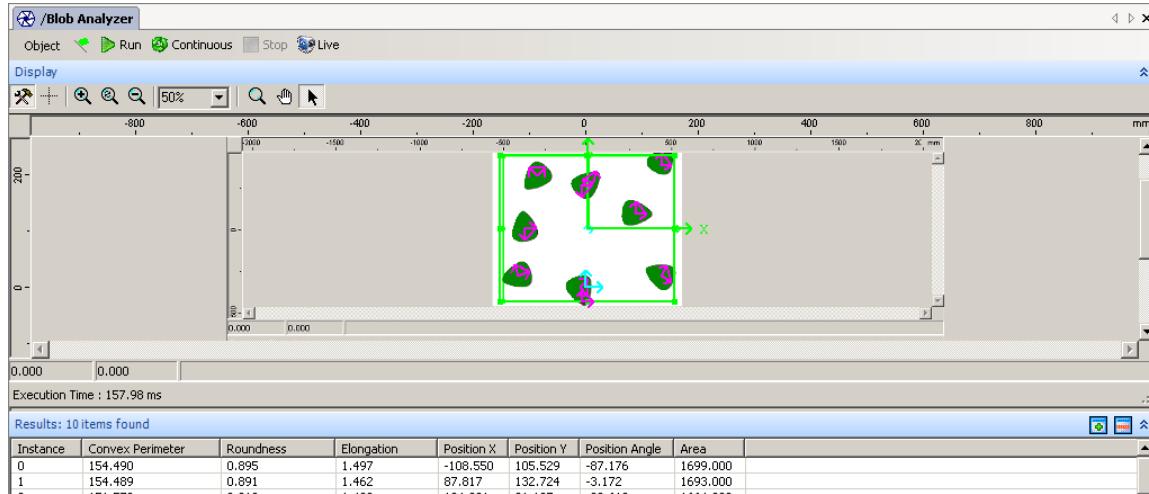
[Configuring Blob Analyzer Properties](#)

### **Blob Analyzer Results**

Results for found blobs appear as a graphic display and in the grid of results, below the display, as shown in the following figure. Found blobs are shown in green. Clicking any of the instance lines in the Results Grid turns the origin for that blob blue.

### **Display Window**

The Display window shows the image being processed, the region of interest selected, and the locations of found instances. The elapsed time is displayed at the bottom of this window.



*Blob Analyzer Results in Display Window and Results Grid*

The Display pane can show Marker, Blob Image, or Both. This is specified in the Results Display Mode property.

### **Grid of Results**

The Grid of Results presents the statistical results for the region of interest analyzed by the Blob Analyzer.

Results are described in the section Description of Blob Analyzer Results on page 418.

### **Enabling Blob Analyzer Results**

Because of the large number of results that can be calculated and output by the Blob Analyzer, only General Results are output by default.

To enable the output of other types of results, the output must be configured in the Advanced Parameters section.

**NOTE:** To optimize the tool execution time, you should enable only the results that you need for your application.

### Description of Blob Analyzer Results

Results are presented below, ordered by output group.

- General Results on page 418
- Perimeter Results on page 419
- Intrinsic Inertia Results on page 420
- Extrinsic Inertia Results on page 421
- Intrinsic Bounding Box Results on page 422
- Extrinsic Bounding Box Results on page 425
- Grey-Level Results on page 427
- Topological Results on page 427

### General Results

**NOTE:** Most Results other than General are enabled by Advanced settings, usually in the Data Collection parameter.

■ Advanced Properties	
Bilinear Interpolation Enabled	True
Blob Sorting	Area
Calculate Blob Angle	True
Custom Sampling Step	Default
■ Data Collection	Perimeter, Chain Code
Hole Filling Enabled	<input checked="" type="checkbox"/> Perimeter Results:
Optimization Mode	<input type="checkbox"/> Grey Level Results:
■ Properties	<input type="checkbox"/> Extrinsic Inertial Results:
Allow Clipped Blobs	<input checked="" type="checkbox"/> Chain Code Results:
Maximum Blob Count	<input type="checkbox"/> Intrinsic Box Results:
Results Display Mode	<input type="checkbox"/> Topological Results:
Segmentation Parameters	
Show Results Graphics	
■ Region Of Interest	
Offset	-106.594 107.699 0.000
■ Relative To	
Search Area	64.450 57.527
■ Tool Links	
Image Source	/System Configuration/Emulation Device Virtual Camera

*Data Collection Property Drop-down Box*

The Blob Analyzer outputs the following General results:

Object	Definition
Elapsed Time	The Elapsed Time is the total execution time of the Blob Analyzer. Elapsed Time is shown immediately below the Display window at each iteration of the Blob Analyzer.
Instance	Identification number of the found blob. Each blob found and output by the Blob Analyzer tool is a blob instance.
Frame/Group	The frame or group to which the result belongs.
Area	Area results are returned in squared calibrated-units or pixels depending on the positioning mode applied to the Blob Analyzer.
Position X	The X coordinate of the center of mass of the blob.
Position Y	The Y coordinate of the center of mass of the blob. <ul style="list-style-type: none"><li>• The center of mass is defined by the average position of the pixels in the blob and takes into account the effect of weighted pixel values, in the case of soft thresholding.</li><li>• In the Blob Results display, the center of mass is displayed as a crosshair and an index number. Position coordinates are referenced in the user-selected coordinate system: Tool, Object, Image, or World.</li></ul>
Position Angle	The angle of the located instance with respect to the X-axis of the selected coordinate system. This requires Calculate Blob Angle to be set to true.

#### **Perimeter Results**

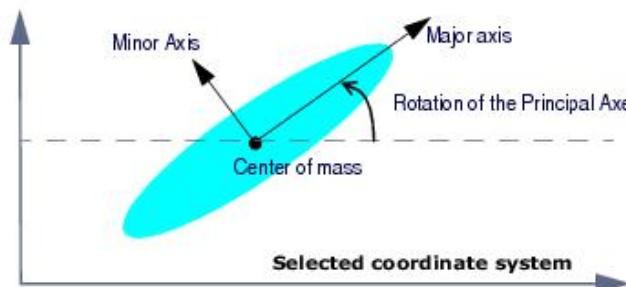
Checking the Perimeter Results box in the Data Collection property enables the following results:

---

Object	Definition
Roundness	The degree of similarity between the blob and a circle. Values are between 0 and 1, where 1 is a perfectly circular blob.
Convex Perimeter	Calculated from the average projected diameter of the blob and is more stable and accurate than the raw perimeter for convex shapes, including rectangular forms.  Convex perimeter is calculated using an approximation of Crofton's theorem. The blob's diameter is determined by projections made at four different orientations: 0, 45, 90 and 135 degrees. The average diameter calculated from these projections is multiplied by PI to obtain the convex perimeter.
Raw Perimeter	The sum of the pixel edge lengths on the contour of the blob. Because the raw perimeter is sensitive to the orientation of the blob with respect to the pixel grid, results may vary greatly. Unless blobs are non-convex, convex perimeter results provide greater accuracy.

### Intrinsic Inertia Results

The intrinsic moments of inertia measure the inertial resistance of the blob to rotation about its principal axes. Because their orientation depends on the coordinate system in which the blob is represented, the principal axes, major and minor, are defined in the section on extrinsic blob properties.



### Intrinsic Inertia Results

Object	Definition
Elongation	<p>The degree of dispersion of all pixels belonging to the blob around its center of mass. The elongation of the blob is calculated as the square root of the ratio of the moment of inertia, about the minor axis, to the moment of inertia about the major axis.</p> $\text{Elongation} = \sqrt{\frac{\text{Inertia Maximum}}{\text{Inertia Minimum}}}$

The following two results require the Calculate Blob Angle property to be true.

Object	Definition
Inertia Minimum	Moment of inertia about the major axis, which corresponds to the lowest moment of inertia.
Inertia Maximum	Moment of inertia about the minor axis, which corresponds to the highest moment of inertia.

### ***Extrinsic Inertia Results***

A moment of inertia of the blob is a measure of the inertial resistance of the blob to rotation about a given axis. Extrinsic moments of inertia measure the moment of inertia about the X-Y axes of the Tool coordinate system.

### ***Extrinsic Moments of Inertia***

A moment of inertia of the blob is a measure of the inertial resistance of the blob to rotation about a given axis. Extrinsic moments of inertia measure the moment of inertia about the x-y axes of the Tool coordinate system.

Object	Definition
Principal Axes	<p>A reference system that is constituted of the major axis and the minor axis. The major axis (X) is the axis about which the moment of inertia is smallest. Conversely, the minor axis (Y) is the axis about which the moment of inertia of the blob is the greatest.</p> <p>The principal axes are orthogonal and are identified by the angle between the X-axis of the region of interest and the major axis of the blob.</p>

Inertia X and Y axes require that the Extrinsic Inertia Results box be checked in the Data Collection property.

Object	Definition
Inertia X-Axis	The moment of inertia about the X-axis of the Tool coordinate system.
Inertia Y-Axis	The moment of inertia about the Y-axis of the Tool coordinate system.  The intrinsic moments of inertia measure the inertial resistance of the blob to rotation about its principal axes. Because their orientation depends on the coordinate system in which the blob is represented, the principal axes, major and minor, are defined in the section on extrinsic blob properties.

### **Rotation of the Principal Axes**

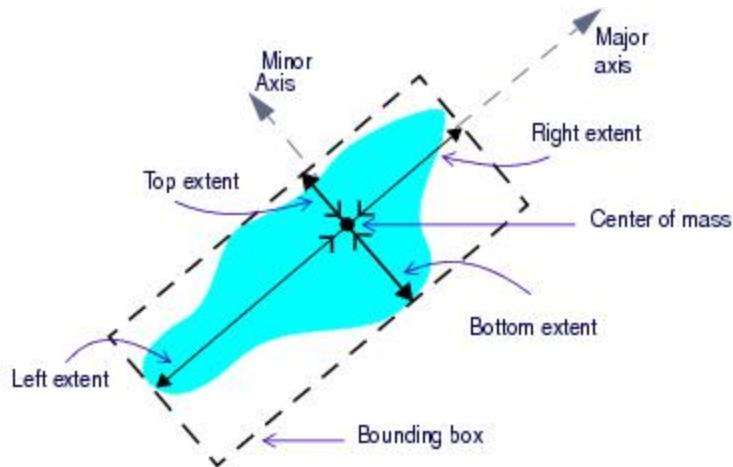
The counterclockwise angle between the X-axis of a selected coordinate system and the major axis.

Object	Definition
Principal Axes Rotation	The angle of axis of the smallest moment of inertia with respect to the X-axis of the selected coordinate system.

### ***Intrinsic Bounding Box Results***

The smallest rectangle enclosing the blob. The width and length of this box define the minor axis and the major axis. Extents measure the distance between a blob's center of mass and the four sides of the bounding box.

---



#### *Intrinsic Bounding Box and Extents*

### **Intrinsic Bounding Box Properties**

Checking the Intrinsic Box Results box in the Data Collection property enables the following results:

Object	Definition
Intrinsic Bounding Box Left	The leftmost coordinate of the bounding box aligned with respect to the X-axis (major axis) of the principal axes.
Intrinsic Bounding Box Right	The rightmost coordinate of the bounding box aligned with respect to the X-axis (major axis) of the principal axes.
Intrinsic Bounding Box Top	The topmost coordinate of the bounding box aligned with respect to the Y-axis (minor axis) of the principal axes.
Intrinsic Bounding Box Bottom	The bottommost coordinate of the bounding box aligned with respect to the Y-axis (minor axis) of the principal axes.
Intrinsic Bounding Box Rotation	The rotation of the intrinsic bounding box corresponds to the counter-clockwise angle between the X-axis of the bounding box (major axis) and the X-axis of the selected coordinate system.

Checking the Extrinsic Inertial Results box in the Data Collection property enables the following results:

Object	Definition
Intrinsic Bounding Box Center X	X-coordinate of the center of the bounding box with respect to the X-axis (major axis) of the principal axes.
Intrinsic Bounding Box Center Y	Y-coordinate of the center of the bounding box with respect to the Y-axis (minor axis) of the principal axes.
Intrinsic Bounding Box Height	Height of the bounding box with respect to the Y-axis (minor axis) of the principal axes.
Intrinsic Bounding Box Width	Width of the bounding box with respect to the X-axis (major axis) of the principal axes.

### **Intrinsic Bounding Box Extents**

Intrinsic extents are the distances between a blob's center of mass and the four sides of the intrinsic bounding box.

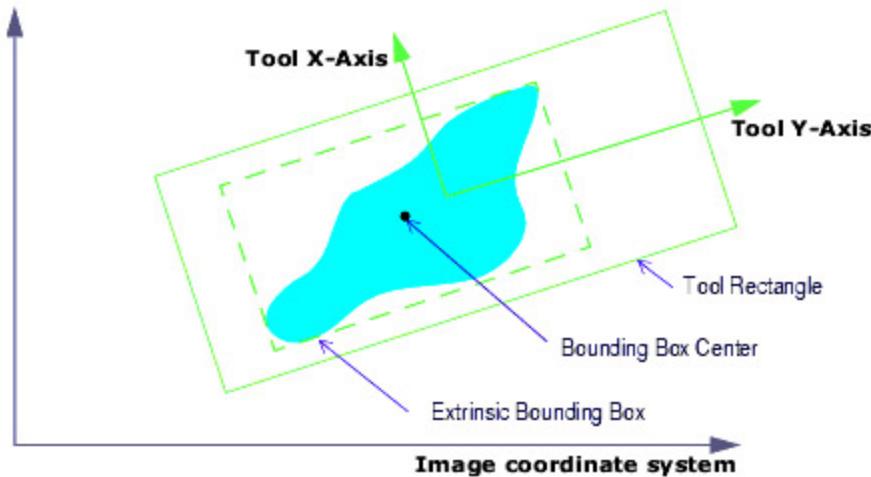
Checking the Intrinsic Box Results box in the Data Collection property enables the following results:

Object	Definition
Intrinsic Extent Top	Distance along the minor axis between the blob's center of mass and the top side of the intrinsic bounding box.
Intrinsic Extent Bottom	Distance along the minor axis between the blob's center of mass and the bottom side of the intrinsic bounding box.
Intrinsic Extent Left	Distance along the major axis between the blob's center of mass and the left side of the intrinsic bounding box.
Intrinsic Extent Right	Distance along the major axis between the blob's center of mass and the right side of the intrinsic bounding box.
Inertia X-Axis	The moment of inertia about the X-axis of the Tool coordinate system.
Inertia Y-Axis	The moment of inertia about the Y-axis of the Tool coordinate system.

Object	Definition
Principal Axes Rotation	The angle of axis of the smallest moment of inertia with respect to the X-axis of the principal axes.

### ***Extrinsic Bounding Box Results***

A bounding box that defines the smallest rectangle, aligned with the Tool coordinate system, that can enclose the blob.



*Extrinsic Bounding Box*

### ***Extrinsic Bounding Box Properties***

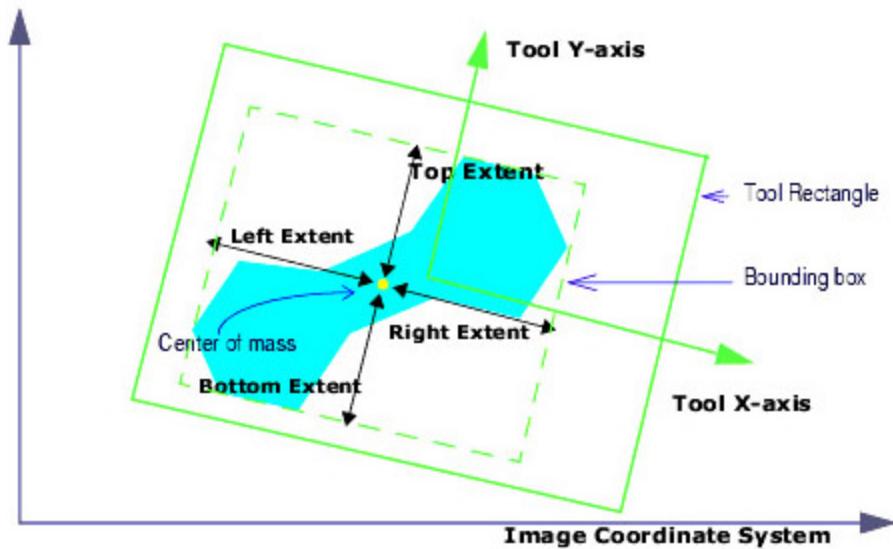
All of the Extrinsic Bounding Box properties require the Calculate Blob Angle property to be true.

Object	Definition
Bounding Box Left	Leftmost coordinate of the bounding box (minimum X value) aligned with respect to the Tool coordinate system.
Bounding Box Right	Rightmost coordinate of the bounding box (maximum X value) aligned with respect to the Tool coordinate system.
Bounding Box Top	Topmost coordinate of the bounding box (maximum Y value) aligned with respect to the Tool coordinate system.

Object	Definition
Bounding Box Bottom	Bottommost coordinate of the bounding box (minimum Y value) aligned with respect to the Tool coordinate system.
Bounding Box Center X	X-coordinate of the center of the bounding box aligned with the Tool coordinate system. This value is returned with respect to the selected coordinate system.
Bounding Box Center Y	Y-coordinate of the center of the bounding box aligned with the Tool coordinate system. This value is returned with respect to the selected coordinate system.
Bounding Box Height	Height of the bounding box with respect to the Y-axis of the Tool coordinate system.
Bounding Box Width	Width of the bounding box with respect to the X-axis of the Tool coordinate system.
Bounding Box Rotation	Rotation of the bounding box with respect to the X-axis of the selected coordinate system.

### Extrinsic Bounding Box Extents

Extents of a blob are the distances between the center of mass and the four sides of the extrinsic bounding box. The following figure illustrates a set of extrinsic extents.



*Extrinsic Extents*

Checking the Intrinsic Box Results box in the Data Collection property enables the following results:

Object	Definition
Extent Left	Distance along the X-axis between the blob's center of mass and the left side of the bounding box.
Extent Right	Distance along the X-axis between the blob's center of mass and the right side of the bounding box.
Extent Top	Distance along the Y-axis between the blob's center of mass and the top side of the bounding box.
Extent Bottom	Distance along the Y-axis between the blob's center of mass and the bottom side of the bounding box.

**Grey-Level Results**

In all cases, grey-level properties apply to pixels included in the blob regardless of weight values attributed by soft thresholding.

Checking the Grey-Level Results box in the Data Collection property enables the following results:

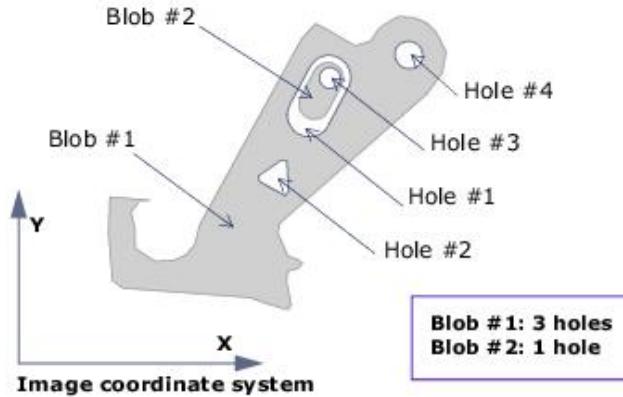
Object	Definition
Mean Grey-Level	The average grey-level of the pixels belonging to the blob.
Minimum Grey-Level	The lowest grey-level pixel found in the blob.
Maximum Grey-Level	The highest grey-level pixel found in the blob.
Grey-Level Range	The difference between the highest and the lowest grey-level found in the blob.
Standard Deviation Grey-Level	The standard deviation of grey-levels for the pixels belonging to the blob.

**Topological Results****Hole Count**

Checking the Topological Results box in the Data Collection property enables the Hole Count results.

---

The Hole Count property returns the number of holes found in each blob. The holes in smaller blobs that are contained within a larger blob are not included in the Hole Count. In other words, the Hole Count does not take into account the hierarchical relationship between blobs. The following figure illustrates such a case, where Blob#1 returns a Hole Count of three, not four.



### *Topological Results*

#### **Related Topics**

[Blob Analyzer](#)

[Configuring Blob Analyzer Properties](#)

[Configuring Blob Analyzer Properties - Advanced](#)

#### **Calculated Arc**

Calculated Arc calculates the circle enclosing an arc based on a specific calculation mode. Possible modes are:

- Three points on the arc
- Center point and one point on the arc

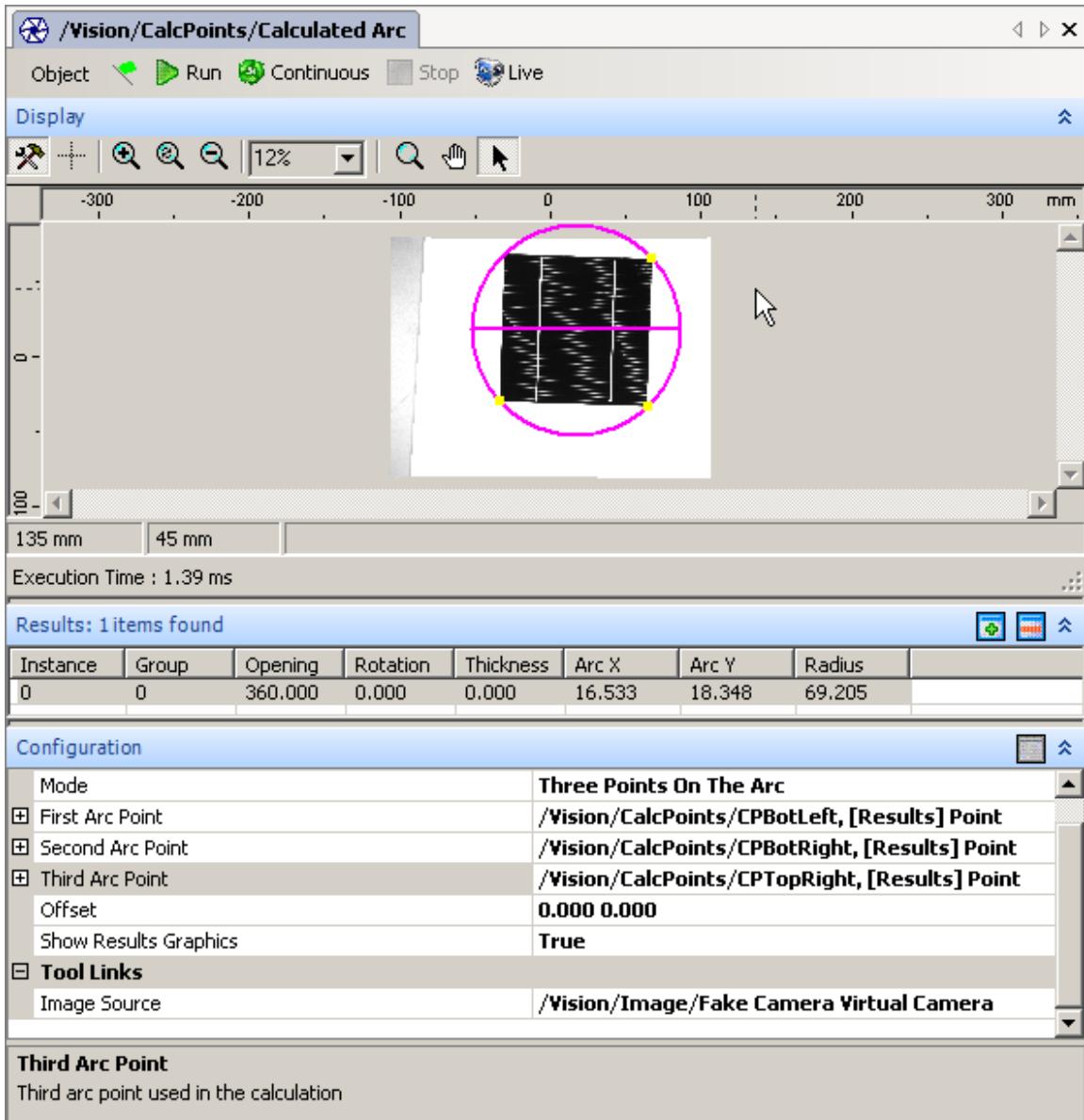
To create a Calculated Arc tool, right-click in the Tree structure, then select:

**New > Vision > Tool > Calculated Arc**

The following example shows a Calculated Arc, generated from three points.

## Calculated Arc

---



*Calculated Arc Object Editor*

### **Configuration**

Object	Definition
<b>Properties</b>	

---

Object	Definition
Mode	Mode of tool execution: 3 points or center and one point.
Center Arc Point	Center arc point used in the calculation.
First Arc Point	First arc point used in the calculation.
Second Arc Point	Second arc point used in the calculation.
Third Arc Point	Third arc point used in the calculation.
Offset	X and Y offset to apply to the calculated results.
Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
<b>Tool Links</b>	
Image Source	Defines the image source used for processing by this vision tool.
<b>Advanced Properties</b>	
Results Logging	Specifies if a results log will be generated. A csv extension generates csv format.
Only Compare Related Results	Use only related results for the comparison. Otherwise all results are used.

### **Results**

Object	Definition
Instance	The index of the located instance
Frame/Group	The frame or group to which the result belongs.
Arc X	The X coordinate of the center of the located instance
Arc Y	The Y coordinate of the center of the located instance
Radius	The radius of the instance
Opening	The opening of the instance (always 360)
Rotation	The rotation of the instance (always 0)
Thickness	The thickness of the instance (always 0)

---

## Calculated Frame

Calculated Frames are used to create a vision frame from other features. Frames allow you to place vision tools on objects that are not always in the same location or orientation. When you create a vision tool, you can specify that it be relative to a vision frame. If the object that defines this vision frame moves, so will the vision frame and the tools that are relative to that frame. You can also create a fixed frame using this tool.

Calculate a frame based on a specific calculation mode. Possible modes are:

- Two lines (X axis line, Y axis line)
- Two points (Origin, +X point)
- A fixed frame of reference
- Single point (Origin point with no rotation)
- Relative to a frame
- One point and a line (An origin point following the angle of the line)

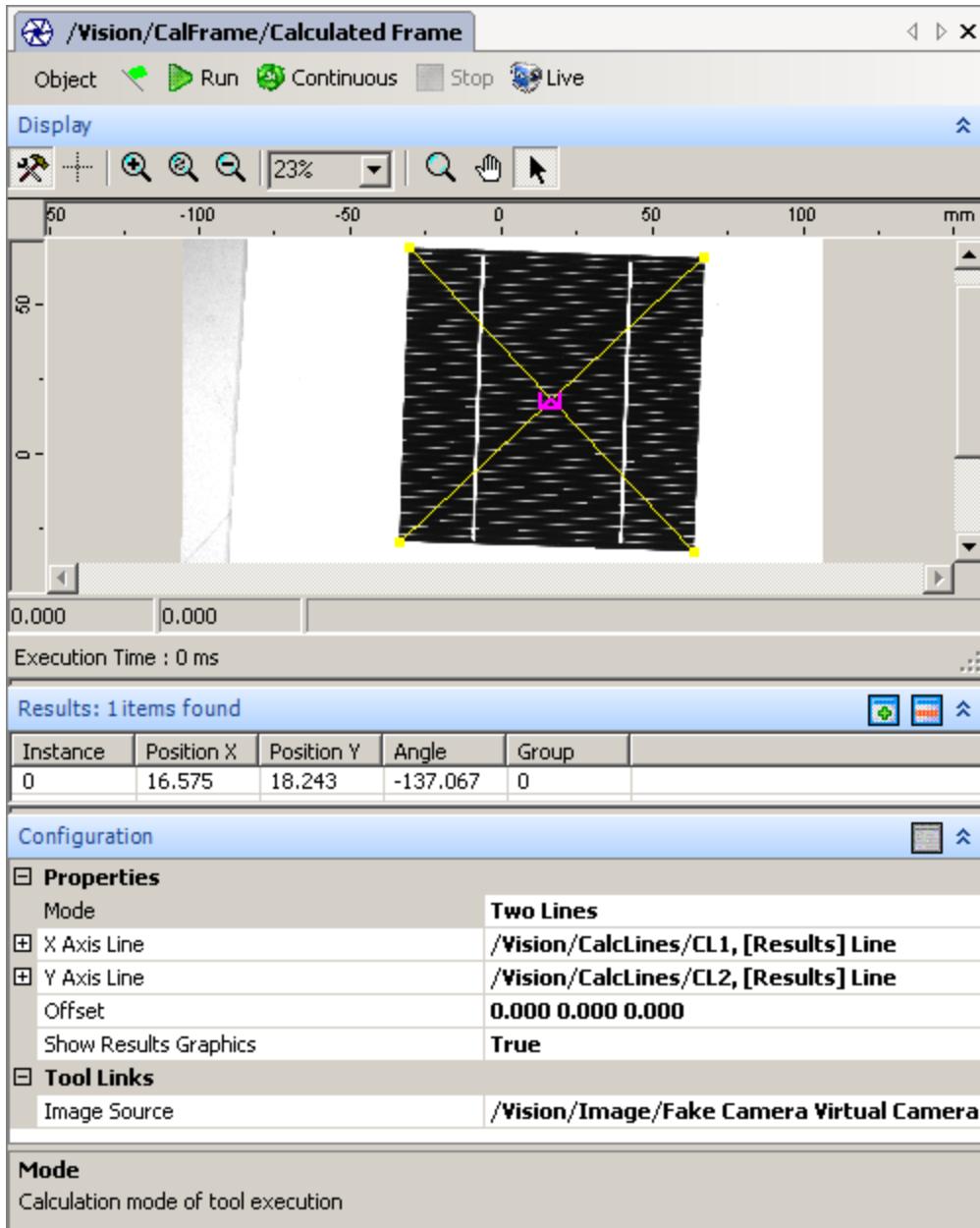
To create a Calculated Frame tool, right-click in the Tree structure, then select:

**New > Vision > Tool > Calculated Frame**

The following example calculates a frame from two lines.

## Calculated Frame

---



Calculated Frame Object Editor

### Configuration

Object	Definition
<b>Properties</b>	

---

Object	Definition
Mode	Mode of tool execution.
Origin Point	Origin point to use in the frame calculation.
Positive X Point	Positive X point to use in the frame calculation.
Offset	X and Y offset to apply to the calculated results.
Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
<b>Tool Links</b>	
Image Source	Defines the image source used for processing by this vision tool.
<b>Advanced Properties</b>	
Results Logging	Specifies if a results log will be generated. A csv extension generates csv format.
Only Compare Related Results	Use only related results for the comparison. Otherwise all results are used.

### **Results**

Object	Definition
Instance	The index of the located instance
Frame/Group	The frame or group to which the result belongs.
Position X	The X coordinate of the located instance
Position Y	The Y coordinate of the located instance
Angle	The angle of the located instance (for two lines, one point and line modes only)

### **Calculated Line**

You can use this tool to create lines from the results of other tools, or to place reference (fixed) lines in the field of view.

A Calculated Line can be created from two points, or from a point and a line. In the latter case, the calculated line will run through the point, and be perpendicular to the line.

Calculate Line generates a line based on a specific calculation mode. Possible modes are:

- Two points
- Perpendicular line, given one line and one point

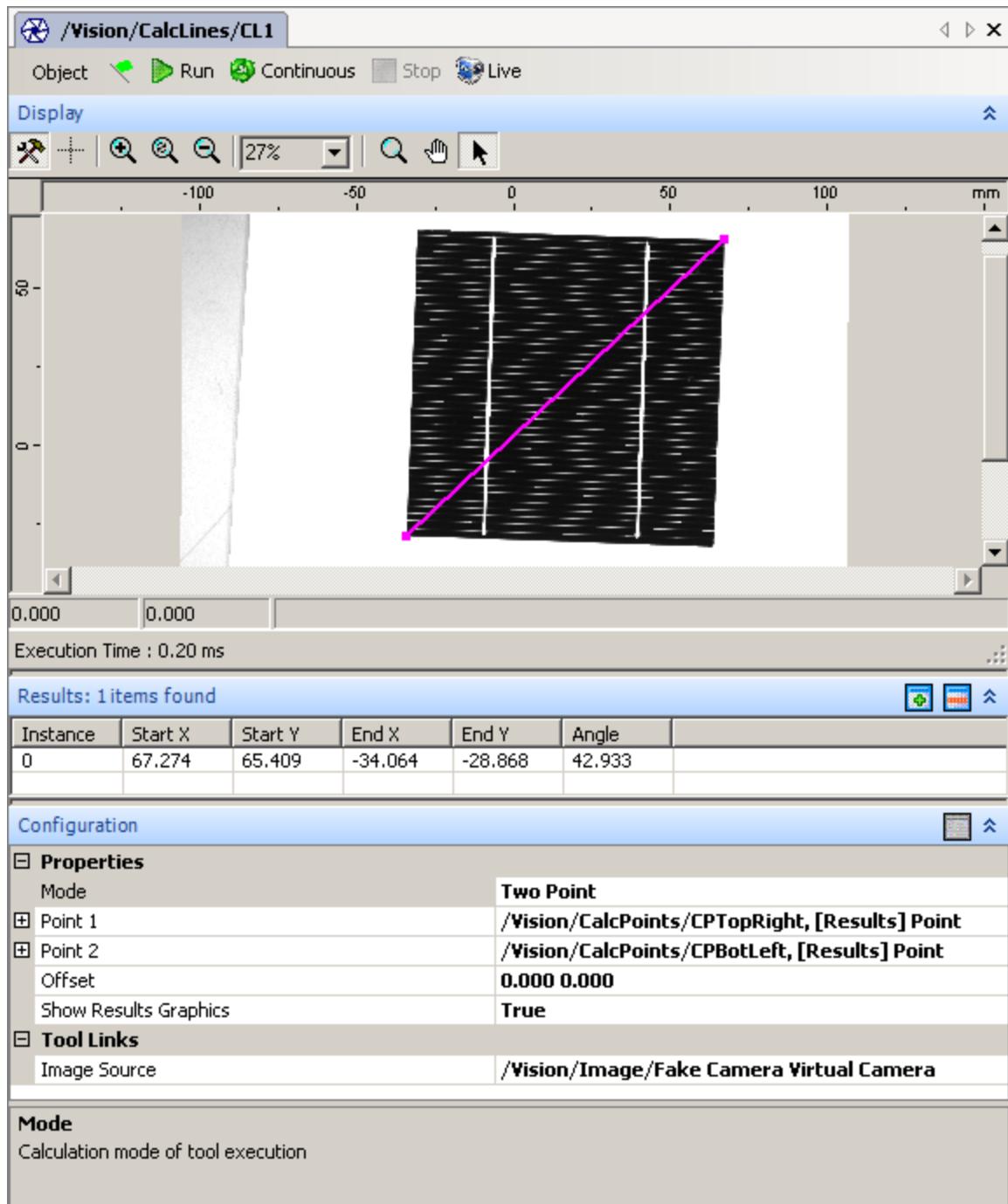
To create a Calculated Line tool, right-click in the Tree structure, then select:

**New > Vision > Tool > Calculated Line**

The following example calculates a line from two points.

## Calculated Line

---



*Calculated Line Object Editor*

---

***Configuration***

<b>Object</b>	<b>Definition</b>
<b>Properties</b>	
Mode	Mode of tool execution: 2 points or perpendicular line.
Point 1	First point used in the calculation.
Point 2	Second point used in the calculation.
Offset	X and Y offset to apply to the calculated results.
Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
<b>Tool Links</b>	
Image Source	Defines the image source used for processing by this vision tool.
<b>Advanced Properties</b>	
Results Logging	Specifies if a results log will be generated. A csv extension generates csv format.
Only Compare Related Results	Use only related results for the comparison. Otherwise all results are used.

***Results***

<b>Object</b>	<b>Definition</b>
Instance	The index of the located instance
Frame/Group	The frame or group to which the result belongs.
Start X	The x coordinate of the start point
Start Y	The y coordinate of the start point
End X	The x coordinate of the end point
End Y	The y coordinate of the end point
Center X	The x coordinate of the center point
Center Y	The y coordinate of the center point
Angle	The angle of the line

## Calculated Point

Often, the most accurate way to calculate a point is to calculate it from other features. For example, Calculated Points can be calculated based on the intersection of two lines, a line and a circle, or midway between two points.

A Calculated Point tool is also used to place a fixed point in the field of view. A fixed point could be used if you want to make all your measurements from a known reference point.

The Calculated Point tool calculates a point based on the calculation mode. Possible modes are:

- Midpoint (midpoint between two points)
- Point and a line (closest point on the line to another point)
- Point and an arc (closest point on the arc to another point)
- Fixed point
- A line and an arc (Line/Arc intersection)
- Two lines (Line-line intersection)
- Two arcs (Arc-arc intersection)

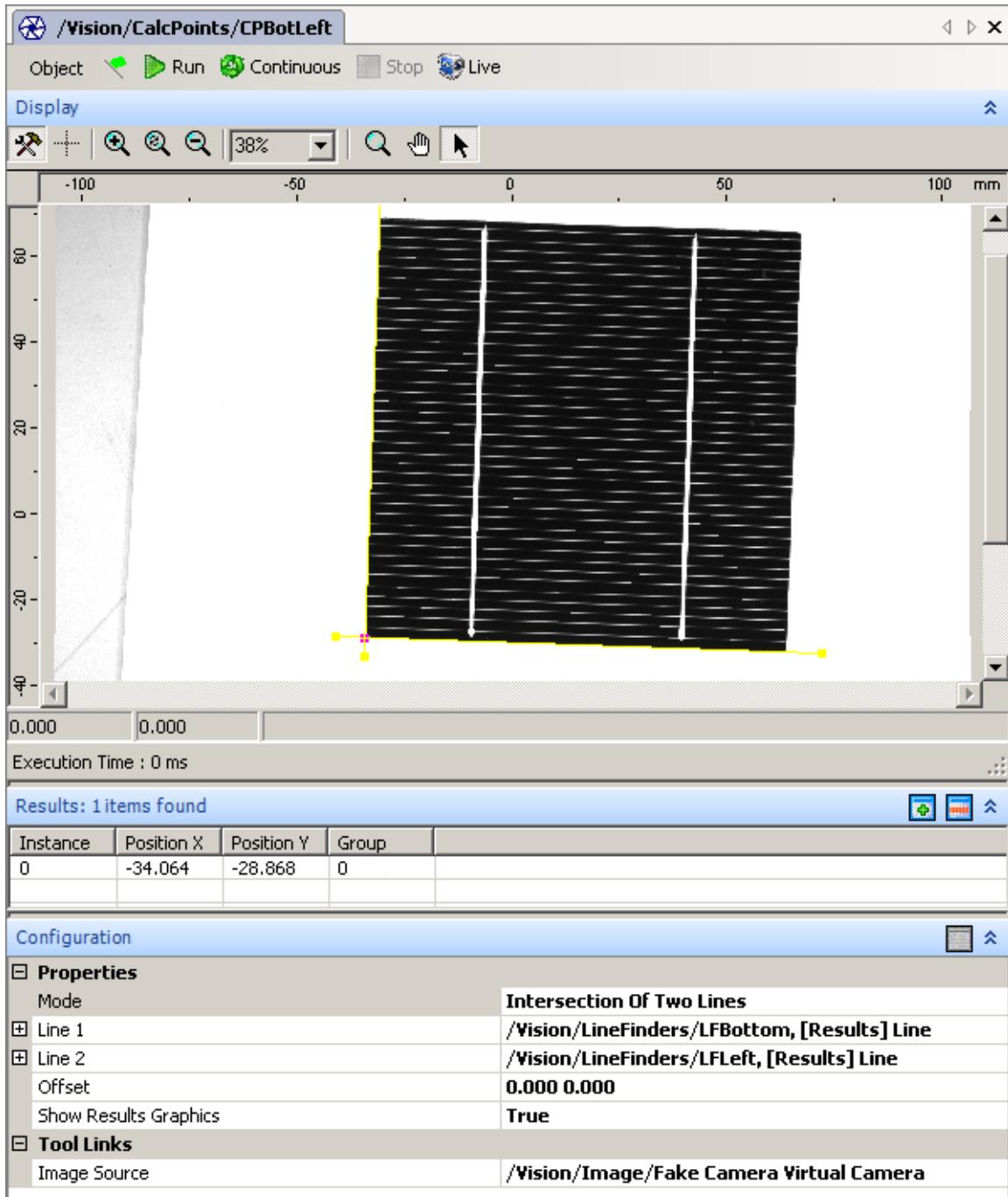
To create a Calculated Point tool, right-click in the Tree structure, then select:

**New > Vision > Tool > Calculated Point**

The following example calculates the bottom-left corner of the wafer, given the bottom and left edges.

## Calculated Point

---



Calculated Point Object Editor

---

***Configuration***

<b>Object</b>	<b>Definition</b>
<b>Properties</b>	
Mode	Mode of tool execution.
Point 1	First point used in the calculation.
Point 2	Second point used in the calculation.
First Arc	First arc used in the calculation.
Second Arc	Second arc used in the calculation.
Line 1	First line used in the calculation.
Line 2	Second line used in the calculation.
Offset	X and Y offset to apply to the calculated results.
Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
<b>Tool Links</b>	
Image Source	Defines the image source used for processing by this vision tool.
<b>Advanced Properties</b>	
Results Logging	Specifies if a results log will be generated. A csv extension generates csv format.
Only Compare Related Results	Use only related results for the comparison. Otherwise all results are used.

***Results***

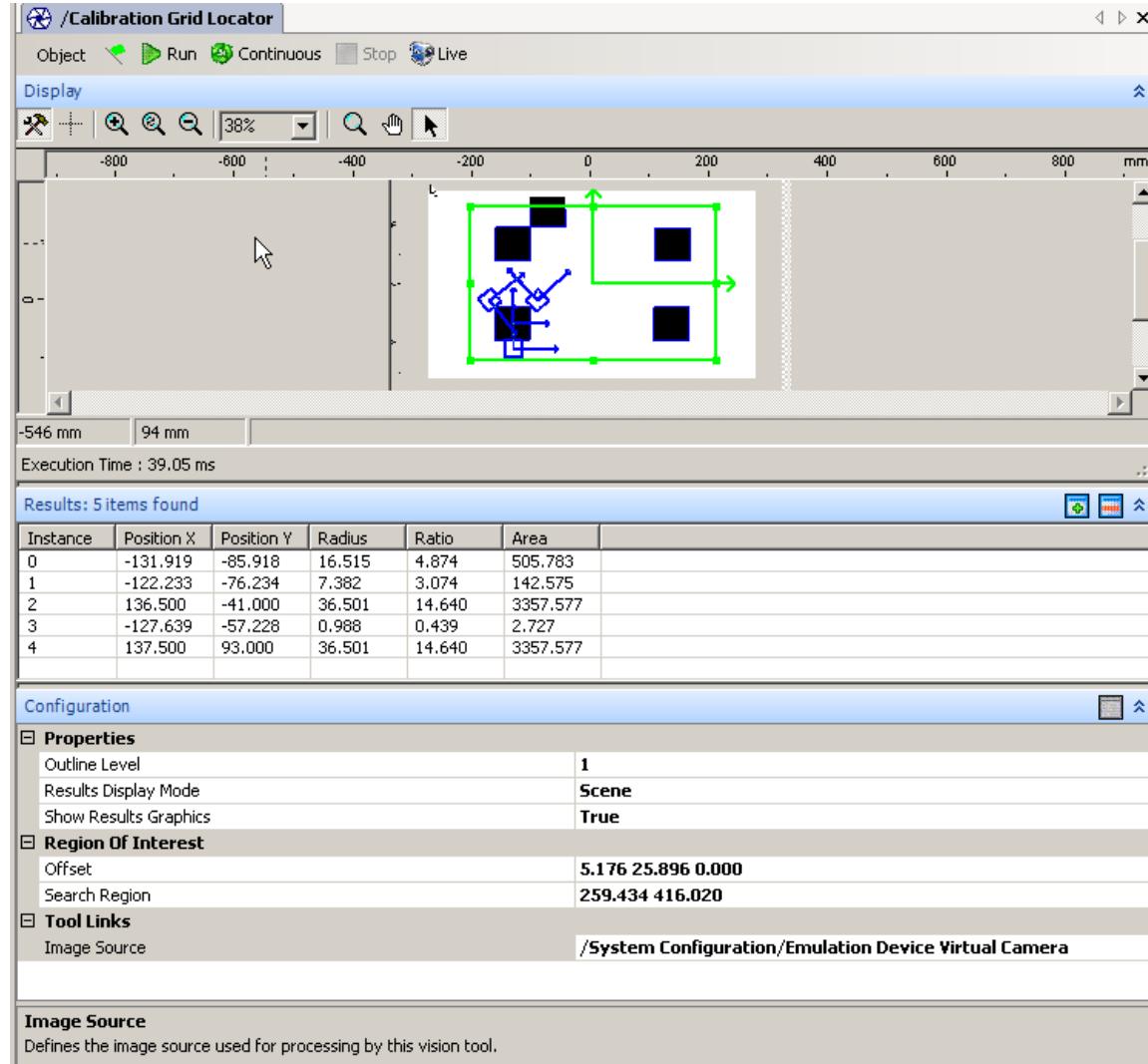
<b>Object</b>	<b>Definition</b>
Instance	The index of the located instance
Frame/Group	The frame or group to which the result belongs.
Point X	The X coordinate of the point
Point Y	The Y coordinate of the point

## Calibration Grid Locator

The Calibration Grid Locator tool is used to locate a collection of dots in the field of view. It is used by the grid calibration procedure.

To create a Calibration Grid Locator tool, right-click in the Tree structure, then select:

**New > Vision > Tool > Calibration Grid Locator**



Calibration Grid Locator Object Editor

***Configuration***

<b>Object</b>	<b>Definition</b>
<b>Properties</b>	
Outline Level	The coarseness of the contours used to build the model at the Outline level.
Results Display Mode	When results are displayed, this defines how the results are rendered in the image display. Marker, Scene, or Marker and Scene.
Search Region	Defines the height and width of the region of interest.
Offset	X and Y offset to apply to the calculated results
Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
<b>Tool Links</b>	
Image Source	Defines the image source used for processing by this vision tool.
<b>Advanced</b>	
Contrast Threshold	Defines the minimum contrast needed for an edge to be detected in the input image and used for arc computation. Expressed in terms of a step in grey-level values. Adaptive Low, Normal, High Sensitivity, or Fixed Value.
Results Logging	Specifies if a results log will be generated. A csv extension generates csv format.

***Results***

<b>Object</b>	<b>Definition</b>
Instance	The index of the located dot, starting at 0
Position X	The X coordinate of the center point of the located dot
Position Y	The Y coordinate of the center point of the located dot
Radius	The radius of the located dot
Ratio	The ratio of height/width of the located dot

## Caliper

---

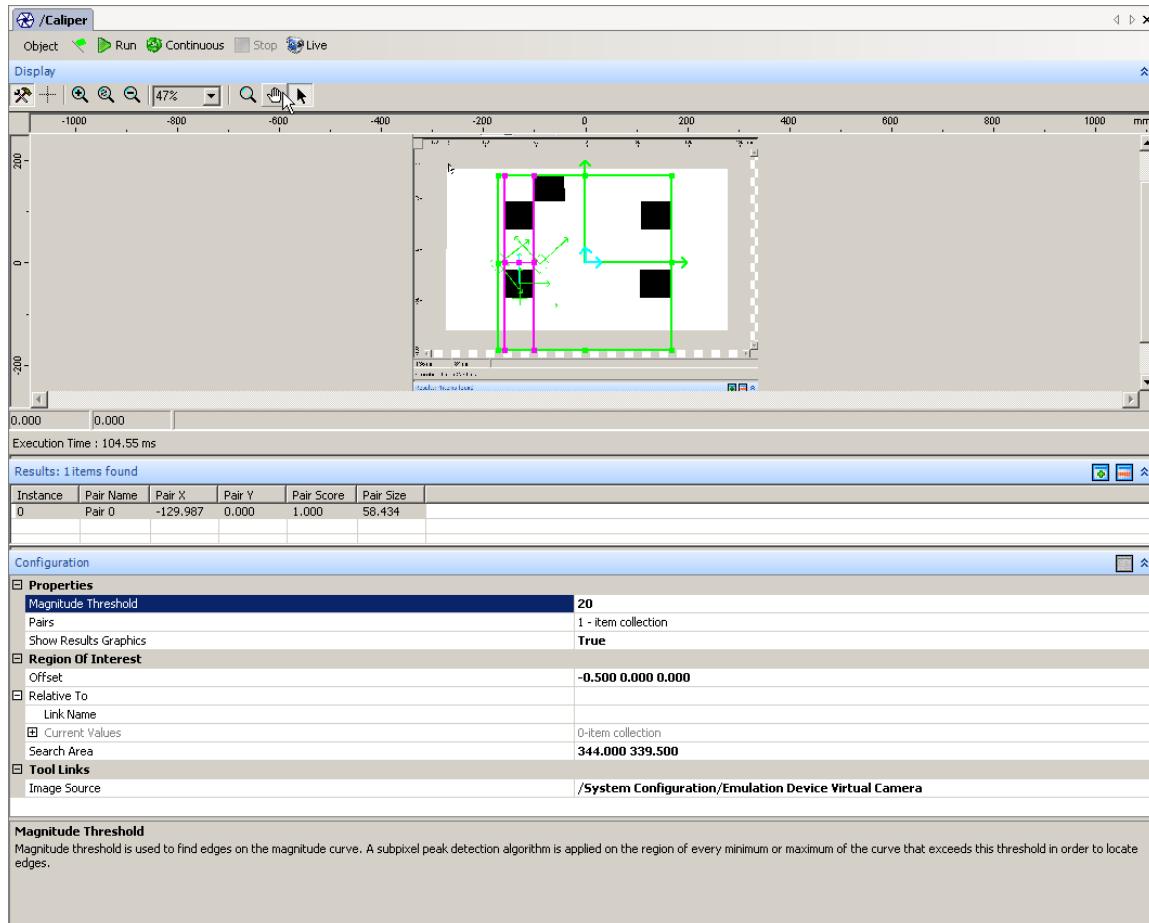
Object	Definition
Area	The area of the located dot

### Caliper

The Caliper tool identifies and measures the gap between one or more edge pairs on an object. The Caliper uses pixel grey-level values within the region of interest to build projections needed for edge detection. After the Caliper detects potential edges, it determines which edge pairs are valid by applying the constraints that are configured for each edge pair. Finally, the Caliper scores and measures each valid edge pair.

To create a Caliper tool, right-click in the Tree structure, then select:

**New > Vision > Tool > Caliper**



*Caliper Object Editor*

***Configuration***

<b>Object</b>	<b>Definition</b>
<b>Properties</b>	
Magnitude Threshold	The Magnitude threshold sets the acceptable magnitude value for potential edges.
Pairs	Collection of transition criteria to search for. Opens the Edge Pair Collection Editor.
Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
<b>Tool Links</b>	
Image Source	Defines the image source used for processing by this vision tool.
<b>Region of Interest (ROI)</b>	
Offset	Specifies the center and rotation of the ROI. X, Y, Degrees.
Relative To	Specifies the tool that will provide the input to this tool. The tool can be dragged from the Folder pane and dropped into this field.
Search Area	Defines the height and width of the region of interest.
<b>Advanced Properties</b>	
Bilinear Interpolation Enabled	Specifies if bilinear interpolation is used to sample the input image. By default, bilinear interpolation is enabled because it improves accuracy.
Custom Sampling Step	When enabled, the tool uses the user-defined sampling step instead of the default optimal sampling step to sample the region of interest from the input image.
Edge Filter Half Width	Width of the edge, in pixels, as it appears in the image.

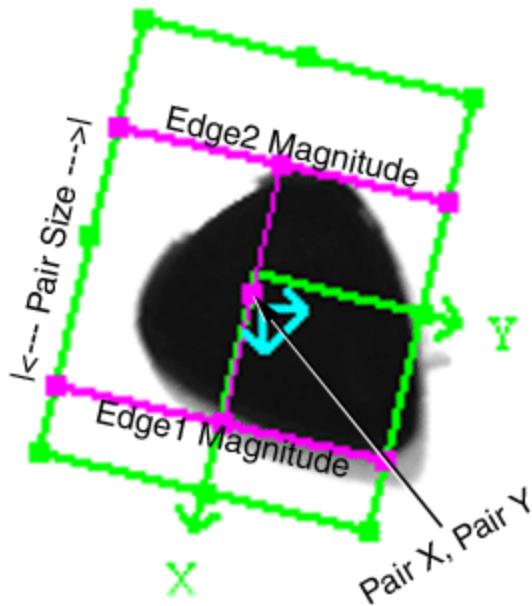
***Results***

<b>Object</b>	<b>Definition</b>
Instance	The index of the located instance

Object	Definition
Frame/Group	The frame or group to which the result belongs.
Pair Name	Name of the edge pair (specified in the Pairs property)
Pair Score	Score of the selected pair. The score of the pair is equal to the mean score of the two edges (Edge 1 Score and Edge 2 Score) that comprise the pair.
Pair Size	The Caliper measure, which is the calculated distance between the pair of edges
Pair X	The X coordinate of the center point of the caliper measure, at the midpoint of the edge pair
Pair Y	The Y coordinate of the center point of the caliper measure, at the midpoint of the edge pair
Edge 1 X	The X coordinate of the first end of the line between the pair edges
Edge 1 Y	The Y coordinate of the first end of the line between the pair edges
Edge 2 X	The X coordinate of the second end of the line between the pair edges
Edge 2 Y	The Y coordinate of the second end of the line between the pair edges
Edge 1 Magnitude	Magnitude of the first edge of the selected pair
Edge 2 Magnitude	Magnitude of the second edge of the selected pair
Edge 1 Magnitude Score	Magnitude score of the first edge of the selected pair
Edge 2 Magnitude Score	Magnitude score of the second edge of the selected pair
Edge 1 Position Score	Position score of the first edge of the selected pair
Edge 2 Position Score	Position score of the second edge of the selected pair
Edge 1 Score	Minimum score to accept an edge as the first edge of the selected pair. The score is computed according to the constraints set in the Score Threshold property.

Object	Definition
Edge 2 Score	<p>Minimum score to accept an edge as the second edge of the selected pair.</p> <p>The score is computed according to the constraints set in the Score Threshold property.</p>

Edges returned as results:



*Returned Edges*

**NOTE:** Before configuring the Caliper, execute the tool at least once and verify that the tool is being positioned correctly in the image.

The display represents the Caliper as a green rectangle, with found edges and caliper measure represented in magenta.

#### **Related Topics**

[Configuring Caliper Properties](#)

[Configuring Caliper Properties - Advanced](#)

## Configuring Caliper Properties

### Edge Properties

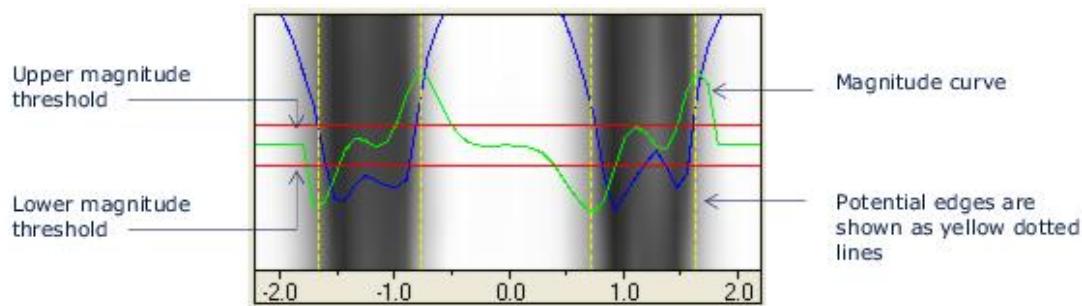
Edge Magnitude expresses the strength of a potential edge.

#### Magnitude Threshold

Sets the acceptable magnitude value for potential edges. This value is expressed as an absolute value. There are two magnitude lines: an upper (positive) threshold and lower (negative) threshold.

Possible values are from 0 - 255.

Edge Magnitude expresses the strength of a potential edge. The (green) magnitude curve, represents magnitude values across the region of interest. Potential edges must have a magnitude greater than the upper threshold, or lower than the lower threshold. See the following figure.



*Interpreting the Magnitude Threshold in the display area*

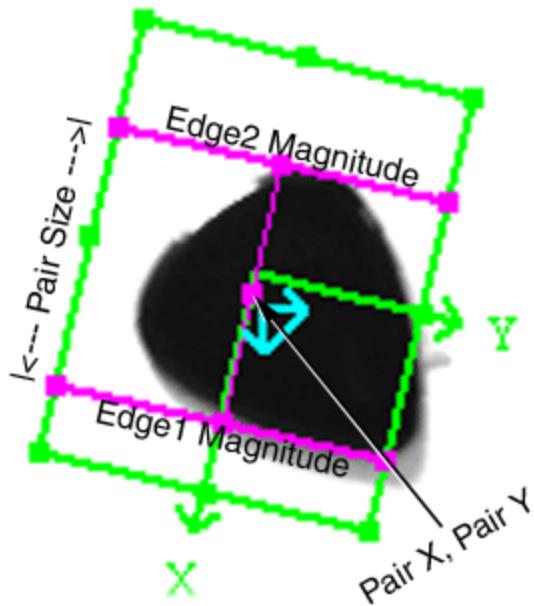
### Configuring Edge Pair Properties

When a caliper is executed, the caliper first applies edge detection constraints to the entire region of interest. Then, the tool applies edge scoring constraints to determine which edges are valid for the caliper measure. If only one valid edge is found, no caliper measure is output.

Pair Settings parameters determine how the tool detects edges and which edge pair are valid.

**NOTE:** Before configuring the caliper, execute the tool at least once and verify that the tool is being positioned correctly in the image.

The display represents the caliper as green, with found edges and caliper measure represented in magenta:



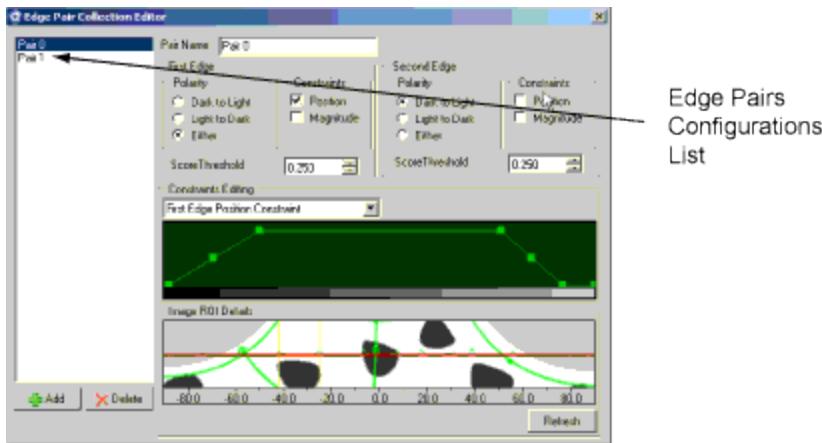
#### Found Edges and Caliper

If the display in the Pair Settings window is blank, or the edges are not properly placed, close the window and verify the following:

- The tool was executed after positioning the tool.
- The tool was executed at least once before opening the Pair Settings window.
- The Location parameters are correct.
- The Y-axis of the tool is parallel to the edges you want to detect.

Constraints are set with the Edge Pair Collection Editor window. From the Object editor window, select

**Configuration > Properties >Pairs >**



### *Edge Pair Collection Editor*

The caliper tool can measure any number of pairs. When the caliper is executed, it first applies edge detection parameters to the entire region of interest. Then, the tool applies pair settings constraints to determine which edges are valid. Results are then calculated for each valid edge pair as well as for individual edges in each edge pair.

### **Pairs Configuration List**

The Pairs list of the Edge Pair Collection Editor contains a list of all the pair configurations for the current caliper tool. This list always contains at least one pair configuration, which is named Pair 0 by default.

Each pair configuration has a name, with polarity and constraints for the first edge and second edge.

From the Edge Pair Collection Editor, you can:

- Access the parameters for each pair configuration
- Add and remove edge pair configurations
- Rename edge pair configurations



#### Pairs List in the Caliper Interface

To access the parameters for an edge pair configuration, click on that pair configuration in the Pairs Configurations List.

To add an edge pair configuration:

1. Under the Pairs Configuration list, click add ().

A pair configuration is added with the default name: "Pair n", where n is the next (unused) integer.

2. Edit the parameters for the new pair configuration.

To remove an edge pair configuration:

1. In the Pairs configuration list, select the pair to be removed.
2. Click delete (.

To rename an edge pair configuration:

1. Click the pair configuration to be renamed in the pairs list.
2. Highlight the Pair Name field for the pair to be renamed.
3. Type a new name for the edge pair.

**NOTE:** This will not affect the parameters of the pair configuration.

To configure edge pair configuration parameters:

1. In the Edge Pair Collection Editor, select a pair configuration from the list. The default name for a first pair is Pair0.
2. The remainder of the window provides parameters for each edge of the caliper edge pair configuration, which are named First Edge and Second Edge.
3. Configure parameters for each edge. Refer to the following sections for help on configuring Pair Settings, and using the display and Function Editor.

## Edge Pair Parameters

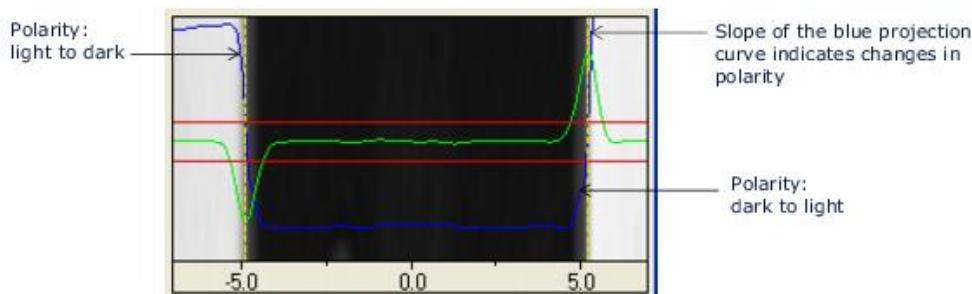
There are two criteria that affect the choice of valid edges: Polarity and Edge Score Constraints.

### Polarity

Polarity corresponds to the change in light values, moving from left to right in the display, along the X-Axis in the region of interest. The caliper applies the Polarity constraint before applying edge-score Constraints.

Polarity does not affect the edge score, but only edges that meet the selected Polarity are retained as valid edges, regardless of their scores.

- Dark to Light will only accept edges occurring at transitions from a dark area to a light area.
- Light to Dark will only accept edges occurring at transitions from a light area to a dark area.
- Either will accept any edge, regardless of its polarity.



*Edge Polarity*

### Edge Score Constraints

There are two types of constraints: Position and Magnitude. You can set the caliper to use only one constraint or both. The graphical Function Editor is provided for viewing and setting each type of constraint.

- If only one constraint is selected, edges are scored based on the selected constraint.
- If both constraints are selected, each constraint accounts for 50% of the edge score.

#### Magnitude Constraint

The Magnitude constraint is based on edge values relative to the Magnitude

Threshold, which is represented in the display by two red lines.

Edges having a magnitude equal to or exceeding the Magnitude Threshold receive a score of 1. Edges with values below the Magnitude Threshold receive a score ranging from 0 to 0.999, according to a manually set magnitude constraint function.

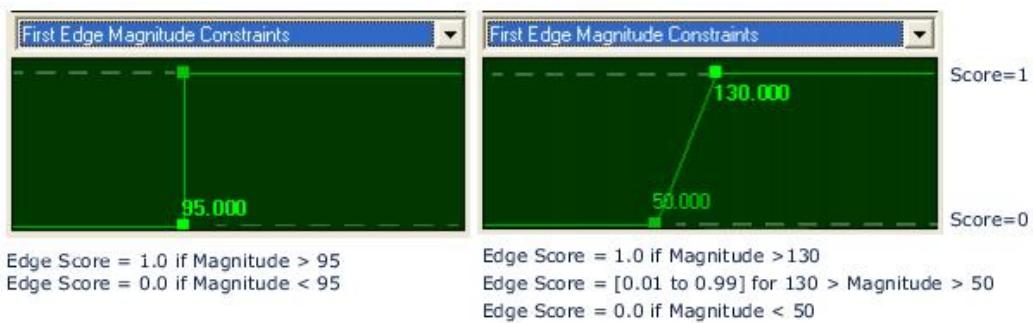
The Magnitude Threshold value can be modified in the Advanced Parameters section of the tool interface.

A Magnitude Constraint must be defined individually for each edge of an edge pair configuration.

The following figure shows examples of two different setups for a magnitude constraint function.

To set a Magnitude Constraint:

1. In the drop-down list above the Function Editor, select First Edge Magnitude Constraint or Second Edge Magnitude Constraint.  
If either edge does not have the Magnitude box checked, it will not be shown in the drop-down list.
  2. In the Function Editor, use the mouse to drag handles and set the magnitude limits. See examples in the following figure.



## *Setting the Magnitude Constraint in the Function Editor*

## Position Constraint

Position constraints restrict the caliper's search for edges to a specific zone of the region of interest.

- It is possible to graphically set a position constraint function when the approximate position of an edge is known beforehand. This is useful for

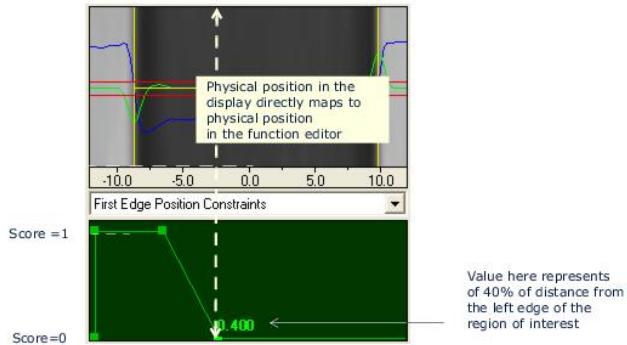
scoring an edge based on its offset from the expected position.

- Values in the Constraint Function Editor indicate relative distance in the region of interest where 0.0 is the leftmost position and 1.0 is the right-most position.

To set a Position Constraint:

1. In the drop-down list above the Function Editor, select First Edge Position Constraint or Second Edge Position Constraint.  
If either edge does not have the Position box checked, it will not be shown in the drop-down list.
2. In the Function Editor, use the mouse to drag handles and set the position limits. See the following figure.

The physical position in the Function Editor corresponds to the same physical position in the display.



#### *Setting the Position Constraint Function Editor*

### Score Threshold

The score threshold sets the minimum acceptable score for a valid edge. The caliper will disregard edges that obtain a score lower than the Score Threshold.

- Caliper constraint scores range from 0 to 1.
- If both Position and Magnitude constraints are enabled, each constraint accounts for 50% of the total edge score.

### Region of Interest (ROI)

The region of interest is the area in which the tool carries out its process. Most tool position parameters can be set through the region of interest section of the tool interface. The following parameters define the tool region of interest.

## Offset

Offset is the center of the ROI, defined by:

X	X coordinate of the center of the region of interest
Y	Y coordinate of the center of the region of interest
Degrees	Angle of rotation of the region of interest

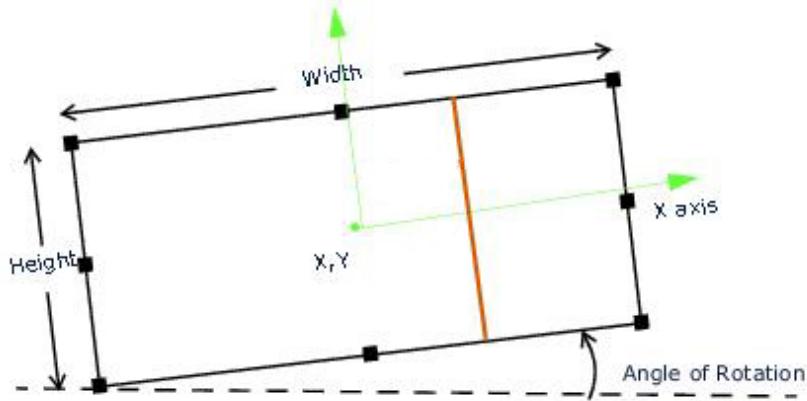
## Relative To

Relative To specifies the tool that will provide the input to this tool. The tool can be dragged from the Tree structure and dropped into this field.

## Search Area

Search Area is the size of the region of interest is defined by:

Height	Height of the region of interest
Width	Width of the region of interest



*Location Properties for the Region of Interest*

## Related Topics

[Caliper Tool](#)

[Configuring Caliper Properties - Advanced](#)

### ***Configuring Caliper Properties - Advanced***

The Advanced Properties section of the Caliper tool interface provides access to advanced Caliper parameters and properties.

#### ***Edge Detection Parameters***

Edge Detection settings configure the parameters that the Caliper will use to find potential edges in the region of interest. The display represents the Caliper region of interest and provides information to assist in configuring Edge Detection parameters.

#### ***Edge Filter Half-Width***

The filtering process attenuates peaks in the magnitude curve that are caused by noise. Edge Filter Half Width should be set to a value approximately equivalent to the width of the edge, in pixels. An incorrect value can cause edges to be incorrectly detected.

#### ***Tool Sampling Parameters***

Sampling refers to the procedure used by the tool for gathering values within the portion of the input image that is bounded by the tool region of interest. Two sampling parameters, the Sampling Step and Bilinear Interpolation, can be used to create the desired balance between speed and accuracy.

#### ***Bilinear Interpolation***

Bilinear Interpolation uses pixel averaging to improve the quality of the image. It is applied before the image is analyzed.

To ensure subpixel accuracy in inspection applications, Bilinear Interpolation should always be set to True (enabled). Non-interpolated sampling (Bilinear Interpolation disabled) should only be used in applications where the speed requirements are more critical than accuracy.

#### ***Custom Sampling Step (1 - 100)***

The Sampling Step is the step used by the tool to sample the area of the input image that is bounded by the tool region of interest. The sampling step represents the height and the width of a sampled pixel.

For applications where a more specific tradeoff between speed and accuracy must be established, the sampling step can be modified by enabling the Custom Sampling Step and setting the desired value.

Object	Definition
Default	Default is the best sampling step computed by the tool. It is based on the average size, in calibrated units, of a pixel in the image. This default sampling step is usually recommended. Default is automatically used by the tool if no other value is specified.
Custom	<p>Custom Sampling Step lets you set a sampling step value other than the default. To select a custom sampling step, Enable must be checked.</p> <ul style="list-style-type: none"> <li>• Increasing the sampling step value reduces the tool accuracy and decreases the execution time.</li> <li>• Reducing the sampling step can increase the tool accuracy but will also increase the execution time.</li> </ul>

## Results Logging

If enabled, the results of the tool will be logged to a file.

If the extension of the destination file is csv, then the log is written in csv format, which opens with Excel. If a different extension is selected, the log is written in a plain text format (similar to the ACE Sight 2 format).

The size of the log file can be set (default size is 5 Mb).

### Related Topics

[Caliper Tool](#)

[Configuring Caliper Properties](#)

## Color Matching

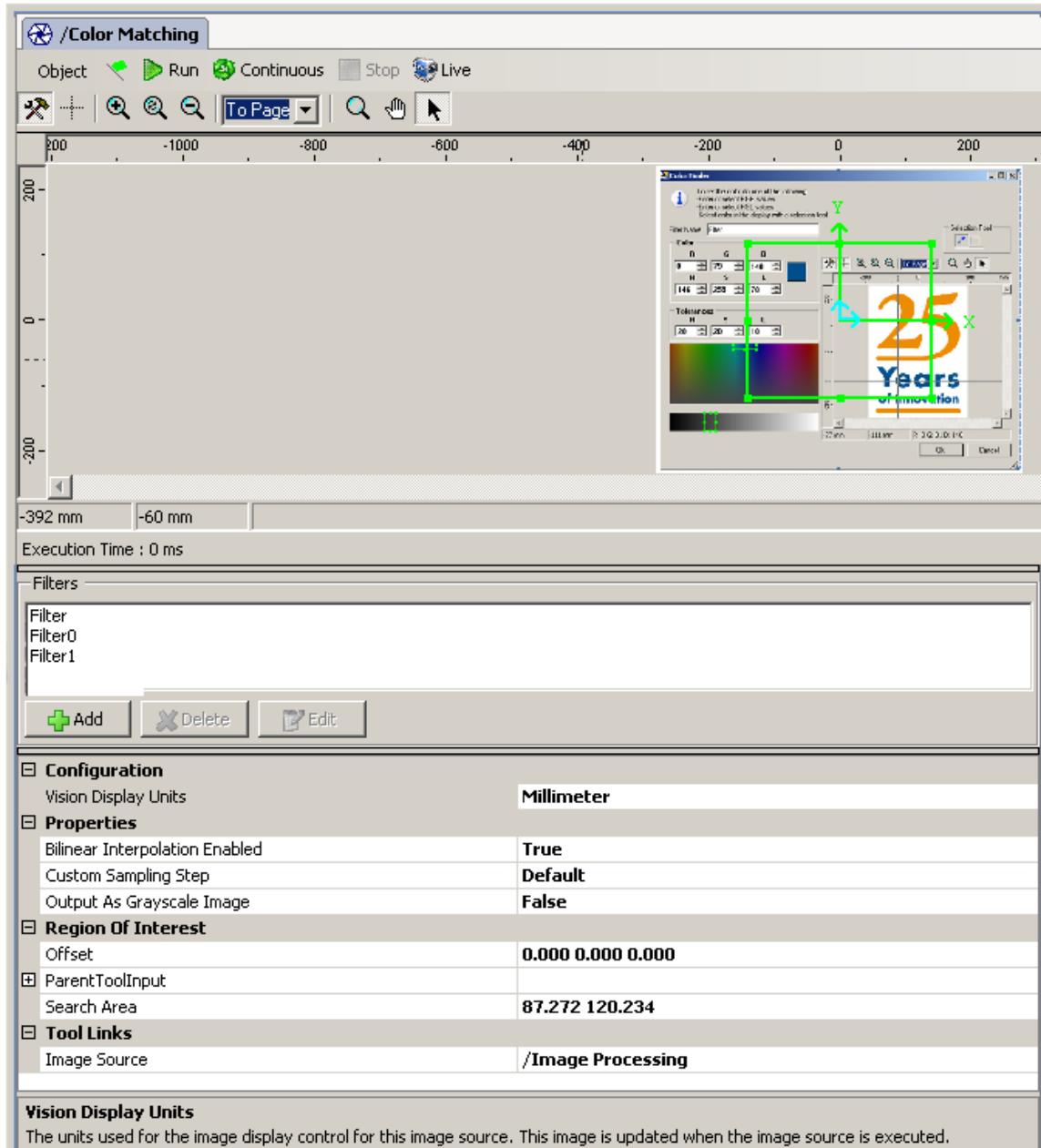
The Color Matching Tool searches and analyzes images to find areas of color that match user-defined filters. Typically, this tool is used to analyze an area on an object for the purpose of verifying if the object meets defined color criteria.

This tool will not be available unless both a vision and a color license are enabled on your license dongle.

To create a Color Matching tool, right-click in the Tree structure, then select:

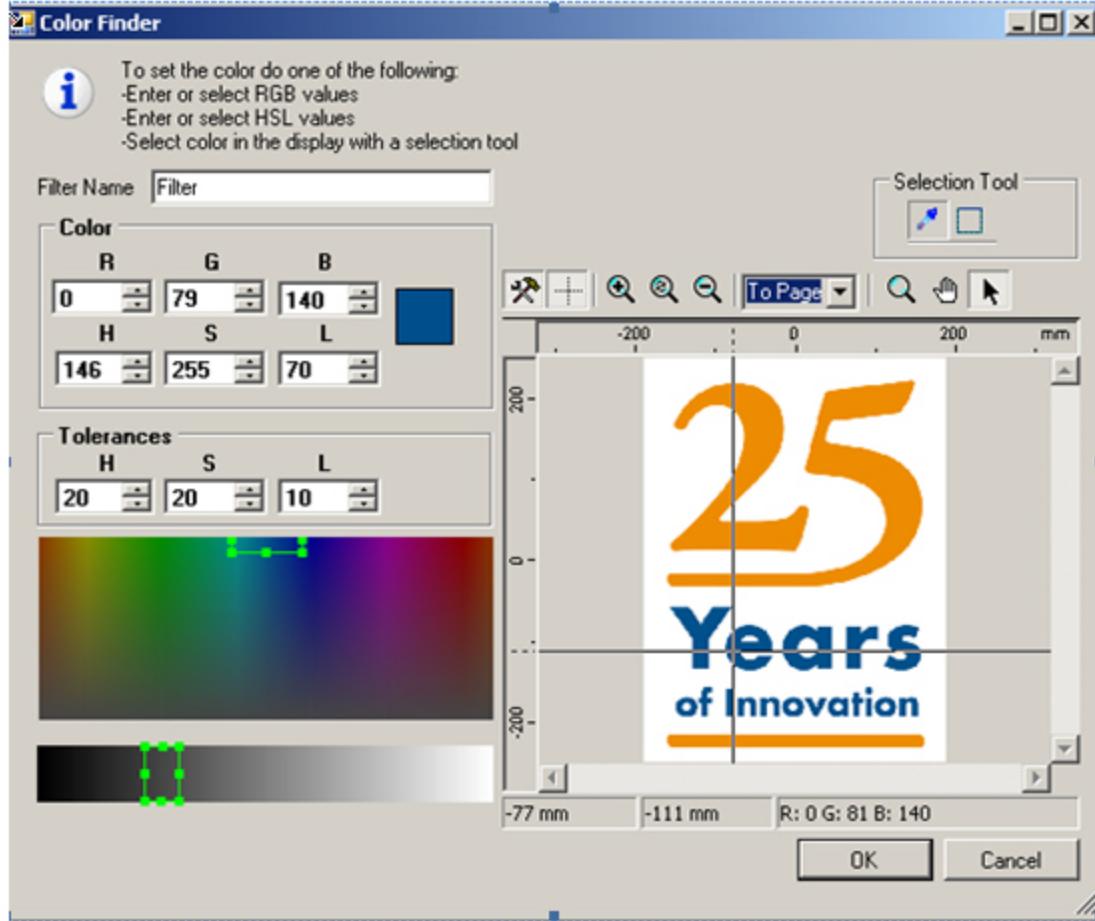
## Color Matching

New > Vision > Tool > Color Matching



Color Matching Object Editor

### ***Creating a Filter***

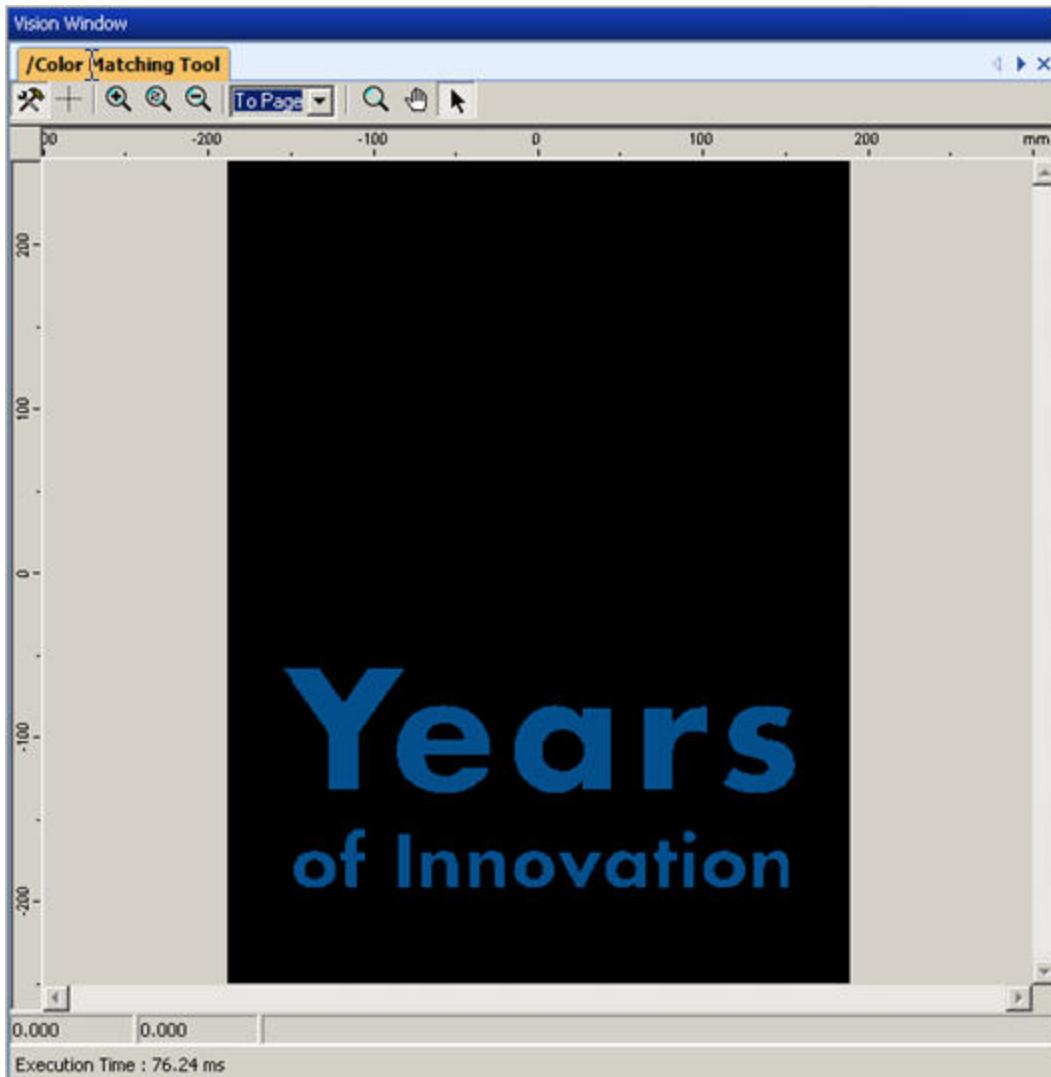


Color Finder Filter

### ***Configuration***

Object	Definition
<b>Configuration</b>	
Vision Display Units	Units for the display for this image source. Default is Millimeter.
<b>Properties</b>	

Object	Definition
Bilinear Interpolation Enabled	Specifies if bilinear interpolation is used to sample the input image. By default, bilinear interpolation is enabled because it improves accuracy.
Custom Sampling Step	When enabled, the tool uses the user-defined sampling step (Custom Sampling Step) instead of the default optimal sampling step to sample the region of interest from the input image.
Output as Grayscale Image	When enabled, converts image to grayscale after color filters are applied.
<b>Region of Interest (ROI)</b>	
Offset	Specifies the center and rotation of the ROI. X, Y, Degrees.
Parent Tool Input	Specifies the tool that will provide the input to this tool. The tool can be dragged from the Folder pane and dropped into this field.
Search Area	Defines the height and width of the region of interest.
<b>Tool Links</b>	
Image Source	Defines the image source used for processing by this vision tool.



*Vision Window Showing Results of the Color Match*

### ***Results***

None.

The Color Matching tool outputs images that can be used by other vision tools.

### ***Related Topics***

[Configuring Color Matching Properties](#)

[Creating and Configuring Color Filters](#)

### ***Configuring Color Matching Properties***

The Properties section of the Color Matching interface provides access to advanced Color Matching Tool parameters and properties.

Sampling refers to the procedure used by the tool for gathering values within the portion of the input image that is bounded by the tool region of interest. Two sampling parameters, the Sampling Step and Bilinear Interpolation, can be used to create the desired balance between speed and accuracy.

#### **Bilinear Interpolation**

Bilinear Interpolation uses pixel averaging to improve the quality of the image. It is applied before the image is analyzed.

To ensure subpixel accuracy in inspection applications, Bilinear Interpolation should always be set to True (enabled). Non-interpolated sampling (Bilinear Interpolation disabled) should only be used in applications where the speed requirements are more critical than accuracy.

#### **Custom Sampling Step (1 - 100)**

The Sampling Step is the step used by the tool to sample the area of the input image that is bounded by the tool region of interest. The sampling step represents the height and the width of a sampled pixel.

For applications where a more specific tradeoff between speed and accuracy must be established, the sampling step can be modified by enabling the Custom Sampling Step and setting the desired value.

Object	Definition
Default	Default is the best sampling step computed by the tool. It is based on the average size, in calibrated units, of a pixel in the image. This default sampling step is usually recommended. Default is automatically used by the tool if no other value is specified.

Object	Definition
Custom	<p>Custom Sampling Step lets you set a sampling step value other than the default. To select a custom sampling step, Enable must be checked.</p> <ul style="list-style-type: none"> <li>Increasing the sampling step value reduces the tool accuracy and decreases the execution time.</li> <li>Reducing the sampling step can increase the tool accuracy but will also increase the execution time.</li> </ul>

### **Output As Grayscale Image**

Indicates if the color image is converted to grey-scale after the color filters are applied.

### **Region of Interest (ROI)**

The region of interest is the area in which the tool carries out its process. Most tool position parameters can be set through the region of interest section of the tool interface. The following parameters define the tool region of interest.

#### **Offset**

Offset is the center of the ROI, defined by:

X	X coordinate of the center of the region of interest
Y	Y coordinate of the center of the region of interest
Degrees	Angle of rotation of the region of interest

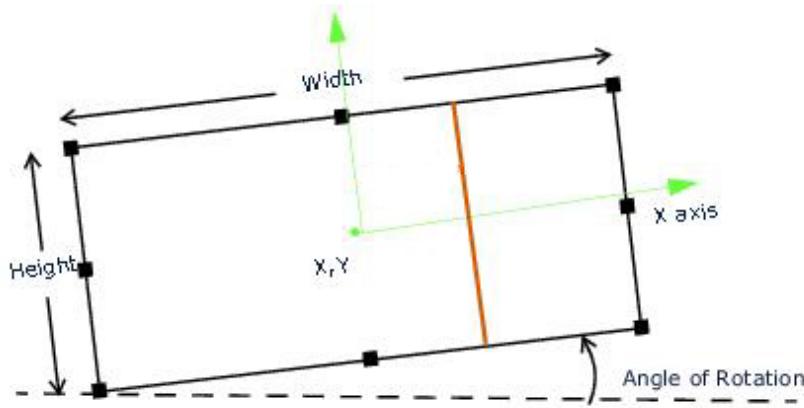
#### **Relative To**

Relative To specifies the tool that will provide the input to this tool. The tool can be dragged from the Tree structure and dropped into this field.

#### **Search Area**

Search Area is the size of the region of interest is defined by:

Height	Height of the region of interest
Width	Width of the region of interest



### *Location Properties for the Region of Interest*

#### **Related Topics**

[Color Matching Tool](#)

[Creating and Configuring Color Filters](#)

#### ***Creating and Configuring Color Filters***

The Color Matching tool analyzes the region of interest by applying all the defined filters to the image within the region of interest. Any number of filters can be added to the Color Matching tool.

The Filters section contains a list of all the filters that are configured for the current tool.

From the Filters list, you can:

- Add color filters
- Edit the color filters
- Delete color filters

To add a filter:

Under the Filters list, click the Add icon. A filter is added with the default name: Filter(n).

To delete a filter:

1. In the Filters list, select the filter to be removed.
2. Click the Delete icon.

To edit a filter:

1. In the Filter list, select the filter.
2. Click Edit or double-click on the filter name.

This opens the Color Finder window for the selected filter.

3. Configure the filter using the display or by entering values. See the following figure.

You can rename the filter from the Color Finder.

### ***Configuring Color Filters in the Color Finder***

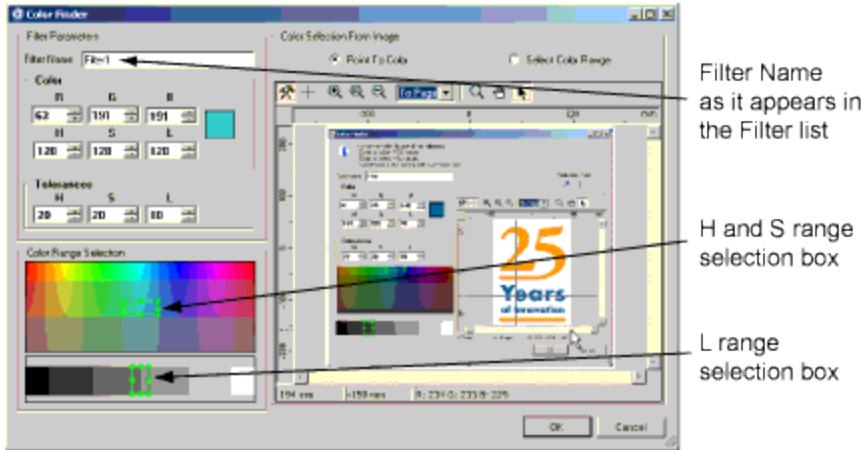
Color filters can be configured and edited in the Color Finder. See the following figure.

A filter contains a color definition, displayed both as RGB and HSL values, and tolerances for variations in hue, saturation, and luminance, with respect to the defined color. See Defining a Color by its RGB values on page 464 and Defining a Color by its HSL Values on page 464.

The Color and the Tolerances can be set by entering values or by using the display and selection tools that are in the Color Finder.

To configure the color filter:

1. Set an initial color in one of the following manners:
  - Pick a specific color in the display: Under Selection Tools, select the "Color Selection" icon. Using the mouse, move the cursor in the image display and click once to select the color of the pixel where the cursor is placed.
  - Pick an average color in the display: Under Selection Tools, select the "Range Selection" icon. Using the mouse, drag an area cursor in the image display. This calculates and selects the average color in the selected area.
  - Set the color RGB or HSL values: Enter values under R, G, and B, or under H, S, and L. The single color box above the "values" area provides a preview of the defined color.
2. Define Tolerances in one of the following manners:
  - Enter values for H, S, and L, in the Tolerances section.
  - In the Color Finder display, resize the bounding boxes to set tolerance values. As illustrated in the following graphic, the bounding box in the multicolored area sets tolerance ranges for hue and saturation. The bounding box in the grey-level area sets a tolerance range for luminance.
3. Click OK to confirm changes and return to the editor.



*Configuring a Color Filter in the Color Finder*

### **Color Values**

The value of a filter can be configured either by its HSL values or its RGB values.

The following table lists a few colors with their corresponding RGB and HSL values

#### **RGB and HSL Values for some Common Colors**

<b>Color Name</b>	<b>RGB Values</b>	<b>HSL Values</b>
White	255, 255, 255	0, 0, 255
Black	0, 0, 0	0, 0, 0
Middle Grey	127, 127, 127	0, 0, 128
Red	255, 0, 0	0, 255, 128
Green	0, 255, 0	85, 255, 128
Blue	0, 0, 255	170, 255, 128

### **Defining a Color by its RGB values**

RGB refers to a mode of describing, or quantifying, a color by its red (R), green (G), and blue (B) values.

- The values for R, G, and B range from 0 (no color) to 255 (maximum color)
- White is defined by R, G, B = 255, 255, 255
- Black is defined by R, G, B = 0, 0, 0
- The table shows the same common colors expressed in RGB and HSL

### **Defining a Color by its HSL Values**

HSL refers to a mode of describing, or quantifying, colors by their Hue, Saturation, and Luminance values.

### **Hue - H**

Hue is the quality of color that we perceive as the color itself, for example: red, green, yellow. The hue is determined by the perceived dominant wavelength, or the central tendency combined wavelengths, within the visible spectrum.

- Hue values range from 0 to 255. These values correspond to a displacement along the color spectrum starting from red =0.
- At H=0, the color is a shade of red, at H=85, the color is a shade of green, at H=170, the color is a shade of blue.

### **Saturation - S**

Saturation is the purity of the color, or the grey in a color. For example a high saturation value produces a very pure, intense color. Reducing the saturation value adds grey to the color.

### **Luminance - L**

Luminance is the brightness of the color, or the amount of white contained in the color. As the value increases the color becomes lighter and tends towards white. As the luminance value decreases the color is darker and tends towards black.

### ***Color Tolerances***

The color filter accepts any color values that are within defined tolerances. Tolerance values can only be expressed in HSL values. The tolerance range for each H, S, and L tolerance is applied to the defined color.

A defined tolerance value is distributed equally above and below the color value to which it applies.

For example, if the Color luminance (L) value is 200 and the Tolerance luminance (L) value is 20, the filter will accept pixels within a range of luminance values from 190 to 210.

### ***Related Topics***

[Color Matching Tool](#)

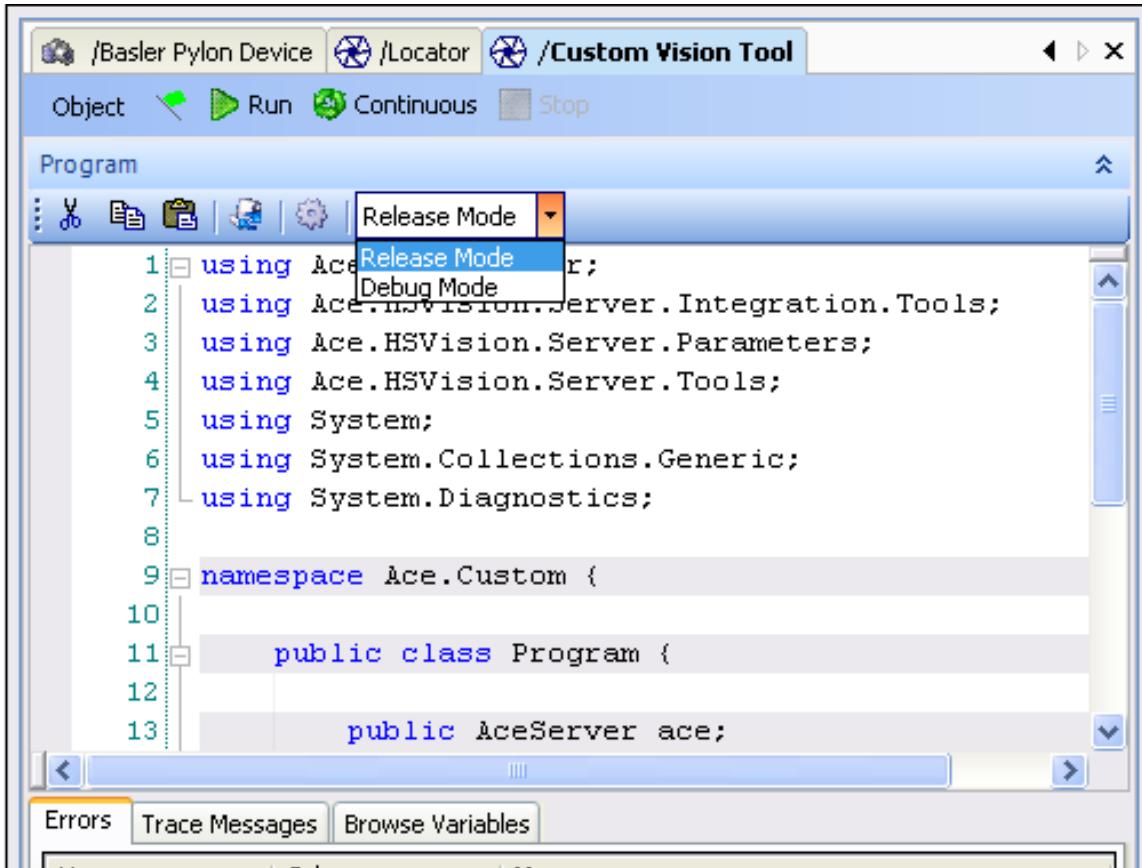
[Configuring Color Matching Properties](#)

### ***Custom Vision Tool***

The Custom Vision tool is used to specify the program that is called when the tool is executed.

From within a Custom Vision tool, other tools can be executed, and return a set of results that are used as the output of the tool.

To create a Custom Vision tool, right-click in the Tree structure, then select:

**New > Vision > Tool > Custom Vision Tool***Custom Vision Main Screen*

If the image source is not defined, you will see the message:

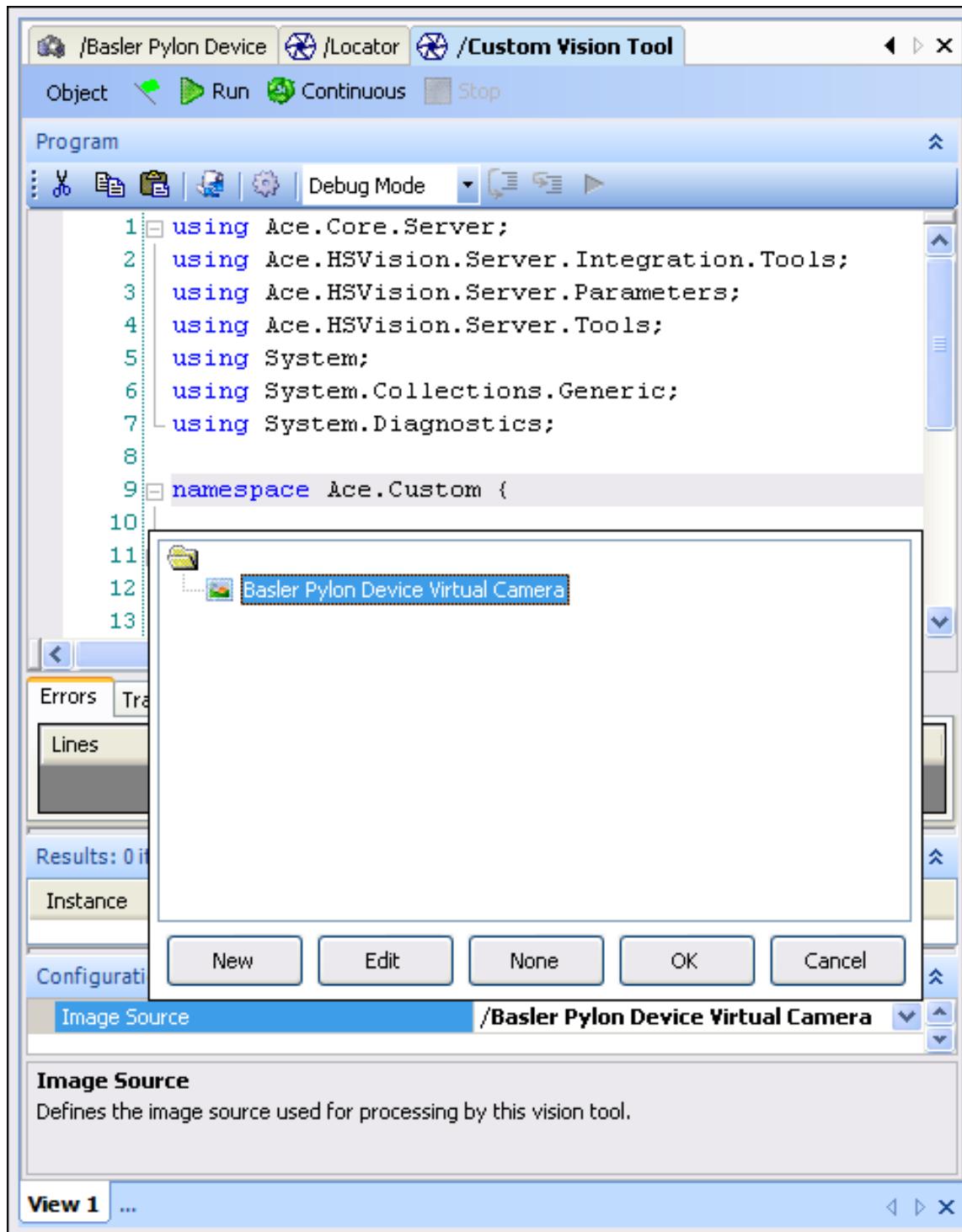
Image source is not defined for tool <tool name>

Where <tool name> is the tool name that does not have an image source defined. Note that you could be executing a tool relative to another tool that does not have the image source defined.

To select a vision device, highlight the Image Source field in Configuration, then click the down arrow to open a new window, which shows all of the available vision devices connected to the system. You can select a specific vision device from this list and associate it with the Custom Vision tool. See the following figure:

## Custom Vision Tool

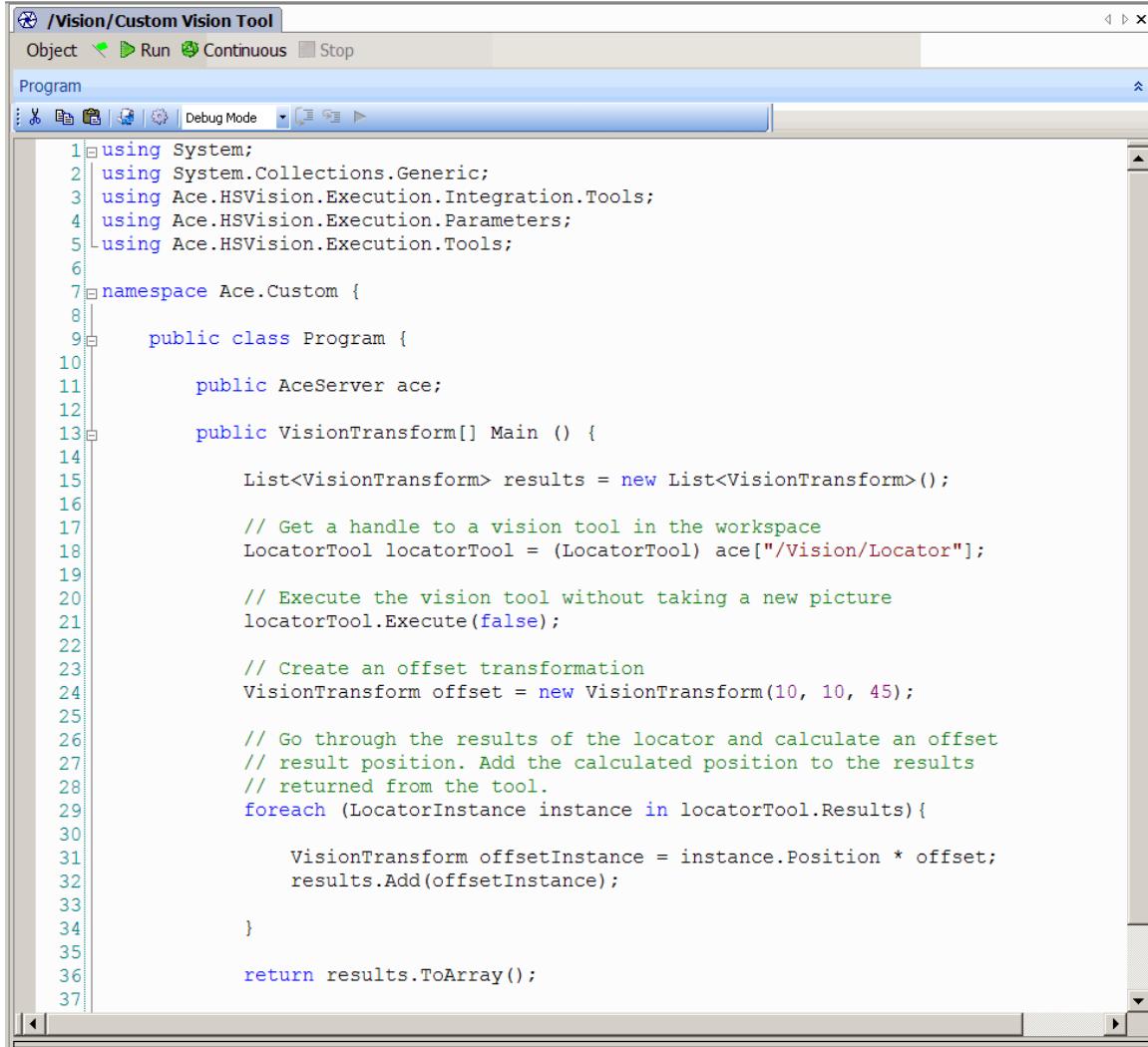
---



## Custom Vision Tool

---

### Image Source Pane



The screenshot shows the 'Custom Vision Tool' application window. The title bar reads '/Vision/Custom Vision Tool'. The menu bar includes 'Object', 'Run' (with options 'Continuous' and 'Stop'), and a toolbar with icons for file operations and 'Debug Mode'. The main area is a code editor displaying C# code for a 'Program' class. The code uses namespaces from the Ace.HSVision library to interact with a workspace and execute vision tools.

```
1 using System;
2 using System.Collections.Generic;
3 using Ace.HSVision.Execution.Integration.Tools;
4 using Ace.HSVision.Execution.Parameters;
5 using Ace.HSVision.Execution.Tools;
6
7 namespace Ace.Custom {
8
9     public class Program {
10
11         public AceServer ace;
12
13         public VisionTransform[] Main () {
14
15             List<VisionTransform> results = new List<VisionTransform>();
16
17             // Get a handle to a vision tool in the workspace
18             LocatorTool locatorTool = (LocatorTool) ace["/Vision/Locator"];
19
20             // Execute the vision tool without taking a new picture
21             locatorTool.Execute(false);
22
23             // Create an offset transformation
24             VisionTransform offset = new VisionTransform(10, 10, 45);
25
26             // Go through the results of the locator and calculate an offset
27             // result position. Add the calculated position to the results
28             // returned from the tool.
29             foreach (LocatorInstance instance in locatorTool.Results){
30
31                 VisionTransform offsetInstance = instance.Position * offset;
32                 results.Add(offsetInstance);
33
34             }
35
36         return results.ToArray();
37     }
38 }
```

Custom Vision Tool Code

### Configuration

Object	Definition
<b>Configuration</b>	
OverlayMarkers	This is covered in the ACE Sight Reference Guide.
<b>Properties</b>	

---

Object	Definition
Results Logging	Specifies if a results log will be generated. A csv extension generates csv format.
Show Name	Is the name associated with the result displayed in the Vision window? This is only evaluated if results graphics are enabled.
Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
<b>Tool Links</b>	
Image Source	Defines the image source used for processing by this vision tool.

### **Results**

Object	Definition
Instance	The index of the located instance
Position X	The X coordinate of the located instance
Position Y	The Y coordinate of the located instance
Angle	The angle of the located instance
Name	The name assigned to a located instance. You choose the names, and specify the criteria for which parts get which name. The names can later be used by ACE PackXpert for separating different types of parts.

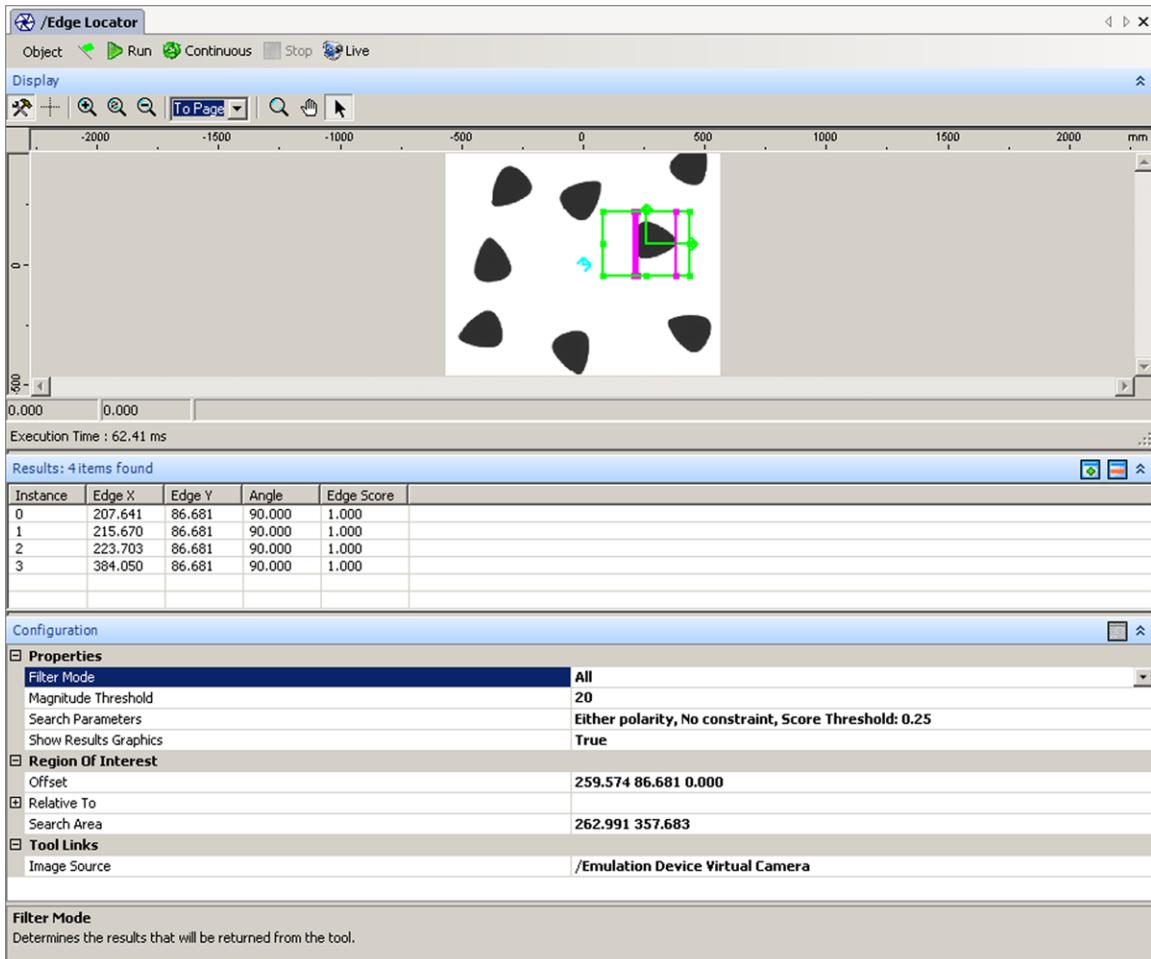
### **Edge Locator**

The Edge Locator tool identifies and measures the position of one or more edges on an object. The Edge Locator uses pixel grey-level values to detect edges found within the region of interest. Once potential edges have been located, the Edge Locator applies the constraints to determine which edges are valid. The Edge Locator determines the position of one or more edges; it does not measure the length of edges detected in the region of interest. To extrapolate and measure a line on an object, use the Edge Finder tool.

To create an Edge Locator tool, right-click in the Tree structure, then select:

**New > Vision > Tool > Edge Locator**

## Edge Locator



Edge Locator Object Editor

### Configuration

Object	Definition
<b>Properties</b>	
Filter Mode	Determines the results that will be returned from the tool (All, First Edge, Last Edge, or Middle).
Magnitude Threshold	Sets the threshold used to find edges on the magnitude curve.
Search Parameters	Set polarity, constraints, and threshold used by the tool.

<b>Object</b>	<b>Definition</b>
Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
<b>Tool Links</b>	
Image Source	Defines the image source used for processing by this vision tool.
<b>Region of Interest (ROI)</b>	
Offset	Specifies the center and rotation of the ROI. X, Y, Degrees.
Relative To	Specifies the tool that will provide the input to this tool. The tool can be dragged from the Folder pane and dropped into this field.
Search Area	Defines the height and width of the region of interest.
<b>Advanced Properties</b>	
Bilinear Interpolation Enabled	Specifies if bilinear interpolation is used to sample the input image. By default, bilinear interpolation is enabled because it improves accuracy.
Custom Sampling Step	When enabled, the tool uses the user-defined sampling step instead of the default optimal sampling step to sample the region of interest from the input image.
Edge Filter Half Width	Sets the half-width of the filter used by the tool to compute an edge magnitude curve from which edges are detected.
Sort Results Enabled	When enabled, the located edges will be sorted in descending order of score values.

### Results

<b>Object</b>	<b>Definition</b>
Instance	The index of the result instance
Frame/Group	The frame or group to which the result belongs.
Edge X	The X coordinate of the center point for the edge segment

Object	Definition
Edge Y	The Y coordinate of the center point for the edge segment
Angle	Angle of the edge segment
Edge Score	Score of the selected edge. The score is computed according the constraints set by the Search parameters.
Magnitude	The Magnitude of the edge indicates its peak value in the magnitude curve. Light to dark is -; dark to light is +.
Magnitude Score	Magnitude Score of the selected edge
Position Score	Position Score of the selected edge
Projection Average	Average Projection
Projection Magnitude	Projection Magnitude

#### ***Additional Menus***

#### **Edge Constraint Editor**

The Edge Locator tool uses a form for configuring the edge constraints. - See Configuring Edge Locator Properties on page 472.

#### ***Related Topics***

[Configuring Edge Locator Properties](#)

[Configuring Edge Locator Properties - Advanced](#)

#### ***Configuring Edge Locator Properties***

#### **Filter Mode**

Determines the edges that will be returned from the tool, as follows:

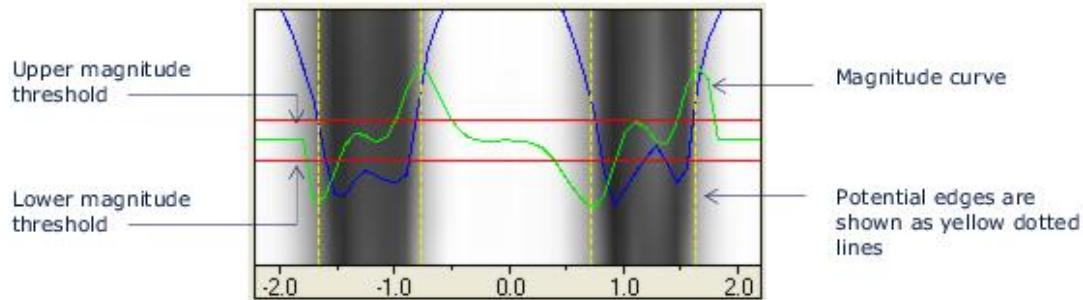
All, First Edge, Last Edge, Middle

#### **Magnitude Threshold**

Sets the acceptable magnitude value for potential edges. This value is expressed as an absolute value. There are two magnitude lines: an upper (positive) threshold and lower (negative) threshold.

Possible values are from 0 - 255.

Edge Magnitude expresses the strength of a potential edge. The (green) magnitude curve, represents magnitude values across the region of interest. Potential edges must have a magnitude greater than the upper threshold, or lower than the lower threshold. See the following figure.



### *Interpreting the Magnitude Threshold in the display area*

- All peaks on the edge magnitude curve that fall between the two red magnitude threshold lines are not considered.
- All peaks on the edge magnitude curve that are above the upper threshold or below the lower threshold are considered as potential edges.
- Sometimes unwanted edges, such as those corresponding to shadows, have magnitude values CLOSE to those of valid edges. When possible, do NOT use the magnitude threshold to reject such edges because in some images, lighting variations may cause edges of interest to also fall below the threshold. Instead, use polarity and position constraints that are not affected by lighting variations from one image to another.

## Edge Constraint Editor

This tool uses a form for defining the constraints.

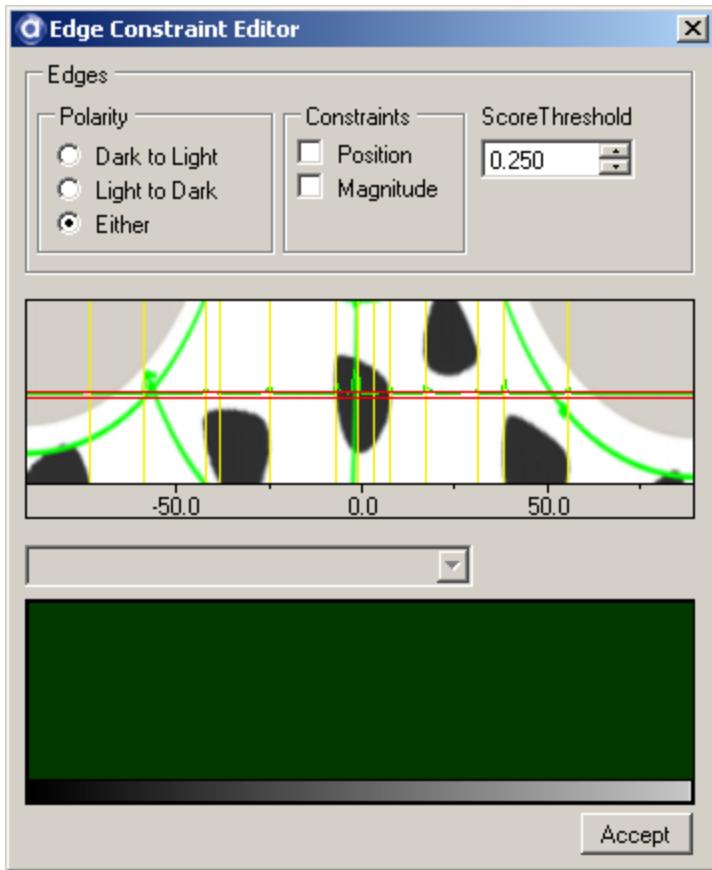
To set constraints, from the object editor:

1. Select **Configuration > Properties > Search Parameters**

Click Search Parameters to select it.

2. Click the browse icon (...

The following window opens.



*Edge Constraint Editor Window*

### **Edge Score**

The tool scores potential edges according to the specified edge parameters. The scoring method restricts the search so that only results for valid edge pairs are returned.

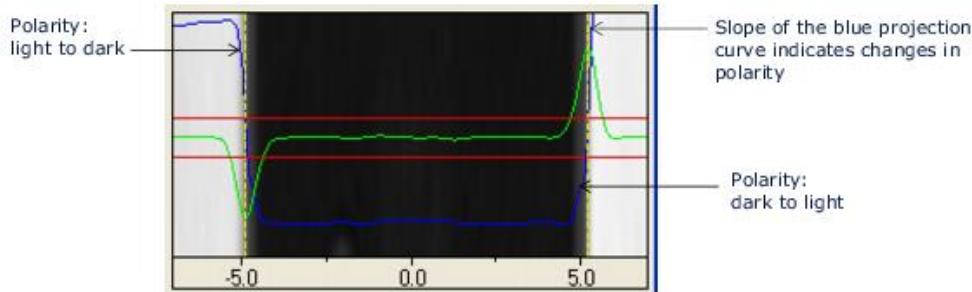
There are two types of parameters that affect the choice of valid edges: Polarity and Edge Score Constraints.

#### **Polarity**

Polarity corresponds to the change in light values, moving from left to right in the display, along the X-Axis in the region of interest. The tool applies the Polarity constraint before applying edge-score Constraints.

Polarity does not affect the Edge Score, but only edges that meet the selected Polarity are output as valid edges, regardless of their scores.

- Dark to Light will only accept edges occurring at transitions from a dark area to a light area.
- Light to Dark will only accept edges occurring at transitions from a light area to a dark area.
- Either will accept any edge, regardless of polarity.



*Edge Polarity*

### Edge Score Constraints

There are two types of constraints: Position and Magnitude. You can set the tool to use only one constraint or both. A graphical function editor is provided for viewing and setting each type of constraint.

- If only one constraint is selected, edges are scored based on the selected constraint.
- If both constraints are selected, each constraint accounts for 50% of the edge score.

### Magnitude Constraint

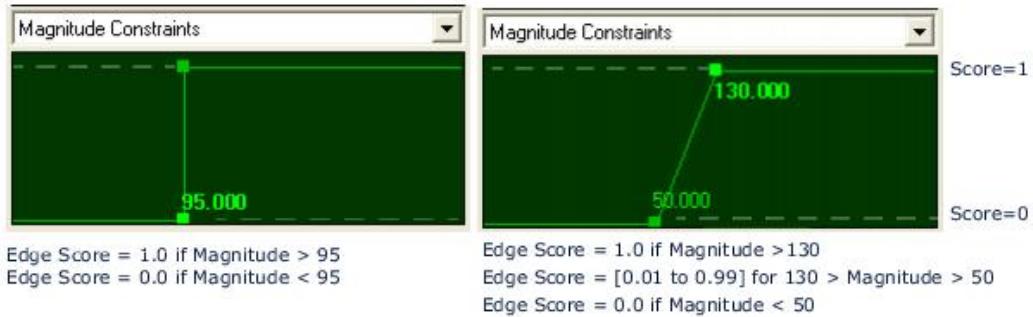
The Magnitude constraint is based on edge values relative to the Magnitude Threshold. Edges having a magnitude equal to or exceeding the Magnitude Threshold are given a score of 1. Edges with values below the Magnitude Threshold receive a score ranging from 0 to 0.999, according to a manually set magnitude constraint function.

The Magnitude Constraint is applied globally to all edges detected.

The following figure shows two different setups for a magnitude constraint function.

To set the Magnitude Constraint:

1. In the drop-down list above the function editor, select Magnitude Constraints.
2. In the Function Editor, use the mouse to drag handles and set the Magnitude limits. See examples in the following figure.



#### *Setting the Magnitude Constraint in the Function Editor*

### **Position Constraint**

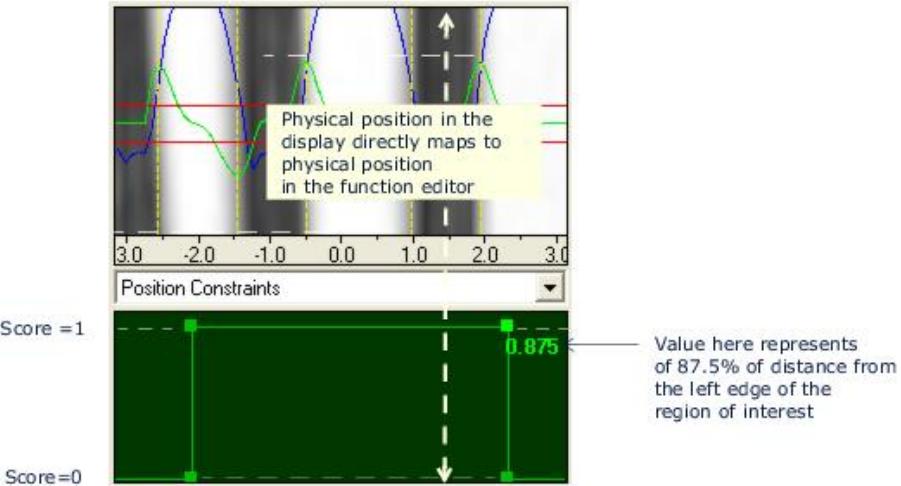
The Position constraint restricts the tool's search for edges to a specific zone of the region of interest.

- It is possible to graphically set a position constraint function when the approximate position of an edge is known beforehand. This is useful for scoring an edge based on its offset from the expected position.
- Values in the Constraint Function Editor indicate relative distance in the region of interest where 0.0 is the left-most position and 1.0 is the right-most position.

To set the Position Constraint:

1. In the drop-down list above the function editor, select Position Constraints.
2. In the Function Editor, use the mouse to drag handles and set the Position limits. See examples in the following figure.

The position in the function editor corresponds to the same position in the display.

*Setting the Position Constraint Function Editor*

### Score Threshold

The score threshold sets the minimum acceptable score for a valid edge. The tool will disregard edges that obtain a score lower than the Score Threshold.

- Scores attributed for constraints range from 0 to 1.
- If both Position and Magnitude constraints are enabled, each constraint accounts for 50% of the total edge score.

### Sort Results

You can enable the Sort Results checkbox to sort the located edges in descending order based on values. By default, Sort Results is not enabled and edges are output in the same left to right order as they appear on the projection curve.

### Region of Interest (ROI)

The region of interest is the area in which the tool carries out its process. Most tool position parameters can be set through the region of interest section of the tool interface. The following parameters define the tool region of interest.

#### Offset

Offset is the center of the ROI, defined by:

X	X coordinate of the center of the region of interest
---	--

Y	Y coordinate of the center of the region of interest
Degrees	Angle of rotation of the region of interest

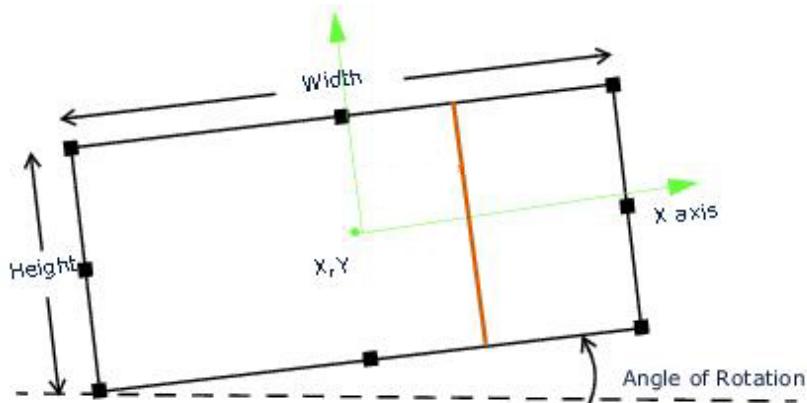
## Relative To

Relative To specifies the tool that will provide the input to this tool. The tool can be dragged from the Tree structure and dropped into this field.

## Search Area

Search Area is the size of the region of interest is defined by:

Height	Height of the region of interest
Width	Width of the region of interest



### *Location Properties for the Region of Interest*

**NOTE:** Before configuring the Edge Locator, execute the tool at least once and verify that the tool is being positioned correctly in the image.

The display represents the Edge Locator as a green rectangle, with found edges represented in magenta.

## Related Topics

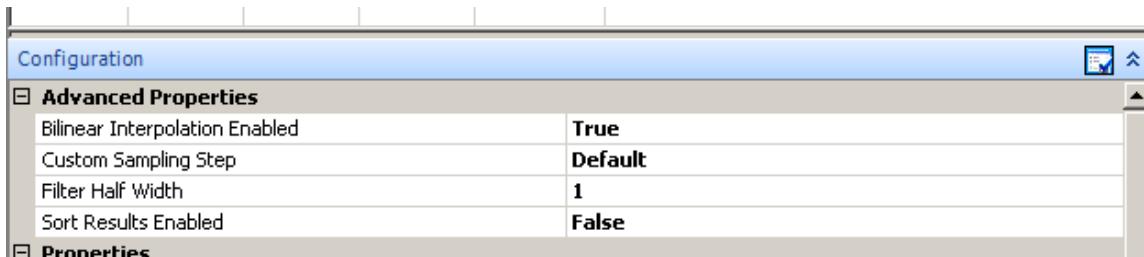
[Edge Locator](#)

[Configuring Edge Locator Properties - Advanced](#)

### **Configuring Edge Locator Properties - Advanced**

The Advanced Properties section of the Edge Locator tool interface provides access to advanced Edge Locator parameters and properties.

Sampling refers to the procedure used by the tool for gathering values within the portion of the input image that is bounded by the tool's region of interest. Two sampling parameters, the Sampling Step and Bilinear Interpolation, can be used as necessary to create a required tradeoff between speed and accuracy.



*Advanced Edge Locator Configuration Properties*

### **Bilinear Interpolation**

Bilinear Interpolation specifies if bilinear interpolation is used to sample the image before it is analyzed for image sharpness.

To ensure subpixel accuracy in inspection applications, Bilinear Interpolation should always be set to true (enabled). Non-interpolated sampling (Bilinear Interpolation disabled) should only be used in applications where the speed requirements are more critical than accuracy.

### **Custom Sampling Step**

Custom Sampling Step is the step size that the tool uses to sample the input image that is bounded by the tool region of interest. The sampling step represents the height and the width of a sampled pixel.

Object	Definition
Default Sampling Step	Default Sampling Step is the best sampling step computed by the tool, based on the average size, in calibrated units, of a pixel in the image. This default sampling step is usually recommended. Default Sampling Step is automatically used by the tool unless Custom Sampling Step is enabled.

Object	Definition
Custom Sampling Step	<p>For specific applications where a more appropriate tradeoff between speed and accuracy must be established, the sampling step can be modified by using a Custom Sampling Step.</p> <ul style="list-style-type: none"><li>• Increasing the sampling step value reduces the tool accuracy and decreases the execution time.</li><li>• Reducing the sampling step can increase the tool accuracy but can also increase the execution time.</li></ul>

### **Filter Half Width**

The filtering process smoothes peaks in the magnitude curve that are caused by noise. Edge Filter Half-Width should be set to a value approximately equivalent to the width of the edge, in pixels, as it appears in the image. An incorrect value can cause edges to be incorrectly detected.

### **Sort Results Enabled**

When Sort Results Enabled is set to false (default) edges are sorted in the order of their location within the region of interest. When set to true, edges are sorted in the order of their score, from highest to lowest.

By default, Sort Results is not enabled and edges are output in the same left to right order as they appear on the projection curve.

### **Results Logging**

If enabled, the results of the tool will be logged to a file.

If the extension of the destination file is csv, then the log is written in csv format, which opens with Excel. If a different extension is selected, the log is written in a plain text format (similar to the ACE Sight 2 format).

The size of the log file can be set (default size is 5 Mb).

#### ***Related Topics***

[Edge Locator Tool](#)

[Configuring Edge Locator Properties](#)

## **Gripper Clearance Tool**

### **Description**

The Gripper Clearance tool is a filter used to determine which parts can be picked without interference. Instances returned as results have passed the filter.

The tool uses histograms (typically one histogram per gripper finger) to determine whether the gripper can pick up a part without interference from another part or edges of the region of interest (ROI). Each set of histograms is applied to each instance in the region of interest.

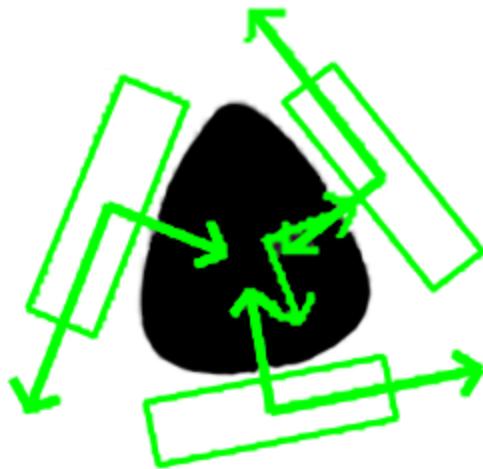
You use the Inspection Configuration field to specify the maximum and minimum Grey Level Mean, Variance, or Histogram pixel count, which are applied to all histograms in the tool. All histograms for an instance must have a value between the maximum and minimum, inclusive, for that instance to pass the filter.

- Grey Level Mean - This is absolute greylevel, and will change with lighting and other environmental changes. For this reason, it is best-suited for situations where these conditions will be constant (a clean, homogenous background and constant lighting intensity).
- Variance - This is the difference, from the mean, of the greylevel values in the histogram. Because it is relative, it is useful when lighting and other conditions may change.
- Histogram pixel count - This specifies the number of pixels, in each histogram, that may fall in the greylevel range that you specify.

To create a Gripper Clearance tool, right-click in the Tree structure, then select:

**New > Vision > Tool > Gripper Clearance**

All histograms are rectangles. Each histogram can be sized, rotated, and moved as needed, to mimic the size and position of the gripper fingers. Editing can be done by modifying Offset and Search Region in the configuration section, or with the mouse in the display window.



*Three Histograms Around a Part*

#### **Related Topics**

See "Configuring Gripper Clearance Properties"

See "Gripper Clearance Results"

[Image Histogram Tool](#)

#### **Configuring Gripper Clearance Properties**

The tab of the selected histogram is highlighted. Clicking on the origin or outline of a histogram with the mouse will select it, highlighting the tab for that histogram. (The cursor will change to either a two- or four-headed arrow near the outline, or a small cross without arrows near the origin.) You can also select a histogram by clicking on its tab.



*Gripper Clearance Tabs with Histogram 2 Selected*

When Adding a new histogram, the tool copies the last (highest numbered) histogram. The height and width are rounded to even numbers, the rotation will be zero, and the offset from the copied histogram will be +20 in both X and Y.

### ***Inspection Configuration***

This field is used to select which criterion to use in deciding if a histogram passes or fails, and the minimum and maximum values for that criterion. The choices are:

- Greylevel (0-255)
- Variance (0-10,000)
- Histogram pixel count (0-10,000)

If the values of all histograms for an instance are within the range you set, the instance will be returned as passing. These are applied to all histograms in the tool.

Although only one of these properties is active as a filter for the histograms at any given time, all three properties can be shown in the Results display.

#### **Grey Level**

This is the mean greylevel of each histogram.

#### **Variance**

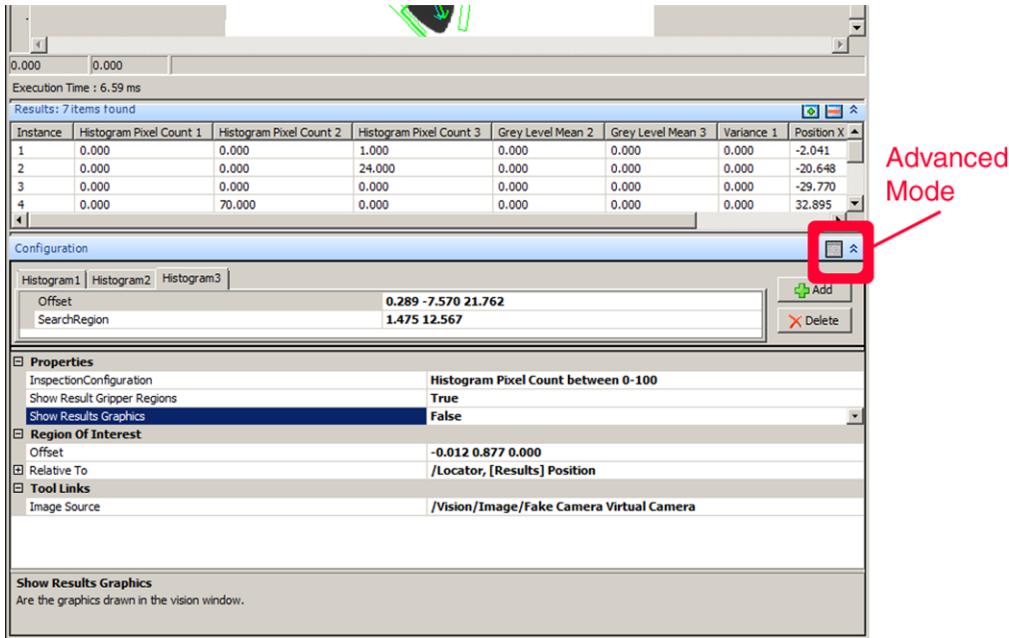
This is the variance, in greylevel value, from the mean greylevel of each histogram.

#### **Histogram pixel count**

This is the number of pixels in each histogram that are within the greylevel range that you set. Set the range for pixel count in this field.

Set the greylevel range of pixels to consider with Threshold Black and Threshold White. These two properties are set in the Advanced Properties section of this tool. Refer to the following figure.

**NOTE:** Thresholds and Tails are all specified in the Advanced Properties section of Configuration. Click the Advanced Mode button to toggle between displaying and hiding Advanced Properties.



*Advanced Mode Button, used to display or hide Advanced Properties*

### Thresholds

Thresholds exclude a range of pixel values from the histogram, according their greylevel value.

#### Threshold Black

The darkest greylevel value to consider when scanning the histogram. Greylevel values below Threshold Black are ignored during the histogram scanning. When a threshold is used and the tool is also configured to remove a percentage of pixels at the dark tail of the histogram (see the Tail Black property), the tail removal process begins to scan the histogram at Threshold Black, instead of starting at 0.

**NOTE:** The Inspection Configuration field ignores this property when using Grey Level Mean for filtering.

#### Threshold White

The lightest greylevel value to consider when scanning the histogram. Greylevel values above Threshold White are ignored during the histogram scanning. When a threshold is used and the tool is also configured to remove a percentage of pixels at the light tail of the histogram (see the Tail White property), the tail removal process begins at Threshold White, instead of starting at 255.

**NOTE:** The Inspection Configuration field ignores this property when using Grey Level Mean for filtering.

### **Tails**

A tail specifies a percentage of pixels to be removed from the dark and light ends of the initial histogram.

#### **Tail Black Grey Level Value**

Tail Black Grey Level Value specifies the percentage of the total number of pixels in the histogram to ignore at the dark end of the greylevel distribution in each histogram. After its creation, the histogram is scanned, starting from greylevel 0. The pixels at the dark end of the histogram are then cleared until the percentage of pixels defined by Tail Black Grey Level Value is reached.

#### **Tail White Grey Level Value**

Tail White Grey Level Value specifies the percentage of the total number of pixels in the histogram to ignore at the light end of the greylevel distribution in each histogram. After its creation, the histogram is scanned, starting from greylevel 255. The pixels at the light end of the histogram are then cleared until the percentage of pixels defined by Tail White Grey Level Value is reached.

### **Custom Menu**

#### **Offset**

Specifies the offset and rotation of the selected histogram from the tool origin. X, Y, Degrees.

#### **Search Region**

Specifies the height and width of the selected histogram.

#### **Add**

Adds a new histogram. If this is not the first histogram in this tool, the previous histogram is replicated, with 0 rotation, and with X and Y values 20 greater than the previous histogram.

#### **Delete**

Removes the selected histogram.

**Tool Links****Image Source**

Input to the Gripper Clearance tool (which virtual camera/tool).

**NOTE:** This needs to be the same source as is used by the Relative To tool.

**Region of Interest (ROI)**

The region of interest is the area in which the tool carries out its process. Most tool position parameters can be set through the region of interest section of the tool interface. The following parameters define the tool region of interest.

**Offset**

Offset is the center of the ROI, defined by:

X	X coordinate of the center of the region of interest
Y	Y coordinate of the center of the region of interest
Degrees	Angle of rotation of the region of interest

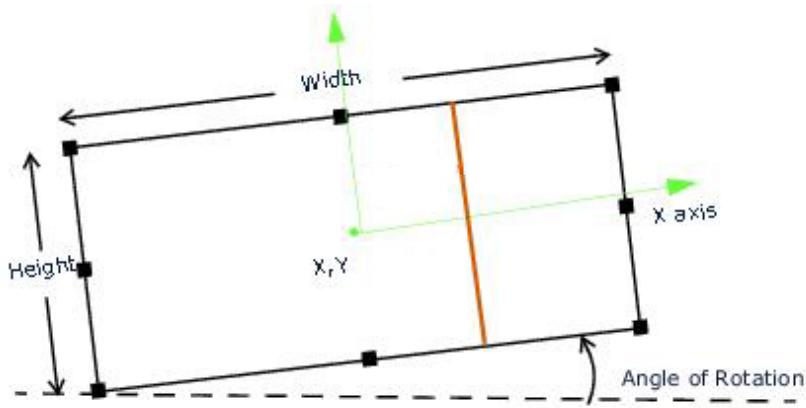
**Relative To**

Relative To specifies the tool that will provide the input to this tool. The tool can be dragged from the Tree structure and dropped into this field.

**Search Area**

Search Area is the size of the region of interest is defined by:

Height	Height of the region of interest
Width	Width of the region of interest



*Location Properties for the Region of Interest*

#### **Display**

#### **Show Result Gripper Regions**

When true, the Vision Window displays the gripper regions. Passed regions will be blue; failed regions will be red.

#### **Show Results graphic**

Specifies if the graphics are drawn in the Vision Window.

#### **Results Logging**

Specifies if a results log will be generated. A csv extension generates csv format.

#### **Related Topics**

See "Gripper Clearance Tool"

See "Gripper Clearance Results"

[Image Histogram Tool](#)

#### **Gripper Clearance Results**

The Gripper Clearance tool filters instances, input from the Relative To tool, based on the values in the tool's histograms for each instance. It returns only instances that pass all histograms for that instance.

---

## Display Window

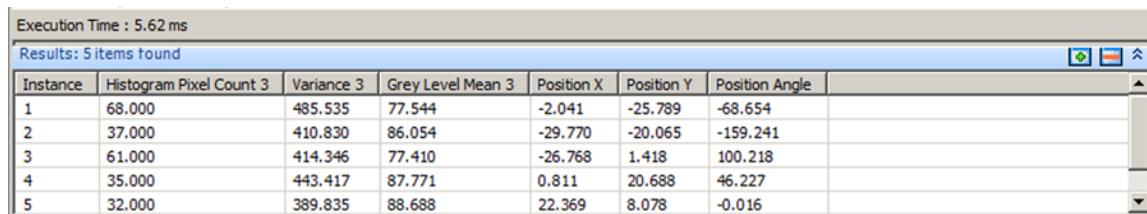
The Display window shows the image being processed, the region of interest selected, and the locations of found instances. The elapsed time is displayed at the bottom of this window.

## Grid of Results

Instances that satisfy all histograms will be reported as results. The Position X, Position Y, and Position Angle are the same for this tool as they are for the Relative To tool (the input to this tool).

Object	Definition
Elapsed Time	Total execution time of the tool. Elapsed Time is not displayed in the grid of results, but is shown at the bottom of the Display window for each iteration of the Image Histogram Tool.
Instance	Index of the result instance.
Position X	X coordinate of the origin of the instance.
Position Y	Y coordinate of the origin of the instance.
Position Angle	Angle of the instance.
Grey Level Mean n,	Mean greylevel for each histogram n, reported for each result instance
Histogram Pixel Count n	Number of pixels from Threshold Black to Threshold White, inclusive, in histogram n.
Variance n	Variance of greylevel values from the mean in histogram n.

Areas outside the region of interest are considered solid, so parts near the edge may not be returned as instances if a histogram goes off the region of interest.



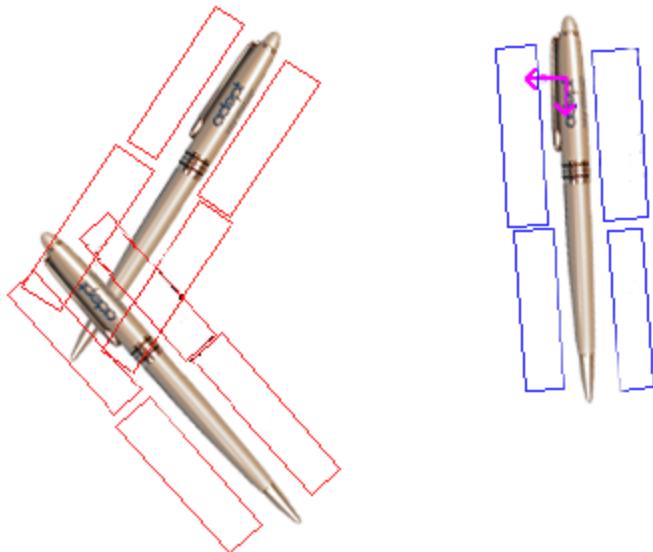
The screenshot shows a software interface titled 'Gripper Clearance Tool'. At the top, it displays 'Execution Time : 5.62 ms' and 'Results: 5 items found'. Below this is a table with the following data:

Instance	Histogram Pixel Count 3	Variance 3	Grey Level Mean 3	Position X	Position Y	Position Angle
1	68.000	485.535	77.544	-2.041	-25.789	-68.654
2	37.000	410.830	86.054	-29.770	-20.065	-159.241
3	61.000	414.346	77.410	-26.768	1.418	100.218
4	35.000	443.417	87.771	0.811	20.688	46.227
5	32.000	389.835	88.688	22.369	8.078	-0.016

*Results of the Gripper Clearance Tool*

## Vision Window

Passed and failed histograms are displayed in the Vision Window if both Show Result Gripper Regions and Show Results Graphics are true. Passed histograms are blue; failed histograms are red. For any given instance, histograms will be either all passed or all failed.



*Vision Window, with Passed and Failed Histograms*

### **Related Topics**

See "Gripper Clearance Tool"

See "Configuring Gripper Clearance Properties"

[Image Histogram Tool](#)

## **Image Histogram**

The Image Histogram tool computes image statistics for all the pixels contained in the tool's region of interest. Pixels can be excluded from the distribution by thresholds or tail functions.

---

(See Configuration on page 492.) The histogram ignores pixels that have been excluded.

Typical uses for the Image Histogram tool include:

- Verifying that the zone around an object is clear of clutter and can therefore be gripped by a robot.
- Verifying and validating the camera iris adjustment or the lighting setup of an application.
- Providing input to the Inspection tool, for filtering the results of the histogram.

To create an Image Histogram tool, right-click in the Tree structure, then select:

**New > Vision > Tool > Image Histogram**

## Image Histogram

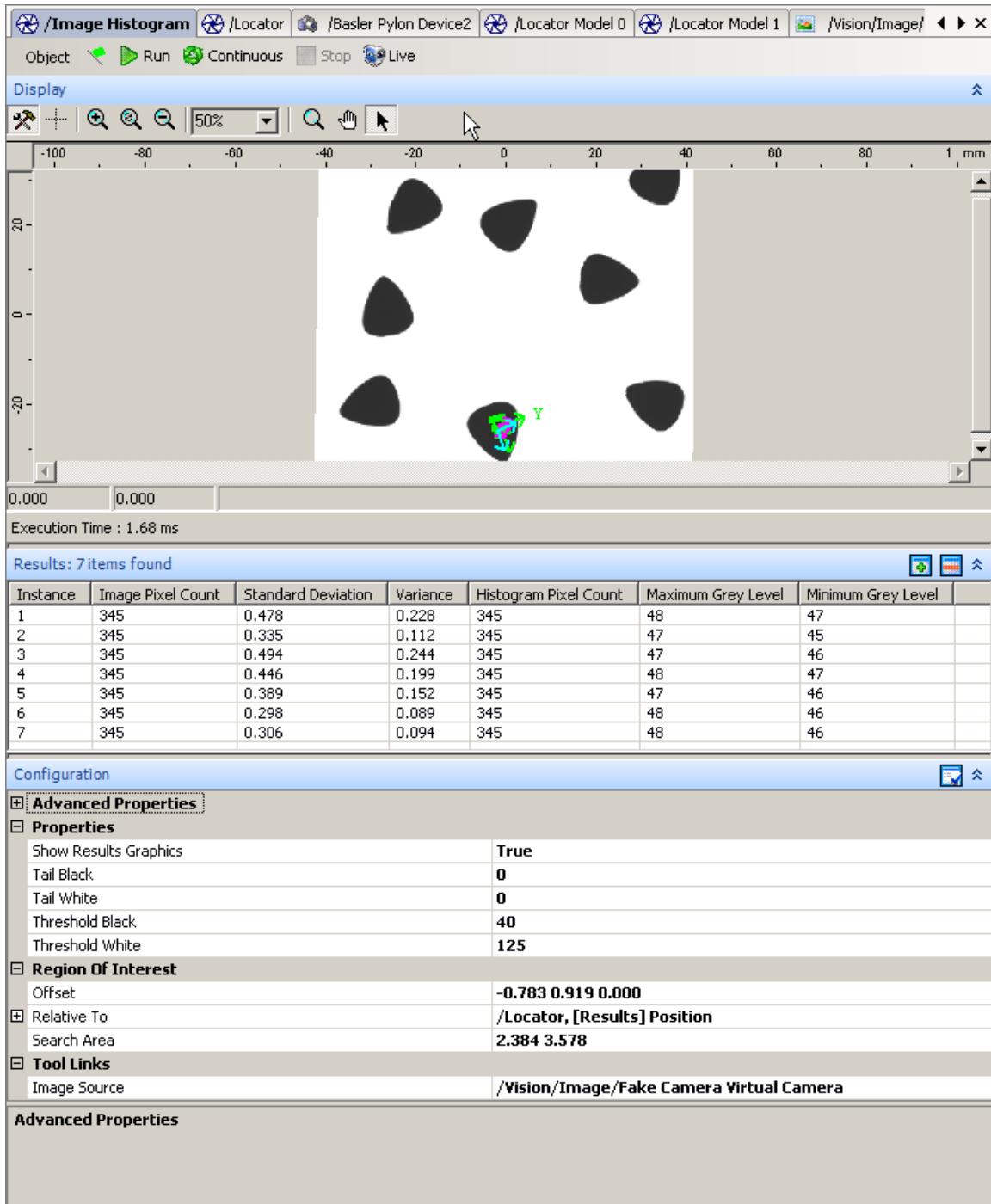


Image Histogram Object Editor

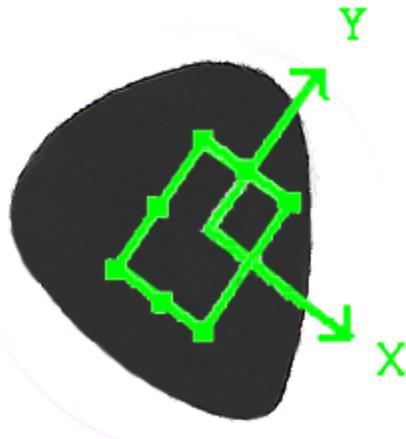


Image Histogram ROI

#### **Configuration**

Object	Definition
<b>Properties</b>	
Threshold Black	Darkest grey-level value to consider when building the histogram.
Threshold White	Brightest grey-level value to consider when building the histogram.
Tail Black	Amount of pixels to ignore at the dark end of the grey-level distribution in the tool region of interest.
Tail White	Amount of pixels to ignore at the bright end of the grey-level distribution in the tool region of interest.
Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.  This will have the same center and angle as the ROI.
<b>Tool Links</b>	
Image Source	Defines the image source used for processing by this vision tool.
<b>Region of Interest (ROI)</b>	

Object	Definition
Offset	Shift of the ROI from X, Y, and angle returned by the Relative To tool, or center and rotation of the ROI: X, Y, and angle, if no tool is specified.
Relative To	Specifies the tool that will provide the input to this tool. The tool can be dragged from the Folder pane and dropped into this field.
LinkName	Selects which results of the Relative To tool to use. Default is Position, when applicable.
Search Area	Height and width of the region of interest.
<b>Advanced</b>	
Bilinear Interpolation Enabled	Specifies if bilinear interpolation is used to sample the input image. By default, bilinear interpolation is enabled because it improves accuracy.
Custom Sampling Step	Step used by the tool to sample the area of the input image.
Image Subsampling	Factor used to sub-sample the greyscale image in the tool region of interest.

### **Results**

See Image Histogram Results on page 497.

### **Related Topics**

[Configuring Image Histogram Properties](#)

[Configuring Image Histogram Properties - Advanced](#)

[Image Histogram Results](#)

[Gripper Clearance Tool](#)

### **Configuring Image Histogram Properties**

The Image Histogram tool calculates grey-level statistics for a selected region of interest. The final histogram, for which the tool calculates the statistics, ignores pixels that have been excluded by thresholds or tails. A grey-level of 0 is black; 255 is white.

**NOTE:** Any pixels within the ROI but beyond the actual image are returned with a value of 0.

Thresholds exclude a range of pixel values from the histogram, according their grey-level value.

### **Black Threshold**

The Black threshold excludes dark pixels that have a grey-level value lower than the threshold value. The excluded pixels are not used to calculate histogram results.

When a threshold is used and the tool is also configured to remove a percentage of pixels at the dark tail of the histogram (see the Tail Black property), the tail-removal process begins to scan the histogram at the bin corresponding to Threshold Black, instead of starting at bin 0.

### **White Threshold**

The White threshold excludes light pixels that have a grey-level value higher than the threshold value. The excluded pixels are not used to calculate histogram results.

When a threshold is used and the tool is also configured to remove a percentage of pixels at the bright tail of the histogram (see the Tail White property), the tail-removal process begins to scan the histogram at the bin corresponding to Threshold White, instead of starting at bin 255.

### **Tails**

A tail specifies the amount pixels to be removed from the dark and light ends of the initial histogram.

### **Tail Black**

Tail Black specifies the amount of dark pixels to be excluded from the histogram, starting from the dark end of the histogram distribution (0). This is expressed as a percentage of the total number of pixels in the tool region of interest before tails are removed.

After its creation, the histogram is scanned, starting from bin 0. The bins at the dark end of the histogram are then cleared until the amount of pixels defined by Tail Black is reached.

### **Tail White**

Tail White specifies the amount of light pixels to be excluded from the histogram, starting from the light end of the histogram distribution (255). This is expressed as a percentage of the total number of pixels in the tool's region of interest before tails are removed.

After its creation, the histogram is scanned, starting from bin 255. The bins at the bright end of the histogram are then cleared until the amount of pixels defined by Tail White is reached.

### **Region of Interest (ROI)**

The region of interest is the area in which the tool carries out its process. Most tool position parameters can be set through the region of interest section of the tool interface. The

following parameters define the tool region of interest.

### Offset

Offset is the center of the ROI, defined by:

X	X coordinate of the center of the region of interest
Y	Y coordinate of the center of the region of interest
Degrees	Angle of rotation of the region of interest

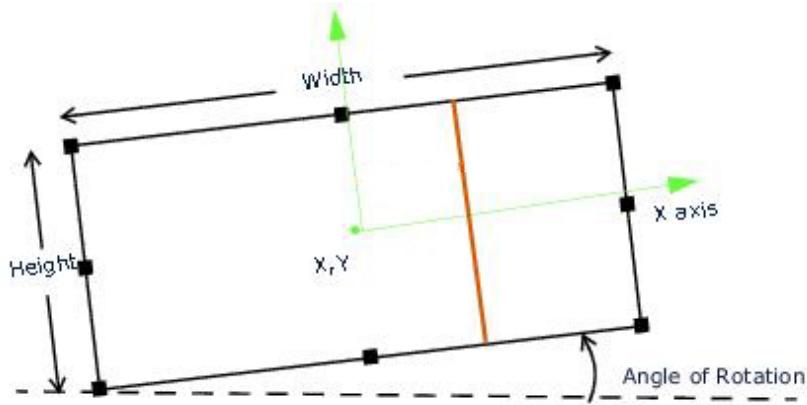
### Relative To

Relative To specifies the tool that will provide the input to this tool. The tool can be dragged from the Tree structure and dropped into this field.

### Search Area

Search Area is the size of the region of interest is defined by:

Height	Height of the region of interest
Width	Width of the region of interest



### *Location Properties for the Region of Interest*

#### **Related Topics**

[Image Histogram](#)

[Configuring Image Histogram Properties - Advanced](#)

[Image Histogram Results](#)

### ***Configuring Image Histogram Properties - Advanced***

The Advanced Properties section of the Image Histogram tool interface provides access to advanced Image Histogram tool parameters and properties.

Sampling refers to the procedure used by the tool for gathering values within the portion of the input image that is bounded by the tool region of interest. Two sampling parameters, the Sampling Step and Bilinear Interpolation, can be used to create the desired balance between speed and accuracy.

#### **Bilinear Interpolation**

Bilinear Interpolation uses pixel averaging to improve the quality of the image. It is applied before the image is analyzed.

To ensure subpixel accuracy in inspection applications, Bilinear Interpolation should always be set to True (enabled). Non-interpolated sampling (Bilinear Interpolation disabled) should only be used in applications where the speed requirements are more critical than accuracy.

#### **Custom Sampling Step (1 - 100)**

The Sampling Step is the step used by the tool to sample the area of the input image that is bounded by the tool region of interest. The sampling step represents the height and the width of a sampled pixel.

For applications where a more specific tradeoff between speed and accuracy must be established, the sampling step can be modified by enabling the Custom Sampling Step and setting the desired value.

Object	Definition
Default	Default is the best sampling step computed by the tool. It is based on the average size, in calibrated units, of a pixel in the image. This default sampling step is usually recommended. Default is automatically used by the tool if no other value is specified.

Object	Definition
Custom	<p>Custom Sampling Step lets you set a sampling step value other than the default. To select a custom sampling step, Enable must be checked.</p> <ul style="list-style-type: none"><li>• Increasing the sampling step value reduces the tool accuracy and decreases the execution time.</li><li>• Reducing the sampling step can increase the tool accuracy but will also increase the execution time.</li></ul>

## **Image Subsampling**

The image subsampling function coarsely resamples the image in the tool region of interest. This is useful when the image does not contain high frequency transitions or textures.

In such cases, the processing time is reduced, without having a negative impact on the accuracy of the results.

With a subsampling factor of 1, the greyscale image is not subsampled. With a subsampling factor of 2, the greyscale image is subsampled in tiles of 2x2 pixels. With a subsampling factor of 3 the greyscale image is subsampled in tiles of 3x3 pixels, and so forth.

## **Results Logging**

If enabled, the results of the tool will be logged to a file.

If the extension of the destination file is csv, then the log is written in csv format, which opens with Excel. If a different extension is selected, the log is written in a plain text format (similar to the ACE Sight 2 format).

The size of the log file can be set (default size is 5 Mb).

### ***Related Topics***

[Image Histogram](#)

[Configuring Image Histogram Properties](#)

[Image Histogram Results](#)

### ***Image Histogram Results***

The Image Histogram Tool outputs statistics about grey-levels for a selected region of interest.

---

## Display Window

The Display window shows the image being processed, the region of interest selected, and the locations of found instances. The elapsed time is displayed at the bottom of this window.

Execution Time : 1.68 ms						
Results: 7 items found						
Instance	Image Pixel Count	Standard Deviation	Variance	Histogram Pixel Count	Maximum Grey Level	Minimum Grey Level
1	345	0.478	0.228	345	48	47
2	345	0.335	0.112	345	47	45
3	345	0.494	0.244	345	47	46
4	345	0.446	0.199	345	48	47
5	345	0.389	0.152	345	47	46
6	345	0.298	0.089	345	48	46
7	345	0.306	0.094	345	48	46

### Histogram Grid of Results

If there are multiple instances input, as shown, you can set the ROI to be relative to one of them, and have the Image Histogram tool look at the same relative ROI on each instance.

## Grid of Results

The grid of results presents the statistical results for the region of interest analyzed by the Image Histogram tool.

The Image Histogram tool outputs the following results:

Object	Definition
Elapsed Time	The total execution time of the tool. Elapsed Time is not displayed in the grid of results, but is shown at the bottom of the Display window for each iteration of the Image Histogram Tool.
Instance	Identification number of each histogram output by the Image Histogram tool.
Frame/Group	The frame or group to which the result belongs.
Minimum Grey-Level	Lowest grey-level value of all pixels in the tool region of interest that are included in the final histogram. Pixels removed from the histogram by tails or thresholds are not included in this calculation. 0 = black; 255 = white.

## Image Histogram

---

Object	Definition
Maximum Grey-Level	Highest grey-level value of all pixels in the tool region of interest that are included in the final histogram. Pixels removed from the histogram by tails or thresholds are not included in this calculation. 0 = black; 255 = white.
Grey-Level Range	Range of grey-level values of the pixels in the tool region of interest that are included in the final histogram. Pixels removed from the histogram by tails or thresholds are not included in this calculation. Grey-Level Range = Maximum Grey-Level Value - Minimum Grey-Level Value + 1.
Median	Median of the grey-level distribution of the pixels in the tool region of interest that are included in the final histogram. Pixels removed from the histogram by tails or thresholds are not included in this calculation.
Variance	Variance of the grey-level distribution of the pixels in the tool region of interest that are included in the final histogram. Pixels removed from the histogram by tails or thresholds are not included in this calculation.
Standard Deviation	Standard deviation of the grey-level distribution of the pixels in the tool region of interest that are included in the final histogram. Pixels removed from the histogram by tails or thresholds are not included in this calculation.
Mean	Mean of the grey-level distribution of the pixels in the tool region of interest
Mode	Grey-level value which corresponds to the histogram bin with the highest number of pixels.
Image Height	Height of the tool region of interest, expressed in pixels.
Image Width	Width of the tool region of interest, expressed in pixels.
Image Pixel Count	Number of pixels in the tool region of interest, before applying tails and thresholds.

Object	Definition
Mode Pixel Count	Number of pixels in the histogram bin which corresponds to the Mode of the grey-level distribution of all pixels in the tool region of interest.
Histogram Pixel Count	Number of pixels in the tool region of interest, after applying tails and thresholds.
Tail Black Grey-Level	Darkest grey-level value that remains in the histogram after the tail is removed.
Tail White Grey-Level	Brightest grey-level value that remains in the histogram after the tail is removed.

#### **Related Topics**

[Image Histogram](#)

[Configuring Image Histogram Properties](#)

[Configuring Image Histogram Properties - Advanced](#)

## **Image Processing**

The Image Processing Tool processes greyscale images by applying arithmetic, assignment, logical, filtering, morphological, or histogram operations. You can also define and apply custom filtering operations. Each Image Processing Tool in an application performs a selected operation on an image called the input image. An image processing operation can also involve another image or a constant, as well as set of processing parameters.

To create an Image Processing tool, right-click in the Tree structure, then select:

**New > Vision > Tool > Image Processing**

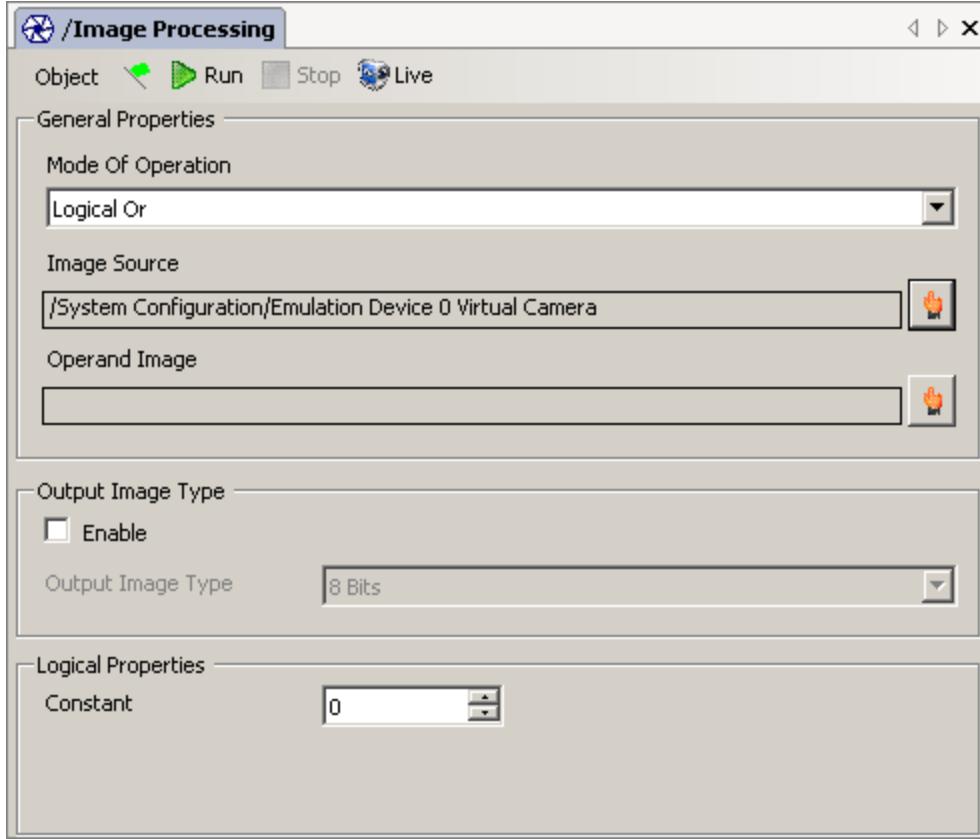


Image Processing Object Editor

### Configuration

Object	Definition
<b>General Properties</b>	
Mode of Operation	Operation applied by the Image Processing tool.
Image Source	Defines the image source used for processing by this vision tool.
Operand Image	The link to the operand vision tool executed for this vision tool (may be None).
<b>Output Image Type</b>	
Enable checkbox	Enables the Output Image Type.

Object	Definition
Output Image Type	Output image type when Output Image Type is enabled. By default, the Image Processing Tool outputs all resulting images as unsigned 8-bit images. 16- and 32-bit are also available.
<b>Other Properties</b>	
The title and contents of Other Properties vary with the Mode of Operation selected. The possibilities are: Arithmetic, Assignment, Logical, Filtering, Morphological, Histogram, and Transform. Not all of the Modes have corresponding Properties.	

### Results

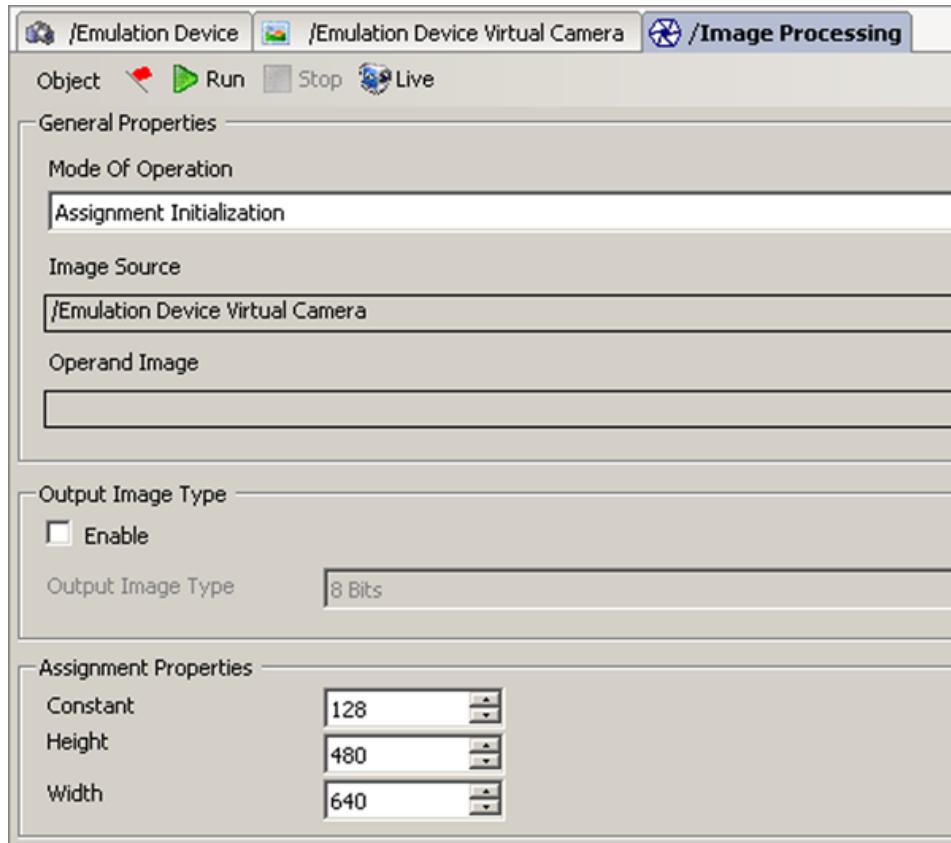
The Image Processing tool outputs images that can be used by other vision tools.

### Custom Editor

When this tool is created, an image buffer is created in the Vision Window with the same name as this tool. (See the following figure.) The results of the tool are displayed in the buffer.



The following figure shows the custom editing form.



*Image Processing Custom Editing Form*

## **Image Types**

The tool can accept unsigned 8-bit, signed 16-bit, and signed 32-bit images as input. The processing is usually performed in the defined type, based on input or operand image, or in a promoted type (signed 16-bit), if needed.

The Image Processing Tool output is of the same type as the input image unless the user overrides the type by setting another value, or an output image already exists.

## **What is an Image Processing Operation**

An image processing operation is a process carried out by the Image Processing Tool on an Input image. The result of an operation is an output image that can be used by other ACE Sight vision tools.

Image Processing operations are typically applied to images before they are processed by other vision tools. Some complex image processing applications may require two or more Image Processing operations.

Some common uses of an image processing tool are:

- Inverting images (negative image)
- Creating a binary image, using a threshold operation
- Sharpening or averaging an image to improve quality.

### **Basic Steps for Configuring an Image Processing Tool**

1. Select the tool that will provide Input images. See [Image Source](#).
2. Select the tool that provides an Operand image, if required. Many operations do not require an operand Image.
3. Select the Operation that will be performed by the tool.
4. In the Advanced Parameters, configure the parameters for the selected operation. See Configuring Image Processing Properties - Advanced on page 522.
5. Test and verify the results. See Image Processing Tool Results.

#### **Image Source**

The Image Source required by the Image Processing Tool is an image (generally provided by another tool). This image will be processed and modified by the Image Processing Tool.

The Image Processing Tool cannot be frame-based, and the tool's region of interest is always the entire input image. Therefore, this tool does not have any Location (positioning) parameters.

**NOTE:** The Image Processing Tool processes greyscale images only. If the Input Image is a color image, the Image Processing Tool may fail to execute, or may execute and output invalid results.

To set the Input:

1. Execute the tool once to make sure that an input image is available.
2. From the Image Source, select the tool that will provide the input image.
3. If the required tool does not appear in the drop-down list, make sure that the required tool (Acquire Image or other) has been added to the Workspace Explorer, above the Image Processing Tool.

#### **Operand Image**

Some image processing operations require an Operand Image. This operand image is provided by another tool or by a vision device. An Operand Image is selected following the same procedure as described in the Input (Source Image).

---

- Some operations require a constant as a second operation. This constant must be defined in the Advanced Parameters section of the tool interface.
- The Operand Image must be set to None (displays as an empty field) if a constant must be applied as an operand. Otherwise any selected Operand Image will override the selected constant.

### ***Modes of Operation***

The selected Operation corresponds to the process that will be applied to the input image by the Image Processing Tool. Each Image Processing Tool can apply a single operation.

Once an operation is selected, parameters related to the operation such as clipping, scale, constant (operand), and others, must be configured in the Properties section. This section will change its label depending on the type of properties (logical, filtering, etc.).

The following table provides a list and short description of the available operations. For more information on a specific operation, see the Configuring Image Processing Properties on page 508 section.

***List of Available Operations***

<b>Object</b>	<b>Definition</b>
<b>Arithmetic Properties</b>	
Arithmetic Addition	Operand value (constant or Operand Image pixel) is added to the corresponding pixel in the input image.
Arithmetic Subtraction	Operand value (constant or Operand Image pixel) is subtracted from the corresponding pixel in the input image.
Arithmetic Multiplication	The input image pixel value is multiplied by the Operand value (constant or corresponding Operand Image pixel).
Arithmetic Division	The input image pixel value is divided by the Operand value (constant or corresponding Operand image pixel). The result is scaled and clipped, and finally written to the output image.
Arithmetic Lightest	The Operand value (constant or Operand Image pixel) and corresponding pixel in the input image are compared to find the maximal value.
Arithmetic Darkest	The Operand value (constant or Operand Image pixel) and corresponding pixel in the input image are compared to find the minimal value.

Object	Definition
<b>Assignment Properties</b>	
Assignment Initialization	All the pixels of the output image are set to a specific constant value. The height and width of the output image must be specified.
Assignment Copy	Each input image pixel is copied to the corresponding output image pixel.
Assignment Inversion	The input image pixel value is inverted and the result is copied to the corresponding output image pixel.
<b>Logical Properties</b>	
Logical And	AND operation is applied to the Operand value (constant or Operand image pixel) and the corresponding pixel in the input image.
Logical NAnd	NAND operation is applied to the Operand value (constant or Operand image pixel) and the corresponding pixel in the input image.
Logical Or	OR operation is applied to the Operand value (constant or Operand image pixel) and the corresponding pixel in the input image.
Logical XOR	XOR operation is applied to the Operand value (constant or Operand image pixel) and the corresponding pixel in the input image.
Logical NOR	NOR operation is applied using the Operand value (constant or Operand image pixel) and the corresponding pixel in the input image.
<b>Filtering Properties</b>	
Filtering Custom	Applies a Custom filter
Filtering Average	Applies an Average filter
Filtering Laplacian	Applies a Laplacian filter
Filtering HorizontalSobel	Applies a Horizontal Sobel filter
Filtering VerticalSobel	Applies a Vertical Sobel filter
Filtering Sharpen	Applies a Sharpen filter
Filtering SharpenLow	Applies a SharpenLow filter

<b>Object</b>	<b>Definition</b>
Filtering Horizontal Prewitt	Applies a Horizontal Prewitt filter
Filtering Vertical Prewitt	Applies a Vertical Prewitt filter
Filtering Gaussian	Applies Gaussian filter
Filtering HighPass	Applies High Pass filter
Filtering Median	Applies a Median filter
<b>Morphological Properties</b>	
Morphological Dilate	Sets each pixel in the output image as the largest luminance value of all the input image pixels in the neighborhood defined by the selected kernel size.
Morphological Erode	Sets each pixel in the output image as the smallest luminance value of all the input image pixels in the neighborhood defined by the selected kernel size.
Morphological Close	Has the effect of removing small dark particles and holes within objects
Morphological Open	Has the effect of removing peaks from an image, leaving only the image background
<b>Histogram Properties</b>	
Histogram Equalization	Equalization operation enhances the Input Image by flattening the histogram of the Input Image
Histogram Stretching	Stretches (increases) the contrast in an image by applying a simple piecewise linear intensity transformation based on the histogram of the Input Image.
Histogram Light Threshold	Changes each pixel value depending on whether they are less or greater than the specified threshold. If an input pixel value is less than the threshold, the corresponding output pixel is set to the minimum acceptable value. Otherwise, it is set to the maximum presentable value.

Object	Definition
Histogram Dark Threshold	Changes each pixel value depending on whether they are less or greater than the specified threshold. If an input pixel value is less than the threshold, the corresponding output pixel is set to the maximum presentable value. Otherwise, it is set to the minimum acceptable value.
<b>Transform Properties</b>	
Transform FFT	Converts and outputs a frequency description of the input image by applying a Fast Fourier Transform (FFT)
Transform DCT	Converts and outputs a frequency description of the input image by applying a Discrete Cosine Transform (DCT)

### **Related Topics**

[Configuring Image Processing Properties](#)

[Configuring Image Processing Properties - Advanced](#)

### ***Configuring Image Processing Properties***

Each Image Processing Tool in application performs a selected operation on an image, called the input image. An image processing operation can also involve another image or a constant, as well as set of processing parameters.

### **Image Types**

The tool can accept unsigned 8-bit, signed 16-bit, and signed 32-bit images as input. The processing is usually performed in the more defined type, based on input or operand image, or in a promoted type (signed 16-bit), if needed.

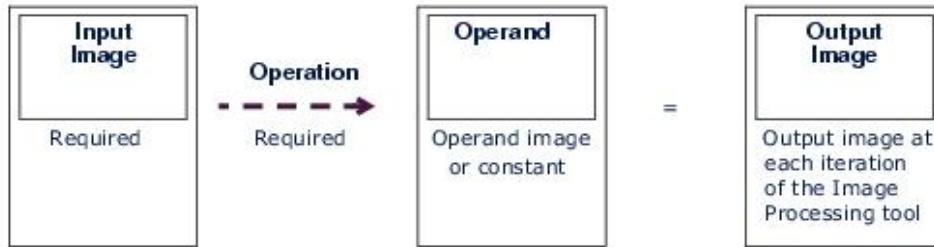
The Image Processing Tool output is of same type as the input image unless:

- the user overrides the type by setting another value or
- an output image already exists: the output image type remains the same unless otherwise specified.

### **Elements of an Operation**

An image processing operation requires at least one operand that is acted upon by an operation. For the Image Processing Tool, this first operand is always the Input image. Some operations require a second operand. This Operand can be an image, called the Operand image, or

a constant. The basic elements of an operation are shown in the following figure. Furthermore, some operations involve other parameters such as clipping, scaling, and filters. Such parameters are discussed under the category of operation to which they apply.



*Basic Elements of an Image Processing Operation*

## **Input Image**

An Input image is required as the first operand. The only type of operation that does not require an Input Image is an Assignment operation.

## **Operation**

The available operations are described in greater detail under the following sections: [Arithmetic Operations](#), [Assignment Operations](#), [Transform Operations](#), [Logical Operations](#), [Filtering Operations](#), [Morphological Operations](#), and [Histogram Operations](#).

## **Operand**

Operations that require a second operand can use either an Operand image or a constant.

### **Operand Image**

An Operand Image is used by an operation that acts on two images. If an Operand image is specified it will override the use of the constant specified for the operation.

- The Image Processing Tool applies logical and arithmetic operators, first to the Input image, secondly to the Operand image.
- The Image Processing Tool can accept unsigned 8-bit, signed 16-bit ,and signed 32-bit images as an Operand image.

### **Constant**

Any constant value specified for an operation will be overridden by an Operand image that has been defined for the operation.

## **Output Image**

The output image is the image resulting from an image processing operation.

- The user can specify the type of output images as either unsigned 8-bit, signed 16-bit, or signed 32-bit images.
- ACE Sight processes other than the Image Processing Tool can only take unsigned 8-bit images as input.

### ***Arithmetic Operations***

Arithmetic operations are performed by promoting the input values of the source pixels (from the Input image) and the Operand values to the highest resolution of the Input Image, Operand image or desired output image type. The results of the operation are converted according to the following rules:

- Destination pixel value = ClipMode (Result \* Scale)
- Destination pixel value is truncated as necessary

## **Clipping Modes**

Two clipping modes are available for arithmetic operations: normal and absolute.

### **Normal Clipping Mode**

Normal Clipping mode forces the value of a destination pixel to a value from 0 to 255 for unsigned 8-bit images, to a value from -327678 to 32767 for signed 16-bit images, or to a value from -2,147,483,648 to 2,147,483,647 for signed 32-bit images. Values that are less than the specified minimum value are set to the minimum value. Values greater than the specified maximum value are set to the maximum value.

### **Absolute Clipping Mode**

The absolute clipping mode takes the absolute value of the result and clips it using the same algorithm as for Normal Clipping mode.

## **Arithmetic Operation Modes**

There are two Arithmetic operation modes. In the first, the operation is applied to every pixel of an input image and the corresponding pixel in the Operand image. The result is written in the output image.

In the second mode, the operand is a constant, and it is used on every pixel of the input image and the result is written in the output image.

The Image Processing Tool supports the following arithmetic operations: Addition, Subtraction, Multiplication, Division, Lightest and Darkest.

## Addition

The operand value (constant or Operand image pixel) is added to the corresponding pixel in the input image. The result is scaled and clipped, and finally written to the output image.

## Subtraction

The operand value (constant or Operand image pixel) is subtracted from the corresponding pixel in the input image. The result is scaled and clipped, and finally written to the output image.

## Division

The input image pixel value is divided by the operand value (constant or corresponding Operand image pixel). The result is scaled and clipped, and finally written to the output image.

## Multiplication

The input image pixel value is multiplied by the operand value (constant or corresponding Operand image pixel). The result is scaled and clipped, and finally written to the output image.

## Lightest (Maximum)

The operand value (constant or Operand image pixel) and corresponding pixel in the input image are compared to find the maximal value. The result is scaled and clipped, and finally written to the output image.

## Darkest (Minimum)

The operand value (constant or Operand image pixel) and corresponding pixel in the input image are compared to find the minimal value. The result is scaled and clipped, and finally written to the output image.

## *Assignment Operations*

Assignment operations promote the input values of the source pixels and the Operand values to the more defined type, based on the input image, the Operand image or the desired output image type. This type of operation does not support scaling or clipping. The Image Processing Tool provides the following assignment operations: Initialization, Copy and Inversion.

### Initialization

All the pixels of the output image are set to a specific constant value. The height and width of the output image must be specified.

### Copy

Each input image pixel is copied to the corresponding output image pixel.

---

## **Inversion**

The input image pixel value is inverted and the result is copied to the corresponding output image pixel.

### ***Transform Operations***

Transform operations convert and output a frequency description of the input image. The available operations are a Fast Fourier Transform (FFT) and a Discrete Cosine Transform (DCT). These transforms can be output as 1D Linear, 2D Linear, 2D Logarithmic or Histogram.

### ***Logical Operations***

There are two logical operation modes. In the first, the operation is applied to every pixel of an input image and the corresponding pixel in the Operand image. The result is written in the output image. In the second mode, the operand is a constant, and it is used on every pixel of the input image and the result is written in the output image. No scaling or clipping is supported for logical operations.

#### **AND**

The logical AND operation is applied using the operand value (constant or Operand image pixel) and the corresponding pixel in the input image. The result is written to the output image.

#### **NAND**

The logical NAND operation is applied using the operand value (constant or Operand image pixel) and the corresponding pixel in the input image. The result is written to the output image.

#### **NOR**

The logical NOR operation is applied using the operand value (constant or Operand image pixel) and the corresponding pixel in the input image. The result is written to the output image.

#### **OR**

The logical Or operation is applied using the Operand value (constant or Operand image pixel) and the corresponding pixel in the input image. The result is written to the output image.

#### **XOR**

The logical XOR operation is applied using the Operand value (constant or Operand image pixel) and the corresponding pixel in the input image. The result is written to the output image.

---

### ***Filtering Operations***

The filtering of an input image using a square, rectangular or linear kernel. The Image Processing Tool provides a set of defined filters as well as a custom filtering operation that apply a user-defined kernel.

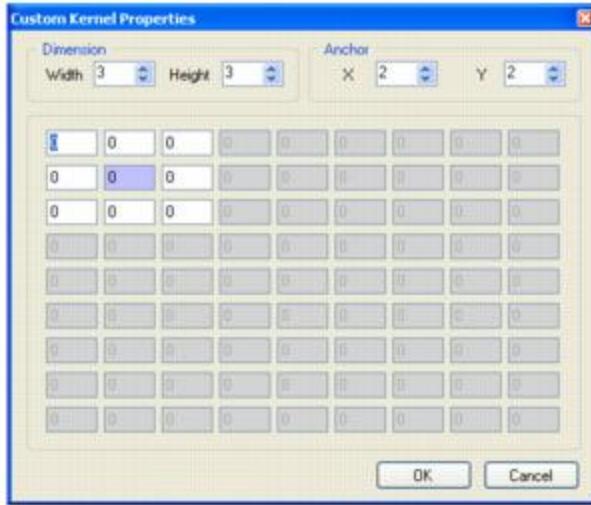
The predefined filters are: Average, Gaussian, Horizontal Prewitt, Vertical Prewitt, Horizontal Sobel, Vertical Sobel, High Pass, Laplacian, Sharpen, SharpenLow and Median.



*Example Of Image After Some Common Filtering Operations*

### **Creating A Custom Filter**

ACE Sight software enables the creation of a Custom Kernel for use in the Image Processing Tool.



*Custom Kernel Dialog of the Image Processing Tool*

To create a custom filter:

1. In the Image Processing Tool interface, expand the Advanced Parameters list.
2. Under Configuration, select the Operation parameter.
3. In the right-hand column, select: hsCustomFilter. This enables the tool to apply a the custom filter you will create in the next steps. Next you must create the custom filter.
4. Under Filtering, select the FilteringCustomKernel parameter.
5. In right-hand column click the Browse (...) icon. This opens the Custom Filter Properties dialog, as shown in the preceding figure.  
In the Dimensions box, enter values for Width and Height of the kernel. Grid boxes in white indicate kernel elements.
6. Enter the required in the value in each box of the kernel grid.
7. In the Anchor box, enter the X and Y positions of the kernel anchor, with respect to the defined kernel. The box indicating the anchor position is identified by a different color.

### Average Filter

The Average operation sets each pixel in the output image as the average of all the input image pixels in the neighborhood defined by the selected kernel size. This has the effect of blurring the image, especially edges.

---

The average filters are designed to remove noise. The kernel size can be 3, 5 or 7. The kernels used by the Image Processing Tool are shown in the following figure.

1	1	1
1	1	1
1	1	1

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

*Average Filtering Kernels*

### Gaussian Filter

The Gaussian operation acts like a low pass filter. This has the effect of blurring the image. Gaussian filters are designed to remove noise. The kernel size can be 3, 5 or 7. The kernels used by the Image Processing Tool are shown in the following figure.

1	2	1
2	4	2
1	2	1

2	7	12	7	2
7	31	52	31	7
12	52	127	52	12
7	31	52	31	7
2	7	12	7	2

1	1	2	2	2	1	1
1	3	4	5	4	3	1
2	4	7	8	7	4	2
2	5	8	10	8	5	2
2	4	7	8	7	4	2
1	31	4	5	4	3	1
1	1	2	2	2	1	1

*Gaussian Filtering Kernels*

### Horizontal Prewitt Filter

The Horizontal Prewitt operation acts like a gradient filter. This has the effect of highlighting horizontal edges (gradients) in the image. The kernel size used is 3 and it is shown in the following figure. The absolute clipping method is usually used with this filtering operation.

1	1	1
0	0	0
-1	-1	-1

*Horizontal Prewitt Filtering Kernel*

### **Vertical Prewitt Filter**

The Vertical Prewitt operation acts like a gradient filter. This has the effect of highlighting vertical edges (gradients) in the image. The kernel size used is 3 and it is shown in the following figure. The absolute clipping method is usually used with this filtering operation.

-1	0	1
-1	0	1
-1	0	1

*Vertical Prewitt Filtering Kernel*

### **Horizontal Sobel Filter**

The Horizontal Sobel operation acts like a gradient filter. This has the effect of highlighting horizontal edges (gradients) in the image. The kernel size can be 3, 5 or 7. The absolute clipping method is usually used with this filtering operation. The kernels used by the Image Processing Tool are shown in the following figure.

1	2	1
0	0	0
-1	-2	-1

1	4	7	4	1
2	10	17	10	2
0	0	0	0	0
-2	-10	-17	-10	-2
-1	-4	-7	-4	-1

1	4	9	13	9	4	1
3	11	26	34	26	11	3
3	13	30	40	30	13	3
0	0	0	0	0	0	0
-3	-13	-30	-40	-30	-13	-3
-3	-11	-26	-34	-26	-11	-3
-1	-4	-9	-13	-9	-4	-1

*Horizontal Sobel Filtering Kernels*

### Vertical Sobel Filter

The Vertical Sobel operation acts like a gradient filter. This has the effect of highlighting vertical edges (gradients) in the image. The kernel size can be 3, 5 or 7. The absolute clipping method is usually used with this filtering operation. The kernels used by the Image Processing Tool are shown in the following figure.

-1	0	1
-2	0	2
-1	0	1

-1	-2	0	2	1
-4	-10	0	10	4
-7	-17	0	17	7
-4	-10	0	10	4
-1	-2	0	2	1

-1	-3	-3	0	3	3	1
-4	-11	-13	0	13	11	4
-9	-26	-30	0	30	26	9
-13	-34	-40	0	40	34	13
-9	-26	-30	0	30	26	9
-4	-11	-13	0	13	11	4
-1	-3	-3	0	3	3	1

*Vertical Sobel Filtering Kernels*

### High Pass

The High Pass operation acts like a circular gradient high pass filter. It essentially extracts high frequency details. This has the effect of highlighting all edges (gradients) in the image. The kernel size can be 3, 5 or 7. The absolute clipping method is usually used with this filtering operation. The kernels used by the Image Processing Tool are shown in the following figure.

-1	-1	-1
-1	8	-1
-1	-1	-1

-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	24	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1

-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1

*High Pass Filtering Kernels*

## Laplacian Filter

The Laplacian operation also acts like a circular gradient filter. This has the effect of highlighting all edges (gradients) in the image. The kernel size can be 3, 5 or 7. The absolute clipping method is usually used with this filtering operation. The kernels used by the Image Processing Tool are shown in the following figure.

-1	-1	-1
-1	8	-1
-1	-1	-1

-1	-3	-4	-3	-1
-3	0	6	0	-3
-4	6	20	6	-4
-3	0	6	0	-3
-1	-3	-4	-3	-1

-2	-3	-4	-6	-4	-3	-2
-3	-5	-4	-3	-4	-5	-3
-4	-4	9	20	9	-4	-4
-6	-3	20	36	20	-3	-6
-4	-4	9	20	9	-4	-4
-3	-5	-4	-3	-4	-5	-3
-2	-3	-4	-6	-4	-3	-2

*Laplacian Filtering Kernels*

## Sharpen Filter

The Sharpen operation sets each pixel in the output image as the subtraction of the average of all the input image pixels in the neighborhood defined by the selected kernel size. This has the effect of sharpening the image, especially edges. The kernel size can be 3, 5 or 7. The kernels used by the Image Processing Tool are shown in the following figure.

-1	-1	-1
-1	9	-1
-1	-1	-1

-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	25	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1

-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1

*Sharpen Filtering Kernels*

## SharpenLow Filter

The SharpenLow operation has the effect of sharpening and smoothing the image at the same time. The kernel size can be 3, 5 or 7. The kernels used by the Image Processing Tool are shown in the following figure.

(1/8) *	-1 -3 -4 -3 -1	-3 0 6 0 -3	-2 -3 -4 -6 -4 -3 -2
-1 -1 -1	-4 6 40 6 -4	-4 -4 9 20 9 -4 -4	-3 -5 -4 -3 -4 -5 -3
-1 16 -1	-3 0 6 0 -3	-6 -3 20 72 20 -3 -6	-4 -4 9 20 9 -4 -4
-1 -1 -1	-1 -3 -4 -3 -1	-3 -5 -4 -3 -4 -5 -3	-2 -3 -4 -6 -4 -3 -2

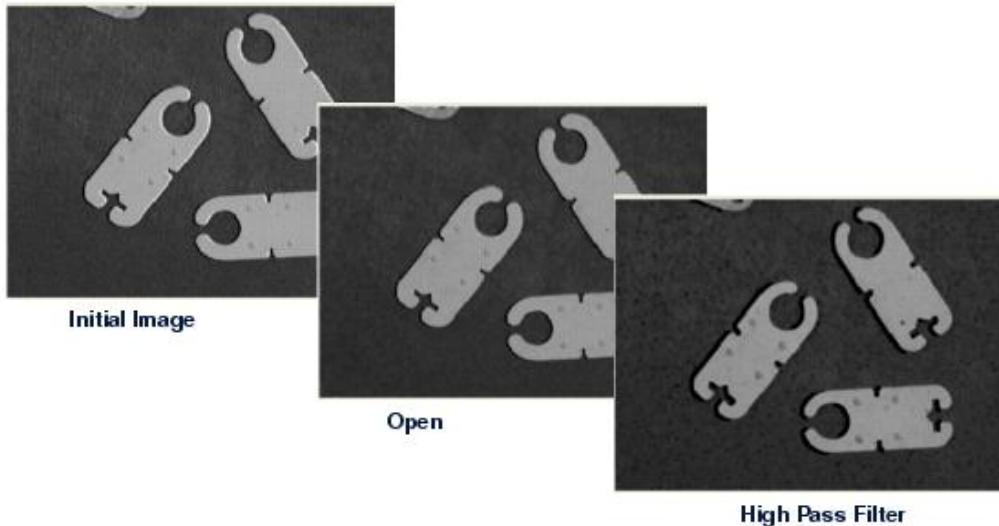
*SharpenLow Filtering Kernels*

## Median

The Median operation sets each pixel in the output image as the median luminance of all the input image pixels in the neighborhood defined by the selected kernel size. This has the effect of reducing impulsive image noise without degrading edges or smudging intensity gradients.

### **Morphological Operations**

Morphological operations are used to eliminate or fill small and thin holes in objects, break objects at thin points or connect nearby objects. These operations generally smooth the boundaries of objects without significantly changing their area. The Image Processing Tool provides the following predefined morphological operations, each of which can only be applied to a 3x3 neighborhood: Dilate, Erode, Close and Open.



*Examples Of Resulting Images After Some Common Morphological Operations*

### **Dilate**

The Dilate operation sets each pixel in the output image as the largest luminance value of all the input image pixels in the neighborhood defined by the selected kernel size. (Currently fixed to 3x3)

### **Erode**

The Erode operation sets each pixel in the output image as the smallest luminance value of all the input image pixels in the neighborhood defined by the selected kernel size. (Fixed to 3x3)

### **Close**

The Close operation is equivalent to a Dilate operation followed by an Erode operation. This has the effect of removing small dark particles and holes within objects.

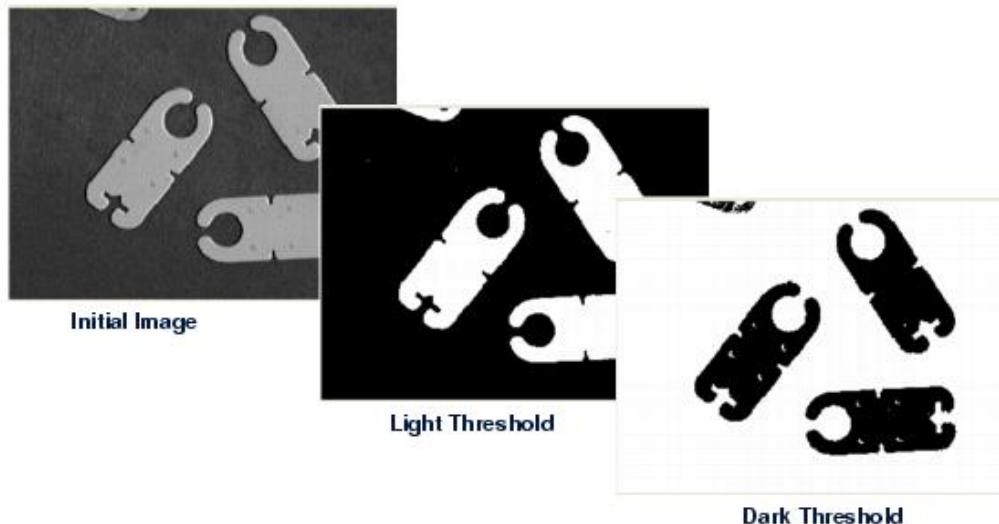
### **Open**

The Open operation is equivalent to an Erode operation followed by a Dilate operation. This has the effect of removing peaks from an image, leaving only the image background.

### **Histogram Operations**

The action of a histogram operation depends on the histogram of the Input Image. The Image Processing Tool provides the following histogram operations, each of which can only be

applied to an unsigned 8-bit image: Equalization, Stretching, Light Threshold and Dark Threshold.



*Example Of Image After Some Common Histogram Operations*

### **Equalization**

The Equalization operation enhances the Input Image by flattening the histogram of the Input Image.

### **Stretching**

The Stretching operation stretches (increases) the contrast in an image by applying a simple piecewise linear intensity transformation, based on the histogram of the input image.

### **Light Threshold**

The Light Threshold operation changes each pixel value depending on whether they are less than or greater than the specified threshold. If an input pixel value is less than the threshold, the corresponding output pixel is set to the minimum acceptable value. Otherwise, it is set to the maximum acceptable value.

### **Dark Threshold**

The Dark Threshold operation changes each pixel value depending on whether they are less than or greater than the specified threshold. If an input pixel value is less than the threshold, the corresponding output pixel is set to the maximum presentable value. Otherwise, it is set to the minimum acceptable value.

---

### **Related Topics**

[Image Processing](#)

[Configuring Image Processing Properties - Advanced](#)

### **Configuring Image Processing Properties - Advanced**

The Advanced Properties section of the Image Processing Tool interface provides access to advanced Image Processing Tool parameters and properties.

#### **Arithmetic Properties**

This is used to set the parameters for an arithmetic operation.

Arithmetic operations are applied in the following manner, depending on the type of operand.

- If the operand is an Operand Image, the operation is applied to every pixel of an input image and the corresponding pixel in the Operand Image. The result is written in the output image.
- If the operand is a constant, the constant is applied on every pixel of the input image and the result is written in the output image.

#### **Arithmetic Constant**

Arithmetic Constant defines a constant that is applied as an operand by an arithmetic operator. This constant is applied only when no valid Operand Image is specified.

#### **Arithmetic Scale**

Arithmetic Scale sets the scaling factor applied by an arithmetic operation. After the operation has been applied, the value of each pixel is multiplied by the Arithmetic Scale value.

#### **Assignment Properties**

This is used to set the properties for an assignment operation.

Assignment operations promote the input values of the source pixels and the Operand values to the more defined type, based on the input image, the Operand image or the desired output image type. This type of operation does not support scaling or clipping

#### **Arithmetic Constant**

Arithmetic Constant defines constant that applied as an operand by an arithmetic operation. This constant is applied only when no valid Operand Image is specified.

---

## Assignment Height

Read-only. Assignment Height is a constant value indicates sets the height, in pixels, of the output image.

## Assignment Width

Read-only. Assignment Width is a constant value that indicates the width, in pixels, of the output image.

### ***Configuration Parameters***

This is used to set the operation applied by the tool as well as parameters related to the type of image output by the Image Processing Tool.

**NOTE:** By default an output image is of same type as the input image, unless an output image of another type already exists. The output image type remains the same unless otherwise specified.

### ***Filtering Properties***

This is used to set the parameters for a filtering operation.

- Filtering operations do not require an operand.
- For more information on filters and the creating custom filters, see the [Filtering Operations](#) section.

### **Filtering Clipping Mode**

Filtering Clipping Mode sets the clipping mode applied by a filtering operation.

Two clipping modes are available: normal and absolute.

#### **Normal Clipping Mode**

Normal Clipping mode forces the value of a destination pixel to a value from:

- 0 to 255 for unsigned 8-bit images,
- -327678 to 32767 for signed 16-bit images,  
or
- -2,147,483,648 to 2,147,483,647 for signed 32-bit images.

Values that are less than the specified minimum value are set to the minimum value. Values greater than the specified maximum value are set to the maximum value.

### **Absolute Clipping Mode**

The absolute clipping mode takes the absolute value of the result and clips it using the same algorithm as for Normal Clipping mode.

### **Filtering Kernel Size**

Filtering Kernel Size sets the size of the kernel applied by a fixed (predefined) filtering operation.

### **Filtering Scale**

Filtering Scale sets the scaling factor applied by a filtering operation. After the operation has been applied, the value of each pixel is multiplied by the Filtering Scale value.

#### ***Histogram Properties***

This is used to set the parameters for a histogram operation.

Histogram operations can only be applied to an unsigned 8-bit image.

### **Histogram Threshold**

Histogram Threshold sets threshold value applied by a histogram thresholding operation

#### ***Logical Properties***

This is used to set the constant operand for a logical operation.

### **Logical Constant**

Logical Constant defines constant that applied as an operand by a logical operation. This constant is applied only when no valid Operand Image is specified.

#### ***Morphological Properties***

This is used to set the parameters for a morphological operation.

Morphological operations are used to eliminate or fill small and thin holes in objects, break objects at thin points or connect nearby objects. These operations generally smooth the boundaries of objects without significantly changing their area.

### **Morphological Neighborhood Size**

Morphological Neighborhood Size neighborhood is the neighborhood by a morphological operation. This value is currently fixed at 3x3. No other values are allowed.

### **Transform Properties**

Transform operations convert and output a frequency description of the input image. The available operations are a Fast Fourier Transform (FFT) and a Discrete Cosine Transform (DCT). These transforms can be output as 1D Linear, 2D Linear, 2D Logarithmic or Histogram.

### **Transform Flags**

Transform Flags sets the flag used by a transform operation, either FFT or DCT.

### **Image Processing Tool Results**

The Image Processing Tool outputs images that can be used by other vision tools. This tool does not output frame results.

### **Viewing Results**

The results for each execution of the tool are represented in the display Vision window.

### **Description of Image Processing Tool Results**

The Image Processing Tool outputs the following results:

#### **Elapsed Time**

The Elapsed Time is the total execution time of the Image Processing Tool.  
Elapsed Time is available at the bottom of the window in which the result is displayed.

#### **Related Topics**

[Image Processing](#)

[Configuring Image Processing Properties](#)

### **Image Sampling**

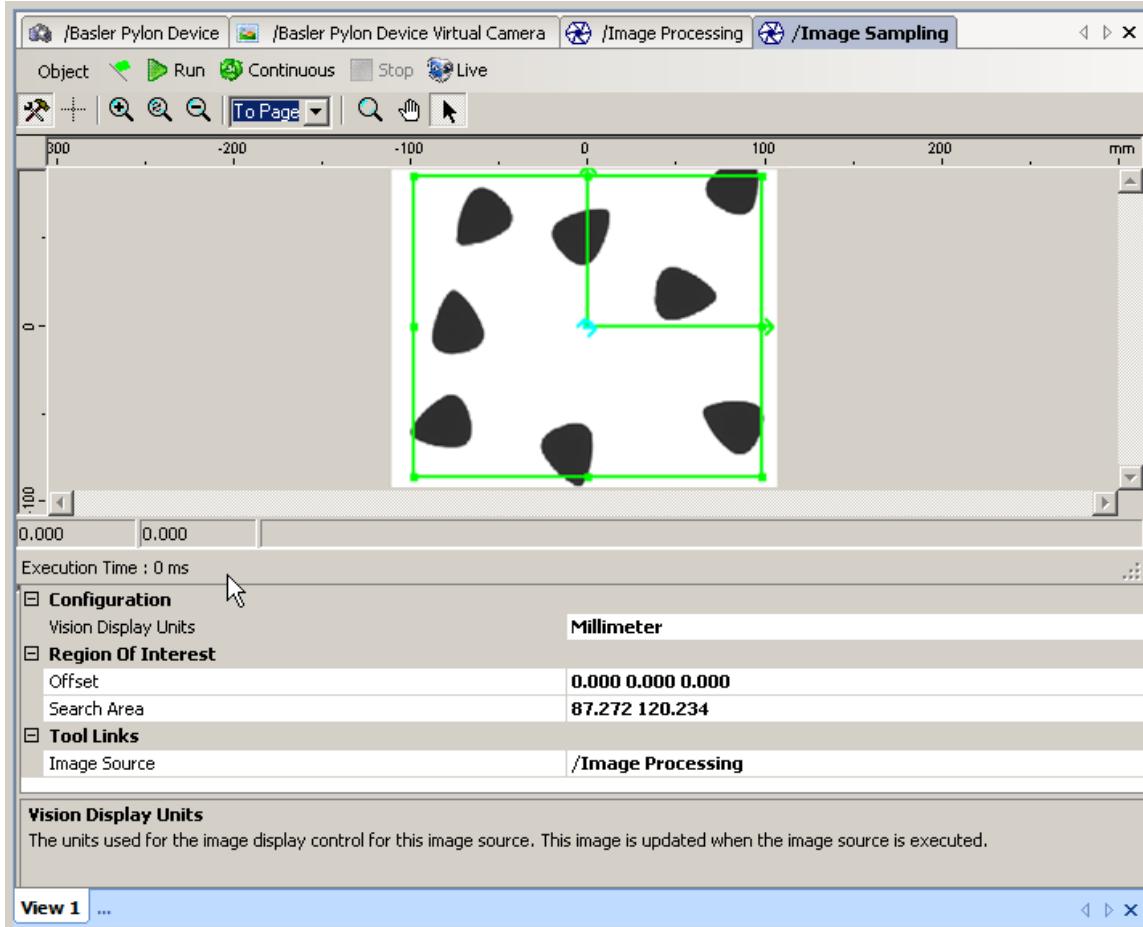
The Image Sampling Tool is used to extract an area of an image and output it as a separate image.

To create an Image Sampling tool, right-click in the Tree structure, then select:

**New > Vision > Tool > Image Sampling**

## Image Sampling

---



*Image Sampling Object Editor*

### **Configuration**

Object	Definition
<b>Configuration</b>	
Vision Display Units	Units used to display image source. Default is Millimeter.
<b>Region of Interest (ROI)</b>	
Offset	Specifies the center and rotation of the ROI. X, Y, Degrees.

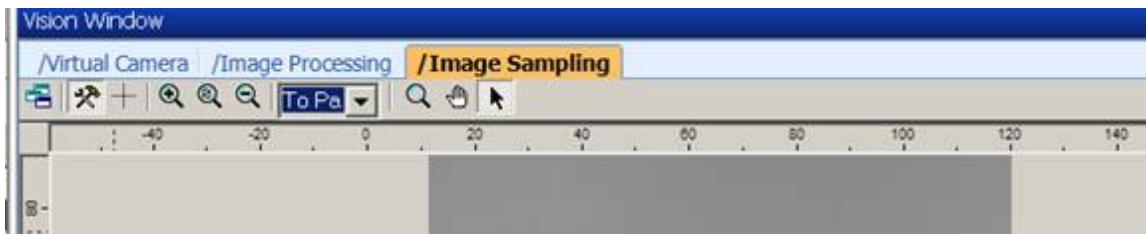
Object	Definition
Search Area	Defines the height and width of the region of interest.
<b>Tool Links</b>	
Image Source	Defines the image source used for processing by this vision tool.

### Results

The Image Sampling tool outputs images that can be used by other vision tools.

### Custom Display

When this tool is created, an image buffer is created in the vision window with the same name as the tool. (See the following figure.) The results of the vision tool are displayed in the buffer associated with this record.



## Image Sharpness

The Image Sharpness Tool computes the sharpness of major edges in a user-defined region of interest.

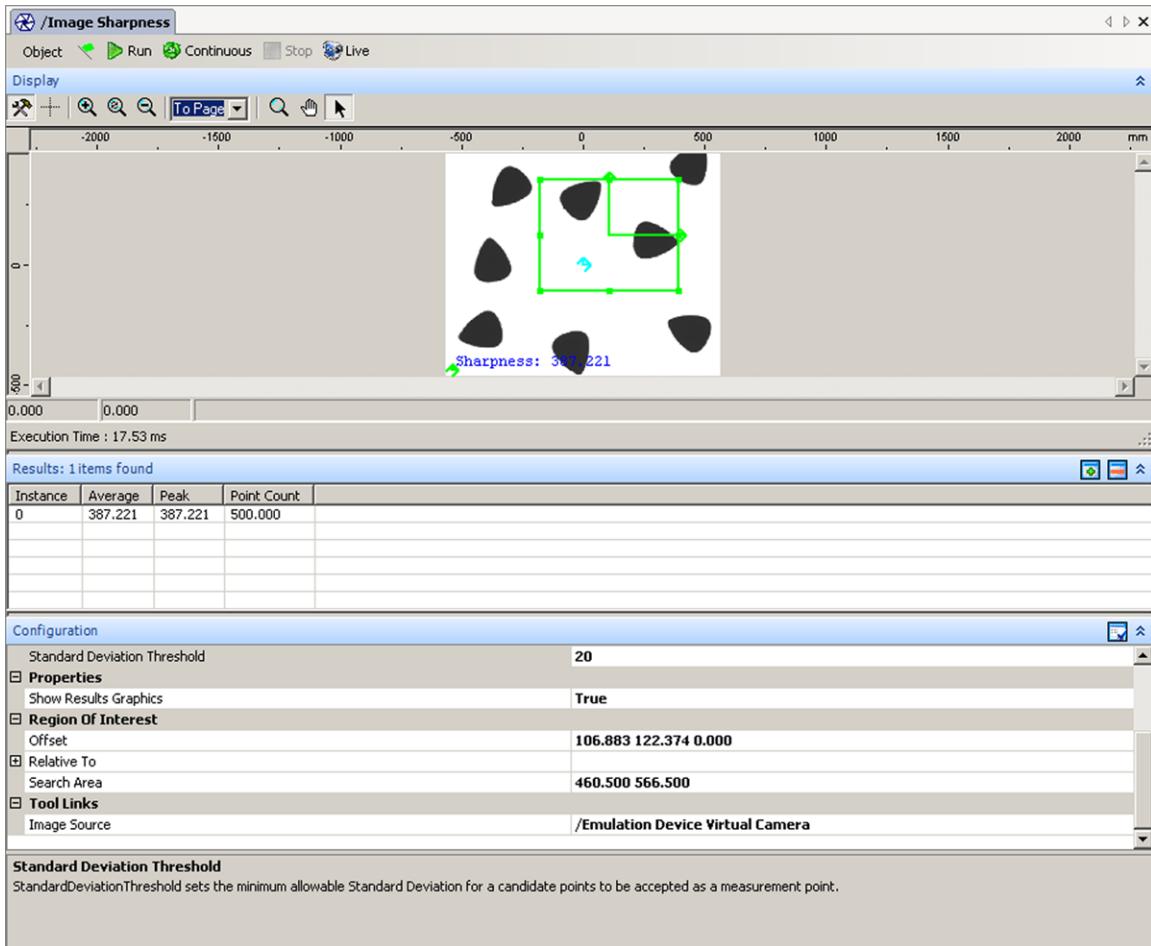
- Typical use of the Image Sharpness Tool is the verification or validation of the sharpness of an image before it is processed by other tools.
- This tool can also be used as a building block for implementing an auto-focus procedure, which consists of a motorized focus lens and uses the sharpness value to close the loop.

To create an Image Sharpness tool, right-click in the Tree structure, then select:

**New > Vision > Tool > Image Sharpness**

## Image Sharpness

---



*Image Sharpness Object Editor*

### **Input**

The Input required by the Image Sharpness tool is an image provided by another tool. Images can be provided by other tools, such as an Image Processing tool, if the purpose of the tool is to analyze the sharpness of processed image.

To set the Input:

1. Execute the tool once to make sure that an input image is available.
2. From the Image Source list under the Tool Links section, select the tool that will provide the input image.

**Configuration**

<b>Object</b>	<b>Definition</b>
<b>Properties</b>	
Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
<b>Tool Links</b>	
Image Source	Defines the image source used for processing by this vision tool.
<b>Region of Interest (ROI)</b>	
Offset	Specifies the center and rotation of the ROI. X, Y, Degrees.
Relative To	Specifies the tool that will provide the input to this tool. The tool can be dragged from the Folder pane and dropped into this field.
Search Area	Defines the height and width of the region of interest.
<b>Advanced Properties</b>	
Bilinear Interpolation Enabled	Specifies if bilinear interpolation is used to sample the input image. By default, bilinear interpolation is enabled because it improves accuracy.
Custom Candidate Points	Sets the maximum number of points that can be used by the tool to calculate the image sharpness.
Custom Sampling Step	When enabled, the tool uses the user-defined sampling step, instead of the default optimal sampling step, to sample the region of interest from the input image.
Kernel Size	Sets the size of the kernel of the operator for the sharpness process. The default setting of 5 (a 5x5 kernel) is generally sufficient for most cases.
Results Display Position	Location of Image Sharpness Average in the display pane.
Standard Deviation Threshold	Sets the minimum allowable Standard Deviation for a candidate point to be accepted as a measurement point.

**Results**

Object	Definition
Instance	Identifies the number of the Instance output by the Image Sharpness Tool.
Average	Average sharpness value calculated for the current instance.
Point Count	Number of points actually used to measure the average sharpness for the current region of interest.
Peak	Maximum average sharpness value computed by the tool

**NOTE:** Before configuring the Image Sharpness Tool, execute the tool at least once and verify that the tool is being positioned correctly in the image.

The display represents the region of interest of the Image Sharpness Tool as a green box.

**Positioning the Image Sharpness Tool**

Positioning the tool defines the area of the image that will be processed by the Image Sharpness Tool.

To position the Image Sharpness tool:

- A blue marker indicates the frame provided by the Frame Input tool. If there is more than one object in the image, make sure that you are positioning the bounding box relative to the object identified by a blue axes marker.
- Enter values in the Search Area parameter, or use the mouse to configure the bounding box in the display.

**Related Topics**

[Image Sharpness Results](#)

[Configuring Image Sharpness Properties - Advanced](#)

**Configuring Image Sharpness Properties - Advanced**

The Image Sharpness Tool operates by first identifying a set of points with high local grey-scale variations at various points and applying an autocorrelation method to calculate an average sharpness factor from these points.

---

Sampling refers to the procedure used by the tool for gathering values within the portion of the input image that is bounded by the tool region of interest. Two sampling parameters, the Sampling Step and Bilinear Interpolation, can be used to create the desired balance between speed and accuracy.

### Bilinear Interpolation

Bilinear Interpolation uses pixel averaging to improve the quality of the image. It is applied before the image is analyzed.

To ensure subpixel accuracy in inspection applications, Bilinear Interpolation should always be set to True (enabled). Non-interpolated sampling (Bilinear Interpolation disabled) should only be used in applications where the speed requirements are more critical than accuracy.

### Custom Sampling Step (1 - 100)

The Sampling Step is the step used by the tool to sample the area of the input image that is bounded by the tool region of interest. The sampling step represents the height and the width of a sampled pixel.

For applications where a more specific tradeoff between speed and accuracy must be established, the sampling step can be modified by enabling the Custom Sampling Step and setting the desired value.

Object	Definition
Default	Default is the best sampling step computed by the tool. It is based on the average size, in calibrated units, of a pixel in the image. This default sampling step is usually recommended. Default is automatically used by the tool if no other value is specified.
Custom	Custom Sampling Step lets you set a sampling step value other than the default. To select a custom sampling step, Enable must be checked. <ul style="list-style-type: none"><li>• Increasing the sampling step value reduces the tool accuracy and decreases the execution time.</li><li>• Reducing the sampling step can increase the tool accuracy but will also increase the execution time.</li></ul>

## Candidate Points

Candidate Points are the points with the highest local greyscale variation in the region of interest. These points are candidates at which a sharpness measurement will be made if the local variation is sufficient. The number of candidate points is, by default, automatically set by the tool based on size of the region of interest.

When the tool is executed, it first scans the region of interest and identifies many candidate points where the local standard deviation is the highest. It then evaluates the sharpness at each of the candidate location that has a local standard deviation above the Standard Deviation Threshold. The locations where the sharpness is actually measured become Measurement Points.

- Candidate points are, by default, set automatically by the tool. When the default, and recommended, Automatic setting is enabled, the tool uses 500 Candidate Points for a region of interest over 320x240 pixels in size. If the area is smaller than 320x240, the number of Candidate Points is equal to:  $\text{width} * \text{height} / (500 / 320 \times 240)$
- The number of candidate points can be set manually by entering a value for the Candidate Point Count parameter.

## Standard Deviation Threshold

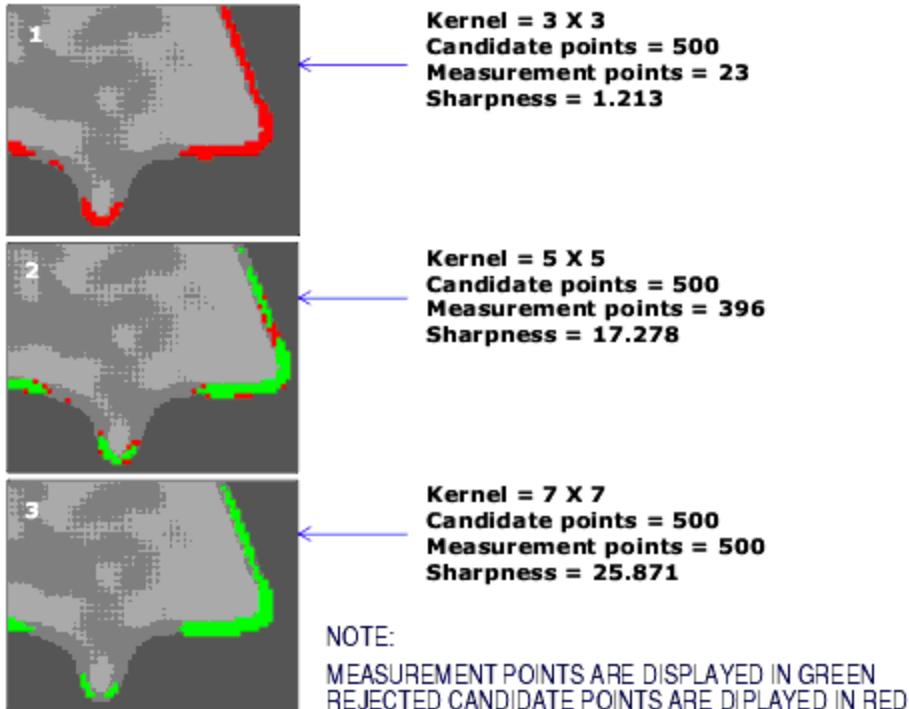
Standard Deviation Threshold sets the minimum standard deviation required for a Candidate Point to be used as a Measurement Point for calculating the average image sharpness.

When the tool is executed, it scans the region of interest and identifies candidate locations where the local standard deviation is the highest. The number of candidate locations is set by Custom Candidate Points. Points having a standard deviation equal to or above the threshold are used by the tool as the measurement points for calculating the average image sharpness.

## Kernel Size

Sets the size of the kernel of the operator for the sharpness process. The default kernel size of 5 is usually appropriate in typical applications. However, the Kernel should be larger than the number of pixels over which a typical contrast is spread.

A larger kernel may be used for blurrier images, for example, in the case where the blurriness is larger than the default kernel value. This is illustrated in the following figure, where the contrast is about 6-8 pixels wide. Note the difference in values obtained with different kernel sizes. With a 7x7 kernel, all the candidate points are used as measurement points. Larger kernels subsequently have almost no impact on the Sharpness result.



#### *Effect of Kernel Operator on the Sharpness Result*

A smaller kernel size, 2 or 3, for example, may be helpful for images with fine details or for images constituted of fine, high-frequency textures.

#### **Region of Interest (ROI)**

The region of interest is the area in which the tool carries out its process. Most tool position parameters can be set through the region of interest section of the tool interface. The following parameters define the tool region of interest.

#### **Offset**

Offset is the center of the ROI, defined by:

X	X coordinate of the center of the region of interest
Y	Y coordinate of the center of the region of interest
Degrees	Angle of rotation of the region of interest

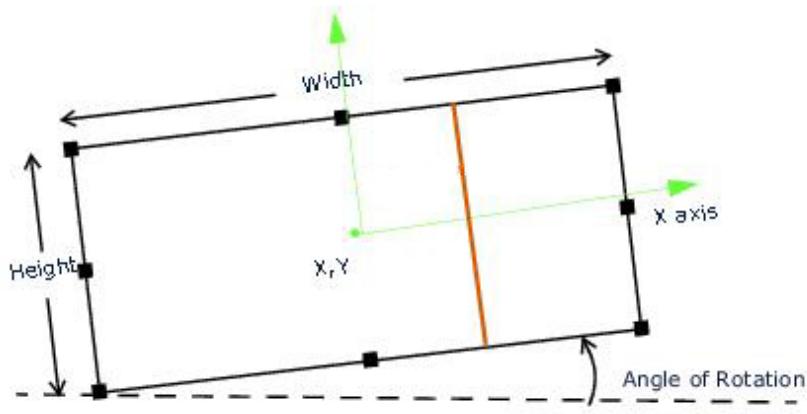
### Relative To

Relative To specifies the tool that will provide the input to this tool. The tool can be dragged from the Tree structure and dropped into this field.

### Search Area

Search Area is the size of the region of interest is defined by:

Height	Height of the region of interest
Width	Width of the region of interest



*Location Properties for the Region of Interest*

### Related Topics

[Image Sharpness](#)

[Image Sharpness Results](#)

### Image Sharpness Results

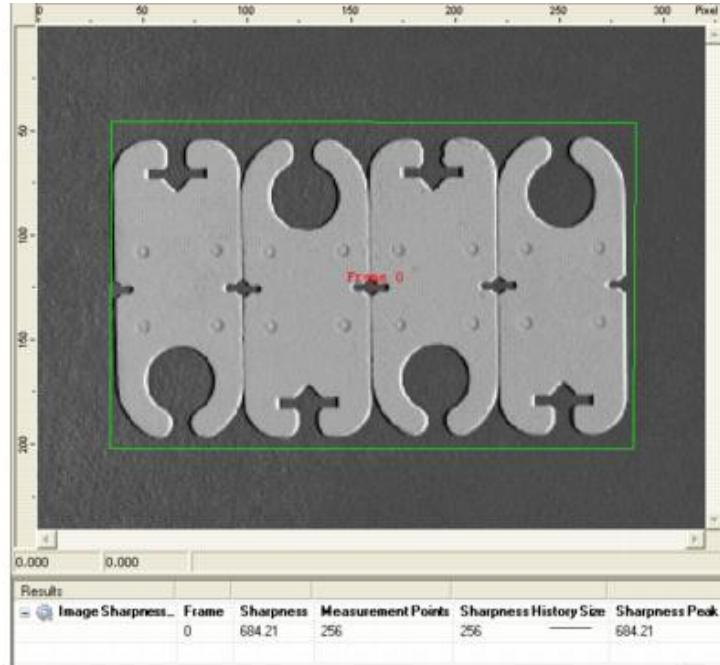
The Image Sharpness Tool outputs read-only results that provide statistical and general information.

### Display Window

The Display window shows the image being processed, the region of interest selected, and the locations of found instances. The elapsed time is displayed at the bottom of this window.

### Grid of Results

The grid of results presents the result values for all caliper measures found by the Image Sharpness Tool.



*Representation of Image Sharpness Tool Results in Display and Results Grid*

The Image Sharpness Tool outputs the following results:

Object	Definition
Elapsed Time	The Elapsed Time is the total execution time of the Image Sharpness Tool. Elapsed Time is not visible in the results grid, but is shown at the bottom of the Display window after each iteration of the Image Sharpness Tool.
Instance	Identifies the number of the Instance output by the Image Sharpness Tool. If the tool is frame-based, this number corresponds to the input frame that provided the positioning.

Object	Definition
Average	The average sharpness value calculated for the current instance. The sharpness value ranges from a maximum of 1000, indicating a very sharp image, to 0, indicating a very blurry image.
Point Count	Point Count is the number of points used to measure the average sharpness for the current region of interest. This can be less than the number of Candidate Points set in the Configuration panel.  Measurement Points are the points used to calculate the average Sharpness result. Only the Candidate Points that meet the Standard Deviation Threshold are retained as measurement points.
Peak	Peak is the maximum average sharpness value computed by the tool since the history was reset.

#### **Related Topics**

[Image Sharpness](#)

[Configuring Image Sharpness Properties - Advanced](#)

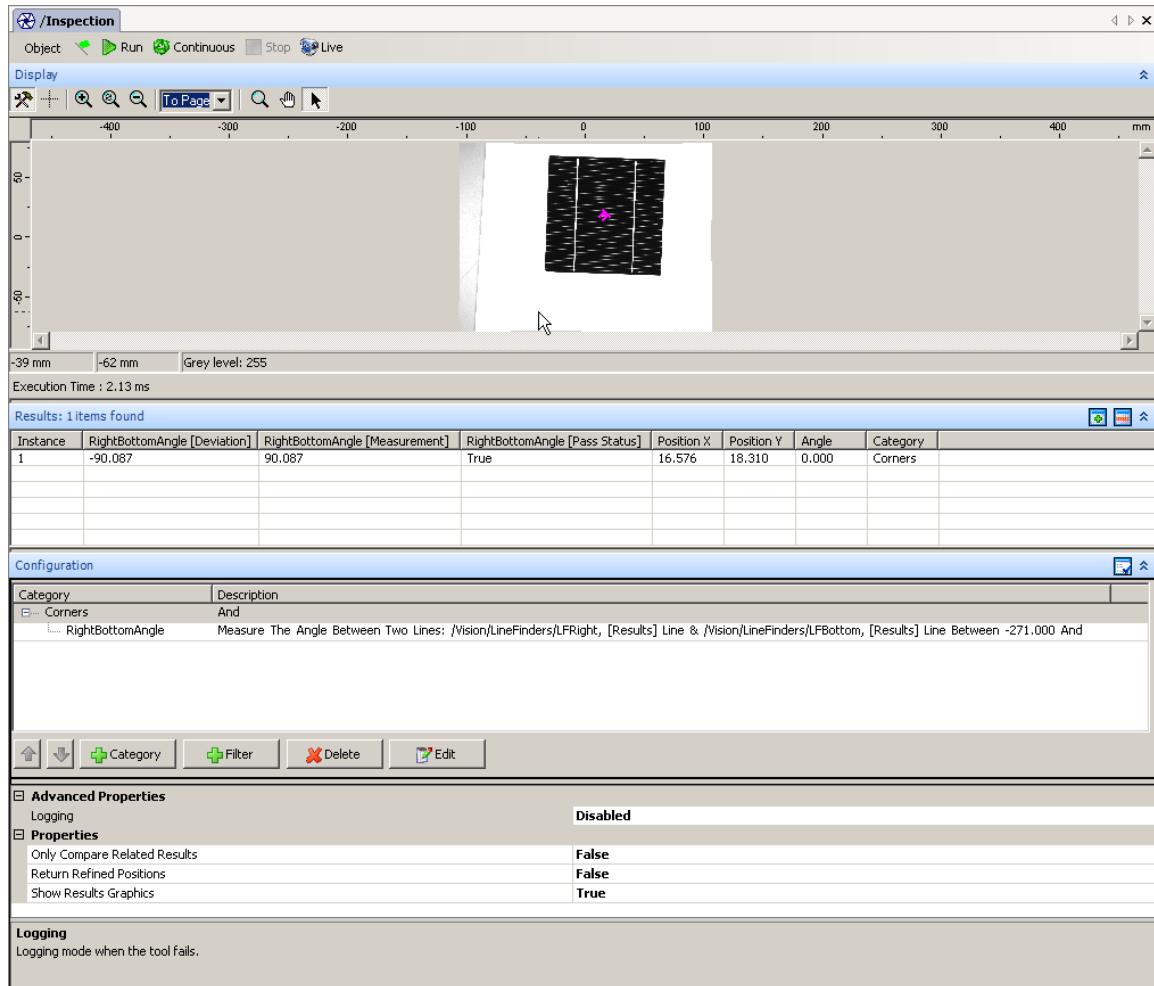
#### **Inspection**

The Inspection tool sorts instances based on the results of other tools and inspection filters.

To create an Inspection tool, right-click in the Tree structure, then select:

**New > Vision > Tool > Inspection**

## Inspection



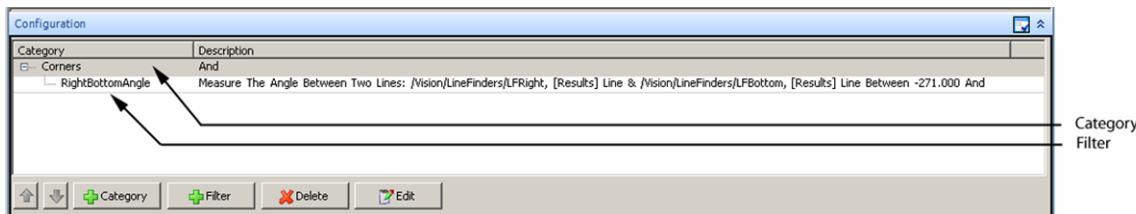
Inspection Tool Object Editor

### Configuration

Object	Definition
<b>Properties</b>	
Only Compare Related Results	When true, only related results will be compared against each other; otherwise all results are compared against each other.
Return Refined Positions	When true, return the transform for the last vision tool; otherwise return the first transform located.

Object	Definition
Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
<b>Advanced Properties</b>	
Image and Filter Logging	Logging mode when the Inspection fails.
Results Logging	Specifies if a results log will be generated. A csv extension generates csv format.

This tool has a modified configuration section. Each inspection filter is displayed in this section:



*Inspection Category with one Filter, shown in Configuration Section*

	Move selected category up or down to determine the order in which they will be used to sort instances. Greyed-out when not applicable (top and bottom)
	Add a new category
	Add a new filter and open the filter editor (a category or filter must be selected)
	Delete the selected category or filter (deleting a category deletes all of its filters)
	Edit the selected category or filter (double-clicking also opens the appropriate editor)

## Categories

The purpose of categories is to sort instances based on the filters in each category. An instance is put into the first category that it passes the filters for. For each instance, categories are evaluated starting with the category listed at the top of the Configuration section.

Each category has a name and an operator associated with it. The operator (AND or OR) is applied to all of the filters within the category to determine which instances are put in that category. The operator associated with a category is displayed in the Description field of the Configuration section, shown in the preceding figure.

A Default Category with an AND operator is created when the tool is created.

- If the category has an AND operator, an instance must pass every filter in the category.
- If the category has an OR operator, the instance only needs to pass one of the category's filters.
- The category to which an instance is assigned is listed in the Category column of the Results section.

If an instance fails all categories, the Results section displays Unassigned in the Category column.

You can change the name and operator of a category by clicking Edit when the category is selected, or by double-clicking the category name.

### Up/Down Arrows ()

Because instances are put into the first category they qualify for, it makes the order of the categories in the Configuration section important. The Up and Down arrows are used to move the categories up or down relative to the other categories.

- If the selected category or filter is at the top of its list, the Up-Arrow will be greyed-out. For the bottom category, the Down-Arrow will be greyed-out.
- Although the arrows move the filters, too, the order of filters within a category is not significant.
- The arrows do not move filters between categories.

**NOTE:** The location results associated with each category are exposed to ACE Sight in a separate frame index for each category. The first category is associated with frame 1, the second category with frame 2, and so on.

### Filters

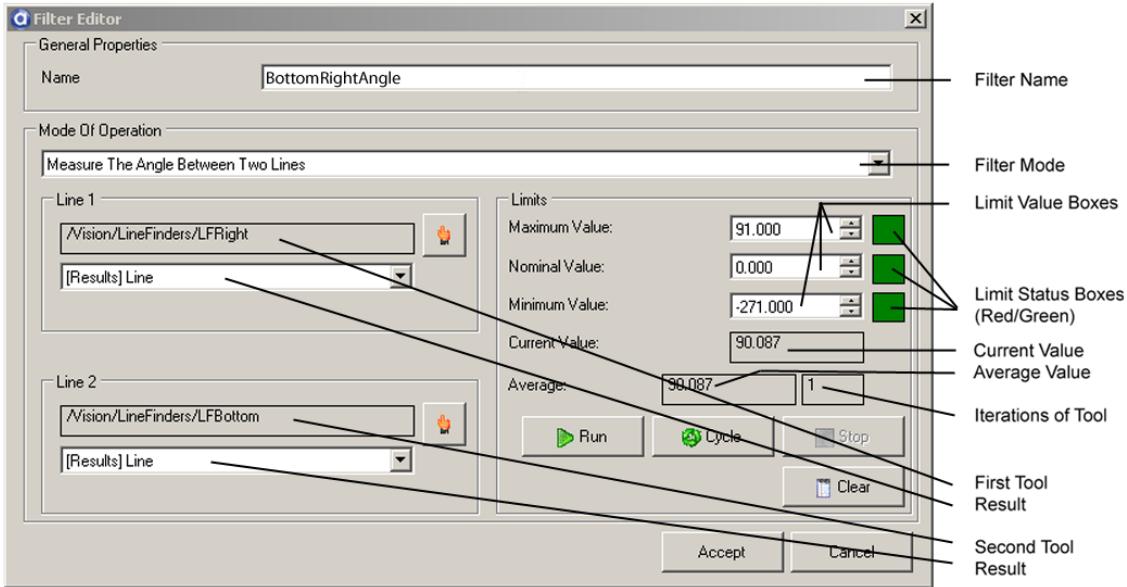
Each filter has a name and belongs to a category. You cannot add a filter until a category has been added, and either the category or a filter in the category is selected (+Filter will be greyed-out).

Filters define the inspection/comparison that will be performed. You can define multiple filters within each category. The inspection performed by a filter is displayed in the Description field of the Configuration section, shown in [Inspection Tool Object Editor](#).

A filter can be modified by selecting the filter and clicking Edit, or by double-clicking the filter in the Configuration section. When a filter is added or edited, the following editing form is displayed:

## Inspection

---



*Inspection Filter Editor*

### Name

Select a name for this filter.

### Mode of Operation

This defines what the filter does. The available modes are:

- Measure the distance between two points
- Measure the distance between a point and a line
- Measure the angle between two lines
- Test the value of a vision result variable
- Vision tool transform results

The data requested in the rest of this form will vary depending on the Mode selected.

### Vision Tools

Once the mode is selected, you should select the appropriate vision tools and properties on the left side of the form. Only tools that provide appropriate

results for the selected mode will be displayed. Some modes require just one vision tool.

### Limits

You set the minimum and maximum limits, and nominal value, to determine how the filter behaves. You can run the tool from this form. When the tool runs, it will show the current and average values for instances that pass the filter.

- Minimum and Maximum determine whether an instance passes or fails the filter.
- Nominal is used in calculating the deviation result. It does not influence passing or failing.

### Average/Current Values

The tool can be run once or repetitively, (by clicking Cycle), from within the filter editor, to test the values set for the limits. This lets you determine the appropriate values for the limits.

- Average is the average value obtained from the tool since Average was last cleared.
- The box to the right of Average displays how many iterations of the tool were used to obtain the average.
- Current Value displays the most recent result of the tool.

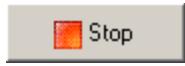
You can click the buttons that follow to run the tool once, run continuously, stop a continuous run, or clear the Average value (so you can start another continuous run).



Run the tool once



Run the tool continuously



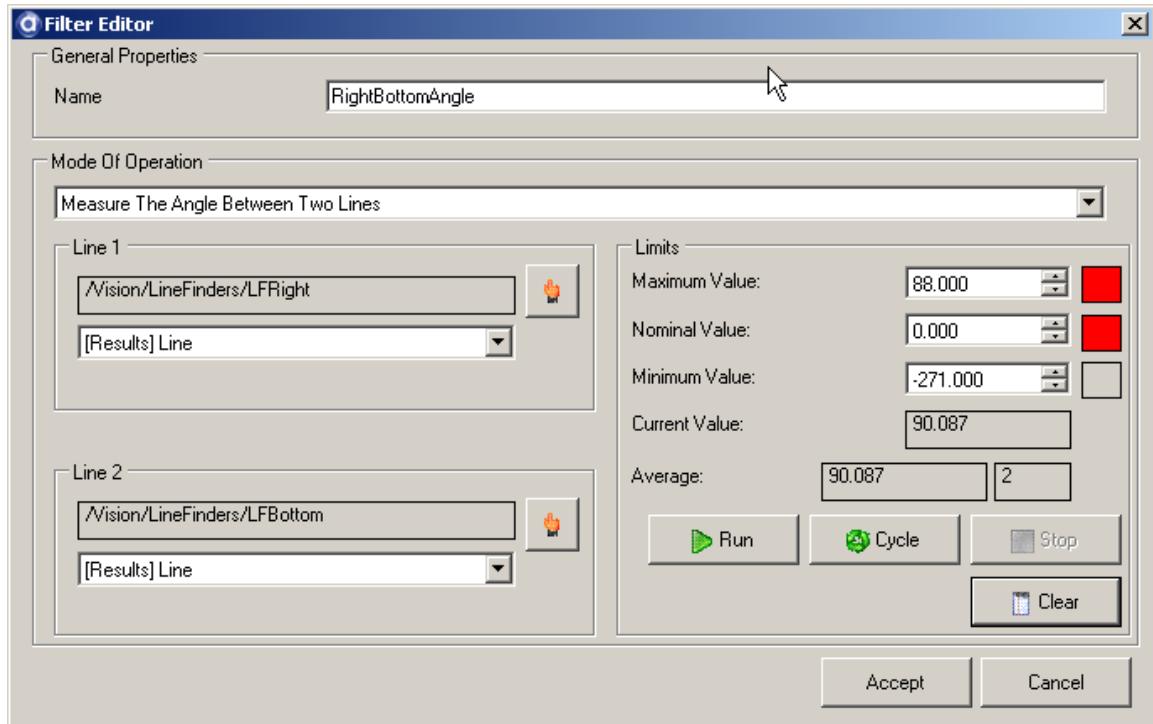
Stop the tool from running



Reset the average value and iteration count

When the tool runs, the current value will be compared against the filter limits.

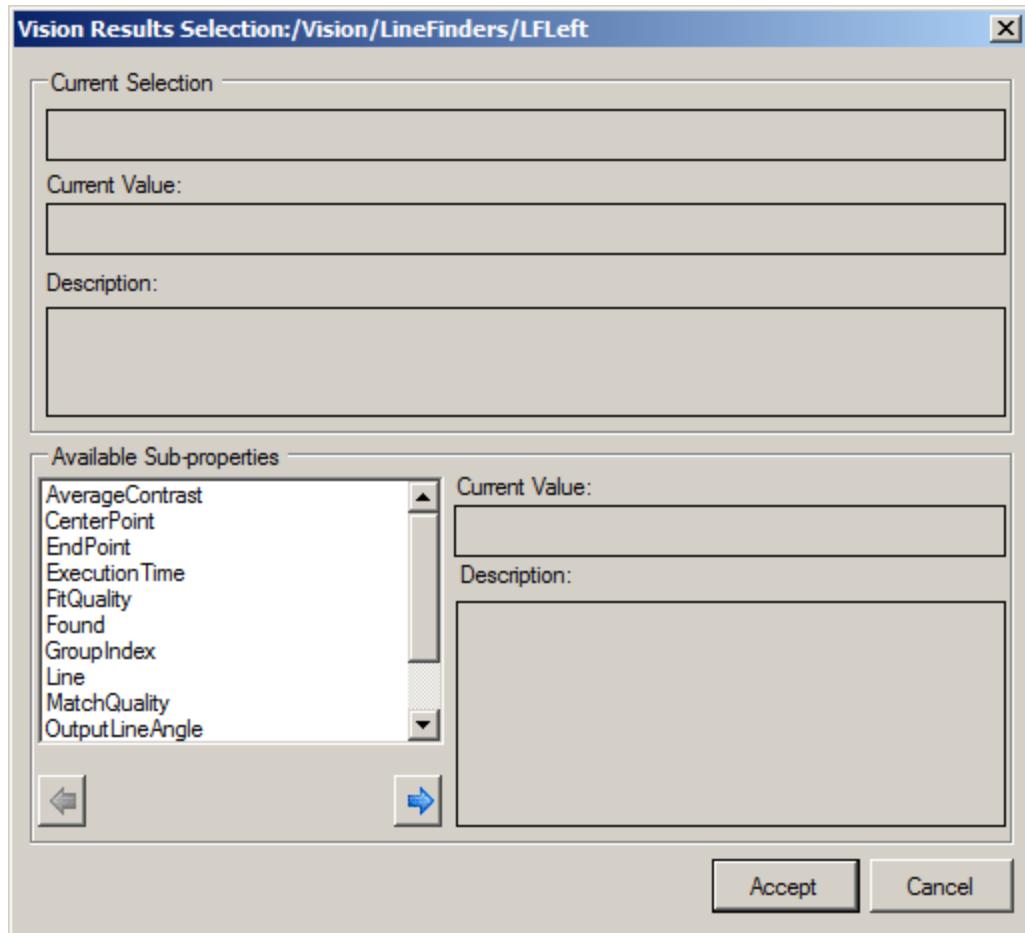
- If the current value is between the limits, the limit indicator boxes will be green.
- If the current value is out of range, the indicator for the limit that was exceeded will change to red, and the other limit indicator will be blank. See the following figure.
- If either limit is exceeded, the Nominal indicator will be red.



*Inspection Filter Editor, with Maximum Limit Exceeded*

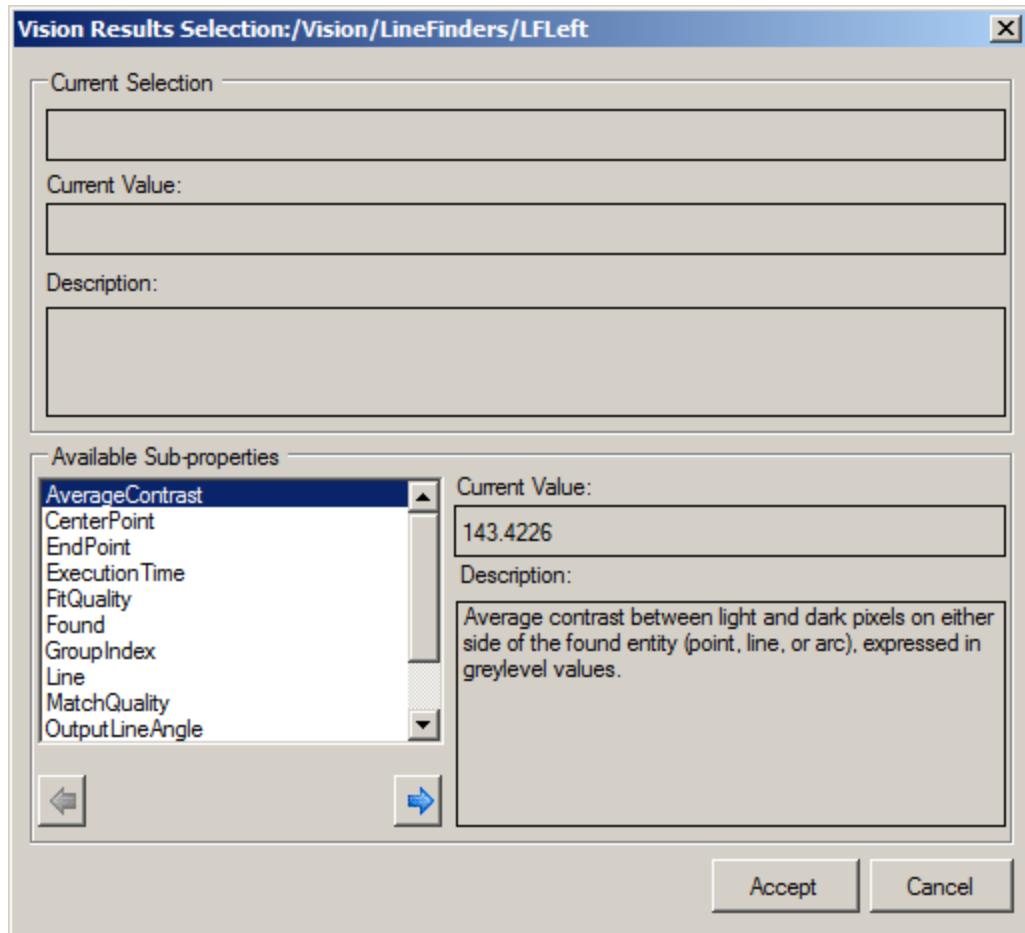
### **Filters Sub-properties Configuration**

If you select Test The Value Of A Vision Result Variable as your Mode Of Operation, you will be asked for a Vision Tool and a Result. The Vision Tool uses the same selection method as other Modes Of Operation, but the Result field has its own window. The window will first display the available sub-properties that can be used for testing.



*Available Sub-properties Window*

Highlighting any of the sub-properties will display a description of that sub-property:

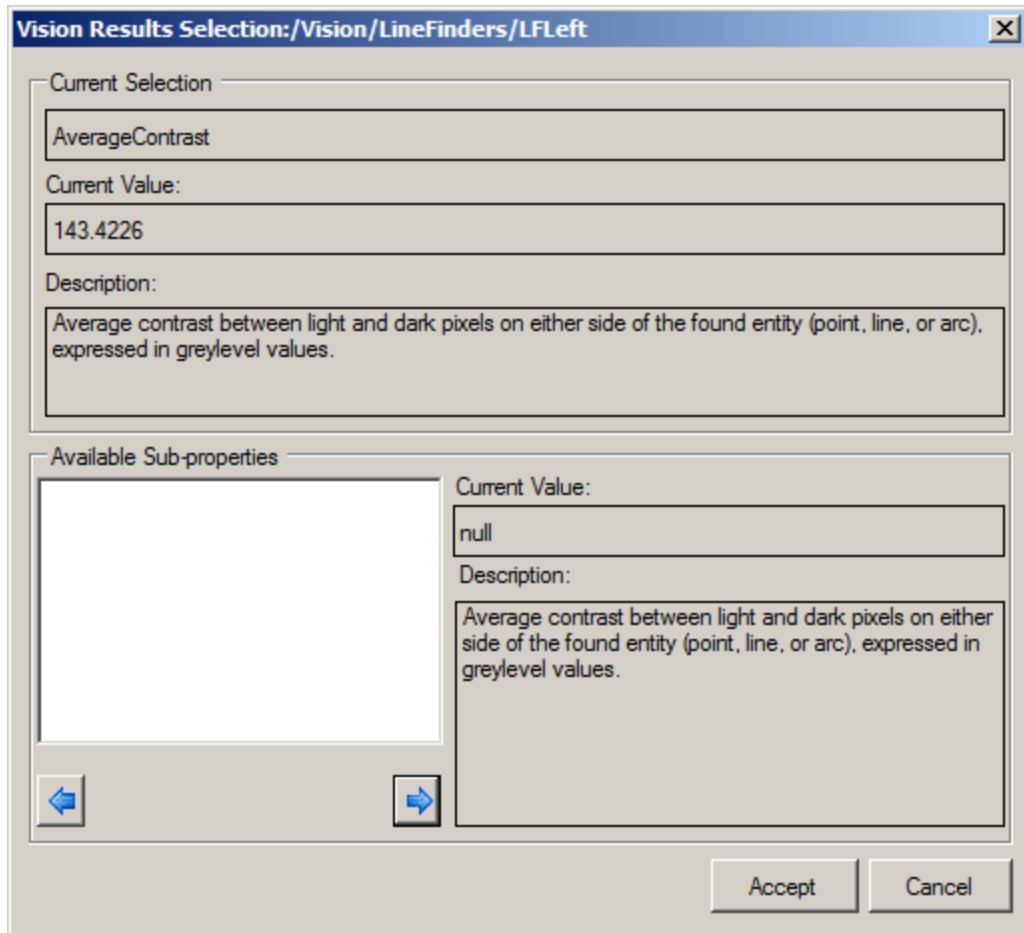


*Sub-property Description*

When you have highlighted the sub-property that you want to test, click the blue right-arrow to select it.

## Inspection

---



*Sub-property Selected*

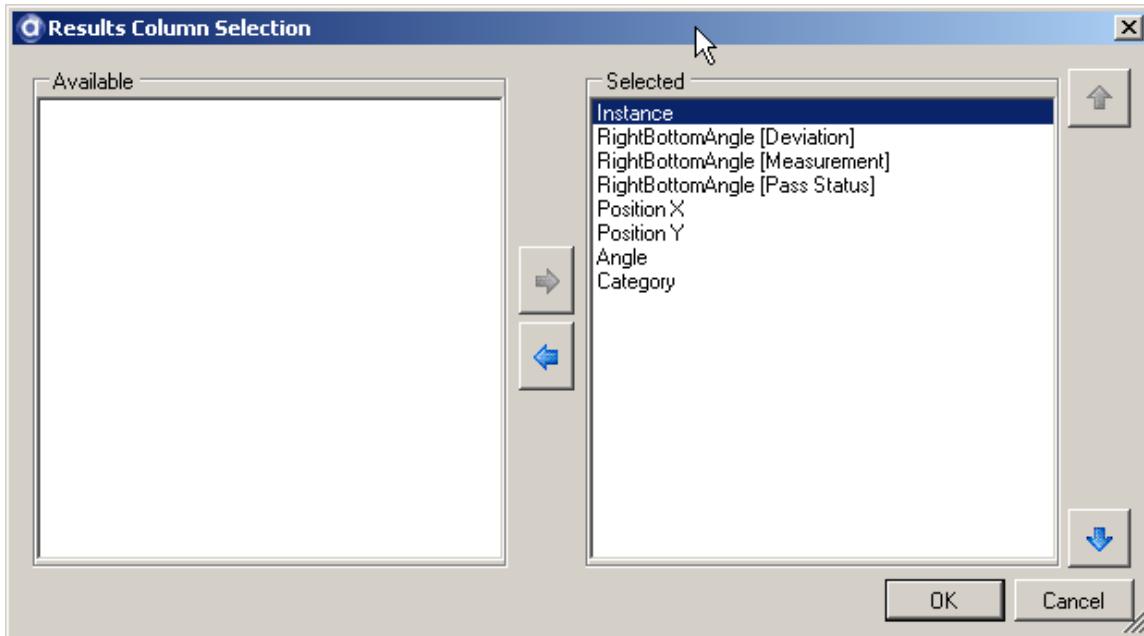
The fields above the available sub-properties pane will be filled in. Click Accept to accept your choice and close the window.

### Results

Object	Definition
Instance	The index of the located instance
<filter> [Deviation]	Deviation for this instance from the named filter
<filter> [Measurement]	Measurement for this instance with the named filter
<filter> [Pass Status]	True/False status for this instance with the named filter

Object	Definition
Position X	The X coordinate of the located instance
Position Y	The Y coordinate of the located instance
Angle	The angle of the located instance
Category	The category the instance was assigned to

The <filter> results won't be displayed unless at least one filter has been defined. The selection window looks like:



*Results Column Selection Window*

The <filter> Pass Status result will display which filters passed and which failed.

#### ***Advanced Properties***

#### **Image and Filter Logging**

Logging provides a means for logging data from the Inspection tool when an inspection fails.

#### **Log Mode**

Log Mode offers three options:

- Log all data
- Log pictures for inspections that fail
- All inspections must fail

The pictures only get logged if all inspections fail.

#### **File Mode**

This offers the choice of storing files by Hour, by Day, or by Month.

#### **Directory**

This lets you select the directory where the files will be stored.

#### **Results Logging**

If enabled, the results of the tool will be logged to a file.

If the extension of the destination file is csv, then the log is written in csv format, which opens with Excel. If a different extension is selected, the log is written in a plain text format (similar to the ACE Sight 2 format).

The size of the log file can be set (default size is 5 Mb).

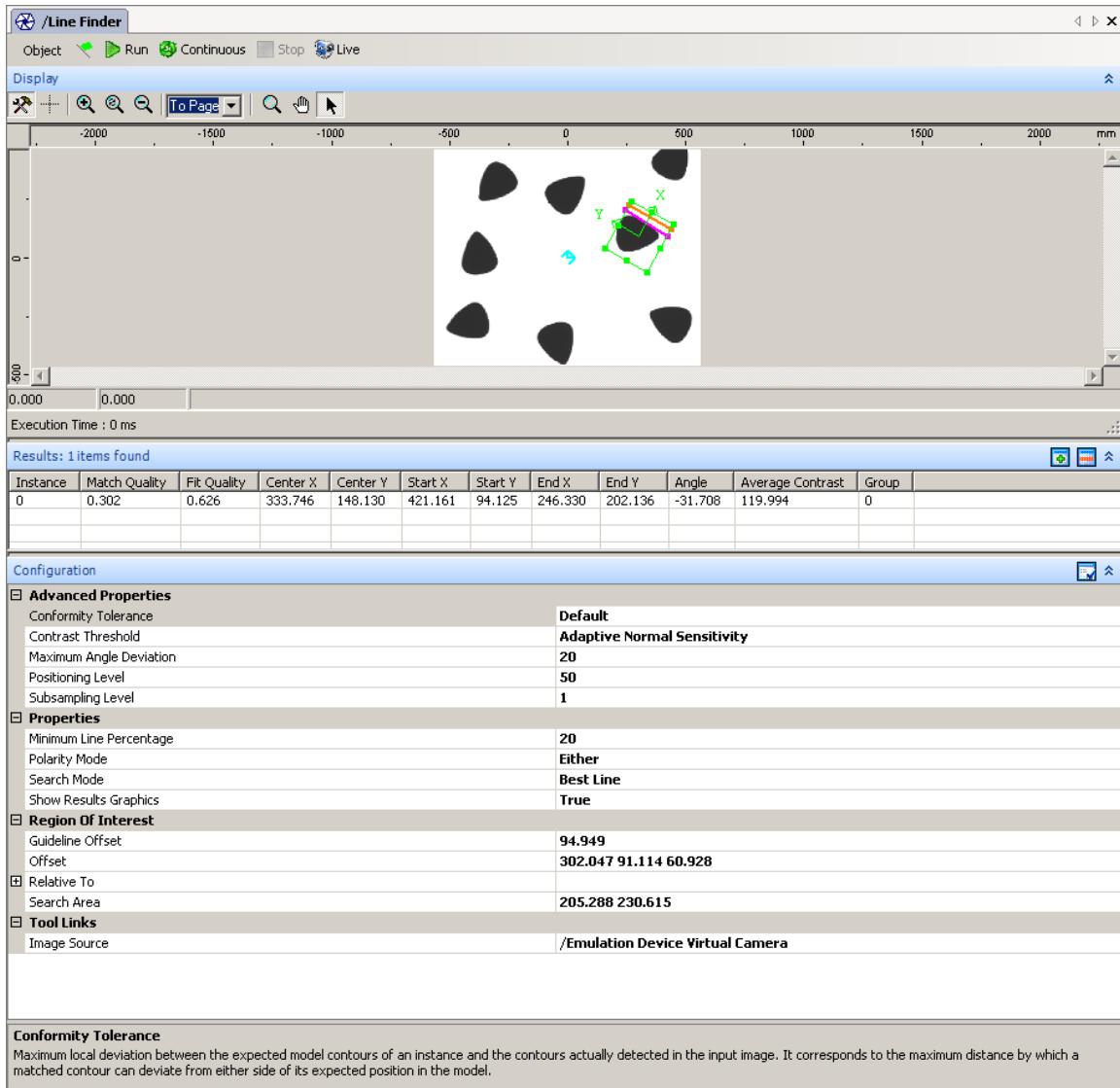
### **Line Finder**

The Line Finder identifies linear features on objects and returns the angle of inclination with the horizontal and coordinates of the end points of the detected lines.

To create a Line Finder tool, right-click in the Tree structure, then select:

**New > Vision > Tool > Line Finder**

## Line Finder



### Line Finder Object Editor

#### Configuration

Object	Definition
<b>Properties</b>	
Minimum Line Percentage	Minimum percentage of line contours that need to be matched for a line hypothesis to be considered as valid.

<b>Object</b>	<b>Definition</b>
Polarity Mode	Selects the type of polarity accepted for finding an entity. Polarity identifies the change in grey-level values from the center of the tool (inside) toward the outside.
Search Mode	Specifies the method used by a Finder tool to select a hypothesis.
Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
<b>Tool Links</b>	
Image Source	Defines the image source used for processing by this vision tool.
<b>Region of Interest (ROI)</b>	
Guideline Offset	Defines the offset, from the center of the region of interest, for the guideline.
Offset	Specifies the center and rotation of the ROI. X, Y, Degrees.
Relative To	Specifies the tool that will provide the input to this tool. The tool can be dragged from the Folder pane and dropped into this field.
Search Area	Defines the height and width of the region of interest.
<b>Advanced Properties</b>	
Conformity Tolerance	Maximum local deviation between the expected model contours of an instance and the contours actually detected in the input image.
Contrast Threshold	Defines the minimum contrast needed for an edge to be detected in the input image and used for line computation.
Maximum Angle Deviation	Maximum deviation in angle allowed for a detected edge to be used for generating an entity hypothesis.
Positioning Level	Configurable effort level of the instance positioning process.
Subsampling Level	Subsampling level used to detect edges that are used by the tool to generate hypotheses.

**Results**

Object	Definition
Instance	The result instance
Start X	The X coordinate of the starting point
Start Y	The Y coordinate of the starting point
End X	The X coordinate of the ending point
End Y	The Y coordinate of the ending point
Center X	The X coordinate of the center point
Center Y	The Y coordinate of the center point
Angle	The angle of the located instance with respect to the horizontal plane
Fit Quality	Normalized average error between the calculated line and the actual edges matched to the line.
Match Quality	Percentage of edges actually matched to the found line.
Average Contrast	Average contrast between light and dark pixels on either side of the found entity (point, line, or arc), expressed in grey-level values.

**Related Topics**[Configuring Line Finder Properties](#)[Configuring Line Finder Properties - Advanced](#)[Line Finder Results](#)**Configuring Line Finder Properties**

Finder tools detect edges in the input images, then uses edges to generate a vectorized description called an entity.

- Line Finder parameters modify the quality and quantity of edges that are generated from the input image.
- Edges are detected parallel to the Y-Axis, moving through the region of interest in a negative-to-positive direction relative to the X-Axis.

**Minimum Line Percentage**

Minimum percentage of line contours that need to be matched for a line hypothesis to be considered as valid.

---

0 < Range ≤ 100.0

### Polarity Mode

Sets the mode that will apply to the search for entities. Polarity identifies the change in grey-level values along the tool's X axis, in the positive direction.

The available modes are:

- Dark To Light: The Line Finder searches only for lines occurring at a dark to light transition in grey-level values.
- Light To Dark: The Line Finder searches only for lines occurring at a light to dark transition in grey-level values.
- Either: The Line Finder searches only for lines occurring either at a light to dark or dark to light transition in grey-level values. This mode will increase processing time.
- Don't Care: The Line Finder searches only for lines occurring at any transition in grey-level values including reversals in contrast, for example, on an unevenly colored background.

### Search Mode

Search Mode specifies the method used by the tool to generate and select a hypothesis.

The available modes are:

- Best Line: Selects the best line according to hypotheses strengths. This mode will increase processing time.
- Line Closest To Guideline: Selects the line hypothesis closest to the Guideline.
- Line With Maximum Negative X Offset: Selects the line hypothesis closest to the Rectangle bound that is at maximum negative X offset.
- Line With Maximum Positive X Offset: Selects the line hypothesis closest to the Rectangle bound that is at maximum positive X offset.

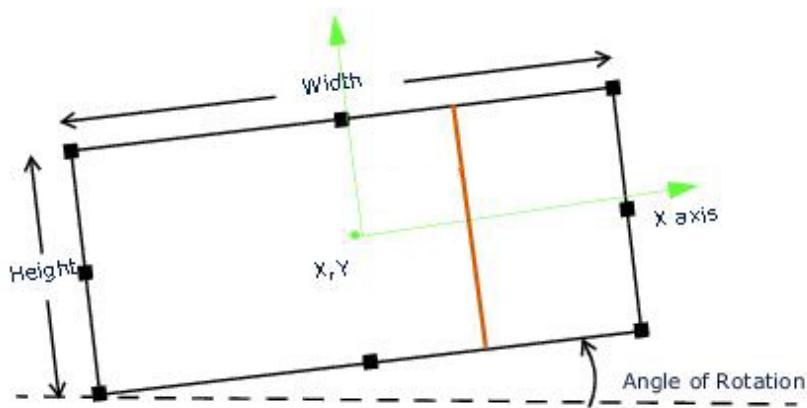
### ***Region of Interest (ROI)***

Most tool position parameters can be set through the ROI section of the tool interface. The following parameters define the tool region of interest.

Object	Definition
<b>Offset</b>	Offset is the offset from the tool X-axis, defined by:
X	The X coordinate of the center of the region of interest.

---

Object	Definition
Y	The Y coordinate of the center of the region of interest.
Degrees	The angle of rotation of the region of interest.
<b>Relative To</b>	Specifies the tool that will provide the input to this tool. The tool can be dragged from the Folder pane and dropped into this field.
<b>Search Area</b>	The tool region of interest is the rectangle that is defined by:
Height	The height of the region of interest.
Width	The width of the region of interest.



#### *Location Properties for the Region of Interest*

##### **Guideline Offset**

The Guideline Offset is the offset from the tool Y-axis. The Guideline marker can be displaced along the X-axis. This marker acts as both a visual guide for positioning the tool and as a constraint for the tool Search Mode.

##### **Related Topics**

[Line Finder Tool](#)

[Configuring Line Finder Properties - Advanced](#)

[Line Finder Results](#)

### ***Configuring Line Finder Properties - Advanced***

#### **Conformity Tolerance**

Conformity Tolerance corresponds to the maximum distance in calibrated units by which a matched edge can deviate from either side of its expected position on the line.

Default Conformity Tolerance is a read-only value that the tool computes by analyzing the calibration, the edge detection parameters, and the search parameters.

This can be a fixed value, or, if disabled, the Default.

To manually set Conformity Tolerance you must check the Enable checkbox.

- If you set a value lower than the Minimum Conformity Tolerance value, the Conformity Tolerance value will be automatically reset to the minimum valid value.
- If you set a value higher than the Maximum Conformity Tolerance value, the Conformity Tolerance value will be automatically reset to the maximum valid value.

#### **Contrast Threshold**

Contrast Threshold sets the minimum contrast needed for an edge to be detected in the input image. The threshold value expresses the step in light values required to detect edges.

This threshold is expressed in terms of a step in grey-level values. The property can be set to Low, Normal, or High Sensitivity, or a fixed value.

- This value can be set manually only when Contrast Threshold Mode is set to Fixed Value.
- Higher values reduce sensitivity to contrast. This reduces noise and the amount of low-contrast edges.
- Lower values increase sensitivity and add a greater amount of edge at the expense of adding more noise. This may generate false detections and/or slow down the search process.

#### **Contrast Threshold Mode**

Contrast Threshold Mode defines how contrast threshold is set. Contrast threshold is the level of sensitivity that is applied to the detection of edges in the input image. The contrast threshold can be either Adaptive, or Fixed.

Adaptive thresholds set a sensitivity level based on image content. This provides flexibility to variations in image lighting conditions and variations in contrast during the Search process.

- Adaptive Low Sensitivity uses a low sensitivity, adaptive threshold for detecting edges. Adaptive Low Sensitivity detects strongly-defined edges and eliminates noise, at the risk of losing significant edge segments.
- Adaptive Normal Sensitivity sets a default sensitivity threshold for detecting edges.

- Adaptive High Sensitivity detects a great amount of low-contrast edges and noise.
- Fixed Value sets an absolute value for the sensitivity to contrast. A typical situation for the use of a fixed value is a setting in which there is little variance in lighting conditions.

The value represents steps in grey-level.

- Higher values reduce sensitivity to contrast. This reduces noise and the amount of low-contrast edges.
- Lower values increase sensitivity and add a greater amount of edge at the expense of adding more noise. This may generate false detections and/or slow down the search process.

### **Maximum Angle Deviation**

Maximum Angle Deviation relates to the deviation in angle between a hypothesis and the line found by the tool. By default the Line Finder accepts a 20 degree deviation. However, the tool uses the defined Maximum Angle Deviation value to test the hypothesis and refine the pose of the found line.

### **Positioning Level**

Positioning Level sets the effort level of the instance-positioning process. The range is from 10 (coarser positioning and lower execution time) to 100 (high-accuracy positioning of lines).

### **Subsampling Level**

Subsampling Level sets the subsampling level used to detect edges that are used by the tool to generate hypotheses.

- High values provide a coarser search with a shorter execution time.
- Lower values can provide a more refined search with slower execution time.
- A higher subsampling value may help improve accuracy in blurry images.

### **Results Logging**

If enabled, the results of the tool will be logged to a file.

If the extension of the destination file is csv, then the log is written in csv format, which opens with Excel. If a different extension is selected, the log is written in a plain text format (similar to the ACE Sight 2 format).

The size of the log file can be set (default size is 5 Mb).

### **Related Topics**

[Line Finder](#)

---

[Configuring Line Finder Properties](#)

[Line Finder Results](#)

### ***Line Finder Results***

#### **Display Window**

The Display window shows the image being processed, the region of interest selected, and the locations of found instances. The elapsed time is displayed at the bottom of this window.

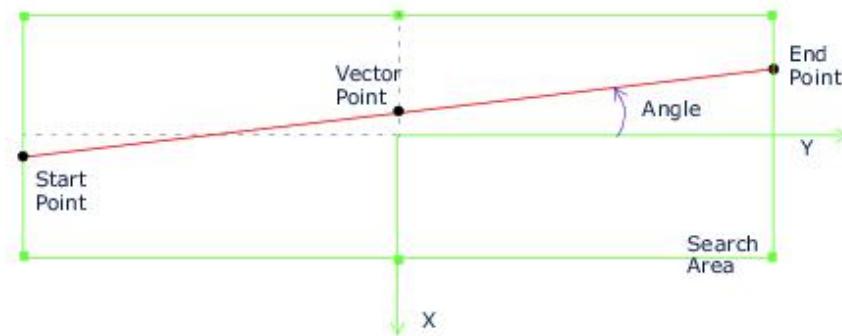
#### **Grid of Results**

The grid of results presents the result values for all instances found by the Line Finder tool.

Results are presented below in the order in which they are output.

Object	Definition
Elapsed Time	The Elapsed Time is the total execution time of the Line Finder. Elapsed Time is not visible in the results grid, but it is displayed at the bottom of the Display window after each iteration of the Line Finder.
Instance	Index of the result instance (numbered from 0).
Frame/Group	The frame or group to which the result belongs.
Vector Position X	X coordinate of the point of intersection between the line and the X axis of the tool bounding box. Occasionally, when the line exits the bounding box without covering the entire bounding box height, the returned Vector Point may be located outside the bounding box boundary.
Vector Position Y	Y coordinate of the point of intersection between the line and the X axis of the tool bounding box. Occasionally, when the line exits the bounding box without covering the entire bounding box height, the returned Vector Point may be located outside the bounding box boundary.
Start Position X	X coordinate of the point at the start of the line segment.
Start Position Y	Y coordinate of the point at the start of the line segment.

Object	Definition
End Position X	X coordinate of the point at the end of the line segment.
End Position Y	Y coordinate of the point at the end of the line segment.
Angle	Angle of the found line. A line is defined as the line passing through Vector Point X and Vector Point Y at the given Angle.
Average Contrast	Average grey-level contrast between light and dark pixels on either side of the found line.
Fit Quality	Fit Quality is the normalized, average error between the calculated line and the actual edges matched to the found line. Fit Quality ranges from 0 to 100, with 100 being the best quality. A value of 100 means that the average error is 0. Conversely, a value of 0 means that the average matched error is equal to the Conformity Tolerance.
Match Quality	Match Quality corresponds to the percentage of edges actually matched to the found line. Match Quality ranges from 0 to 100, with 100 being the best quality. A value of 100 means that edges were matched to 100% of the points along the found line. Similarly, a value of 20 means edges were matched to 20% of the points along the found line.

*Line Finder Results*

**Related Topics**

[Line Finder](#)

[Configuring Line Finder Properties](#)

[Configuring Line Finder Properties - Advanced](#)

**Locator**

NOTE: The Locator tool will not work until you have created a Locator Model for it. See "Locator Model"

The Locator identifies objects based on models, which describe the geometry of the objects.

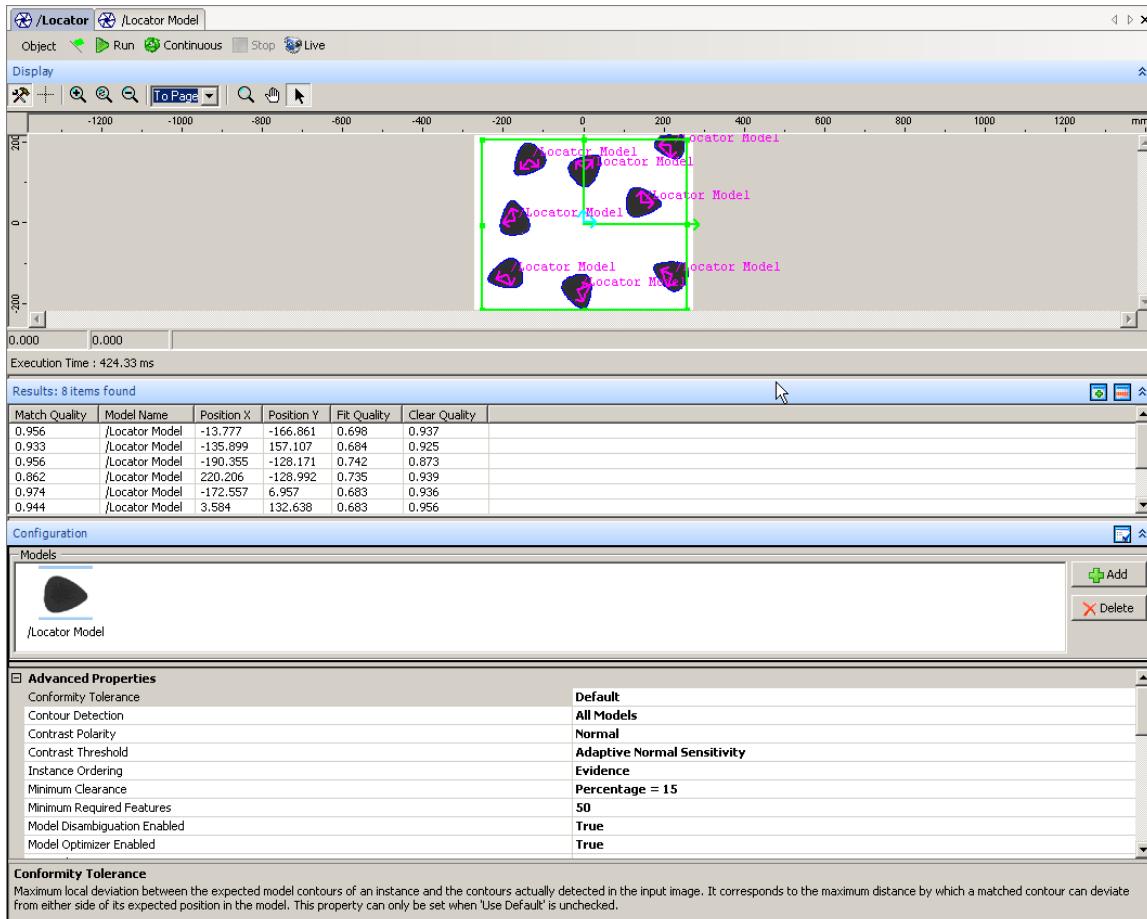
- Because of its speed, accuracy, and robustness, the Locator is the ideal "frame-provider" for ACE Sight inspection tools.
- A Locator can be frame-based. A frame-based Locator requires the input of another tool in the application, preferably another Locator. A model-based Locator can be used to locate features, "sub-features," or "sub-parts" on a parent object.

To create a Locator tool, right-click in the Tree structure, then select:

**New > Vision > Tool > Locator**

## Locator

---



### Locator Object Editor

#### Configuration

Object	Definition
<b>Properties</b>	
Maximum Instance Count	Sets the maximum number of object instances that are searched for in the input image. All of the object instances matching the search constraints are output, up to a maximum of Maximum Instance Count. They are ordered according to the Instance Ordering property.

<b>Object</b>	<b>Definition</b>
Minimum Model Percentage	Sets the minimum percentage of model contours that need to be matched in the input image in order to consider the object instance as valid.
Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
<b>Tool Links</b>	
Image Source	Defines the image source used for processing by this vision tool.
<b>Region of Interest (ROI)</b>	
Offset	Specifies the center and rotation of the ROI. X, Y, Degrees.
Relative To	Specifies the tool that will provide the input to this tool. The tool can be dragged from the Folder pane and dropped into this field.
Search Area	Defines the height and width of the region of interest.
<b>Advanced Properties</b>	
Conformity Tolerance	Sets the maximum local deviation between the model contours and the contours detected in the input image. Allows a selection of Default or a fixed Tolerance within a range.
Contour Detection	Specifies the configuration of the contour detection parameters.
Contrast Polarity	Sets the type of polarity accepted for object recognition.
Contrast Threshold	Sets the minimum contrast needed for an edge to be detected in the input image.
Instance Ordering	Sets the order in which the instances are processed and output
Minimum Clearance	When enabled, sets the minimum percentage of the model bounding box area that must be free of obstacles to consider an object instance as valid.

<b>Object</b>	<b>Definition</b>
Minimum Required Features	Sets the minimum percentage of required features that must be recognized in order to consider the object instance as valid.
Model Disambiguation Enabled	When set to true (default), the Locator applies disambiguation to discriminate between similar models and between similar hypotheses of a single object. When set to false, the Locator does not apply disambiguation.
Model Optimizer Enabled	Specifies whether Models can be optimized interactively.
Nominal Rotation	Specifies the range of rotation allowed for an instance to be valid.
Nominal Scale Factor	Nominal Scale Factor sets the required scale factor for an object to be recognized.
Output Symmetric Instances	When true, all the symmetric poses of the object instances are output. When false, only the best quality of the symmetric poses of the object instance is output.
Percentage Of Points To Analyze	Percentage of points on the contour that will be analyzed.
Positioning Level	Configurable effort level of the instance positioning process.
Recognition Level	Configurable effort level of the search process.
Search Based On Outline Level Only	When set to true, the Locator does not use the models at the Detail level for the positioning process.
Show Model Name	When set to true, and Show Results Graphics is enabled, the model name is displayed in the Vision Window.
Timeout	Time (in milliseconds) after which the Locator tool aborts its search process.

**Results**

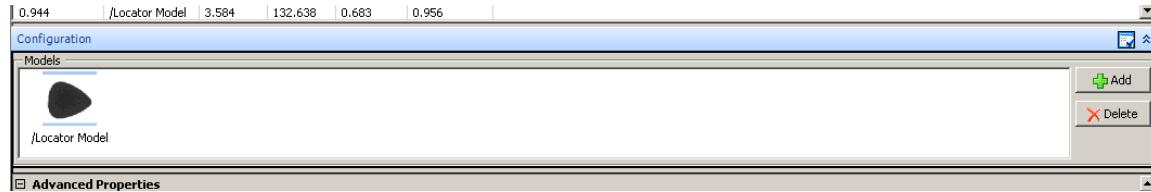
<b>Object</b>	<b>Definition</b>
Instance	Index of the result instance, starting with 0.

<b>Object</b>	<b>Definition</b>
Frame/Group	The frame or group to which the result belongs.
Model Index	Index of the model located for each instance.
Model Name	Name of the model located for each instance.
Fit Quality	Normalized average error between the matched model contours and the actual contours detected in the input image. Fit quality ranges from 0 to 100, with 100 being the best quality. A value of 100 means that the average error is 0. Conversely, a value of 0 means that the average matched error is equal to Conformity Tolerance.
Match Quality	Amount of matched model contours for the selected object instance. Match quality ranges from 0 to 100, with 100 being the best quality. A value of 100 means that 100% of the model contours were successfully matched to the actual contours detected in the input image.
Clear Quality	Measure of the clear area surrounding the specified object instance. Clear quality ranges from 0 to 100, with 100 being the best quality. A value of 100 means that the instance is completely free of obstacles. If Minimum Clearance is Disabled, this value is 100.
Frame/Group	
Position X	X coordinate of the origin of the located instance.
Position Y	Y coordinate of the origin of the located instance.
Angle	Angle of the located instance.
Scale Factor	Relative size of the located instance, with respect to its associated model.
Symmetry	Index of the object instance of which the selected object instance is a symmetry. Output Symmetric Instances must be true.
Time	Time needed to recognize and locate the selected object instance, expressed in milliseconds.

### ***Additional Menus***

The locator tool uses a custom configuration display to graphically show the models associated with the tool. You can drag/drop models onto the tool display.

The Locator Model editor can be opened by double-clicking on the model in the Models section.



*Locator Model Display Window*

### ***Related Topics***

[Configuring Locator Properties - Advanced](#)

[Locator Model](#)

### ***Configuring Locator Properties - Advanced***

The Advanced Properties section of the Locator tool interface provides access to advanced Locator parameters and properties.

The Locator detects edges in the input images, then uses the edges to generate a vectorized description of the image, called a Scene. The Locator tool generates source contours on two coarseness levels: Outline and Detail.

Edge Detection parameters modify the quality and quantity of contours that are generated from the input image.

### **Contour Detection**

Contour Detection sets how the contour detection parameters are configured. For most applications, Contour Detection should be set to All Models. Custom contour detection should only be used when the default values do not work correctly.

Custom can specify Outline Level, Detail Level, and Tracking Inertia.

- When set to All Models, the contour detection parameters are optimized by analyzing the parameters that were used to build all the models that are currently active.
- When set to Custom, the contour detection parameters are set manually to integer values.

The Locator process generates two levels of contours. The coarser Outline Level contour is used to generate hypotheses of potential instances. The finer Detail Level contour is used to confirm the hypotheses and refine the location of the instances.

For each level, coarseness values range from 1 to 16. At level 8 for example, the resolution is 8 times lower than an image at full resolution (level 1).

### **Outline Level**

The Outline Level contour is used to rapidly identify potential instances of the object.

- The higher the setting, the coarser the contour resolution.
- To improve search speed, set the Outline Level to the highest value at which contour geometry is still distinctive and stable.

### **Detail Level**

The Detail Level contour is used to confirm recognition and refine the position of valid instances.

- The higher the setting, the coarser the contour resolution.
- For images that are not in perfect focus, better results will be obtained with a higher value for the Detail Level.
- To obtain high-accuracy object location, it is preferable to use images with sharp edges and set the lowest coarseness value for the Detail Level.

### **Tracking Inertia (0 - 1)**

Increasing the Tracking Inertia can help close small gaps and connect contours that would otherwise be broken into smaller sections.

### **Search Based On Outline Level Only**

Search Based On Outline Level Only restricts the search to using only the Outline Level source contours to search, recognize, and position object instances. Detail Level contours are ignored completely.

Setting this to true can increase speed, with a possible loss of accuracy, and with the possibility of detecting false instances.

- An Outline-based search is useful for time-critical applications that do not require a high-positioning accuracy, or that need only to check for presence/absence of objects.
- To be effective, this type of search requires clean runtime images that provide high-contrast contours with little or no noise or clutter.

## Contrast Polarity

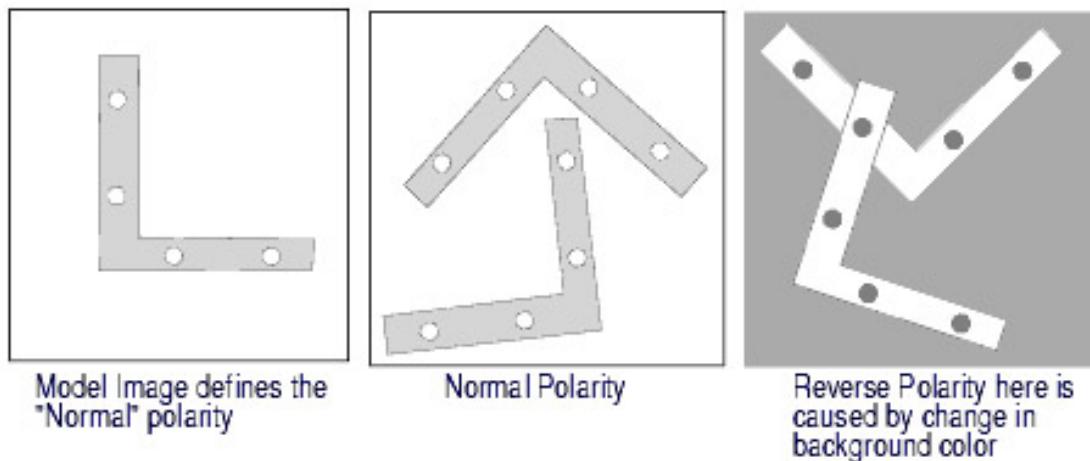
Contrast Polarity defines the polarity change in grey-level values between an object and its background. This polarity can be dark to light or light to dark. The reference polarity for an object is defined by its Model. Polarity is always defined with respect to the initial polarity in the image on which the Model was created.

### Normal polarity

Searches only for object instances having the same object-to-background polarity as the Model. For example, if the model is a dark object on a light background, the Locator searches only for dark objects on a light background.

### Reverse

Restricts the Locator to search only for objects having a object-to-background polarity that is inverse to that in the Model. For example, if the model is a dark object on a light background, the Locator searches only for light objects on a dark background.



*Contrast Polarity - Model, Normal, and Reverse*

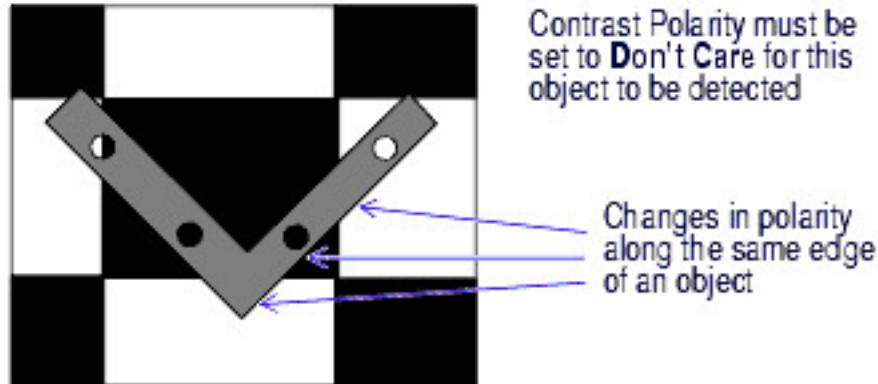
### Normal And Reverse

Enables the Locator to search for all cases that present either Normal or Reverse polarity. This will not take into account cases where polarity is reversed at various locations along the edges of an object.

### Do Not Care

Detects complete contrast reversal, as well as local contrast reversal, on

sections of the object contour, such as shown in the "checkerboard background" of the following example.



*Contrast Polarity - Local Changes in Polarity*

### **Contrast Threshold**

Contrast Threshold sets the minimum contrast needed for an edge to be detected in the input image. The threshold value expresses as the step in grey-level values required to detect contours. This threshold can be set to Low, Normal, High, or a fixed value.

- Higher values reduce sensitivity to contrast. This reduces noise and the amount of low-contrast edges.
- Lower values increase sensitivity and add a greater amount of edge at the expense of adding more noise. This may generate false detections and/or slow down the search process.

Contrast threshold is the level of sensitivity that is applied to the detection of contours in the input image. The contrast threshold can be either Adaptive or Fixed.

Adaptive thresholds set a sensitivity level based on image content. This provides flexibility to variations in image lighting conditions and variations in contrast during the Search process.

#### **Adaptive Low Sensitivity**

Sets a low sensitivity, adaptive threshold for detecting contours. Adaptive Low Sensitivity detects strongly defined contours and eliminates noise, at the risk of losing significant contour segments.

#### **Adaptive Normal Sensitivity**

Sets a default sensitivity threshold for detecting contours.

### **Adaptive High Sensitivity**

Detects the most low-contrast contours and noise.

### **Fixed Value**

Sets an absolute value for the sensitivity to contrast. A typical situation for the use of a fixed value is a setting in which there is little variance in lighting conditions.

- The numerical fixed value, from 1 - 255, corresponds to the minimum step in grey-level values required to detect contours.
- A lower value adds more contours at the expense of adding more noise, which may generate false detections and/or slow down the search process.

### ***Search***

Search parameters are constraints that restrict the Locator's search process.

### **Conformity Tolerance**

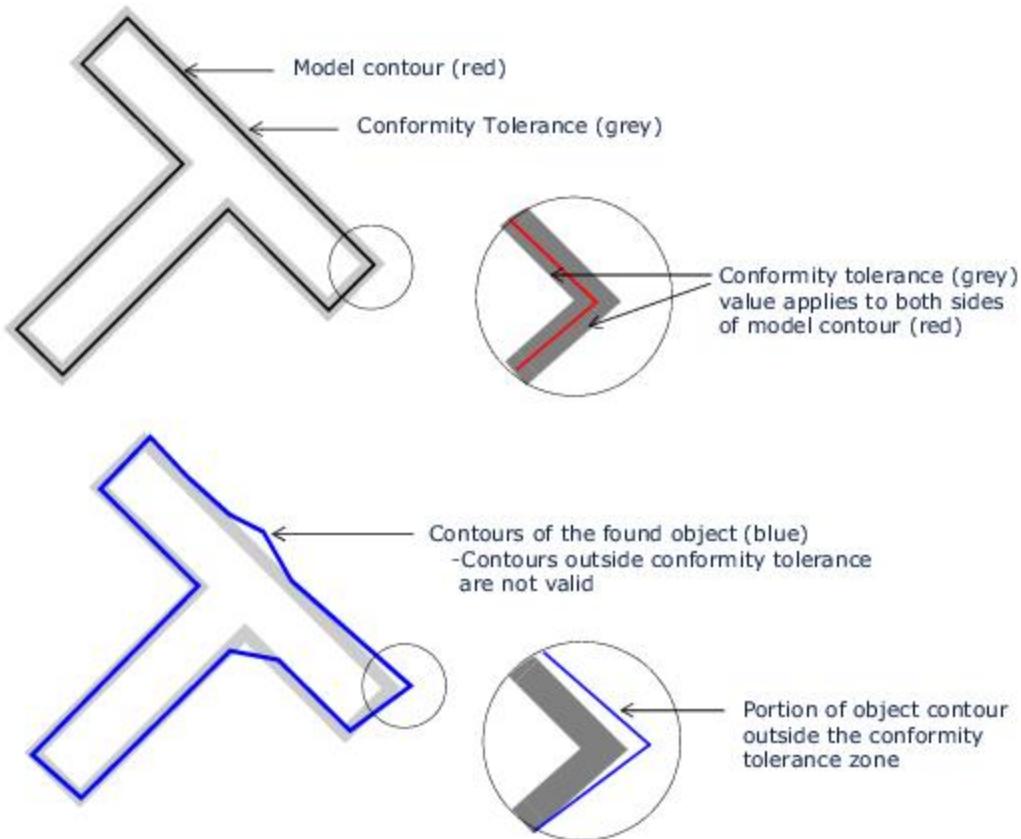
Conformity Tolerance defines the maximum allowable local deviation of instance contours from the expected model contours. Its value corresponds to the maximum distance by which a matched contour can deviate from either side of its expected position in the model.

Portions of the contour that are not within the Conformity Tolerance range are not considered to be valid. Only the contours within Conformity Tolerance are recognized and calculated for the Minimum Model Recognition search constraint.

Tolerance is either Default or user-specified.

## Locator

---



### Conformity Tolerance

**NOTE:** To manually set a Conformity Tolerance value, you must first uncheck the Use Default box.

#### Use Default

The default value is computed by the Locator tool by analyzing the calibration, the contour detection parameters, and the search parameters.

If the Use Default box is checked, the default value will be used.

Unchecking the Use Default box allows you to manually set the Conformity Tolerance value.

#### Range Enabled

Enables the use of the manually-set tolerance value.

### **Tolerance**

The maximum distance in calibrated units by which a matched contour can deviate from either side of its expected position. A value from 1 - 100.

### **Nominal Rotation**

Rotation constrains the rotation range the Locator will search for.

It can specify whether the rotation of a recognized instance must fall within the range set or be equal to the nominal value set.

When enabled, a nominal value is applied (no rotation is accepted); otherwise, the range is used.

By default, the Locator finds objects within a range of rotation values between -180 and 180 degrees. Allowable rotation can be restricted by decreasing the range of rotation values. The rotation range spans counterclockwise from the minimum to the maximum angle.

#### **Use Nominal**

When Use Nominal is checked, the Nominal value is applied. Otherwise, the range set by Maximum and Minimum is used.

If you want to search for an instance at a nominal rotation but need to compute its actual rotation, disable the Use Nominal checkbox and enter a small range, such as 89 to 91.

#### **Nominal**

Search for objects at a specific angle of rotation. Note that, when a nominal value is used, the Locator does not compute the actual rotation of instances; instead, the instances are positioned using the Nominal rotation value. Nominal is a value, in degrees, from -180 to +180.

#### **Maximum**

A value, in degrees, from -180 to +180.

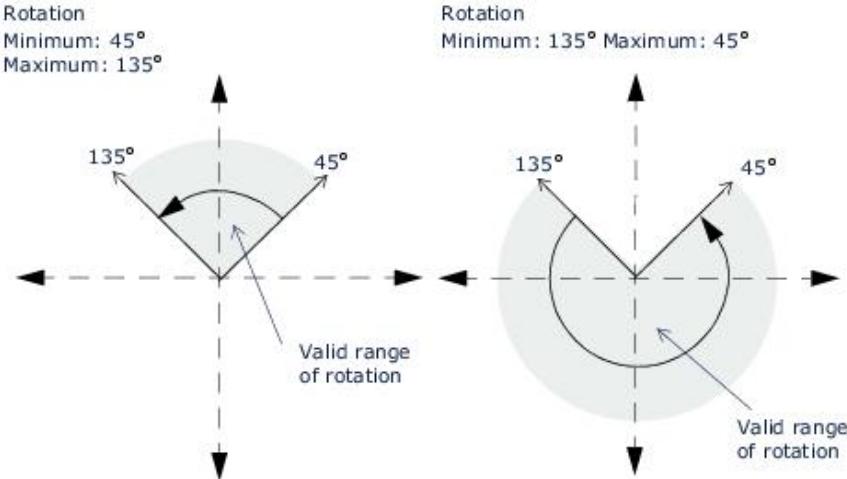
#### **Minimum**

A value, in degrees, from -180 to +180.

---

## Locator

---



### *Rotation Range - Minimum to Maximum Angle*

#### **Rotation Considerations**

- By default, the range is selected, with a full search range from 180 to -180. This means that the Locator will search for objects in all orientations.
- The rotation range spans counterclockwise from the specified minimum angle to the specified maximum angle. The preceding graphic illustrates the impact of selecting a minimum and maximum angle.
- Check Use Nominal to search for objects at a specific angle of rotation. When a Nominal value is applied, the Locator does not compute the actual rotation of instances; instead, the instances are positioned using the Nominal rotation value.
- If you want to search for an instance at a Nominal rotation but need to compute its actual rotation, disable the Use Nominal checkbox and enter a small range, such as 89 to 91.
- If a nominal value is used with objects that present a slight variation in rotation, the objects may be recognized and positioned with reduced quality because their true rotation will not be measured. In such a case, it is preferable to configure a narrow rotation range, such as  $\pm 1$  degree, instead of a nominal value.

#### **Nominal Scale Factor**

Nominal Scale Factor sets the required scale factor for an object to be recognized.

This constrains the range of scale factors the Locator will search for.

The scale of objects to be located can be set at a fixed nominal value, or as a range of scale values.

The default setting for the scale factor is a nominal value of 1, which applies to most situations.

If objects have a slight variation in scale, the objects may be recognized and positioned with reduced quality because their true scale factor will not be measured. In such a case, it is preferable to configure a narrow scale range, instead of a nominal value.

**NOTE:** Using a large scale factor can significantly slow down the search process. This range should be configured to include only the scale factors that are actually expected for a given application. The scale factor range is one of the parameters that has the most significant impact on search speed.

### **Use Nominal**

Search for a specific scale factor. Note that when a nominal value is used, the Locator does not compute the actual scale of instances; instead, the instances are positioned using the Nominal scale value. When Nominal is checked, the Nominal scale factor is applied.

Disable the Use Nominal checkbox to specify the range of scale the Locator will search for. If you want to search for an instance at a nominal scale factor, but need to compute its actual scale, disable the Nominal checkbox and enter a small range, such as 0.99 to 1.01.

#### **Nominal**

A value, from 0.1 to 10. A Nominal scale factor of 1 is set by default.

#### **Maximum**

The maximum scale factor that the Locator will look for. This must be less than or equal to 10, and greater than or equal to Minimum.

#### **Minimum**

The minimum scale factor that the Locator will look for. This must be greater than or equal to 0.1, and less than or equal to Maximum.

### **Positioning Level**

Allows you to modify the positioning accuracy. The default setting of 5 is the optimized and recommended setting for typical applications. Positioning Level can be from 0 to 10.

A value of 0 will provide coarser positioning and faster execution time. Conversely, a value of 10 will provide high-accuracy positioning of object instances.

- Positioning Level has only a slight impact on the execution speed.
- In applications where accuracy is not critical, decreasing the value can provide a slight improvement in speed.

### **Recognition Level**

Recognition Level allows you to slightly modify the level of recognition effort. The default setting of 5 is the optimized (and recommended) setting for typical applications. Recognition Level can be from 0 to 10.

A value of 0 will lead to a faster search that may miss instances that are partly blocked. Conversely, a value of 10 is useful for finding partly blocked objects in cluttered or noisy images, or for models made up of small features at the Outline Level.

- When changing the recognition level, test your application to find the optimum speed at which the process will still find all necessary objects within the image.
- If recognition level is too low (quick) some instances may be ignored.
- If recognition level is too high (exhaustive), your application may run too slowly.
- Recognition level does not affect positioning accuracy.

### ***Model***

#### **Show Model Name**

Determines if the model name is displayed in the Vision Window. This is only evaluated if Show Results Graphics is true.

#### **Model Disambiguation Enabled**

When Model Disambiguation Enabled is set to true (default), the Locator applies disambiguation to discriminate between similar models and between similar hypotheses of a single object. When set to false, the Locator does not apply disambiguation.

#### **Model Optimizer Enabled**

Model Optimizer Enabled specifies if the models can be optimized interactively using the Model Optimizer. When set to true, the models can be optimized. When set to false, the models cannot be modified.

To build an optimized Model, you must provide an initial model of the object, and then execute the Locator tool. While optimization is enabled, each new instance of the object found by the Locator is analyzed and compiled into the optimized model that is currently in progress. Strong features that recur frequently in the analyzed instances are retained in the optimized model. Once the model is considered satisfactory, the optimized model can be saved.

---

Note that the initial image is retained in the model; the image is not modified by the optimization process.

### **Percentage Of Points To Analyze**

Percentage Of Points To Analyze sets the percentage of points on a model contour that are actually used to locate instances. This can be from 0.1 to 100%.

For example, when Percentage Of Points To Analyze is set to the default 50% value, one out of two points are used. Increasing this value can increase the accuracy of the optimized model but incurs a longer optimization time.

Higher values are more accurate and slower than lower values.

### **Minimum Required Features**

Minimum Required Features, set through the Model Editor, are features that need to be recognized for the Locator to accept a valid instance of an object. Although, typically, the presence of all such features is required, the Minimum Required Features parameter lets you set a lower number of required features.

Range:  $0.1 \leq \text{Range} \leq 100$ .

Note that this parameter is expressed in terms of the number of required features in a model. It does not consider the amount of contour each required feature represents in the model.

### ***Instance Output Constraints***

#### **Instance Ordering**

Instance Ordering sets the order in which object instances are output.

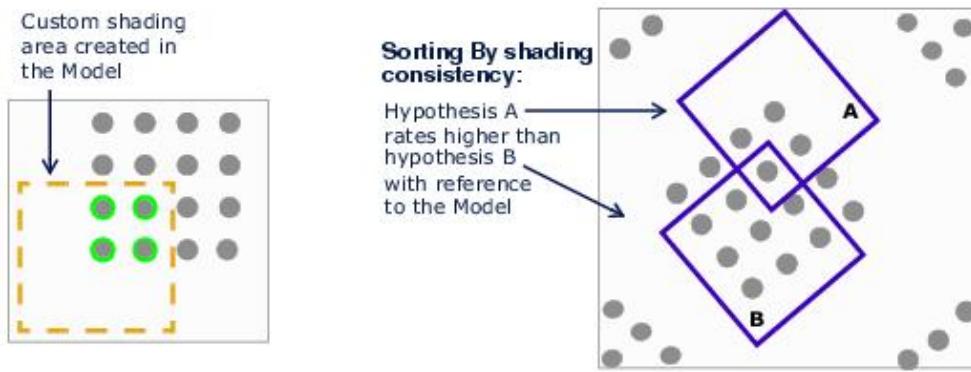
Selecting either Image Distance or World Distance also requires providing X, Y coordinates.

- At the default Evidence setting, the instances are ordered according to their hypothesis strength.
- Instances can be output in the order they appear in the image: Left To Right, Right To Left, Top To Bottom, and Bottom To Top. This feature is useful for pick-and-place applications in which parts that are farther down a conveyor must be picked first.
- The Quality setting orders instances according to their Match Quality. Instances having the same Match Quality are subsequently ordered by their Fit Quality. This setting can significantly increase the search time because the Locator tool cannot output instance results until it has found and compared all instances to determine their order. The time required to output the first instance corresponds to the total time needed to search the image and analyze all the potential instances. The time for additional instances is zero because the search process is already complete.
- Image Distance orders instances according to their proximity to a user-defined point in

the image coordinate system. The Reference X and Reference Y coordinates of the point are expressed in pixels.

- World Distance orders instances according to their proximity to a user-defined point in the World coordinate system. The Reference X and Reference Y coordinates of the point are expressed in the user-selected length units.
- Shading Consistency orders instances according to the custom shading area created in the model. If no Custom Shading Area is defined in the model, the Locator uses the entire model area for shading analysis.

This mode is useful when the shading information, in addition to the normal contour information, can assist in discriminating between very similar hypotheses. This is a requirement for color processing of models and also often used for BGA application, as illustrated in the following figure.



#### *Instance Ordering - Shading Consistency*

### **Output Symmetric Instances**

The Output Symmetric Instances setting determines how the Locator will handle symmetrical or nearly symmetrical objects.

- When set to false, the search process will output only the first best quality instance of a symmetric object.
- When set to true, the search process will output the results for all possible symmetries of a symmetric object. This can significantly increase execution time when there are many possible symmetries of an object, for example, if the object is circular.

### **Timeout**

The Timeout setting controls the elapsed time after which the Locator aborts its search process. This timeout period does not include the model learning phase. When the Timeout is

reached (and Timeout is Enabled), the instances already recognized are output by the Locator, and the search is stopped. Timeout can be from 1 to 30,000 milliseconds.

### **Minimum Clearance**

The Minimum Clearance sets the minimum percentage of the model bounding box area that must be free of obstacles to consider an object instance as valid. To enable this property, the Enabled box must be checked. Enabling Minimum Clearance may significantly increase the search time; it is intended for use in pick-and-place applications. When enabled, Minimum Clearance also activates the computation of the Clear Quality result for each instance.

### **Overlap Configuration**

If enabled, the results of the tool will be checked to see if any instances overlap. If any instances overlap, they will be excluded from the results.

### **Results Logging**

If enabled, the results of the tool will be logged to a file.

If the extension of the destination file is csv, then the log is written in csv format, which opens with Excel. If a different extension is selected, the log is written in a plain text format (similar to the ACE Sight 2 format).

The size of the log file can be set (default size is 5 Mb).

#### **Related Topics**

[Locator Tool](#)

[Locator Model](#)

### **Locator Model**

Locator Models describe the geometry of an object to be found by the Locator tool.

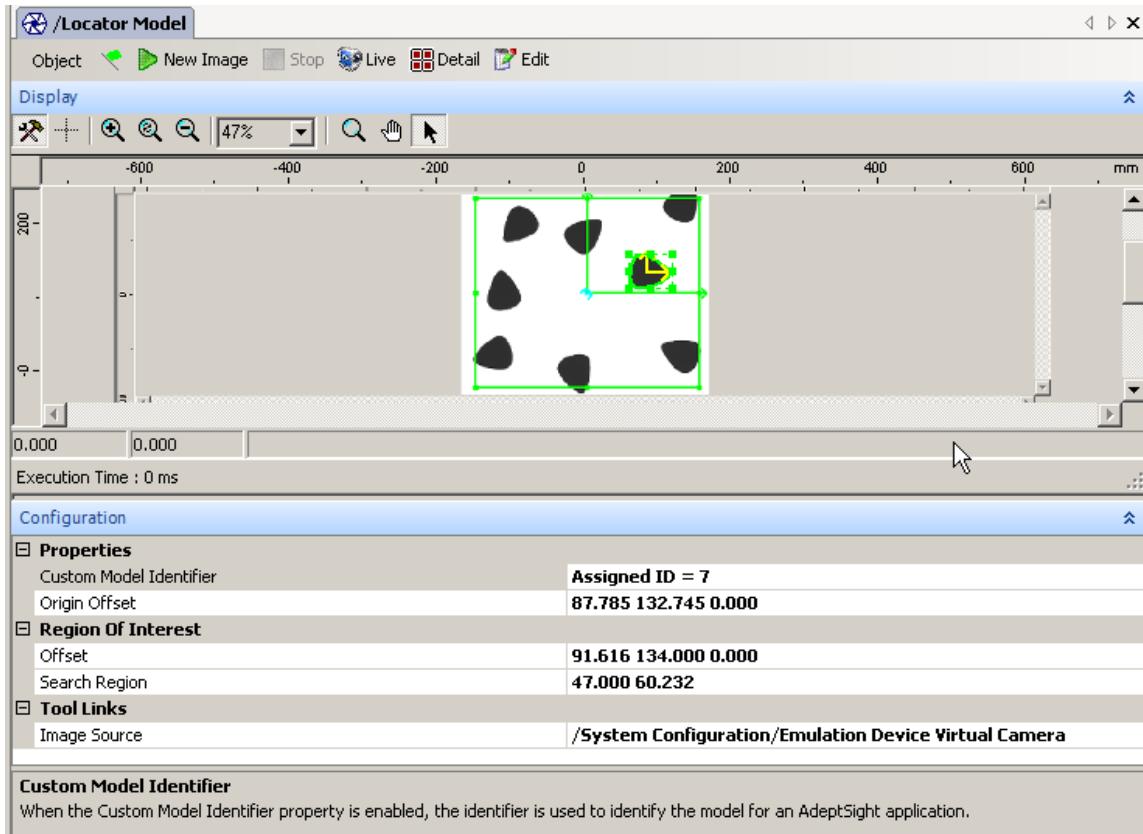
To create a Locator Model, right-click in the Tree structure, then select:

**New > Vision > Tool > Locator Model**

**NOTE:** You should first calibrate the camera before you create any Locator models.

## Locator Model

---

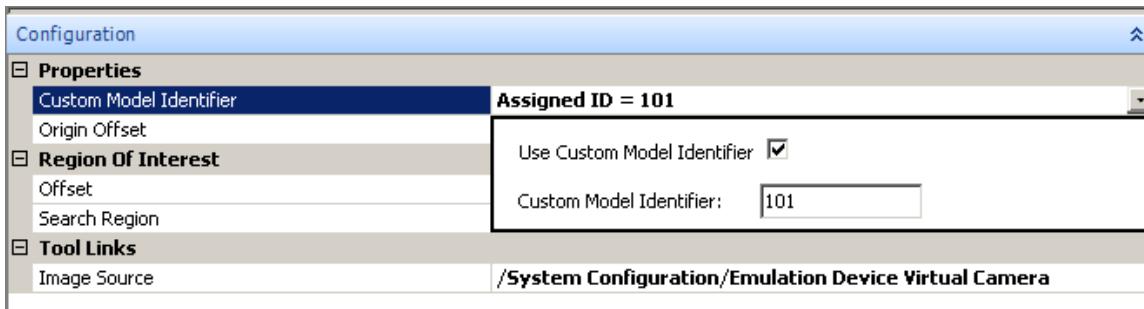


### Properties

Object	Definition
<b>Properties</b>	
Custom Model Identifier	When the Custom Model Identifier property is enabled, the identifier is used to identify the model for an ACE Sight application.
Search Region	Defines the height and width of the region of interest.

Object	Definition
Origin Offset	Defines the position of the model's origin in the World coordinate system. The origin, defined by the X, Y, and Degrees properties, is used by the Locator to express the pose of instances of the model. The model's origin is also used as the pivot point around which the rotation of the instance is measured. This origin defines the object coordinate system that can be used to express results of model-based inspection tools.
<b>Region of Interest (ROI)</b>	
Offset	Specifies the center and rotation of the ROI. X, Y, Degrees.
Search Region	Defines the height and width of the region of interest.
<b>Tool Links</b>	
Image Source	Defines the image source used for processing by this vision tool.

The Custom Model Identifier can be any integer from 0 to 999999999. See the following figure:



*Custom Model Identifier Drop-down Box*

The Use Custom Model Identifier box must be checked before a value can be entered for the identifier. If this box is not checked, the Locator tool will assign numbers automatically.

### **Results**

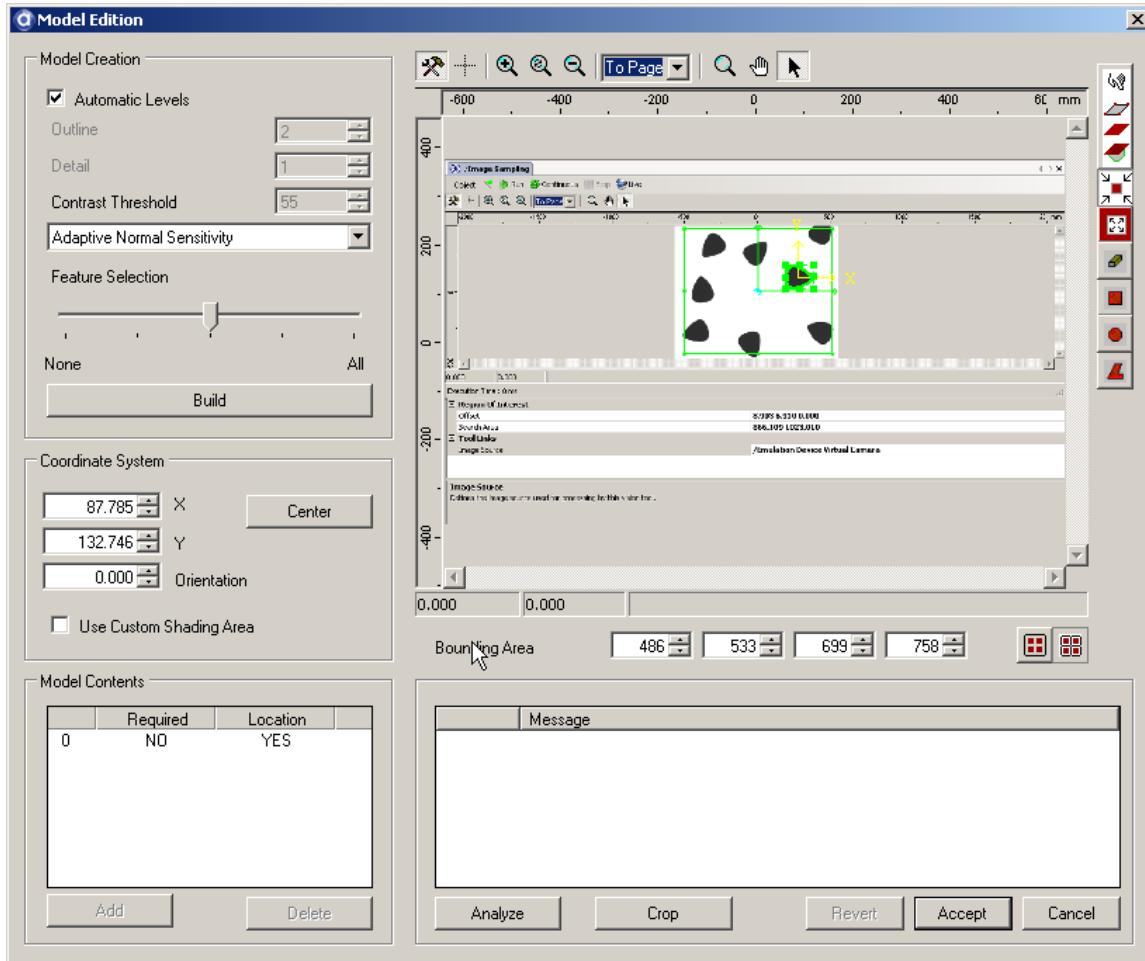
None.

### **Additional Menus**

The Locator Model editor can be opened from the Models window in the Locator by double-clicking the model name.

The tool shows the trained model in either outline or detail mode. Clicking toggles between the detail and outline display mode.

Advanced editing capabilities are available by clicking to display the advanced editing form:



*Locator Model, Advanced Editing*

## Set the Bounding Area

The Bounding Area tells the Locator Model where to look for the model. Anything not included inside the Bounding Area will not be considered a part of the model. For this reason, it is best to set the Bounding Area as close to the model part as possible, without cropping any edges of the part.

The model Bounding Area can be set by dragging the corners of the green box in the display window, or by modifying the numbers in the Bounding Area boxes (bottom, top, left, right). You can either enter the appropriate numbers, or use the up/down arrows for each value. When dragging the edges of the Bounding Area, it is usually best to magnify the view by clicking the magnifying glass icon.

## Setting the Origin

The model's origin tells the software where the part should be picked up, and in what orientation. You can set the origin in three ways:

- Manually drag the yellow origin indicator to the spot on the part where you want it. Adjust the orientation by dragging one of the origin arrowheads (X or Y).

This may be the best method for irregularly-shaped parts, particularly if their mass is off-center.

The coordinates and orientation of the origin can be manually entered in the Coordinate System pane.

- Click Center in the Coordinate System pane. This will center the origin in the model's bounding box.

Depending on the part shape, and how closely you set the bounding box, this may not be appropriate.

- Select a row in the Model Contents pane, and click Center in the Coordinate System pane.

Each row in the Model Contents pane represents a found instance. When you select a row, that instance will be highlighted in red. If there are multiple instances, you can use this to select and delete any unwanted instances. When a Model Contents row is selected, you can delete the associated instance by clicking Delete. When there is only one instance, select the row in the Model Contents pane and click Center in the Coordinate System pane. The origin will be centered on the instance.

Click Accept when you are done setting the Bounding Area and origin.

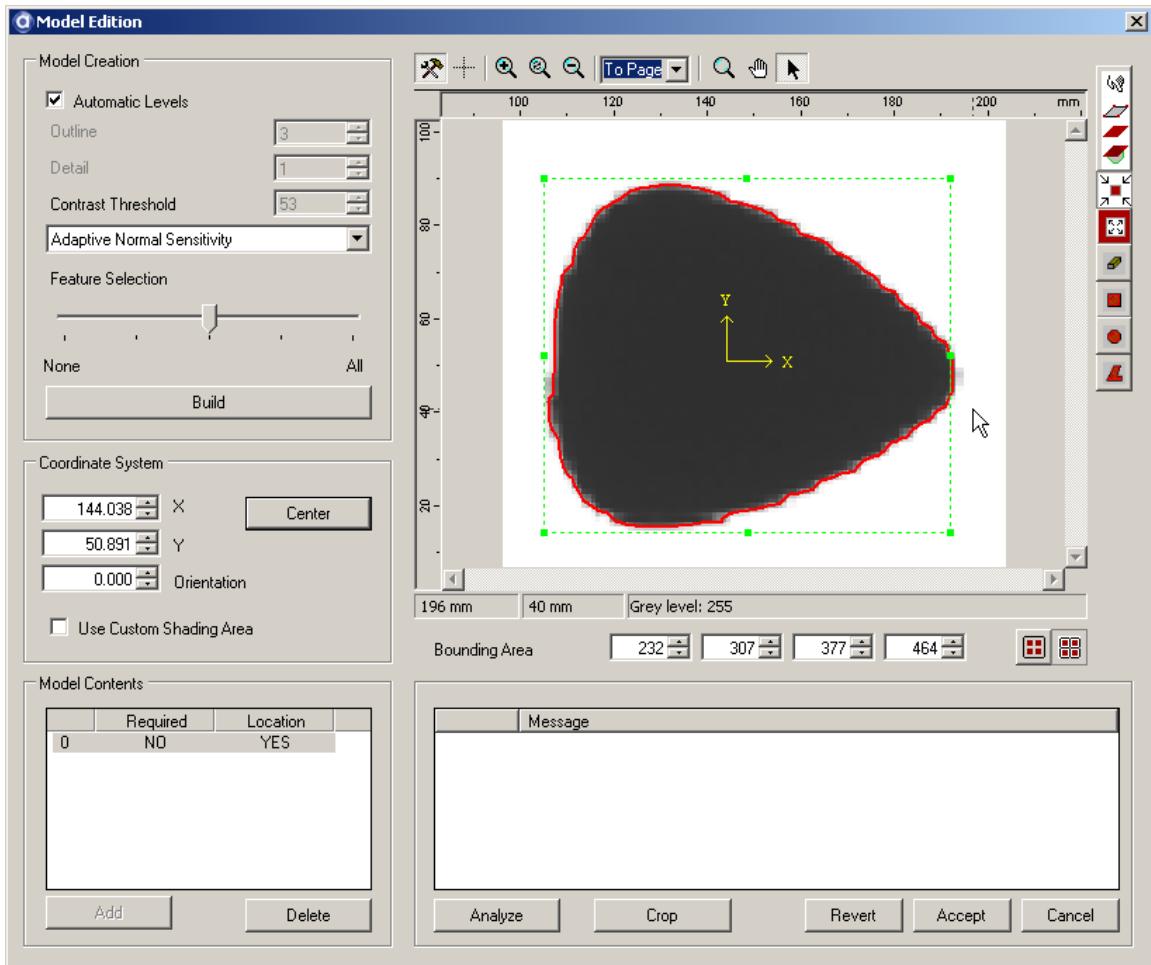
## Cropping the Model

You can crop the image containing the model to just the bounded area by clicking Crop. This eliminates any instances that are near the model, but not actually a part of the model. See the following figure.

---

## Locator Model

---



*Locator Model, After Cropping*

## Setting Advanced Properties in the Model Editor

### Use Custom Shading Area

Enabling the Use Custom Shading Area check box allows you to manually define an area of the model that the Locator will use for Shading Consistency analysis.

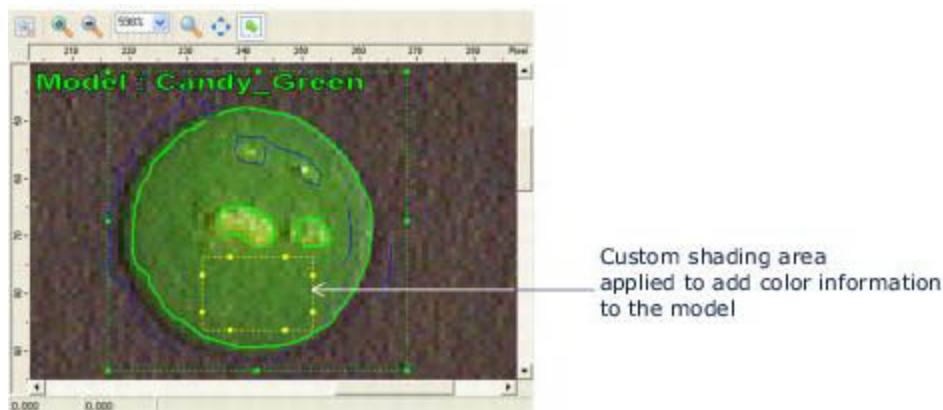
- The Locator analyzes shading consistency by comparing the custom area in the Model to the corresponding area on a found instance.
- A Custom Shading Area is used by the Locator when the Instance Ordering parameter is set to Shading Consistency. If Use Custom Shading Area is not enabled, and

Instance Ordering set to Shading Consistency, the Locator uses the entire Model area for shading analysis.

- Shading Consistency must be enabled to create Models that are based on color. In such a case, the shading consistency analysis can help to discriminate between objects that are very similar in color. For details on creating color models and configuring a color Locator, see Configuring a Color Locator Tool.

To set a custom shading area:

1. Enable the Use Custom Shading Area check box. This will display a yellow bounding box in the display.
2. Use the mouse to drag the shading area bounding box to the appropriate area on the model. The bounding box cannot be rotated, only displaced and resized in the X-Y directions.



*Applying a Custom Shading Area to Add Color Information to a Model*

### **Related Topics**

[Locator Tool](#)

[Configuring Locator Properties - Advanced](#)

### **Pattern Locator**

The Pattern Locator identifies instances of a pattern occurring within an Image.

The Pattern Locator is best suited for applications that require the detection of low contrast and/or small features such as letters, numbers, symbols and logos on a part. Patterns that can provide high-contrast and well-defined contours should be modeled and found by a Locator tool.

Typical uses for the Pattern Locator include:

## Pattern Locator

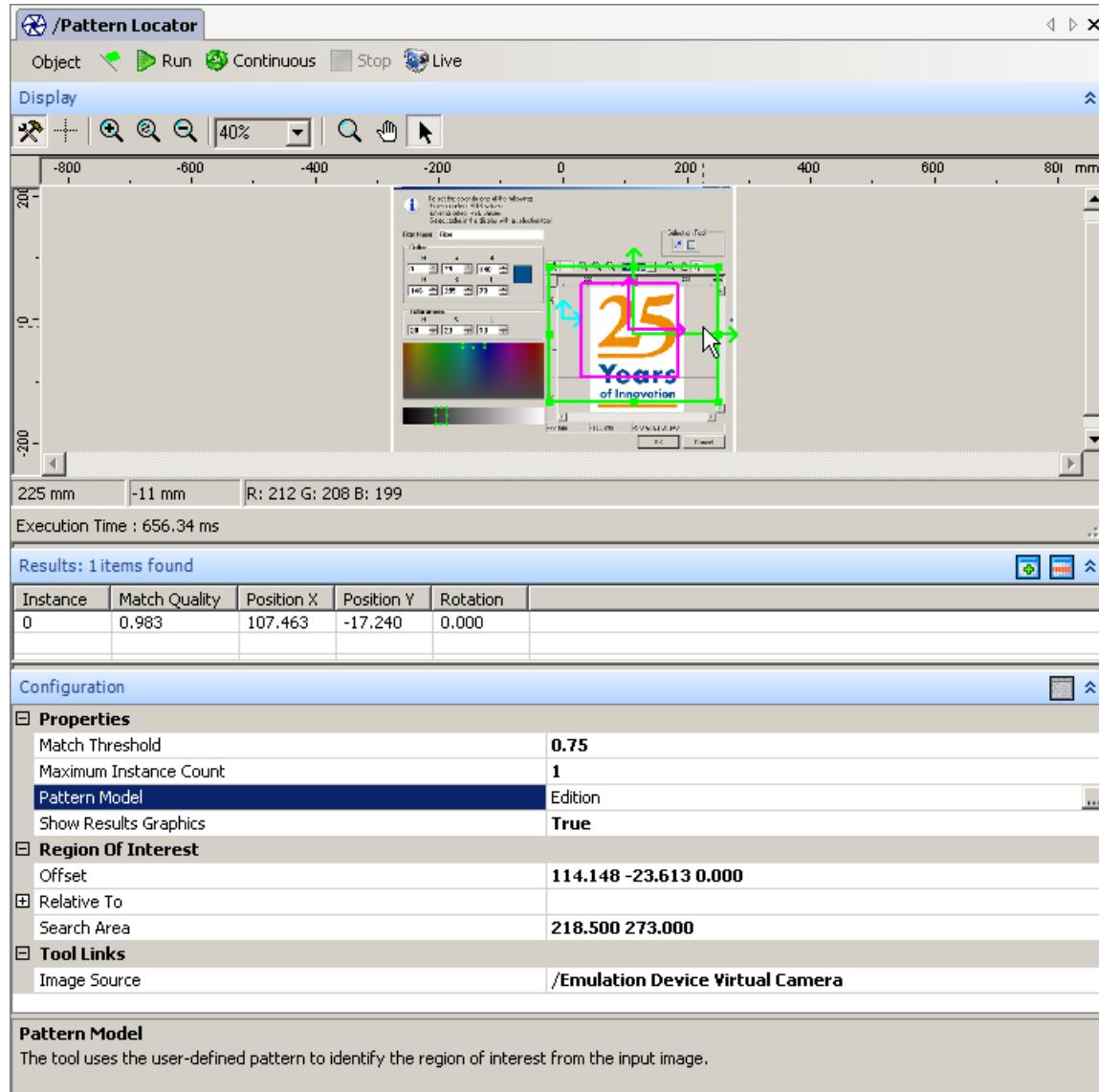
---

- Detecting the presence or absence of a greyscale pattern on a modeled object (Locator).
- Disambiguating objects having the same contours by their greyscale features.

The Pattern Locator does not support rotated patterns and should generally be used as a model-based inspection tool for detecting the presence of small greyscale patterns on small areas in the image or on an object.

To create a Pattern Locator tool, right-click in the Tree structure, then select:

**New > Vision > Tool > Pattern Locator**



*Pattern Locator Object Editor***What is a Pattern?**

A pattern is defined as grid of pixels having a specific arrangement of grey-level values. A pattern must be created for each Pattern Locator tool.

***Properties***

Object	Definition
<b>Properties</b>	
Match Threshold	Sets the Match Threshold of the Instance found by the tool (0 to 1).
Maximum Instance Count	Sets the maximum number of instances that the tool returns.
Pattern Model	The tool uses the user-defined pattern to identify the region of interest from the input image.
Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
<b>Region of Interest (ROI)</b>	
Offset	Specifies the center and rotation of the ROI. X, Y, Degrees.
Relative To	Specifies the tool that will provide the input to this tool. The tool can be dragged from the Folder pane and dropped into this field.
Search Area	Defines the height and width of the region of interest.
<b>Advanced</b>	
Bilinear Interpolation Enabled	Specifies if bilinear interpolation is used to sample the input image. By default, bilinear interpolation is enabled because it improves accuracy.
Custom Sampling Step	When enabled, the tool uses the user-defined sampling step instead of the default optimal sampling step to sample the region of interest from the input image.

**Results**

Object	Definition
Instance	Index of the result instance, starting with 0.
Match Quality	Amount of matched pattern contours for the selected object instance.
Position X	The X coordinate of the origin of the located instance.
Position Y	The Y coordinate of the origin of the located instance.
Rotation	The angle of the located instance.

**Related Topics**

[Configuring Pattern Locator Properties](#)

[Configuring Pattern Locator Properties - Advanced](#)

[Pattern Locator Results](#)

***Configuring Pattern Locator Properties*****Match Threshold**

Match Threshold sets the minimum match strength required for a pattern to be recognized as valid. A perfect match value is 1.

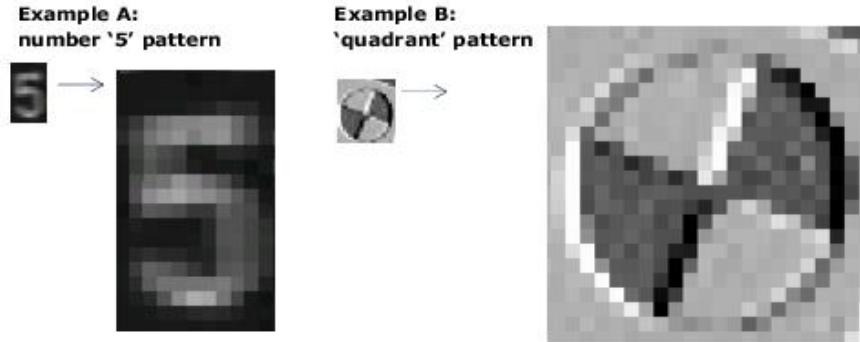
- If the match threshold is too high, many pattern instances may be rejected.
- If the match threshold is too low, too many false pattern instances may be detected.

**Maximum Instance Count**

Maximum Instance Count sets the maximum number of instances that the Pattern Locator will search for. You should set this value to no more than the expected number of instances.

---

## Pattern Model



### Pattern Examples

## Basic Steps for Configuring a Pattern Locator

1. Select the tool that will provide input images.
2. Position the Pattern Locator tool.
3. Create and edit a Pattern.
4. Test and verify the results.
5. Configure the Advanced properties, if required.

## Input

The Input required by the Pattern Locator is an image provided by another tool, such as the Image Processing Tool.

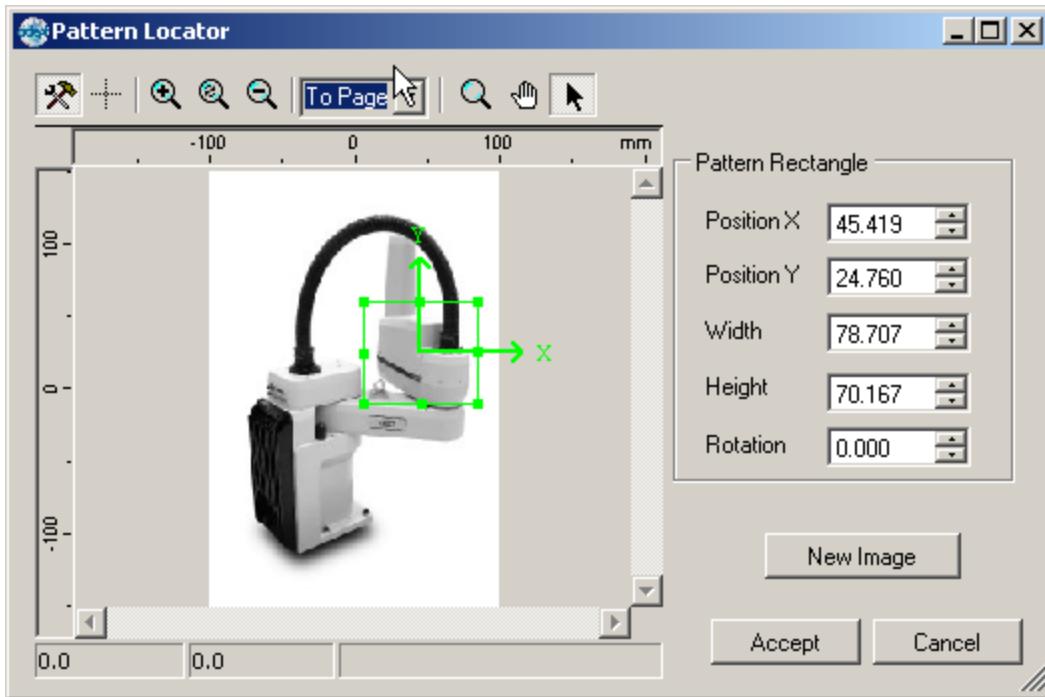
### To set the Input:

1. Execute the tool once to make sure that an input image is available.
2. From the Input drop-down list, select the tool that will provide the input image.

## Location

Location parameters define the position of the tool's region of interest in which the tool carries out its process. The positioning mode is defined by the Frame Input parameter.

---



*Positioning the Pattern Locator Tool*

### **Creating the Pattern Model**

Each Pattern Locator tool can store a single pattern model. This model will be saved when you save the tool.

The Pattern Locator searches for the pattern model within the tool region of interest, but does not search for rotated patterns.

The pattern model can be created on any image that contains the required pattern.

- The pattern model does not have to be created from a pattern that is in the tool region of interest.
- The pattern model can be on any image that contains a pattern.
- The rotation (orientation) and size of the pattern model affect the success of the pattern finding process.
- Creating a pattern "destroys" an existing pattern.
- To erase the current pattern and create a new one, click New.
- To edit or reposition an existing sample pattern, click Edit.

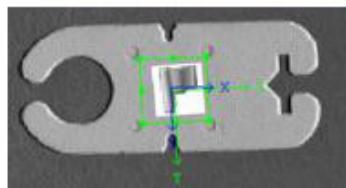
**To create and position a pattern model:**

1. Click New. This opens the Pattern Edition mode.
2. Enter values in the Location dialog, or use the mouse to configure the bounding box in the display.

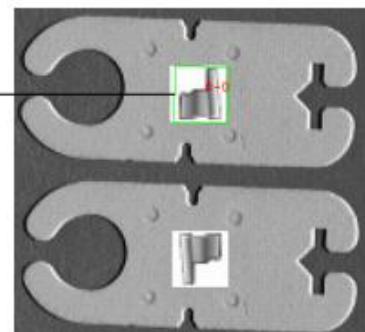
**NOTE:** Correct size and rotation are critical to ensure successful finding of patterns:

- The bounding box should be just large enough to encompass the pattern.
- The X-Y axes marker defines the orientation of the pattern. Make sure the X-Y axes of the Pattern region of interest are aligned in the correct orientation with respect to the Pattern Locator region of interest. See the following figure.

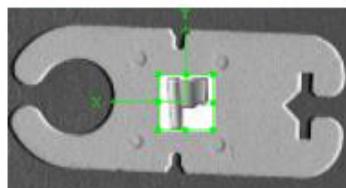
Pattern Locator Region of Interest



Found Pattern



Pattern Region of Interest



Pattern is found relative to alignment of Pattern Region of Interest with Pattern Locator Region of Interest

*Setting the rotation of the sample pattern*

**Editing the Pattern**

Once the pattern model is created, it is temporarily saved to memory. The model will be saved when you save the tool. Changes to the pattern model can be made by at any time.

**To edit or modify the pattern model:**

1. Under the Pattern section, click Edit. This opens the existing sample in

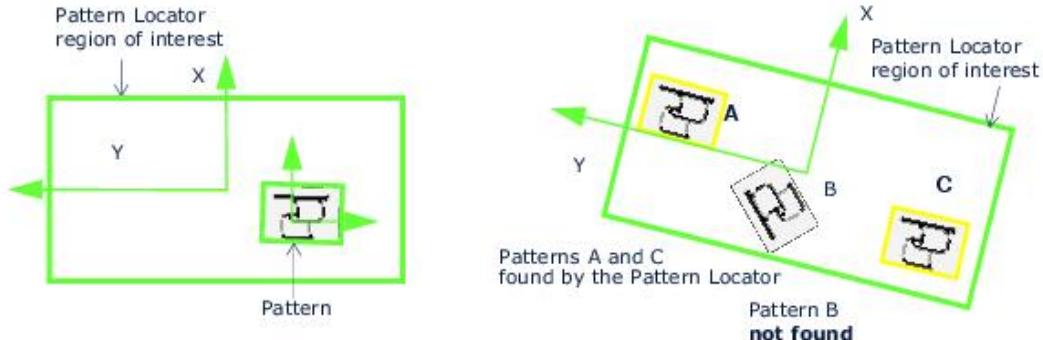
Pattern Edition mode.

2. Enter values in the Location dialog, or use the mouse to configure the bounding box in the display.
3. To change the orientation of the pattern, rotate the X-Y axes marker or enter values in the Rotation text box.

### Orienting Patterns Models

The Pattern Locator finds patterns that are aligned with the Pattern Locator region of interest.

- The axes marker of the Pattern region of interest sets the orientation of the pattern. When the tool searches for pattern instances, it searches for only patterns with X-Y axes that are aligned with the X-Y axes of the tool region of interest.
- Only patterns that are rotated within less than  $\pm 20$  degrees can be found within the region of interest.
- The following figure illustrates an example of a correctly oriented pattern, as well as the effect of the pattern rotation relative to the tool rotation on Pattern Locator results.



*Correct orientation of patterns models*

### Sizing Patterns

The size of the bounding box sets the size of the pattern. The bounding box should be just large enough to contain the pattern. The Pattern Locator only finds patterns that are aligned with the region of interest.

- Patterns that are too large can unnecessarily increase processing time.
- Patterns that are too large can often result in false detections.

- The minimum size of a pattern is fixed as 3x3 pixels.

#### **Related Topics**

[Pattern Locator](#)

[Configuring Pattern Locator Properties - Advanced](#)

[Pattern Locator Results](#)

#### **Configuring Pattern Locator Properties - Advanced**

The Advanced Properties section of the Pattern Locator interface provides access to advanced Pattern Locator parameters and properties.

#### **Tool Sampling Parameters**

Sampling refers to the procedure used by the tool for gathering values within the portion of the input image that is bounded by the tool region of interest. Two sampling parameters, the Sampling Step and Bilinear Interpolation, can be used to create the desired balance between speed and accuracy.

#### **Bilinear Interpolation**

Bilinear Interpolation uses pixel averaging to improve the quality of the image. It is applied before the image is analyzed.

To ensure subpixel accuracy in inspection applications, Bilinear Interpolation should always be set to True (enabled). Non-interpolated sampling (Bilinear Interpolation disabled) should only be used in applications where the speed requirements are more critical than accuracy.

#### **Custom Sampling Step (1 - 100)**

The Sampling Step is the step used by the tool to sample the area of the input image that is bounded by the tool region of interest. The sampling step represents the height and the width of a sampled pixel.

For applications where a more specific tradeoff between speed and accuracy must be established, the sampling step can be modified by enabling the Custom Sampling Step and setting the desired value.

Object	Definition
Default	Default is the best sampling step computed by the tool. It is based on the average size, in calibrated units, of a pixel in the image. This default sampling step is usually recommended. Default is automatically used by the tool if no other value is specified.
Custom	Custom Sampling Step lets you set a sampling step value other than the default. To select a custom sampling step, Enable must be checked. <ul style="list-style-type: none"><li>• Increasing the sampling step value reduces the tool accuracy and decreases the execution time.</li><li>• Reducing the sampling step can increase the tool accuracy but will also increase the execution time.</li></ul>

### **Region of Interest (ROI)**

The region of interest is the area in which the tool carries out its process. Most tool position parameters can be set through the region of interest section of the tool interface. The following parameters define the tool region of interest.

#### **Offset**

Offset is the center of the ROI, defined by:

X	X coordinate of the center of the region of interest
Y	Y coordinate of the center of the region of interest
Degrees	Angle of rotation of the region of interest

#### **Relative To**

Relative To specifies the tool that will provide the input to this tool. The tool can be dragged from the Tree structure and dropped into this field.

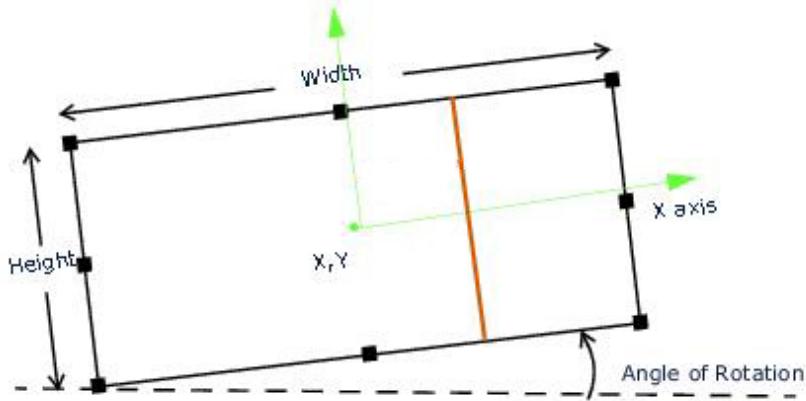
#### **Search Area**

Search Area is the size of the region of interest is defined by:

---

---

Height	Height of the region of interest
Width	Width of the region of interest



#### *Location Properties for the Region of Interest*

### **Results Logging**

If enabled, the results of the tool will be logged to a file.

If the extension of the destination file is csv, then the log is written in csv format, which opens with Excel. If a different extension is selected, the log is written in a plain text format (similar to the ACE Sight 2 format).

The size of the log file can be set (default size is 5 Mb).

#### ***Related Topics***

[Pattern Locator](#)

[Configuring Pattern Locator Properties](#)

[Pattern Locator Results](#)

#### ***Pattern Locator Results***

### **Display Window**

The Display window shows the image being processed, the region of interest selected, and the locations of found instances. The elapsed time is displayed at the bottom of this window.

## Grid of Results

The grid of result presents the statistical results for the region of interest analyzed by the Pattern Locator.

Results are presented below in the order in which they are output.

Object	Definition
Elapsed Time	The Elapsed Time is not visible in the results grid but is output to the results log for each iteration of the Pattern Locator.
Instance	The Instance is the index number of the located pattern instance, starting at 0. Each pattern instance outputs a frame that can be used by a frame-based tool for which the Pattern Locator is a frame-provider.
Frame/Group	The frame or group to which the result belongs.
Match Quality	The Match Quality value ranges from 0 to 100, with 100 being the best quality. A value of 100 means that 100% of the Pattern Model was successfully matched to the found pattern instance.
Position X	Position X is the X coordinate of the center of the Pattern region of interest, with respect to the selected Coordinate System.
Position Y	Position Y is the Y coordinate of the center of the Pattern region of interest, with respect to the selected Coordinate System.
Rotation	Rotation is the rotation of the Pattern Locator region of interest, with respect to the selected Coordinate System. Rotation IS NOT calculated for individual patterns.

### ***Related Topics***

[Pattern Locator](#)

[Configuring Pattern Locator Properties](#)

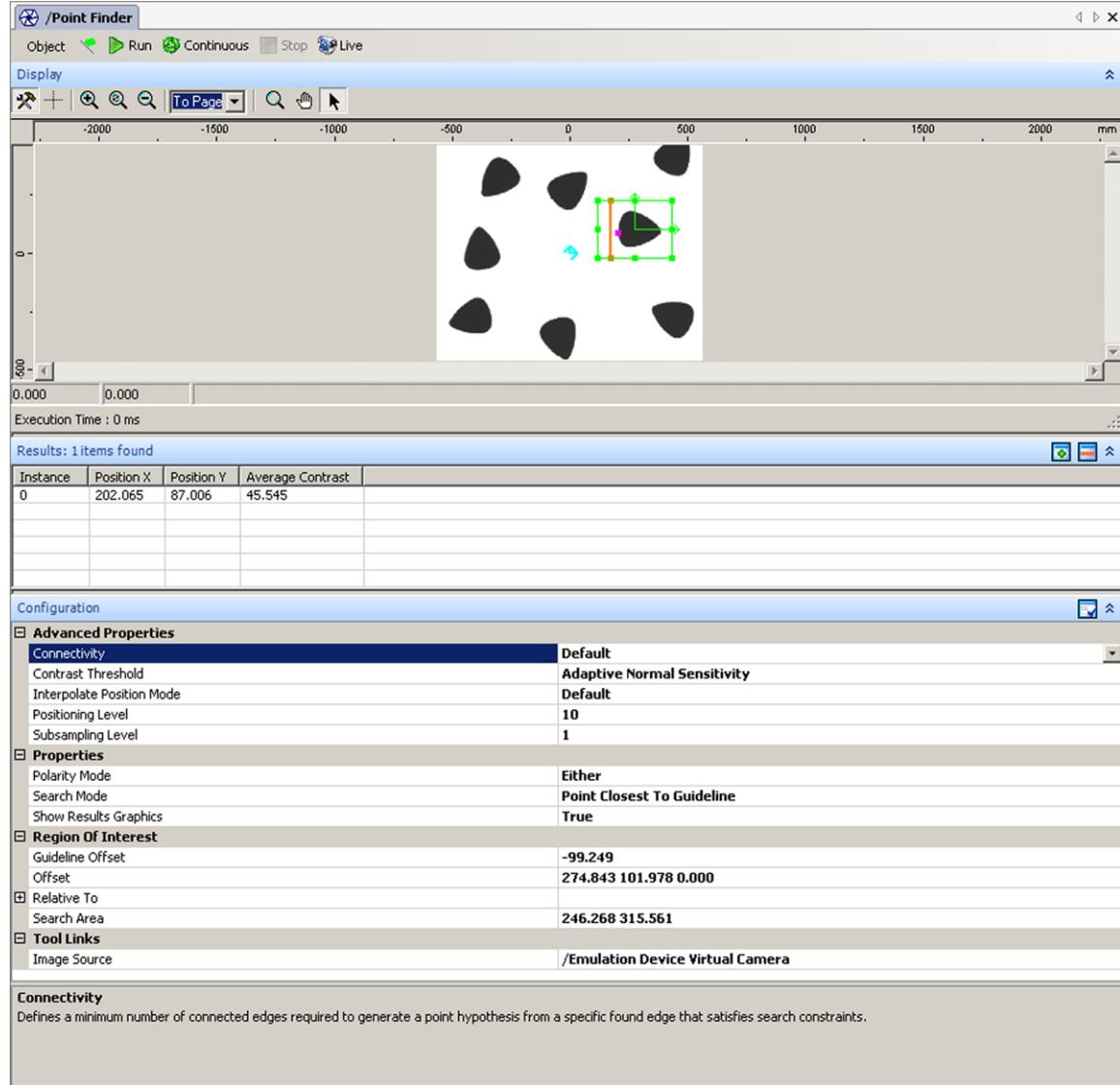
[Configuring Pattern Locator Properties - Advanced](#)

## Point Finder

The Point Finder finds point-like features on objects and returns the coordinates of the found points.

To create a Point Finder tool, right-click in the Tree structure, then select:

**New > Vision > Tool > Point Finder**



*Point Finder Object Editor*

***Configuration***

Object	Definition
<b>Properties</b>	
Polarity Mode	Selects the type of polarity accepted for finding a line. Polarity identifies the change in grey-level values in the positive direction along the tool X axis.
Search Mode	Specifies the method used by the tool to select a hypothesis.
Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
<b>Tool Links</b>	
Image Source	Defines the image source used for processing by this vision tool.
<b>Region of Interest (ROI)</b>	
Guideline Offset	Defines the offset, from the center of the region of interest, for the guideline.
Offset	Specifies the center and rotation of the ROI. X, Y, Degrees.
Relative To	Specifies the tool that will provide the input to this tool. The tool can be dragged from the Folder pane and dropped into this field.
Search Area	Defines the height and width of the region of interest.
<b>Advanced Properties</b>	
Connectivity	Specifies the minimum number of connected edges required to generate a point hypothesis.
Contrast Threshold	Defines the minimum contrast needed for an edge to be detected in the input image.
Interpolate Position Mode	Sets the mode used by the Point Finder to compute a point hypothesis
Positioning Level	Sets the configurable effort level of the entity-positioning process.
Subsampling Level	Set the subsampling level used to detect edges that are used by the tool to generate hypotheses.

**Results**

Object	Definition
Instance	Index of the result instance.
Frame/Group	The frame or group to which the result belongs.
Position X	The X coordinate of the located instance.
Position Y	The Y coordinate of the located instance.
Average Contrast	Average contrast between light and dark pixels on either side of the found entity (point, line, or arc), expressed in grey-level values.

**NOTE:** The tool searches for a point on an edge that is parallel to the Y-Axis, moving through the region of interest in a negative-to-positive direction relative to the X-Axis. Best results are generally obtained by when the guideline is placed on, or very close to, the point to be found. Use Search and Edge Detection parameters (Advanced Parameters) to further configure and refine the finding of the correct point entity.

**Related Topics**

[Configuring Point Finder Properties](#)

[Configuring Point Finder Properties - Advanced](#)

**Configuring Point Finder Properties**

The Properties section of the Point Finder tool interface provides access to Point Finder parameters and properties.

**Polarity Mode**

PolarityMode sets the mode that will apply to the search for entities. Polarity identifies the change in grey-level values along the tool X axis, in the positive direction.

The available modes are

- Dark To Light: The Point Finder searches only for point instances occurring at a dark to light transition in grey-level values.
- Light To Dark: The Point Finder searches only for point instances occurring at a light to dark transition in grey-level values.
- Either: The Point Finder searches only for point instances occurring

either at a light to dark or dark to light transition in grey-level values. This mode will increase processing time.

- Don't Care: The Point Finder searches only for point instances occurring at any transition in grey-level values, including reversals in contrast, for example, on an unevenly colored background.

### **Search Mode**

Search Mode sets the mode used by the tool to generate and select a hypothesis.

The available mode are:

- Point Closest To Guideline: Selects the point hypothesis closest to the Guideline.
- Point With Maximum Negative X Offset: Selects the point hypothesis closest to the region of interest boundary that is at maximum negative X offset.
- Point With Maximum Positive X Offset: Selects the point hypothesis closest to the region of interest boundary that is at maximum positive X offset.

### **Region of Interest (ROI)**

The region of interest is the area in which the tool carries out its process. Most tool position parameters can be set through the region of interest section of the tool interface. The following parameters define the tool region of interest.

#### **Offset**

Offset is the center of the ROI, defined by:

X	X coordinate of the center of the region of interest
Y	Y coordinate of the center of the region of interest
Degrees	Angle of rotation of the region of interest

#### **Relative To**

Relative To specifies the tool that will provide the input to this tool. The tool can be dragged from the Tree structure and dropped into this field.

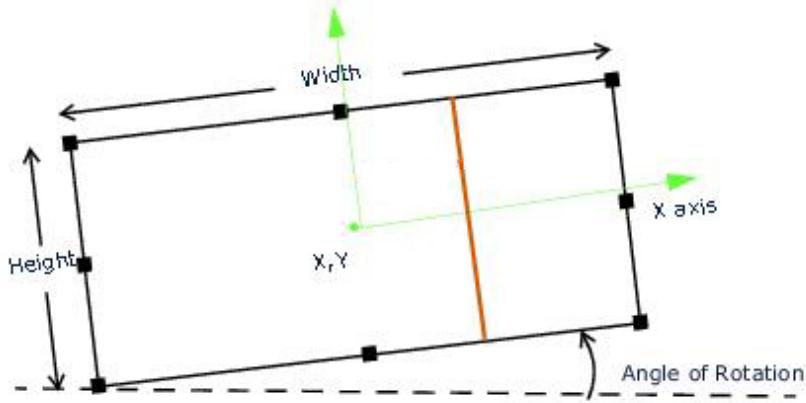
#### **Search Area**

Search Area is the size of the region of interest is defined by:

---

---

Height	Height of the region of interest
Width	Width of the region of interest



#### *Location Properties for the Region of Interest*

#### **Related Topics**

[Point Finder](#)

[Configuring Point Finder Properties - Advanced](#)

#### ***Configuring Point Finder Properties - Advanced***

The Advanced Properties section of the Point Finder tool interface provides access to advanced Point Finder parameters and properties.

#### **Connectivity**

Connectivity is the minimum number of connected edges required to generate a point hypothesis.

By default, Connectivity is disabled. When enabled, you can set the minimum number of connected edges that are required to generate a point hypothesis.

#### **Contrast Threshold Mode**

Contrast Threshold defines how contrast threshold is set. Contrast threshold is the level of sensitivity that is applied to the detection of edges in the input image. This threshold is expressed in terms of a step in grey-level values. The contrast threshold can be either Adaptive or Fixed.

Adaptive thresholds set a sensitivity level based on image content. This provides flexibility to variations in image lighting conditions and variations in contrast during the Search process.

- Adaptive Low Sensitivity uses a low sensitivity adaptive threshold for detecting edges. It detects strongly defined edges and eliminates noise, at the risk of losing significant edge segments.
- Adaptive Normal Sensitivity sets a default sensitivity threshold for detecting edges.
- Adaptive High Sensitivity detects a great amount of low-contrast edges and noise.
- Fixed Value sets an absolute value for the sensitivity to contrast. A typical situation for the use of a fixed value is a setting in which there is little variance in lighting conditions.

Fixed Value Contrast Threshold sets the minimum contrast needed for an edge to be detected in an input image. The threshold value expresses the step in grey-level values required to detect edges.

- Higher values reduce sensitivity to contrast. This reduces noise and the amount of low-contrast edges.
- Lower values increase sensitivity and add a greater amount of edge at the expense of adding more noise. This may generate false detections and/or slow down the search process.

### **Interpolate Position**

Interpolate Position sets the mode used by the tool to compute a point hypothesis. By default, Interpolate Position is disabled. When Enabled, you can select one of the following modes

- Corner: The tool will compute a hypothesis that fits a corner point to interpolated lines from connected edges.
- Intersection: The tool will compute a hypothesis that is an intersection between the search axis and connected edges of an interpolated line.

### **Positioning Level**

Positioning Level sets the effort level of the instance positioning process. A value of 0 will provide coarser positioning and lower execution time. Conversely, a value of 10 will provide high accuracy positioning of Point instances.

### **Subsampling Level**

Subsampling Level sets the subsampling level used to detect edges that are used by the tool to generate hypotheses.

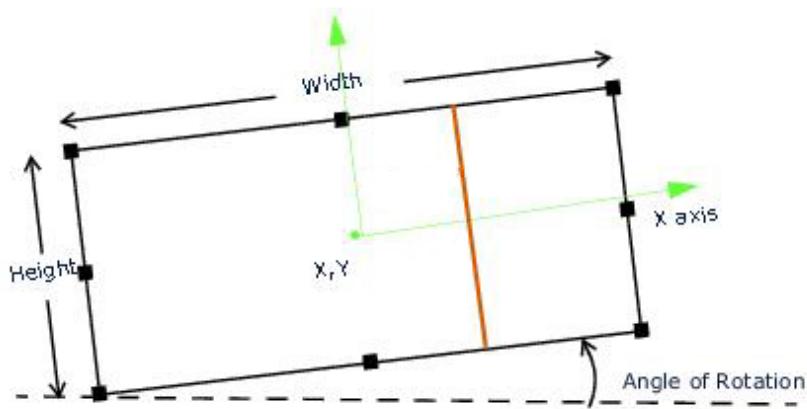
- High values provide a coarser search with a shorter execution time.
- Lower values can provide a more refined search with slower execution time.

- A higher subsampling value may help improve accuracy in blurry images.

### ***Region of Interest (ROI)***

Most tool position parameters can be set through the ROI section of the tool interface. The following parameters define the tool region of interest.

Object	Definition
<b>Offset</b>	Offset is the offset from the tool X-axis, defined by:
X	The X coordinate of the center of the region of interest.
Y	The Y coordinate of the center of the region of interest.
Degrees	The angle of rotation of the region of interest.
<b>Relative To</b>	Specifies the tool that will provide the input to this tool. The tool can be dragged from the Folder pane and dropped into this field.
<b>Search Area</b>	The tool region of interest is the rectangle that is defined by:
Height	The height of the region of interest.
Width	The width of the region of interest.



*Location Properties for the Region of Interest*

### ***Guideline Offset***

The Guideline Offset is the offset from the tool Y-axis. The Guideline marker can be displaced along the X-axis. This marker acts as both a visual guide for positioning the tool and as a constraint for the tool Search Mode.

## **Properties**

### **Polarity Mode**

Polarity Mode sets the mode that will apply to the search for entities. Polarity identifies the change in grey-level values along the tool X axis, in the positive direction.

The available modes are

- Dark To Light: The Point Finder searches only for point instances occurring at a dark to light transition in grey-level values.
- Light To Dark: The Point Finder searches only for point instances occurring at a light to dark transition in grey-level values.
- Either: The Point Finder searches only for point instances occurring either at a light to dark or dark to light transition in grey-level values. This mode will increase processing time.
- Don't Care: The Point Finder searches only for point instances occurring at any transition in grey-level values including reversals in contrast, for example, on an unevenly colored background.

### **Search Mode**

Search Mode sets the mode used by the tool to generate and select a hypothesis.

The available modes are:

- Point Closest To Guideline: Selects the point hypothesis closest to the Guideline.
- Point With Maximum Negative X Offset: Selects the point hypothesis closest to the region of interest boundary that is at maximum negative X offset.
- Point With Maximum Positive X Offset: Selects the point hypothesis closest to the region of interest boundary that is at maximum positive X offset.

### **Results Logging**

If enabled, the results of the tool will be logged to a file.

---

If the extension of the destination file is csv, then the log is written in csv format, which opens with Excel. If a different extension is selected, the log is written in a plain text format (similar to the ACE Sight 2 format).

The size of the log file can be set (default size is 5 Mb).

***Related Topics***

[Point Finder](#)

[Configuring Point Finder Properties](#)

**Remote Vision Tool**

The Remote Vision tool lets an application, such as an ACE PackXpert packaging application, run a vision tool on a remote PC.

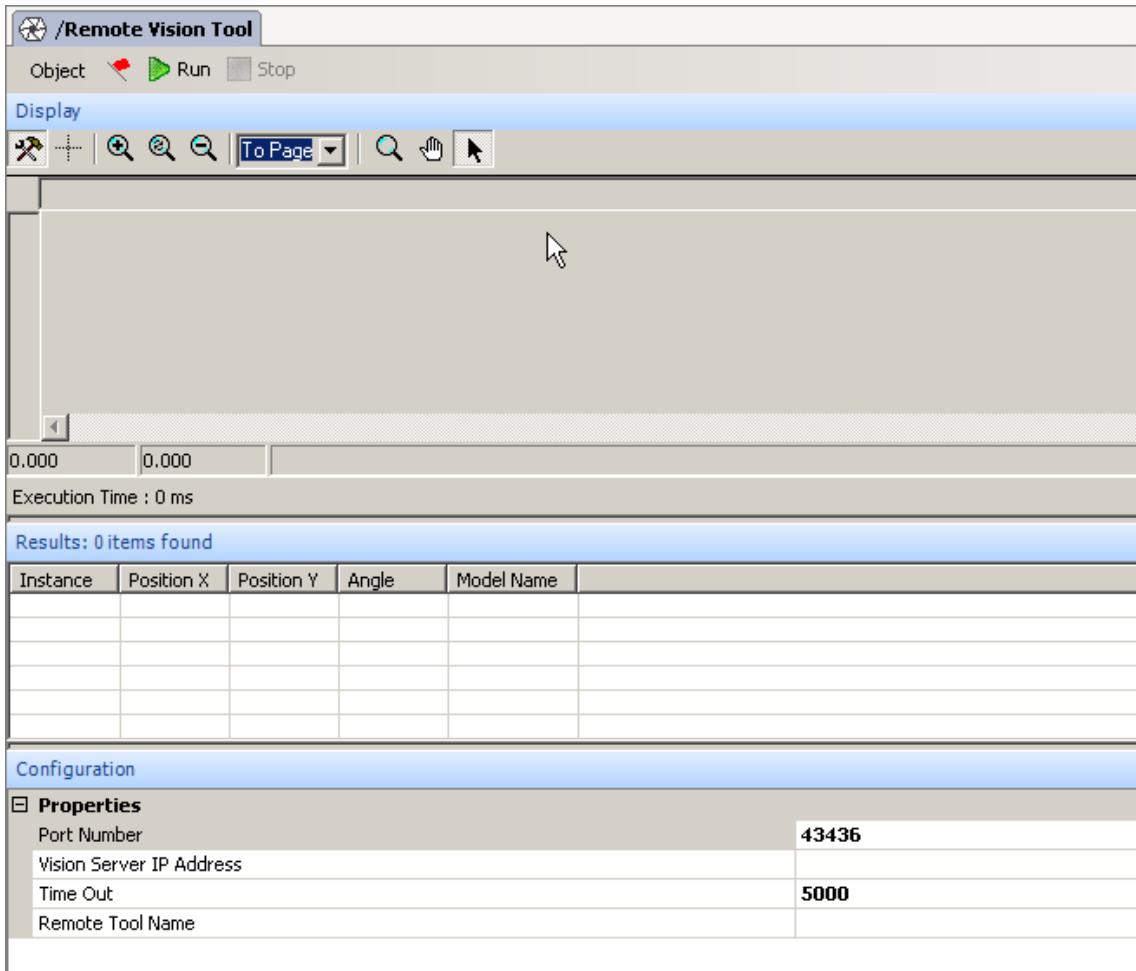
Because image processing is computationally intense, this can offload a significant portion of the processing load from the main ACE PC to a remote PC.

To create a Remote Vision tool, right-click in the Tree structure, then select:

**New > Vision > Tool > Remote Vision Tool**

## Remote Vision Tool

---



### Properties

Object	Definition
Port Number	The TCP/IP port used to communicate with the Remote Vision Server.
Vision Server IP Address	The IP address of the Remote Vision Server running the vision tool.
Time Out	The maximum time duration (in milliseconds) allowed for the remote operation to complete.
Remote Tool Name	The name of the vision tool to be activated remotely.

Remote Vision Tool Object Editor

---

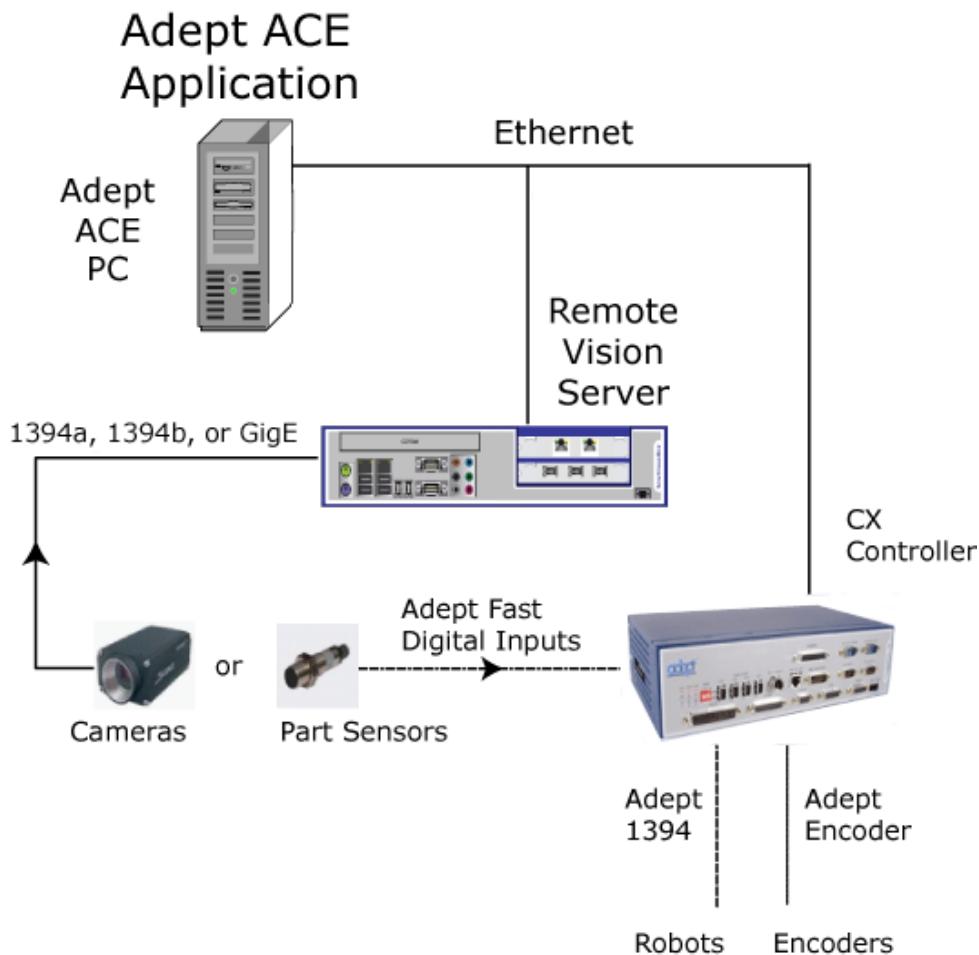
**Results**

Object	Definition
Instance	The index of the located instance.
Position X	The X coordinate of the located instance.
Position Y	The Y coordinate of the located instance.
Angle	The angle of the located instance.
Model Name	Name of the model located.

The ACE PC can run multiple instances of the Remote Vision tool, offloading multiple vision tools to a remote PC. The remote PC returns the results of the vision tool to the ACE PC.

Offloading the image processing to a remote PC allows a single instance of ACE, on the main PC, to manage larger ACE applications.

- The ACE PC is the main PC; it runs the ACE application. This is required for all ACE installations.
- The Remote Vision Server is an optional, secondary PC, used to off-load processing of vision tools.
  - The Remote Vision Server is passive: it waits for a request from the ACE application to run a vision tool.
  - The Remote Vision Server requires a separate USB dongle to run vision tools.
  - Any tool that can be run on the ACE PC can be run on the Remote Vision Server.
  - Multiple Remote Vision Servers can be controlled from the ACE PC (on multiple remote PCs).

*Remote Vision System Components*

### ***Basic Steps for Configuring Remote Vision***

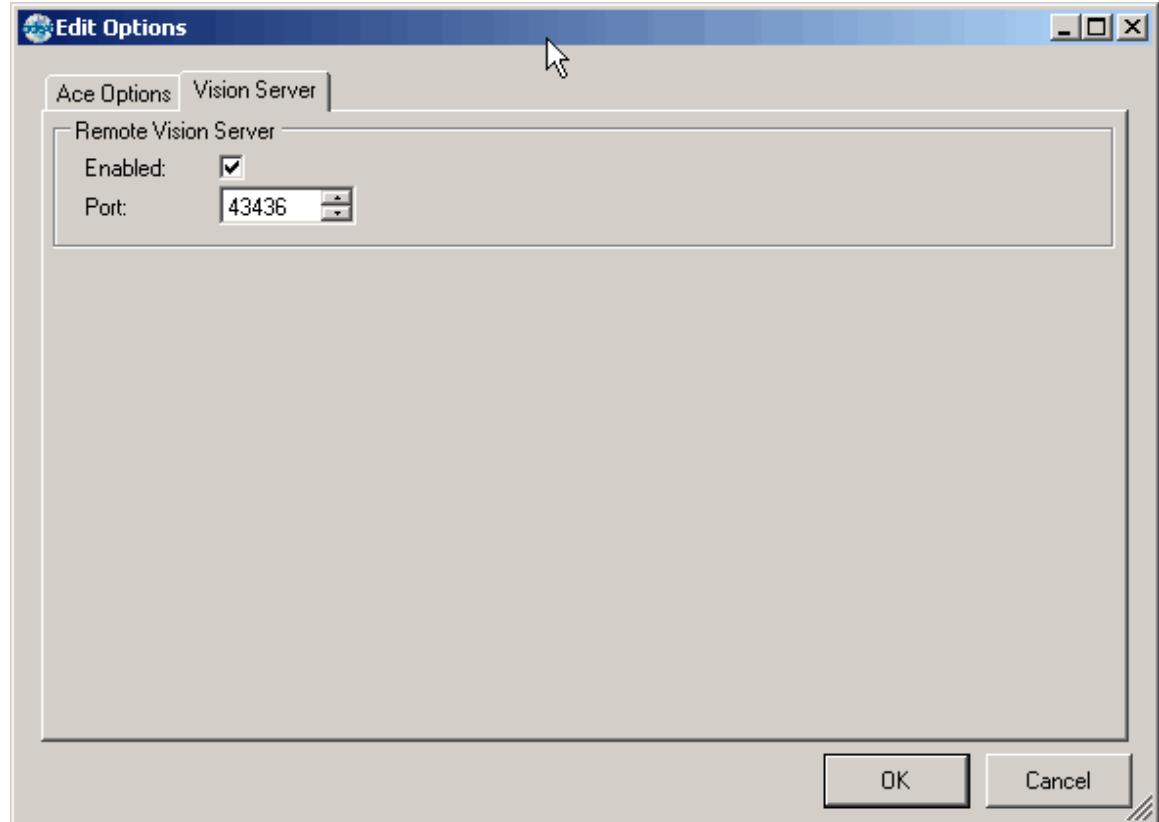
Prerequisites:

- ACE must be loaded on both the ACE PC and the Remote Vision Server.
- Both the ACE PC and the Remote Vision Server must have a separate, vision-enabling USB dongle.
- Vision tools running on a Remote Vision Server can only access cameras physically attached to that remote server.

On the Remote Vision Server:

From the main ACE screen:

1. Select **Tools > Options . . . > Vision Server**



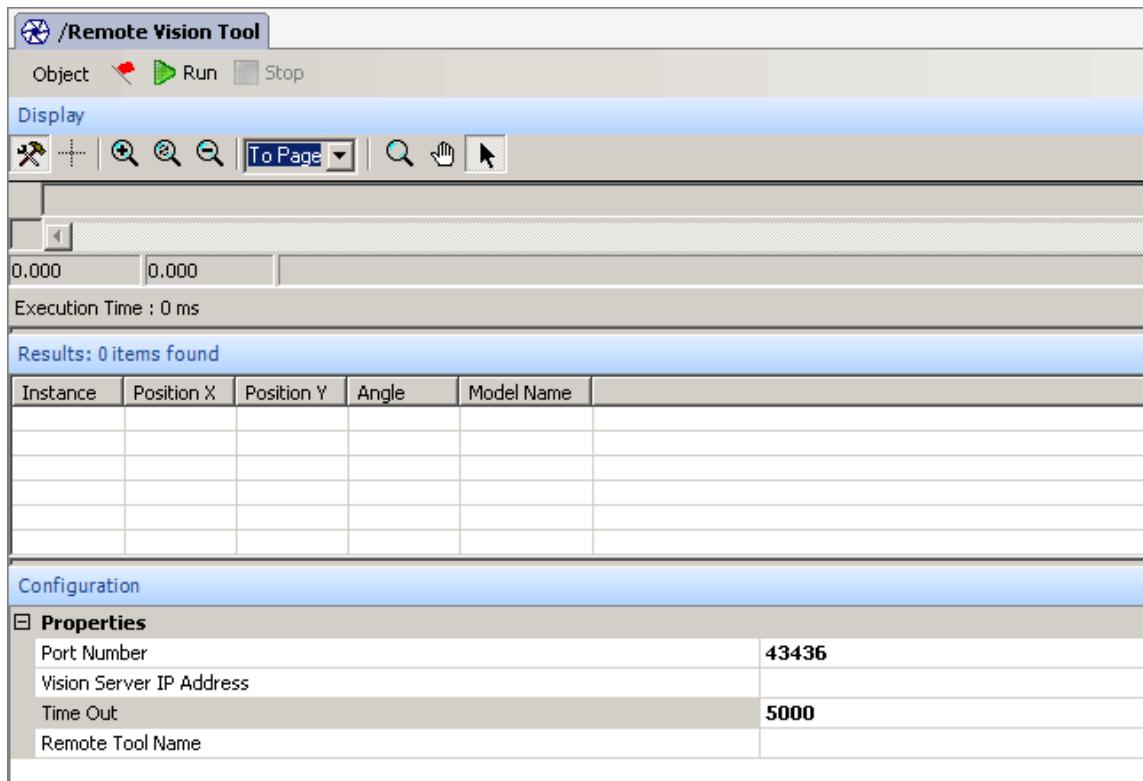
*Vision Server Port Selection*

2. Check the Enabled box.  
The default port address will be displayed.
3. Click OK.

On the ACE (non-remote) PC:

1. Add a Remote Vision tool.

A configuration screen will be displayed:

*Remote Vision Configuration Screen*

2. In the Port Number field, enter the port number of the Remote Vision Server.
3. In the Vision Server IP Address field, enter IP address of the Remote Vision Server.  
The Remote Tool Name list will be populated with the ACE Sight vision tools available on the Remote Vision Server.
4. In the Time Out field, specify a time out, in milliseconds. This is the maximum time to wait for the remote tool to complete its processing.
5. In the Remote Tool Name field, select the vision tool to run on the remote PC.  
The field will be populated with the available tools on the server.
6. Run the Remote Vision tool (click Run).

The results of the remotely-run vision tool will be transmitted from the Remote Vision Server to the ACE PC when that tool is run, through the Remote Vision tool.

- When the Remote Vision tool is run from the Remote Vision tool editor, the image will be displayed on the editor Vision Window.
- When run from an ACE application, such as the ACE PackXpert, the image will not be displayed on the Vision Window.

**NOTE:** When running a vision tool through the Remote Vision tool, Live mode will not be transmitted to the ACE PC. Any configuration that needs to be performed in Live mode, such as camera focusing, should be done on the Remote Vision Server before running the Remote Vision tool.

## ACE Sight Special Tools

### ***ACE Sight Sequences***

ACE Sight Sequences let you see the order and dependency of vision tools that will be executed, and give V+ and MicroV+ a means for retrieving results from vision tools.

The ACE Sight Sequence is an object in the ACE Workspace Explorer.

See "ACE Sight Sequence"

### ***Communication Tool***

The Communication Tool is a tool for conveyor tracking applications. The purpose of the Communication Tool is to provide instructions to the controller for handling parts that must be picked or manipulated by a robot.

The Communication Tool processes the input instances, typically from an Overlap Tool, by applying region-of-interest parameters. It acts as a filter:

- Instances that are passed by the tool are sent to the controller queue.
- Instances that are not output to the controller, because they are outside the region of interest, or because the queue is full, are rejected. These instances are passed to the next tool in the sequence.

See "Communication Tool".

### ***Overlap Tool***

The purpose of the Overlap Tool is to make sure that parts moving on a conveyor belt are recognized only once.

Because a part found by an input tool may be present in multiple images acquired by the camera, the Overlap Tool is needed to ensure that the robot is not instructed to pick up or process the same part more than once.

See "Overlap Tool".

### ***Gripper Offset Tool***

The Gripper Offset Tool defines where on the part a robot can pick up a part, giving the relationship between the pick point, the part model, and the robot flange center.

See "Gripper Offset".

### ***Related Topics***

[Communication Tool](#)

[Communication Tool Configuration](#)

[Overlap Tool](#)

[ACE Sight Sequences](#)

[Gripper Offset Tool](#)

## ACE Sight Sequence

ACE Sight Sequences let you see the order and dependency of vision tools that will be executed, and give V+ and MicroV+ a means for retrieving results from vision tools.

The ACE Sight Sequence is an object in the ACE Workspace Explorer.

**NOTE:** This is not the same as the ACE Sight: Sequencing, which builds a control sequence for controlling robots in a workspace. The object built by the Sequencer will usually include the appropriate ACE Sight Sequence.

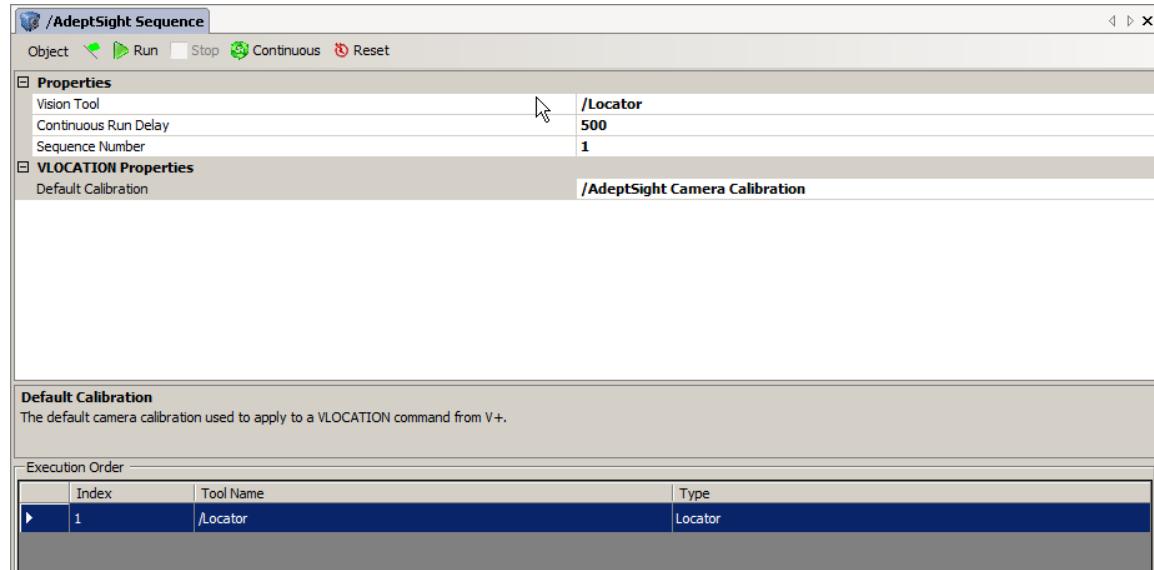
It is configured using its object editor.

1. To add an ACE Sight Sequence, right-click in the Tree structure of the Workspace Explorer and select:

### New > Vision > ACE Sight > ACE Sight Sequence

The object is added to the ACE workspace and the ACE Sight Sequence object editor opens.

2. Use the controls within the object editor to configure the ACE Sight Sequence for your application.



ACE Sight Sequence Object Editor

### Description

The ACE Sight Sequence is an object in the ACE Workspace Explorer.

The Sequence object shows the list of tools that will be executed as part of the sequence, the order in which they will be executed, and the Index associated with each one. The Index is the execution order of each tool.

A sequence maps a V+ sequence number to the "top-level" vision tool. "Top-level" means the vision tool with the highest number in the sequence, which will be the last tool executed. This enables you to access an ACE Sight sequence with the sequence\_id parameter from a V+ program instruction or function.

Note that the sequence cannot be modified from the Sequence Display window. It just shows the order in which the tools will be executed, based on the parent tool specified in each tool. The actual order of a sequence is determined when you specify the Related To parameter for each of the tools to be included in the sequence. When you add a Sequence object to the workspace, the Vision Tool parameter determines the Top-Level tool, and all the tools you specified as the Related To parameter in the chain under that will automatically show up as members of the sequence, in the order you set up.

In a sequence, you specify a robot-to-camera calibration (Default Calibration). The calibration is applied to any result accessed by a VLOCATION transformation function.

**NOTE:** V+ can access the results of intermediate (not only top-level) tools when a sequence is executed, because each tool has an index that can be accessed through V+. The Default Calibration is applied to all results, even if they are not the top-level tool.

### Configuration

Object	Definition
<b>Properties</b>	
Vision Tool	Top-level vision tool this sequence references
Continuous Run Delay	Amount of time (in ms) to delay between execution of the vision sequence in continuous mode
Sequence Number	The number associated with the sequence. This sequence number is used in V+ to reference this vision operation.
<b>VLOCATION Properties</b>	
Default Calibration	Default camera calibration used to apply to a VLOCATION transformation function from V+

The object editor provides the following icons:

- Run  
Runs the sequence once.
- Stop  
Stops the sequence, if it is running.
- Run Continuously  
Repeatedly runs the sequence.
- Reset  
Resets the communications queue (flushes it of instances on the PC that have been identified to be sent to V+, but have not actually been sent) and resets the Overlap tool tracking of previously-located instances.

## Communication Tool

The Communication Tool is a tool for conveyor tracking applications. The purpose of the Communication Tool is to provide instructions to the controller for handling parts that must be picked or manipulated by a robot.

**NOTE:** Conveyor tracking requires an Adept SmartController CX motion controller.

The Communication Tool is configured using its object editor.

1. To add a Communication Tool, right-click in the Tree structure of the Workspace Explorer and select:

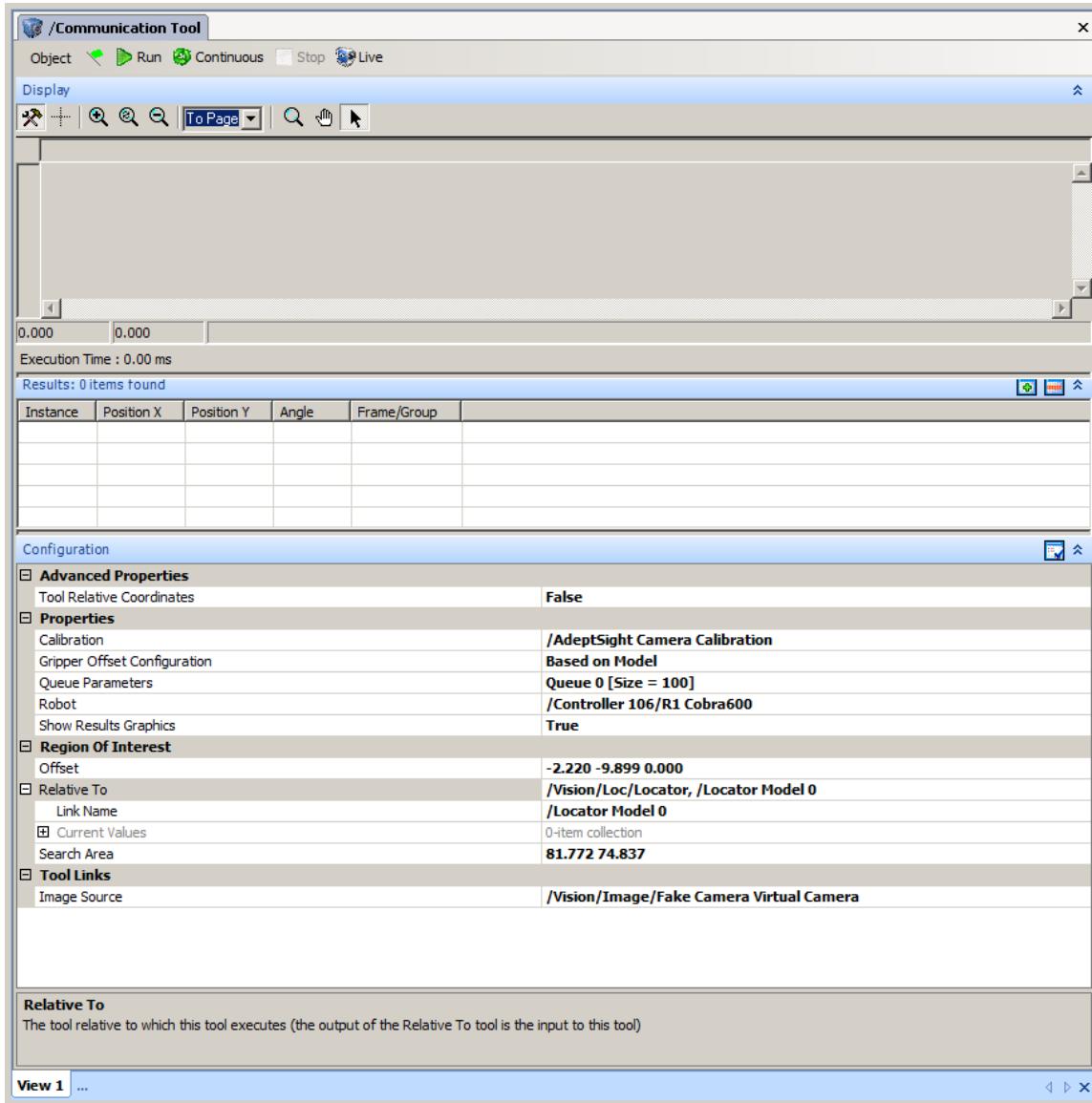
### **New > Vision > ACE Sight > Communication Tool**

The tool is added to the ACE workspace and the Communication Tool object editor opens.

2. Use the controls within the object editor to configure the Communication Tool for your application.

## Communication Tool

---



*Communication Tool Objec Editor*

### **How It Works**

The Communication Tool typically receives instances from an Overlap Tool, which prevents different images of the same instance from being interpreted as different instances. The Communication Tool processes the input instances by applying region-of-interest parameters. The Communication Tool acts as a filter:

- Instances that are passed by the tool are sent to the controller queue.
- Instances that are not output to the controller, because they are outside the region of interest, or because the queue is full, are rejected. These instances are passed to the next tool in the sequence.

### ***Requirements for Using the Communication Tool***

The Communication Tool requires that:

- The camera, robot, and conveyor belt are calibrated.
- The connection to the controller is active.
- The conveyor belt and the controller have been correctly assigned to a camera, in the ACE Sight software vision project.
- A valid Conveyor Tracking License is in place.

### ***Order of the Communication Tool in a Vision Sequence***

In a simple pick-and-place application, one or more Communication Tools are inserted at the end of a sequence, frequently just after the Overlap Tool.

In a sequence that requires inspection of parts before they are picked by a robot, the Communication Tool must be placed after one or more Inspection tools. In such a case, the Inspection tools provide valid instances (parts that have passed inspection) to the Communication Tool.

### ***Multiple Communication Tools***

In many applications, it may be useful to use two or more Communication Tools.

**NOTE:** Each tool must have its "Relative To" property set to the preceding tool, so any parts not queued by one tool are passed to the next tool.

Examples:

- Two Communication Tools handling either side of a conveyor belt. Each Communication Tool sends instances to a robot that picks parts on one side of the belt only.
- Two (or more) Communication Tools so that the subsequent tools can process instances that were rejected by the preceding tools because the queue was full. Each tool will send its passed parts to a different queue, so any parts missed by a robot because its queue is full will be picked by a subsequent robot.

**Properties**

Object	Description
<b>Properties</b>	
Calibration	Specifies the robot-to-camera calibration applied to the results.
Gripper Configuration	<p>Specifies if a gripper offset should be applied to the results. If so, gives the index into the gripper offset table, and specifies whether to use a default value or a value associated with the model being used.</p> <p>The subparameters are:</p> <ul style="list-style-type: none"> <li>• Selection Mode This can be Disabled, Use Default, or Use Model Identifier</li> <li>• Default Offset Index</li> <li>• Model Offset Index</li> </ul> <p>For Model Offset Index (Use Model Identifier), the Gripper Offset table will be displayed, and you will only be allowed to chose a valid index from that table.</p>
Queue Parameters	<p>Specifies the queue the instances are sent to.</p> <p>The subparameters are:</p> <ul style="list-style-type: none"> <li>• Queue Index</li> <li>• Queue Size</li> <li>• Queue Update</li> <li>• Use Soft Signal (on or off)</li> <li>• Soft Signal (value)</li> </ul>
Robot	Specifies which robot the instances will be sent to.
Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
<b>Region of Interest (ROI)</b>	
Offset	Specifies the center and rotation of the ROI. X, Y, Degrees.
Relative To	Specifies the tool that will provide the input to this tool. The tool can be dragged from the Folder pane and dropped into this field.

Object	Description
Search Area	Defines the height and width of the region of interest.
<b>Advanced</b>	
Is Running In Sequence	If the tool is running in a sequence, it will read the required latches and update the internal tracking structures. If it is not running in a sequence, the latches are not read, and the results are not tracked.
Tool Relative Coordinates	Indicates that locations should be returned relative to the robot tool tip position when the picture was taken. This is only used if the selected calibration is for an object attached to a robot (upward-looking camera).

### **Results**

Object	Description
Instance	Index of the result instance, starting with 0.
Position X	The X coordinate of the origin of the located instance.
Position Y	The Y coordinate of the origin of the located instance.
Angle	The angle of the located instance.
Group	The Group Index of the located instance.

### **Related Topics**

[Configuring Communication Tool Properties](#)

[Overlap Tool](#)

### **Configuring Communication Tool Properties**

The basic steps for configuring a Communication Tool are:

1. Select the tool that will provide input images.
2. Select the robot that will handle or pick the instance output by the Communication Tool.
3. Position the region of interest.

4. Set Queue and Gripper parameters.
5. Test and verify results.

### **Input**

The input required by the Communication Tool is typically provided by an Overlap Tool. The input can also be provided by other tools that output instances, such as a Results Inspection tool or a Locator.

### **Robot**

The Robot parameter selects the robot that will handle or pick the instances output by the Communication tool.

### **Region of Interest (ROI)**

The region of interest is the area in which the tool carries out its process. Most tool position parameters can be set through the region of interest section of the tool interface. The following parameters define the tool region of interest.

#### **Offset**

Offset is the center of the ROI, defined by:

X	X coordinate of the center of the region of interest
Y	Y coordinate of the center of the region of interest
Degrees	Angle of rotation of the region of interest

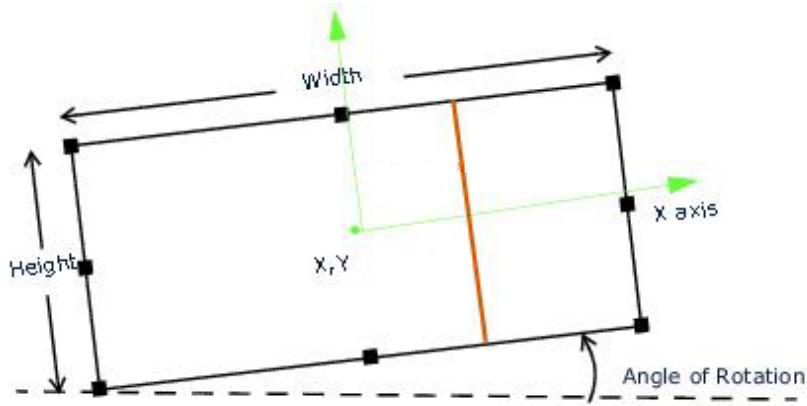
#### **Relative To**

Relative To specifies the tool that will provide the input to this tool. The tool can be dragged from the Tree structure and dropped into this field.

#### **Search Area**

Search Area is the size of the region of interest is defined by:

Height	Height of the region of interest
Width	Width of the region of interest



#### *Location Properties for the Region of Interest*

Modifying the region of interest is useful for applications in which two or more robots pick or handle objects on different sides of the belt. For example, a application could use one Communication Tool configured to output objects on the right side of the belt to Robot A, and a second Communication Tool configured to output instances on the left side of the belt to Robot B.

The region of interest can be the entire image or a portion of the input image. It can be set in one of the following ways:

- Enter or select values for the Offset and Search Area parameters: Position X, Position Y, Angle, Width, and Height.
- Resize the bounding box directly in the display.

The rectangle represents the tool region of interest. Drag the mouse to select the portion of the image that should be included in the ROI.

## **Queue Parameters**

The Communication Tool sends instances that pass its criteria to its queue, which is set up with the following parameters:

### **Queue Index (0 - 100)**

The Queue Index identifies the queue to which instances will be sent.

Two different Communication Tools cannot write to the same queue on a controller. If there are multiple Communication Tools, either on the same or different PCs, each tool must be assigned a unique queue index.

### **Queue Size (1 - 100)**

Queue Size specifies the number of instances that can be written to the queue. The ideal queue size varies greatly. It may require some trial and error to optimize this value for a specific application and environment.

### **Queue Update**

Queue Update specifies how often the Communication Tool will write new instance data to the queue on the controller. The recommended setting is After Every Instance.

After Every Instance

The After Every Instance setting pushes each instance to the queue, separately, as it becomes available. This minimizes the time until the first instance is available for use by the V+ application program. If a large number of instances are located, then it can take longer to push all of the data to the controller.

After Last Instance

The After Last Instance setting sends multiple instances in one data packet. This mode minimizes the total data transfer time, but can increase the time until the first instance is available for use, since the robot is inactive during the time that the PC is writing to the queue.

Note that the two Queue Update settings are effectively the same when only one instance is found.

### **Use Soft Signal**

This is a checkbox that enables the Soft Signal.

### **Soft Signal (0 - 2512)**

This sets the value of the Soft Signal to use when Use Soft Signal is enabled.

The signal can be used by V+ to synchronize the controller and the PC. This signal instructs the controller that all instances detected by the input tool have been sent to the controller.

## **Gripper Configuration**

### **Selection Mode**

This specifies one of Disabled, Use Default, or Use Model Identifier for determining the offset index of the gripper.

### **Offset Index (1 - 1000)**

This specifies the index in the gripper offset table to apply as the gripper offset.

---

Default Offset Index is used for Use Default.

Model Offset Index is used for Use Model Identifier. This gives you a sub-menu for building a table of Model ID - Offset Index values.

#### **Related Topics**

[Communication Tool](#)

[Overlap Tool](#)

### **Overlap Tool**

The purpose of the Overlap Tool is to make sure that parts moving on a conveyor belt are recognized only once.

Because a part found by the Locator (or other input tool) may be present in multiple images acquired by the camera, the Overlap Tool is needed to ensure that the robot is not instructed to pick up or process the same part more than once.

**NOTE:** Conveyor tracking requires an Adept SmartController CX motion controller.

The Overlap Tool is configured using its object editor.

1. To add an Overlap Tool, right-click in the Tree structure of the Workspace Explorer and select:

**New > Vision > ACE Sight > Overlap Tool**

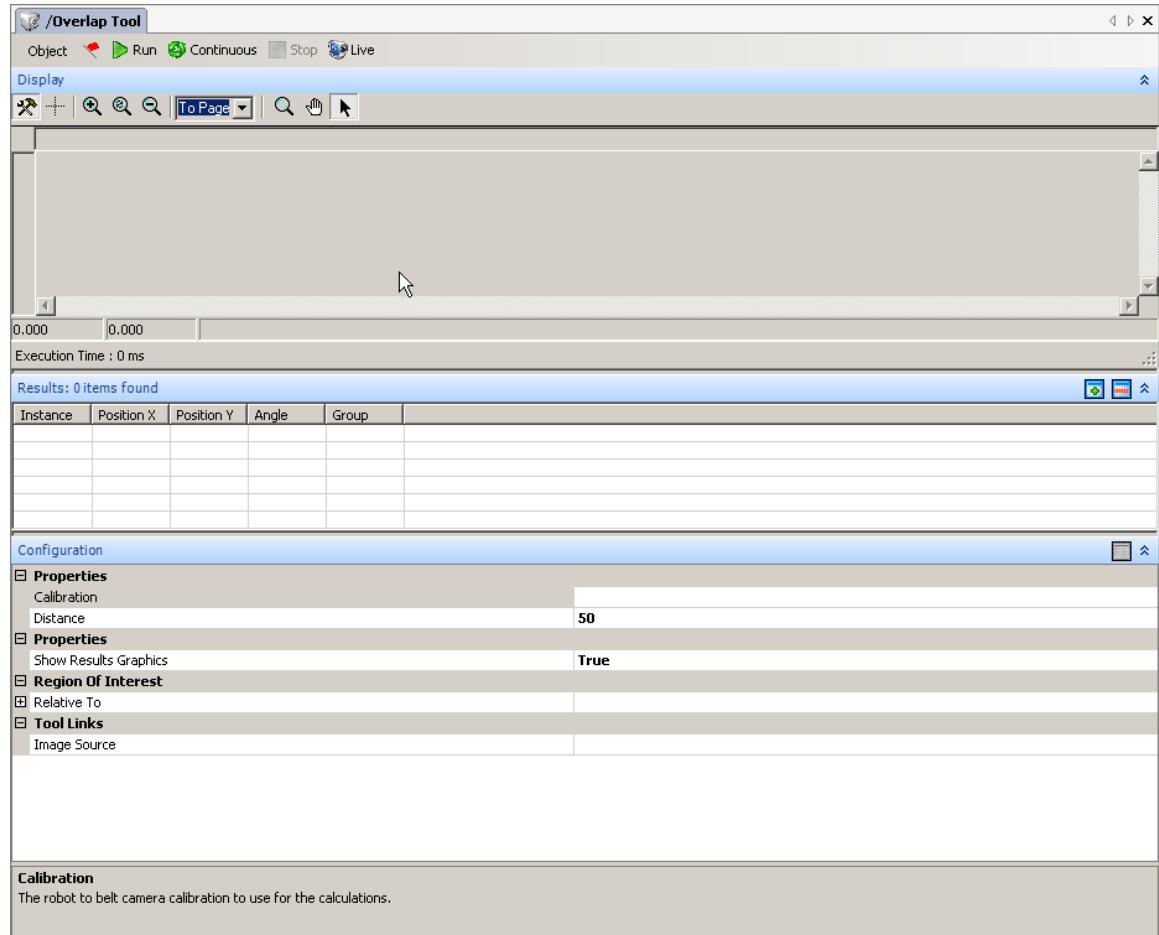
The tool is added to the ACE workspace and the Overlap Tool object editor opens.

2. Use the controls within the object editor to configure the Overlap Tool for your applic-

## Overlap Tool

---

ation.



*Overlap Tool Object Editor*

### **How It Works**

The Overlap Tool filters results. If an instance in the image is a new instance (Pass result) it is passed on to the next tool in the sequence. If an instance is already known, it is rejected (Fail result), and is not sent to the next tool in the sequence. This avoids "double-picking" or "double-processing" of the object.

### **Requirements for Using the Overlap Tool**

The Overlap Tool will only function in a conveyor-tracking environment. This tool executes correctly only if:

- The camera, robot, and conveyor belt are calibrated.
- The connection to the Adept SmartController is active.
- The tool is receiving latched values from the input tool. The latch must be wired to the controller and properly configured.
- The conveyor belt and the controller have been correctly assigned to a camera, in the ACE Sight vision project.
- A valid Conveyor Tracking License is in place.

### **Order of the Overlap Tool in an ACE Sight Sequence**

The Overlap Tool should be placed near the beginning of a sequence, just after the input tool, and before any inspection tools in the sequence. This ensures that the same instance is not processed multiple times by the inspection tools.

#### ***Basic Steps for Configuring the Overlap Tool***

1. Select the tool that will provide the input.

This can be any tool that returns a transform value.

2. Select the Distance parameter.

This specifies how far an instance must be from the expected location of a known instance, in a different image, for it to be considered a new instance. Distance is specified in mm. It should be as small as possible without causing double-picks. 5 mm is a recommended starting value for Distance.

- If Distance is too large, nearby instances will be interpreted as duplicates of a different instance, and some will not be picked.
- If Distance is too small, two transforms will be interpreted as different instances, and the system will try to double-pick the object.

**NOTE:** Rotation is ignored by the Overlap Tool. Only the difference in XY is considered.

3. Test the Overlap Tool by executing the sequence and verifying results.

#### ***Input***

The Input required by the Overlap Tool can be provided by any tool that returns a transform instance.

---

## Configuration

Object	Description
<b>Properties</b>	
Calibration	The robot-to-belt camera calibrations to use for the calculations
Distance	The minimum distance required between parts
<b>Region of Interest (ROI)</b>	
Relative To	Specifies the tool that will provide the input to this tool. The tool can be dragged from the Folder pane and dropped into this field.
<b>Advanced</b>	
Is Running In Sequence	If the tool is running in a sequence, it will read the required latches and update the internal tracking structures. If not running in a sequence, the latches are not read, and the results are not tracked.

## Results

Object	Description
Instance	Index of the result instance, starting with 0
Position X	The X coordinate of the origin of the located instance
Position Y	The Y coordinate of the origin of the located instance
Angle	The angle of the located instance
Group	The group index of the located instance

## Related Topics

[Communication Tool](#)

## Gripper Offset

The Gripper Offset Tool defines where on the part a robot can pick up a part, giving the relationship between the pick point, the part model, and the robot flange center.

A gripper has two types of offsets:

- The offset from the center of the tool flange to the gripper tip(s)

This offset is established when the gripper itself is defined. Right-click in the Tree structure, then select:

**New > Device > Tool > IO Driven Gripper**

This is documented in the ACE User's Guide. This offset will be applied to that gripper whenever it is used, including calculating values for the Gripper Offset Table.

- The offset, on a part to be picked, from the actual pick point to the part origin

This is defined in the Gripper Offset Table. Right-click in the Tree structure, then select:

**New > Vision > ACE Sight > Gripper Offset Table**

The tool is added to the ACE workspace and the Gripper Offset Tool object editor opens.

Use the controls within the object editor to configure the Gripper Offset Tool for your application.

***Related Topics***

[Gripper Offset Table](#)

---

### **Gripper Offset Table**

**NOTE:** A gripper offset is not always needed to pick a part, so you may not need to build a Gripper Offset Table for your application.

The Gripper Offset Table:

- Tells the robot where, in relation to the origin of a part, it must grip the part.
- Describes a transform, expressed as: (x, y, z, yaw, pitch, roll).

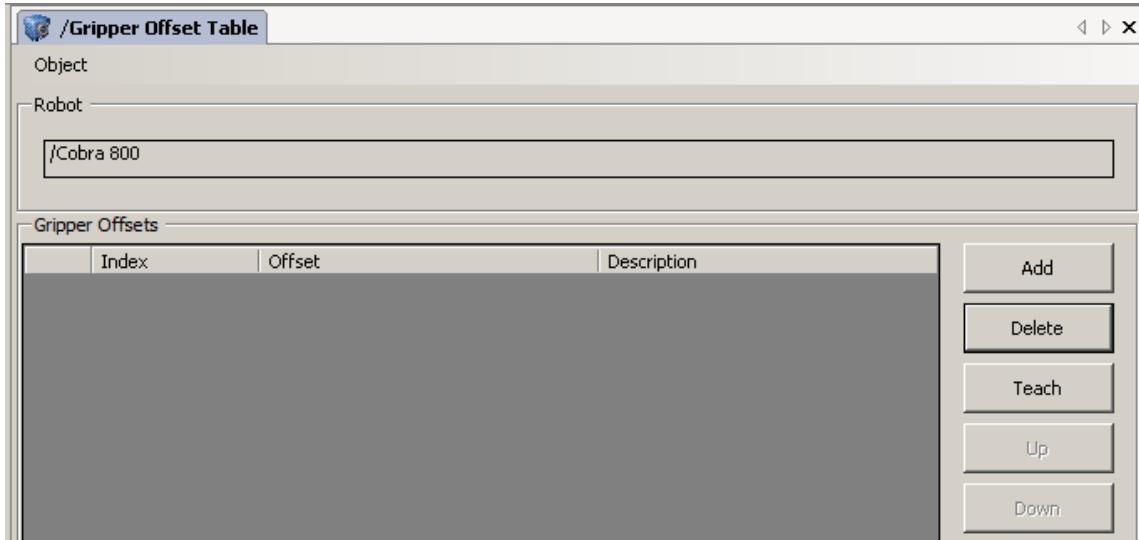
**NOTE:** A part can have more than one Gripper Offset (if it can be picked at more than one spot).

- Contains offset data used by V+ or MicroV+.
- Is assigned to a robot. No robot can have more than one table.

These instructions also apply in Emulation Mode.

To open the Gripper Offset Table, right-click in the Tree structure, then select:

**New > Vision > ACE Sight > Gripper Offset Table**



*The Gripper Offset Table Initial Screen*

### **Object Editor**

The Object Editor screen starts by asking for the robot that this table is to be assigned to.

---

---

It then lists an index (used in V+ or MicroV+), along with the offset transform, and an optional description.

New offsets are added by clicking Add. Existing offsets can be deleted by clicking Delete.

Clicking Teach takes you through the steps needed to calculate a gripper offset. This is covered in the following section.

Offsets can be moved up or down in the table by clicking Up or Down.

## Teach Command

**NOTE:** Gripper offsets can be taught or typed in.

The Gripper Offset Teach command is presented as a wizard that walks you through the steps required for defining Gripper offsets in the offset table.

To teach a Gripper Offset:

1. Click on the entry in the table you want to teach, to select it.  
If there are no entries, or if you want to teach a new offset, click Add.
2. Click Teach.
3. Follow the instructions in the wizard.

If you have offset transforms in a Gripper Offset Table for another robot, you can copy and paste a transform from one robot's offset table to another.

## Use in V+and MicroV+

ACE Sight retrieves the gripper offset from the gripper offset table associated with a specific robot through the VLOCATION command, using ACE Sight Property 10100.

**NOTE:** The gripper offset gives the transform from the pick spot back to the instance origin. You must use the inverse of the offset for the robot to pick from that location.

### Related Topics

[Gripper Offset](#)

## ACE Sight Calibrations

This section covers two three basic calibrations:

- Belt-to-robot calibration
- Camera-to-robot calibration

**NOTE:** The Basic camera calibration should have already been performed. See [Standalone Camera Calibration](#).

- Latch-to-robot calibration

### **When do I Calibrate?**

Calibration needs to be performed once for each different setup. If you make changes to the setup, specifically to the robot, belt, or camera position, parameters, or configuration, then you must recalibrate the new setup.

### **What Order?**

**NOTE:** All devices being calibrated must be connected and functioning properly.

1. ACE Sight Belt Calibration : This calibrates a conveyor belt to a robot.

It consists of an interview wizard followed by a calibration wizard.

You must run this calibration before the ACE Sight Camera Calibration.

**NOTE:** ACE Sight Belt Calibration is not needed if there is no conveyor belt in the system.

When the ACE Sight Belt Calibration is completed, ACE Sight generates an ACE Sight Belt Calibration object, which will be needed for the ACE Sight Camera Calibration.

See "ACE Sight Belt Calibration".

2. ACE Sight Camera Calibration : This calibrates a camera to a robot, making the system aware of the location of the camera's field of view with respect to the robot.

It consists of an interview wizard followed by a calibration wizard.

When the ACE Sight Camera Calibration is completed, ACE Sight generates an ACE Sight Camera Calibration object.

See "ACE Sight Camera Calibration".

3. ACE Sight Latch Calibration : This calibrates a latch to a robot.

It consists of an interview wizard followed by a calibration wizard.

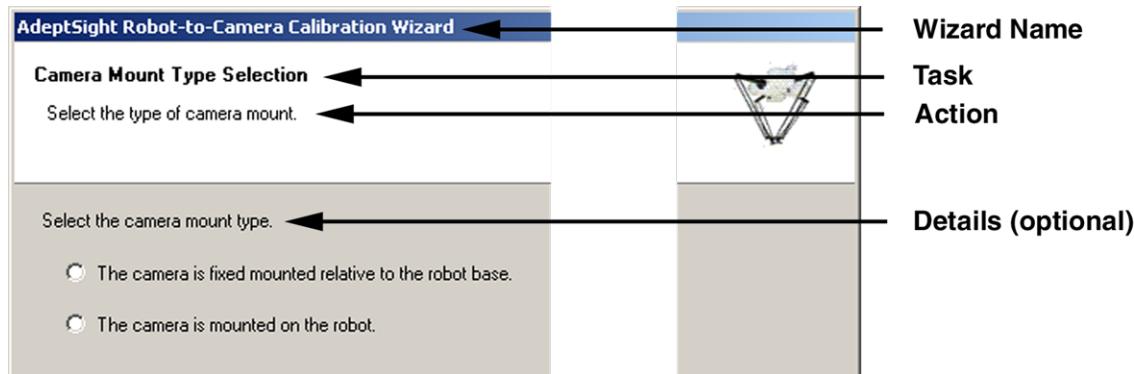
When the ACE Sight Latch Calibration is completed, ACE Sight generates an ACE Sight Latch Calibration object.

The calibration interview wizards gather sufficient information about your system to use the correct parameters during their respective calibration wizards.

### ***Wizard Screens***

The ACE Sight Belt Calibration Wizard and ACE Sight Camera Calibration Wizard use similar formats.

A sample calibration wizard screen is illustrated in the following screen shot:



*Sample Calibration Interview Wizard Screen*

#### **Wizard Name**

The title in the blue bar identifies which wizard is executing. It is the context of the screen. In the example, this is "ACE Sight Robot-to-Camera Calibration Wizard".

#### **Task**

The heading tells you what you will be doing in this screen: "Camera Mount Type Selection".

#### **Action**

The line under the task heading tells you what you must do, or poses the question that this task will answer: "Select the type of camera mount".

#### **Action Details**

If needed, details are given in the inner window: "Select the camera mount type".

At the bottom of the Wizard screens you will usually find **I/O, Power, Previous, Next, Cancel, and Help** buttons.

Some of these, particularly **Previous**, may be greyed-out when that action is not appropriate. (Many procedures require that you click **Cancel** and start over, rather than backing up with **Previous**.)

#### **Related Topics**

[Standalone Camera Calibration](#)

[ACE Sight Belt Calibration Interview Wizard](#)

[ACE Sight Belt Calibration Wizard](#)

[ACE Sight Camera Calibration Interview Wizard](#)

[ACE Sight Camera Calibration Wizard - Automated, no Belt](#)

[ACE Sight Camera Calibration Wizard - Automated, w/ Belt](#)

[ACE Sight Camera Calibration Wizard - Automated Upward-Looking](#)

[ACE Sight Camera Calibration Wizard - Automated, Arm-mounted](#)

[ACE Sight Camera Calibration Wizard - Manual, no Belt](#)

[ACE Sight Camera Calibration Wizard - Manual w/Belt](#)

[ACE Sight Camera Calibration Wizard - Manual, Upward-looking](#)

[ACE Sight Camera Calibration Wizard - Manual, Arm-mounted](#)

## **ACE Sight Belt Calibration**

ACE Sight Belt Calibration calibrates a robot to a conveyor belt.

This calibration is necessary when the robot will handle parts that are moving on a conveyor belt.

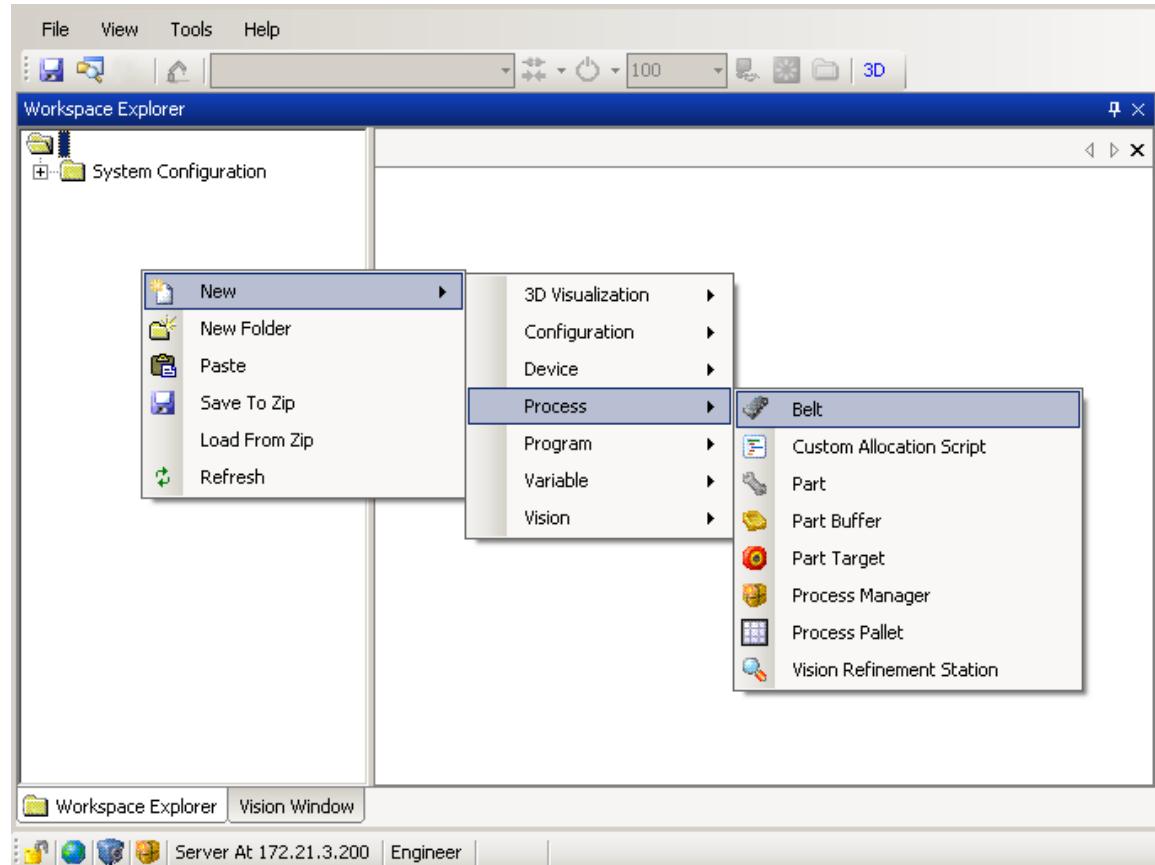
This calibration is available in Emulation Mode.

#### **Requirements**

- The robot, controller, and belt must be correctly connected and functioning.
- The PC running the ACE Sight software must be connected to the controller for the robot and belt.
- The belt, as well as the robot and gripper, must be defined in the ACE Sight software.

To add a belt to the Workspace, right-click in the Tree structure of the Workspace Explorer and select:

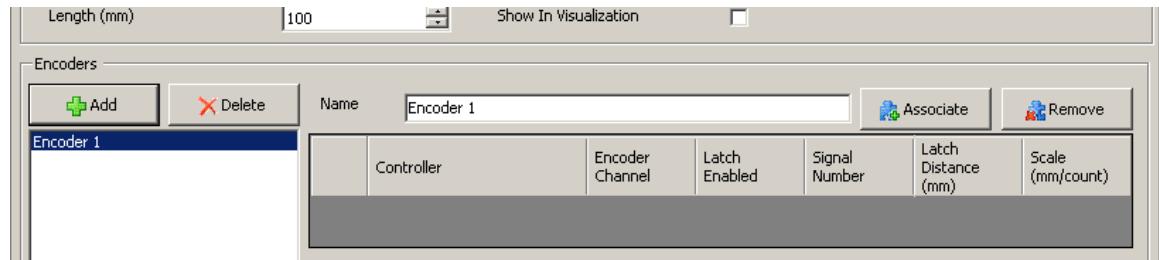
**New > Process > Belt.**



*Adding a Belt Object*

In the Belt Object Editor:

1. Click Add (under Encoders).
2. Click Associate. See the following figure.



*Add and Associate Buttons*

3. Click the controller, to select it.  
The selected controller will be highlighted.
4. Click OK.

### **Next Steps**

Run the ACE Sight Belt Calibration Interview Wizard, then the ACE Sight Belt Calibration Wizard.

### **Related Topics**

[Calibration Overview](#)

[ACE Sight Belt Calibration Interview Wizard](#)

[ACE Sight Belt Calibration Wizard](#)

[ACE Sight Camera Calibration Interview Wizard](#)

[ACE Sight Camera Calibration Wizard - Automated, no Belt](#)

[ACE Sight Camera Calibration Wizard - Automated, with Belt](#)

[ACE Sight Camera Calibration Wizard - Automated Upward-Looking](#)

[ACE Sight Camera Calibration Wizard - Manual, no Belt](#)

[ACE Sight Camera Calibration Wizard - Manual with Belt](#)

[ACE Sight Camera Calibration Wizard - Manual, Upward-looking](#)

### ***ACE Sight Belt Calibration Interview Wizard***

The Belt Calibration Interview Wizard gathers the parameters necessary for you to run the ACE Sight Belt Calibration Wizard. It does not automatically run that wizard.

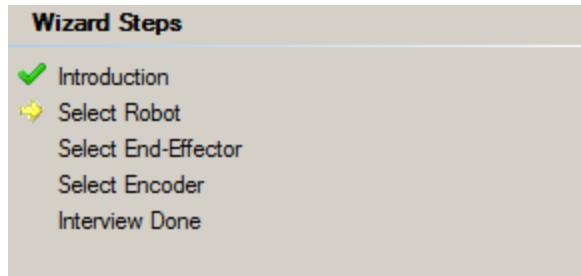
To run the interview wizard, right-click in the Tree structure of the Workspace Explorer and select:

**New > Vision > ACE Sight > ACE Sight Belt Calibration**

**Procedure**

All screens have the title "ACE Sight Robot-to-Belt Calibration Wizard".

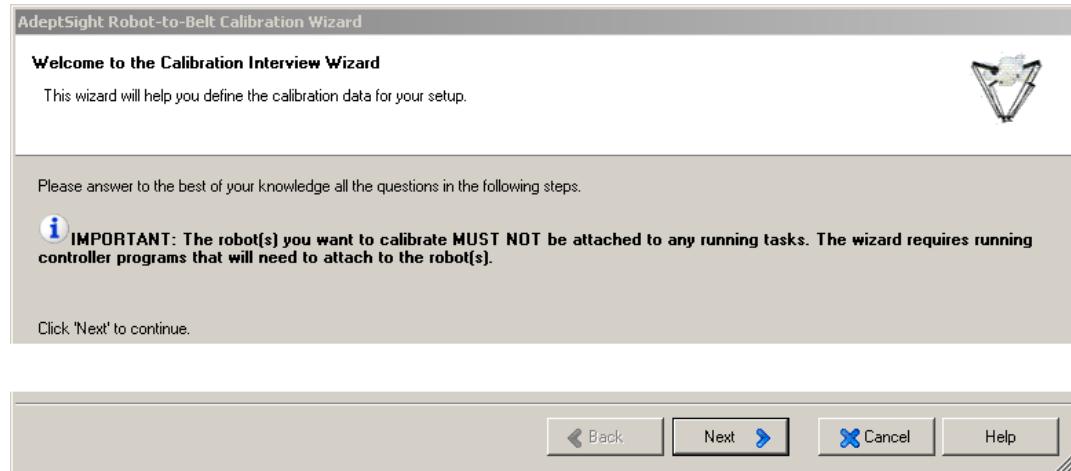
A list of tasks that will be performed is displayed in the left pane as you go through the calibration:



*List of Wizard Tasks*

The screens in this procedure are:

- Welcome



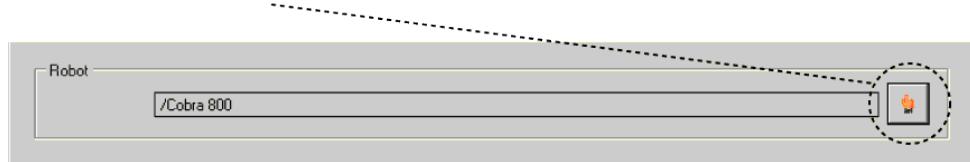
*Belt Calibration Welcome Screen*

Click Next.

- Robot Selection for Calibration

Select the robot that will be used for this calibration.

1. Click on the browse icon to display a list of available robots.



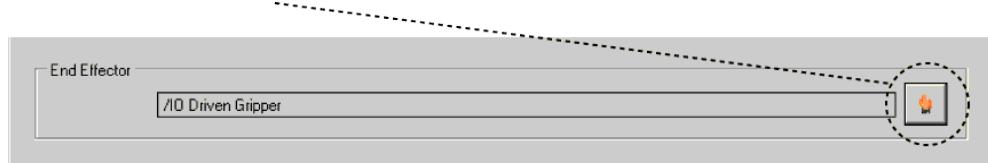
2. Select the robot that you wish to use.
3. Click OK.
4. Click Next.

- End-Effector Selection for Calibration

**NOTE:** This will default to the end-effector of the selected robot.

Select the end-effector of the selected robot.

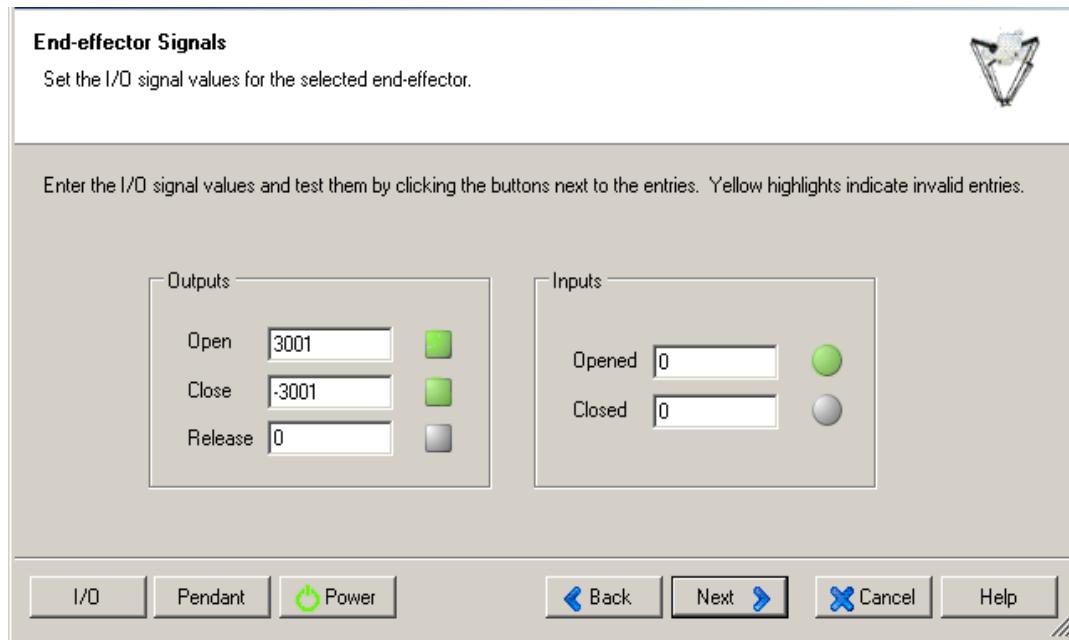
1. Click on the browse icon to display a list of available end-effectors.



2. Select the end-effector that you wish to use.
3. Click OK.
4. Click Next.

- End-effector Signals

Set the I/O signal values for the selected end-effector.



#### Setting the End-effector Signal Values

Green indicates an enabled signal. Black indicates a disabled signal. Yellow in a text field indicates an invalid value.

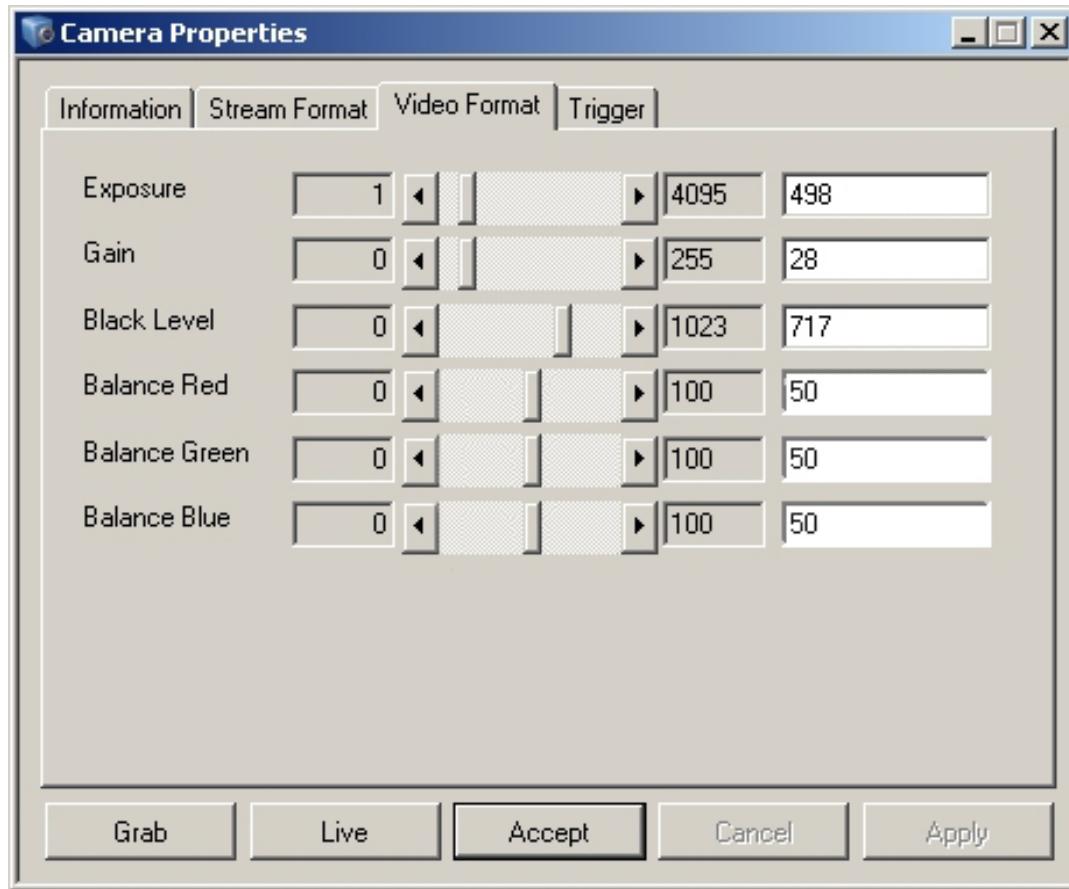
**NOTE:** These signals need to be set for the Belt Calibration Wizard to work, since you will need to be able to move the belt with them.

- Select existing or new camera

The wizard will let you add a new camera at this point if you choose to.

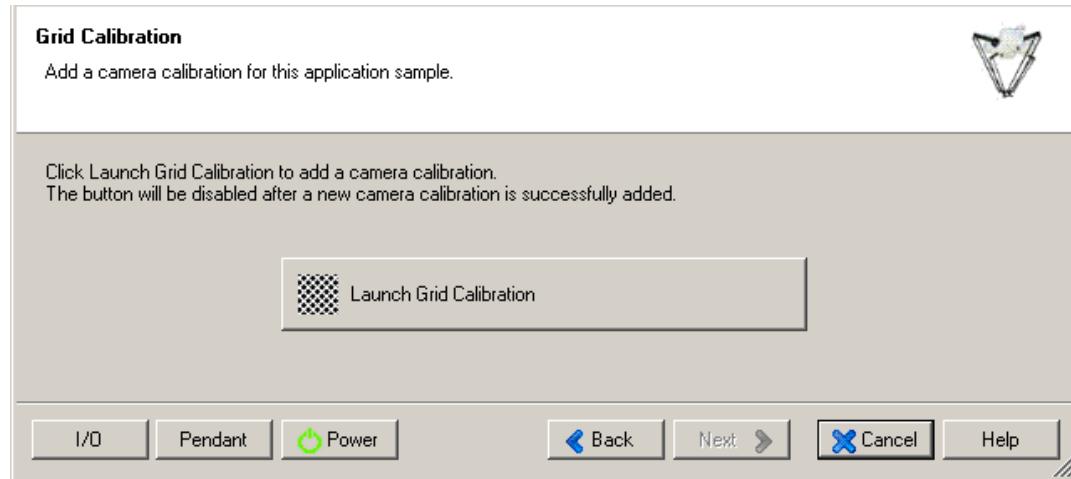
- Camera properties

Edit these, if needed. Refer to See "Camera Properties".



*Setting the Camera Properties*

- Launch Grid Calibration

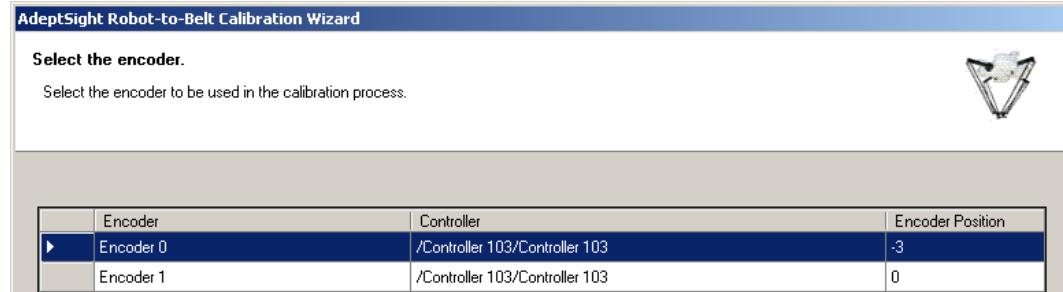


### *Launching the Grid Calibration*

You will need a grid of dots for this calibration. Refer to See "Standalone Camera Calibration".

- Select the encoder

The SmartController's only required connection with a conveyor belt is the belt's encoder. In some installations, the controller will have control over belt motion.



### *Selecting the Encoder*

- Turn on power
- Teach Picture-taking Position
- Teach the Vision tool
- Interview completed

### **Next Step**

You can now run the ACE Sight Belt Calibration Wizard. From the Workspace Belt Calibration Object, displayed after the interview completes, click on Calibration Wizard. See ACE Sight Belt Calibration on page 635.

### **Related Topics**

[Calibration Overview](#)

[ACE Sight Belt Calibration](#)

[ACE Sight Belt Calibration Wizard](#)

[ACE Sight Camera Calibration Interview Wizard](#)

[ACE Sight Camera Calibration Wizard - Automated, no Belt](#)

[ACE Sight Camera Calibration Wizard - Automated, w/ Belt](#)

[ACE Sight Camera Calibration Wizard - Automated Upward-Looking](#)

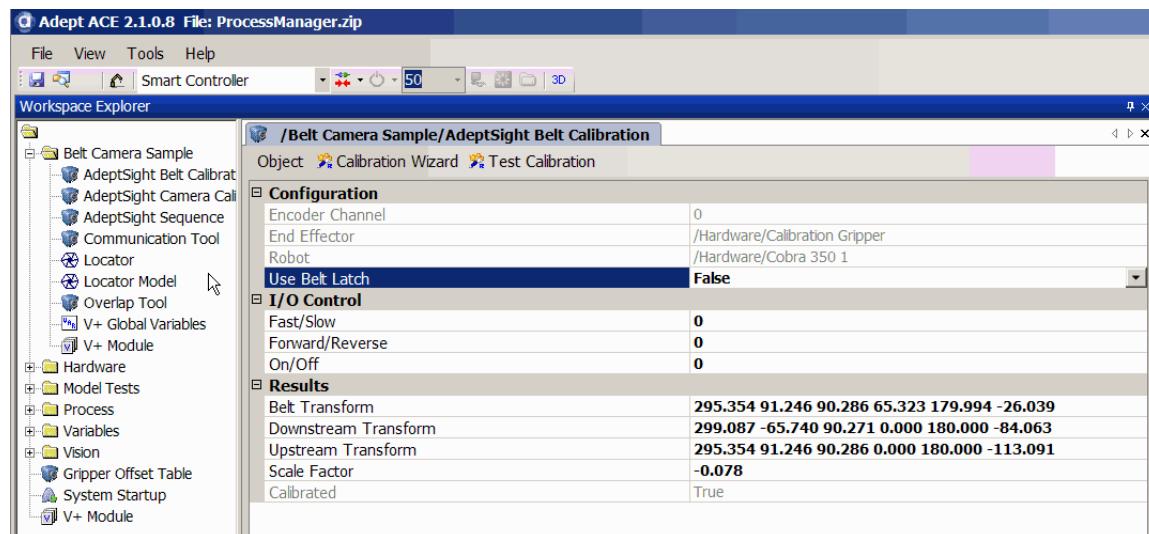
[ACE Sight Camera Calibration Wizard - Manual, no Belt](#)

[ACE Sight Camera Calibration Wizard - Manual w/Belt](#)

[ACE Sight Camera Calibration Wizard - Manual, Upward-looking](#)

### **ACE Sight Belt Calibration**

After the ACE Sight Belt Calibration Interview Wizard completes, it opens the Object editor for the ACE Sight Belt Calibration.

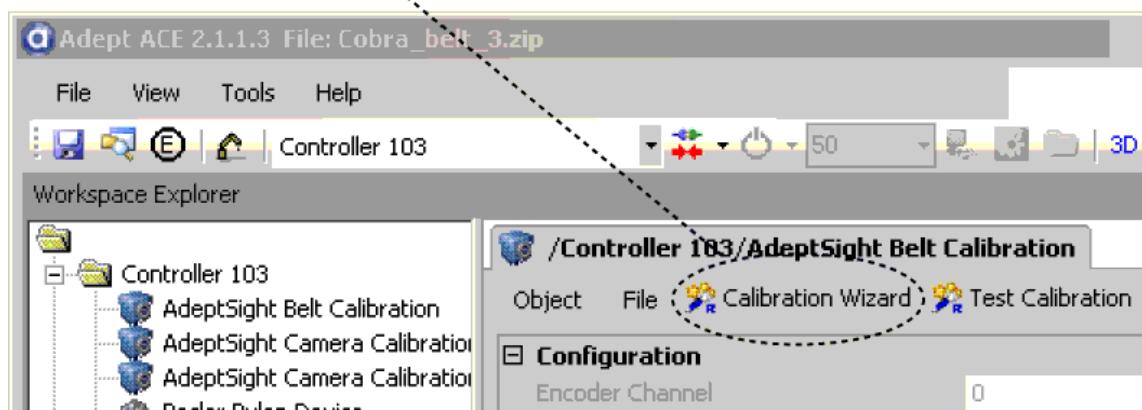


*ACE Sight Belt Calibration Object Editor*

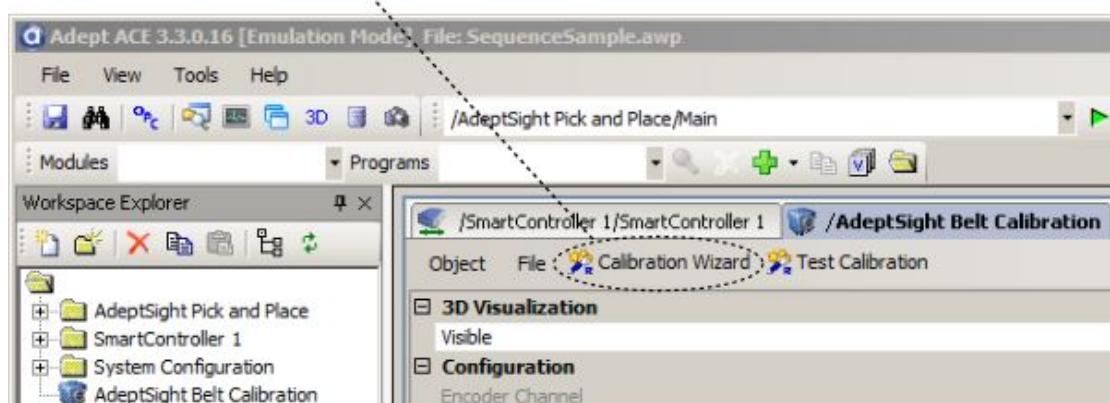
**ACE Sight Belt Calibration Wizard**

The ACE Sight Belt Calibration Wizard calibrates a robot to a conveyor belt. The output of this wizard will be a Belt Calibration object.

Click on Calibration Wizard after running the Calibration Interview Wizard



Click on Calibration Wizard after running the Calibration Interview Wizard



*Launching the Calibration Wizard*

**Requirements**

- The robot and belt must be correctly connected and functioning.
- The ACE Sight Belt Calibration Interview Wizard must have completed successfully.

## What this Wizard Does

This calibration is necessary when the robot will handle parts that are moving on a conveyor belt.

With this wizard you will:

- Establish the relationship between the belt, its encoder, and the robot
- Specify the upstream and downstream limits on the belt
- Specify the downstream pick limit on the belt
- Establish the usable width of the belt

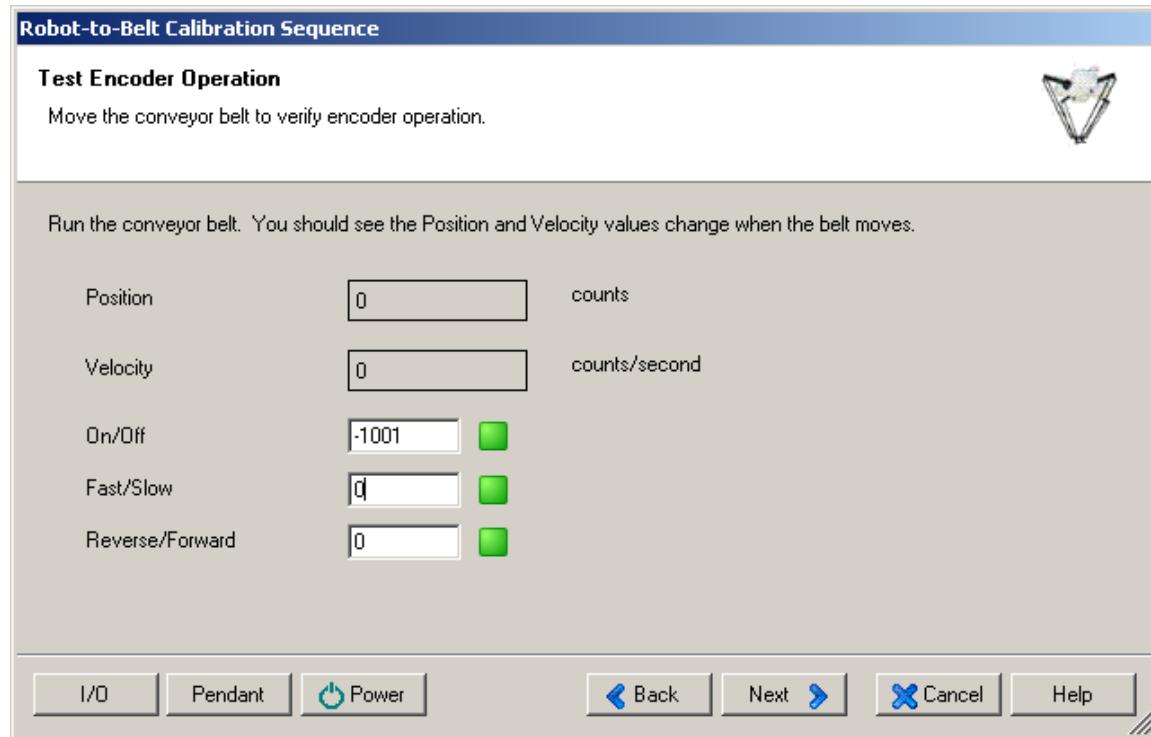
### Procedure

All screens have the title "Robot-to-Belt Calibration Sequence".

The screens in this procedure are:

- Select End-effector
- Test Encoder Operation

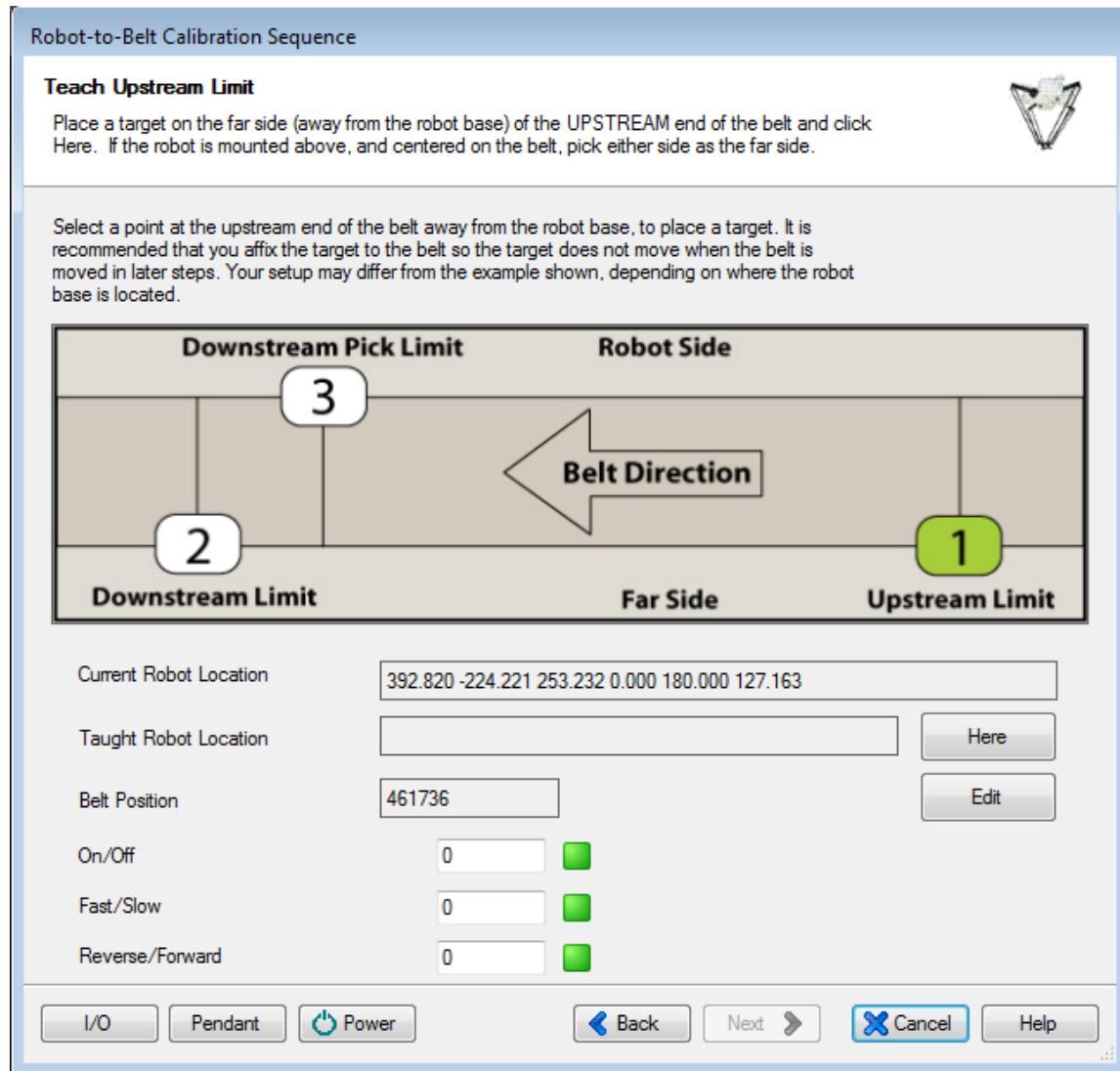
This lets you confirm that the belt encoder is communicating with the controller.



*Testing the Encoder Operation*

You can click On/Off to move the belt. You need to have set the I/O values in the Interview Wizard.

- Teach Upstream Limit (#1 in the following figure)



*Upstream, Downstream, and Downstream Pick Limits*

Upstream limit is the farthest point, towards the start of the belt, that the robot is allowed to move. This teaches the upstream limit, as well as the far side of the belt, width-wise (away from the robot).

The target should be taped or otherwise fastened to the belt to prevent the target from moving, in relation to the belt, during calibration.

1. Place the target on the belt at the upstream limit, away from the robot (width-wise on the belt).

If the robot is mounted above the belt, and is centered on the belt, pick either side of the belt.

2. Move the robot tip to the target.

Click Pendant to display the pendant window. This lets you to move the robot, to center the tip over the target.

3. Click Here.

- Teach Downstream Limit (#2 in preceding figure)

Downstream limit is the farthest point, towards the end of the belt, that the robot is allowed to move.

1. Move the belt, without touching the target, so that the target stops at the downstream limit.

2. Move the robot to the target.

3. Click Here.

- Teach Downstream Pick Limit (#3 in preceding figure)

Downstream pick limit is the farthest point, towards the downstream limit, that the robot is allowed to pick an object. It will be between the upstream and downstream limits. This also teaches the near side of the belt, width-wise (nearest the robot). The difference between the far side and near side establishes the usable width of the belt.

1. Place the target at the downstream pick point, near the robot (widthwise on the belt).

If the robot is mounted above the belt, and is centered on the belt, pick the side of the belt opposite from the side you picked before.

2. Move the robot to the target.

3. Click Here.

- Test Calibration

1. Close the Pendant window (click Cancel).

**NOTE:** This procedure will fail if the Pendant window is left open.

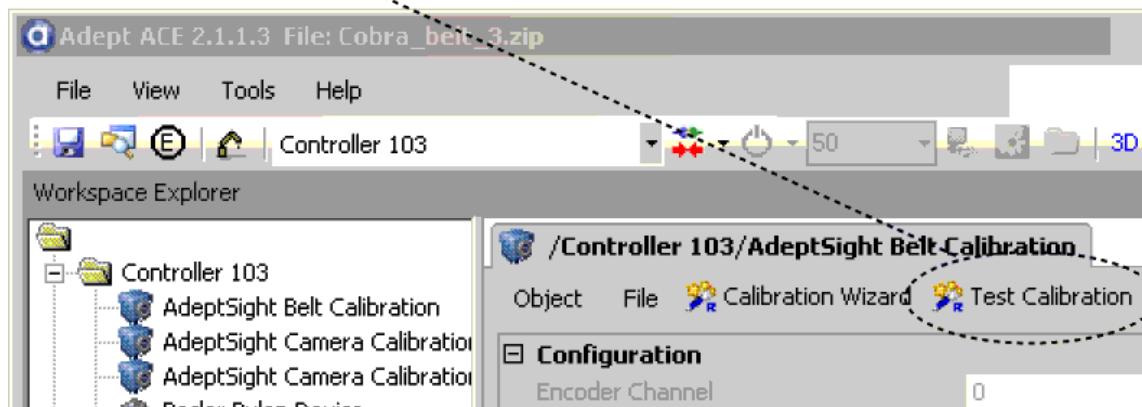
2. Center the robot tool tip over a part on the belt.
3. Click Start Tracking.
4. Advance the belt.

The robot should track the part on the belt.

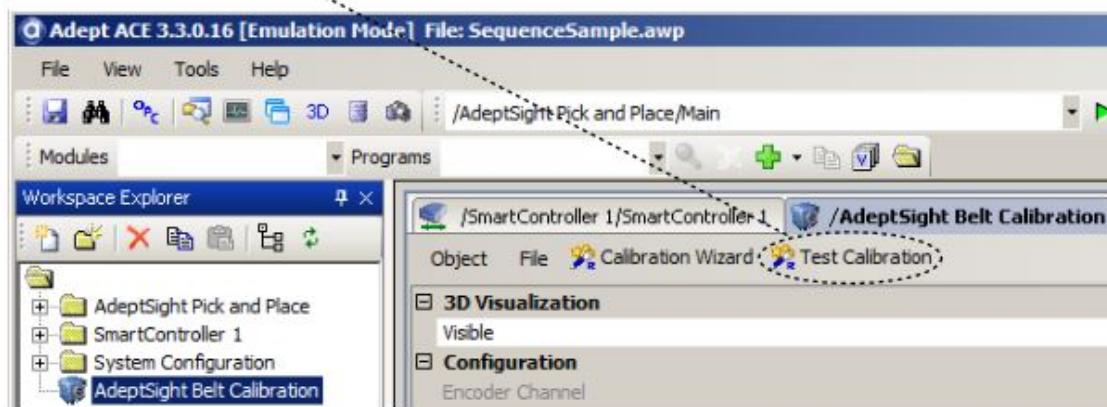
### ***Test Calibration***

The test procedure repeats the test performed at the end of the belt calibration wizard.

Click on Test Calibration after running the Calibration Wizard



Click on Test Calibration after running the Calibration Wizard



### *Testing the Calibration*

1. Close the Pendant window, if it is open (click Cancel).
2. Click Start Tracking.
3. Place a part under the robot.
4. Advance the belt.

The robot should track the part.

### ***Related Topics***

[Calibration Overview](#)

[ACE Sight Belt Calibration](#)

[ACE Sight Belt Calibration Interview Wizard](#)

[ACE Sight Camera Calibration Interview Wizard](#)

[ACE Sight Camera Calibration Wizard - Automated, no Belt](#)

[ACE Sight Camera Calibration Wizard - Automated, w/ Belt](#)

[ACE Sight Camera Calibration Wizard - Automated Arm-mount](#)

[ACE Sight Camera Calibration Wizard - Automated Upward-looking](#)

[ACE Sight Camera Calibration Wizard - Manual, no Belt](#)

[ACE Sight Camera Calibration Wizard - Manual w/Belt](#)

[ACE Sight Camera Calibration Wizard - Manual Arm-mount](#)

[ACE Sight Camera Calibration Wizard - Manual, Upward-looking](#)

## **ACE Sight Camera Calibration**

ACE Sight Camera Calibration calibrates a robot to a camera. This calibration is necessary if you will be using vision with a robot.

Because there are a number of ways to mount a camera, and the option of a conveyor belt, there are eight different camera calibration wizards.

### ***Requirements***

- The robot, controller, belt (if used), and camera must be correctly connected and functioning.  
The Cobra i-Series robots do not need a SmartController motion controller.
- The camera itself must be calibrated. See Standalone Camera Calibration on page

320.

- The PC running the ACE Sight software must be connected to the controller for the robot (and belt).
- The ACE Sight Belt Calibration Wizard must have completed successfully, if a conveyor belt will be used (a belt calibration object must exist in the workspace). See ACE Sight Belt Calibration on page 627.

### **Next Step**

Run the ACE Sight Camera Calibration Interview Wizard. See ACE Sight Camera Calibration Interview Wizard on page 642.

### **Related Topics**

[Calibration Overview](#)

[ACE Sight Belt Calibration](#)

[ACE Sight Camera Calibration Interview Wizard](#)

[ACE Sight Camera Calibration Wizard - Automated, no Belt](#)

[ACE Sight Camera Calibration Wizard - Automated, w/ Belt](#)

[ACE Sight Camera Calibration Wizard - Automated Upward-Looking](#)

[ACE Sight Camera Calibration Wizard - Automated, Arm-mount](#)

[ACE Sight Camera Calibration Wizard - Manual, no Belt](#)

[ACE Sight Camera Calibration Wizard - Manual w/Belt](#)

[ACE Sight Camera Calibration Wizard - Manual, Upward-looking](#)

[ACE Sight Camera Calibration Wizard - Manual, Arm-mount](#)

### ***ACE Sight Camera Calibration Interview Wizard***

The ACE Sight Camera Calibration Interview Wizard acquires the data necessary to generate the correct ACE Sight Camera Calibration Wizard. It does not automatically run that wizard.

### **What this Wizard Does**

With this wizard you will:

- Select the robot, gripper, and virtual camera
- Specify the camera mounting (fixed-mounted, arm-mounted)

## Requirements

- The robot and camera must be correctly connected and functioning.
- The camera itself must be calibrated. See Standalone Camera Calibration on page 320.
- The ACE Sight Belt Calibration Wizard must have completed successfully, if a belt will be used.

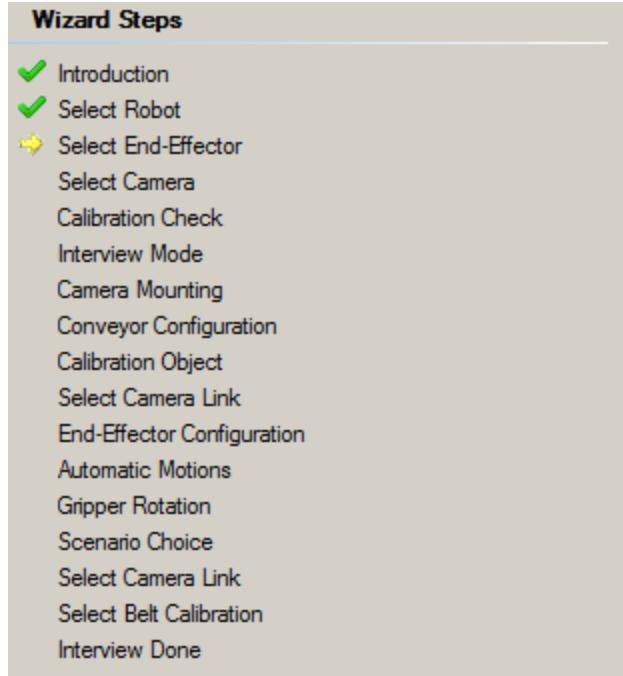
## Procedure

To add an ACE Sight Calibration Object to the workspace, right-click in the Tree structure of the Workspace Explorer and select:

**New > Vision > ACE Sight > ACE Sight Camera Calibration**

All screens have the title "ACE Sight Robot-to-Camera Calibration Wizard".

A list of tasks that will be performed is displayed in the left pane as you go through the calibration:

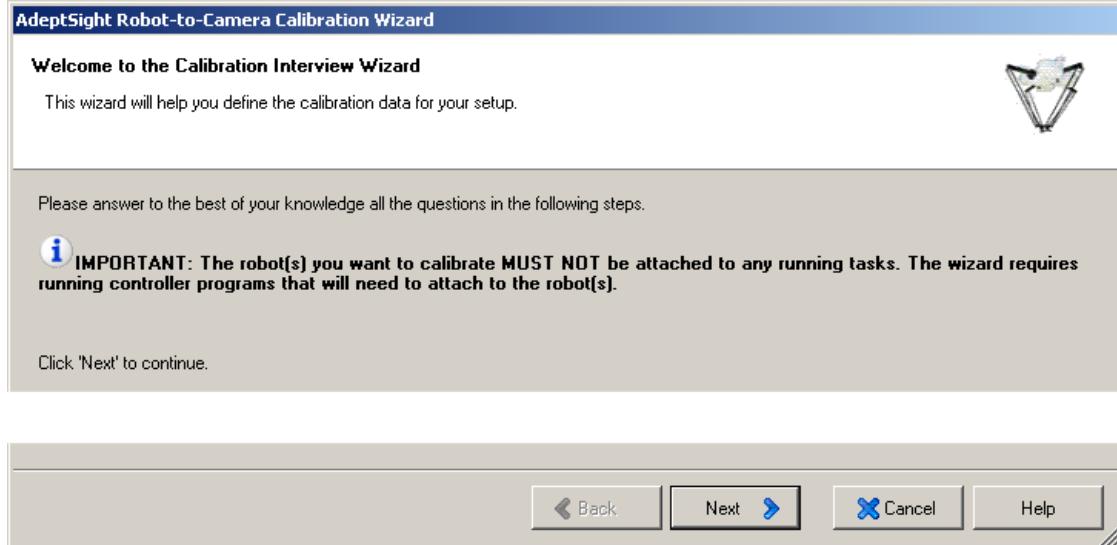


### *List of Wizard Tasks*

The screens in this procedure are:

---

- Welcome to the Calibration Interview Wizard

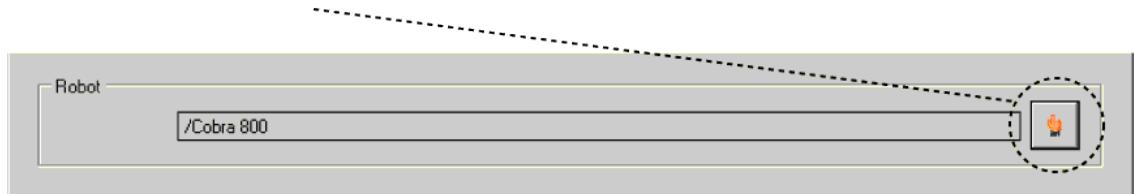


*Camera Calibration Welcome Screen*

- Robot Selection for Calibration

Select the robot that will be used for this calibration.

1. Click on the browse icon to display a list of available robots.



2. Select the robot that you wish to use.

3. Click OK.

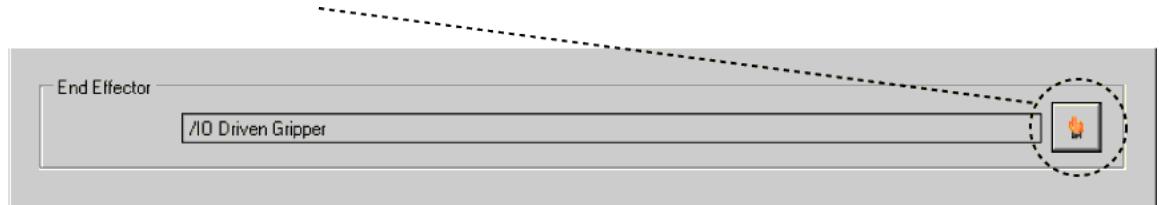
4. Click Next.

- End-Effector Selection for Calibration

**NOTE:** This will default to the end-effector of the selected robot.

Select the end-effector that will be used for this calibration.

1. Click on the browse icon to display a list of available end-effectors.



2. Select the end-effector that you wish to use.

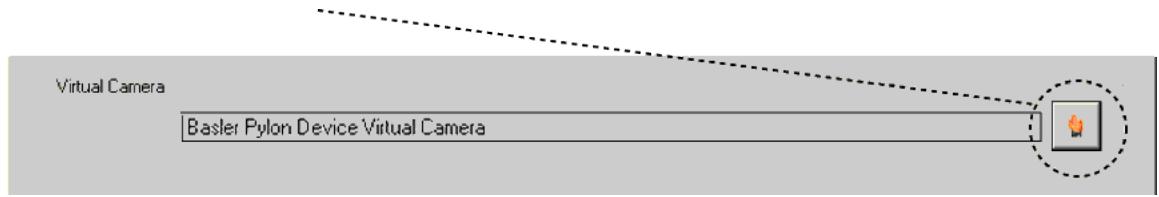
3. Click OK.

4. Click Next.

- Camera Selection for Calibration

Select the camera that will be used for this calibration.

1. Click on the browse icon to display a list of available cameras.



2. Select the camera that you wish to use.

3. Click OK.

4. Click Next.

- Choose Interview Mode

Specify either:

- Answer questions and let the wizard select the correct scenario (Recommended)

or

- Select the correct calibration options from a list

If you chose to answer the wizard's questions, the following information will be requested:

- Specify Camera Mount Type

Please select the option that applies.

- The camera is fixed mounted relative to the robot base.
- The camera is mounted on the robot.

- Define the camera link (arm-mount only)

Please specify the link the camera is mounted on the robot.

- The camera is mounted on the link #2 of a Cobra robot.
- Link '1'
- Link '2'
- Link '3'
- Link '4'
- Link '5'

- Will a conveyor be part of this calibration? (fixed-mount only)

**NOTE:** If you have a belt that hasn't been calibrated, you will not be able to complete this camera calibration.

Answering Yes to this includes the belt in the camera calibration.

- What is the interaction with the calibration object? (fixed-mount only)

Please select the option that applies.

- The calibration object will be picked/pointed by the robot tool.
- The calibration object will be attached to the robot tool.

- Specify End-Effector Type

Please select the options that applies.

- The robot is equipped with an end-effector than can grip and move the object
- The robot has a tool than can point at the object.

- Is the Robot Free to Move?

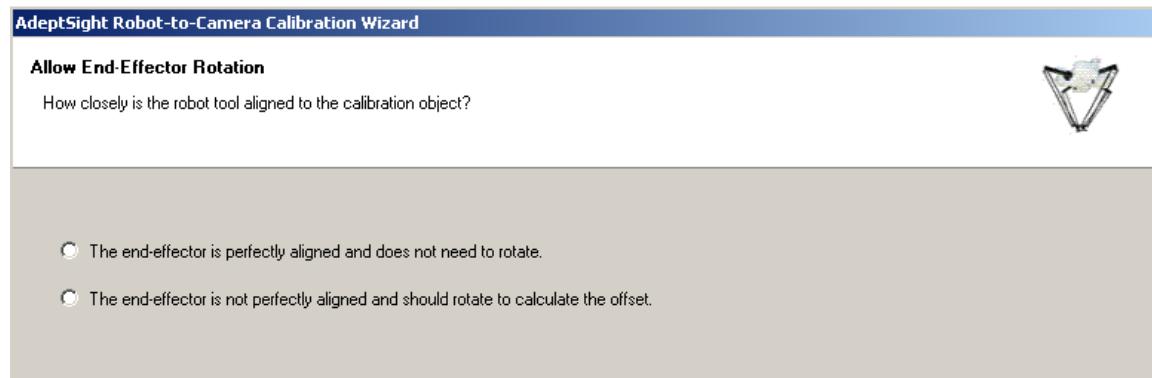
- Robot can move freely in workspace

**NOTE:** This will run the calibration in Automated mode.

- There are obstacles in the workspace or the work surface is not parallel to the robot XY plane. I want to move the robot manually.

**NOTE:** This will run the calibration in Manual mode.

- Allow End-Effector Rotation

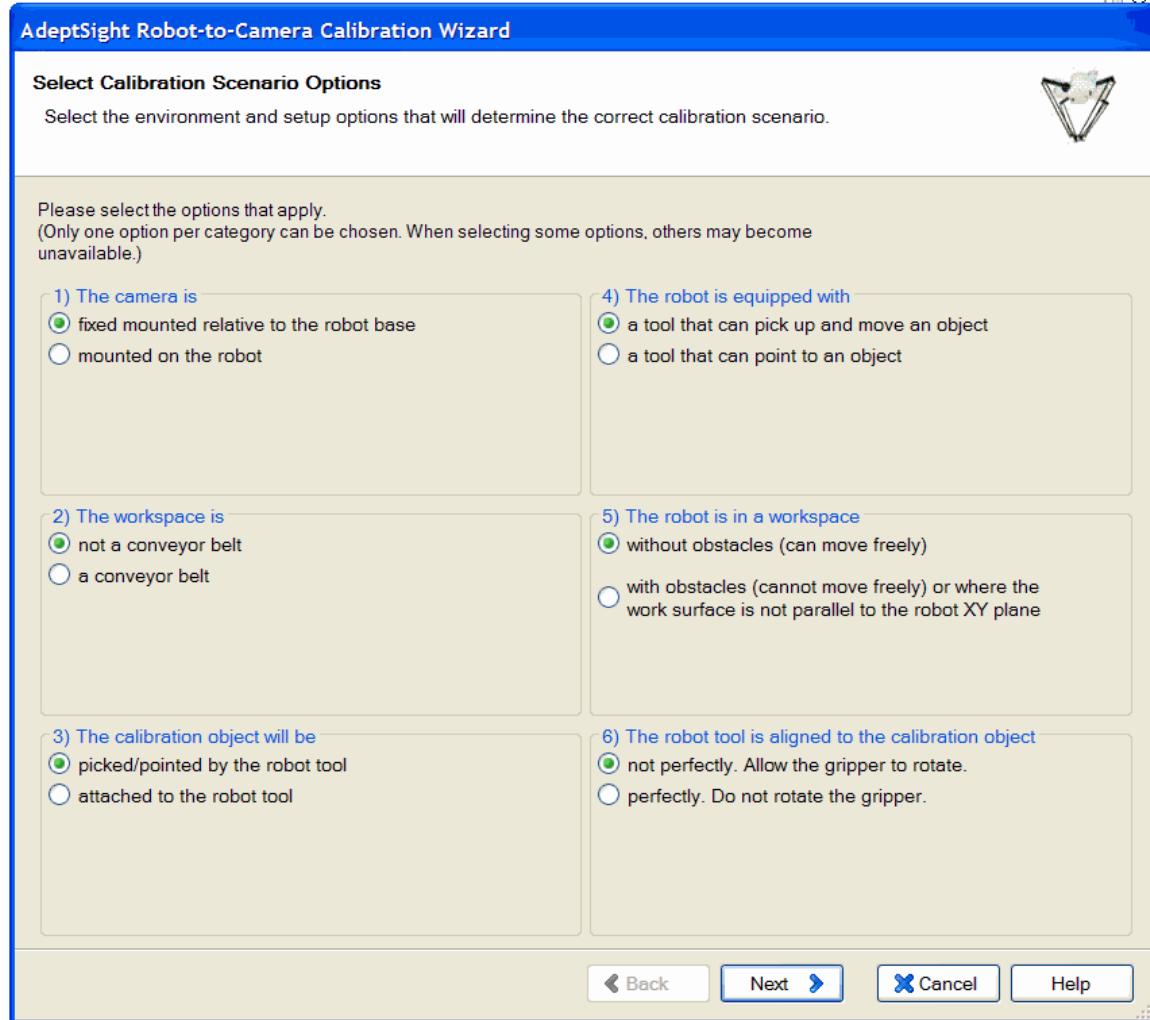


- Select the Robot-to-Belt Calibration (conveyor belt systems only)

Select the ACE Sight Belt Calibration object, generated when you ran the Belt Calibration Wizard.

**NOTE:** If you did not run the ACE Sight Belt Calibration before this calibration, there will be no Belt Calibration object, and this interview will fail here if you said that there would be a conveyor belt as part of this calibration.

If you chose to select the calibration options from a list, you will be presented a screen similar to the following:



### *Specifying Calibration Options from a List*

1. Specify how the camera is mounted.
2. Specify whether a conveyor belt is to be used.
3. Specify if the calibration object is attached to the robot tool, or picked or pointed to.
4. Specify whether the robot can pick an object or point to an object.
5. Specify if there are obstacles within the workspace that must be

considered.

6. Specify whether the tool should rotate during calibration.

- Interview Completed

There are a number of fields that are not covered in the interview wizard. These are usually left at their defaults, but you can change their values if needed.

## **Vision Parameters**

These parameters specify what vision tools are used in the calibration process.

### **Calibration Target Size**

When using a user-defined vision tool, the size of the target, in mm, must be defined. It is used to calculate how far the robot can move the calibration target in the camera field of view. Default is 50 mm.

### **Use Vision Tool**

When true, the calibration algorithm will use the vision tool specified by the user to locate the calibration target.

### **Vision Tool**

The vision tool used to locate the calibration target in the calibration process.

## **Properties**

### **Align Z with Tool**

A flag indicating whether to align the Z axis to the nominal tool axis. If true, the Z axis of the camera offset will align with the tool axis. If false, the Z axis of the camera offset will be determined by the plane of the taught point cluster.

### **Keyword Mapping**

Describes how the calibration is referenced when using V+ ACE Sight keywords for: BeltCalibrationFrame, BeltCalibrationUpstreamLimit, BeltCalibrationDownstreamLimit, InverseKinematics, ImageOriginRobot, VisionOriginRobot, ImageOriginBelt, VisionOriginBelt.

### **Camera Index**

The camera index is passed through the "toolID" parameter in the V+ ACE Sight keyword. Default is 0.

## **Robot Index**

The robot index is passed through the "resultIndex" parameter in the V+ ACE Sight keyword. Default is 0.

## **Read Robot Position Latch**

When a picture is taken with this camera, should the latched robot position be used? If false, the current position is used. This parameter is only used for arm- and table-mounted cameras. Default is false.

## **Rotation Configuration**

Describes the range of travel for the robot theta rotation used by the automated calibration wizards. If the starting angle is negative and the ending angle is positive, the gripper will be rotated through the -180/180 degree boundary. If the starting angle is positive and the ending angle is negative, the gripper will be rotated through the 0 degree boundary.

Full range of motion, or Limit Range, From, To, and Number of Points as parameters.

## **Settling Time**

The amount of time, in milliseconds, the robot will wait after completing a motion when performing an automated calibration. Default is 500 ms.

## **Straight Line Motions**

If True, the robot will use straight-line motions in automatic mode.

If False, the robot will use joint-interpolated motions in automatic mode.

You can now run the ACE Sight Camera Calibration Wizard from the Camera Calibration Object window.

## **Related Topics**

[Calibration Overview](#)

[ACE Sight Camera Calibration](#)

[ACE Sight Camera Calibration Wizard - Automated, no Belt](#)

[ACE Sight Camera Calibration Wizard - Automated, w/ Belt](#)

[ACE Sight Camera Calibration Wizard - Automated Upward-Looking](#)

[ACE Sight Camera Calibration Wizard - Automated, Arm-Mount](#)

[ACE Sight Camera Calibration Wizard - Manual, no Belt](#)

[ACE Sight Camera Calibration Wizard - Manual w/Belt](#)

[ACE Sight Camera Calibration Wizard - Manual, Upward-looking](#)

[ACE Sight Camera Calibration Wizard - Manual, Arm-Mount](#)

### **ACE Sight Camera Calibration Wizard - Automated, no Belt**

This topic covers calibrating an arm- or fixed-mounted camera to a robot, without a conveyor belt.

#### **Requirements**

- The robot and camera must be correctly connected and functioning.
- The camera itself must be calibrated. See Standalone Camera Calibration on page 320.
- The ACE Sight Camera Calibration Interview Wizard must have completed successfully.

#### **What this Wizard Does**

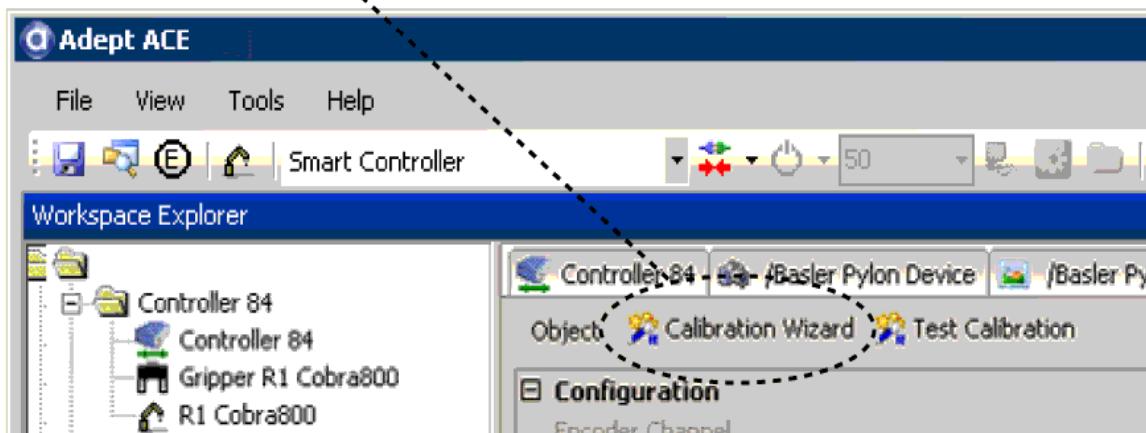
With this wizard you will:

- Teach the picture-taking position, the vision model, and the placement position
- Perform a robot-to-camera calibration

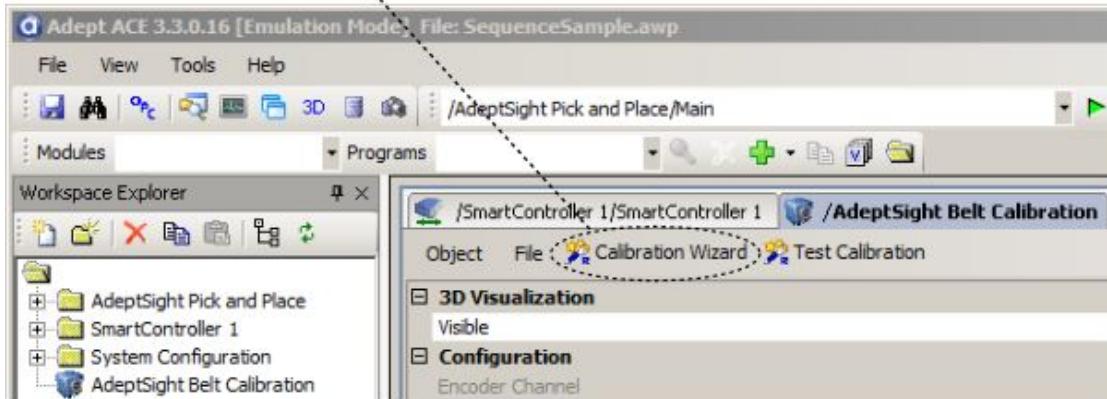
#### **Procedure**

From the Camera Calibration Object window, click Calibration Wizard.

Click on Calibration Wizard after running the Calibration Interview Wizard.



Click on Calibration Wizard after running the Calibration Interview Wizard



The screen titles will all say "Robot-to-Fixed or Arm-Camera Automated Calibration Sequence".

The screens in this procedure are:

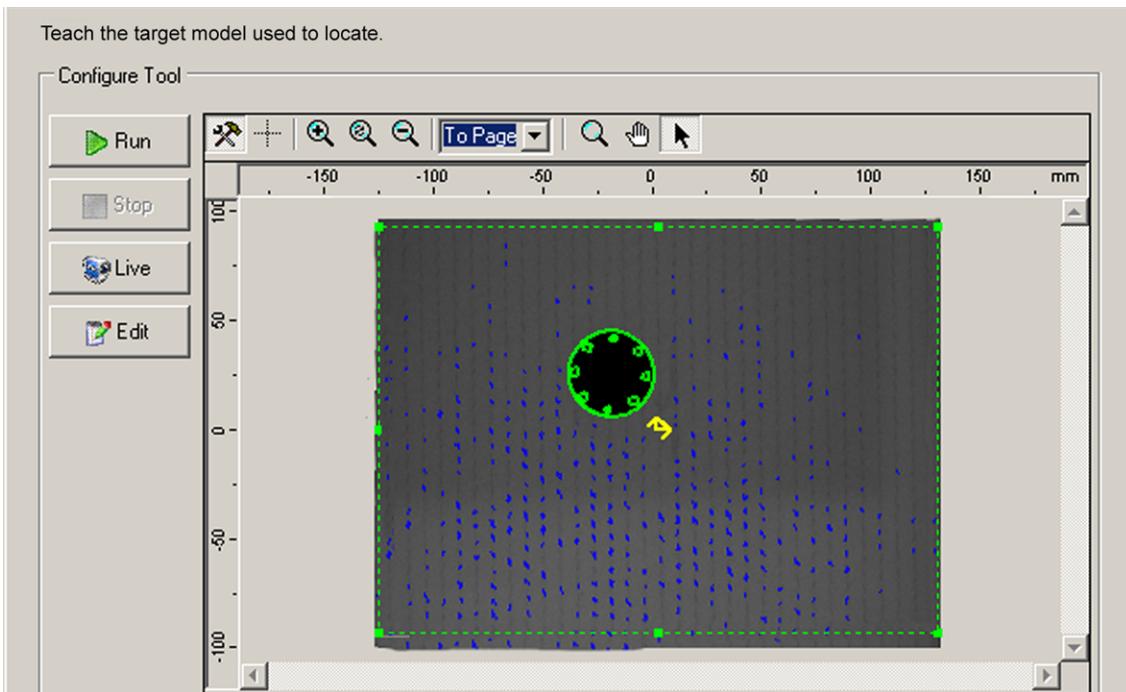
- Select the End-Effector



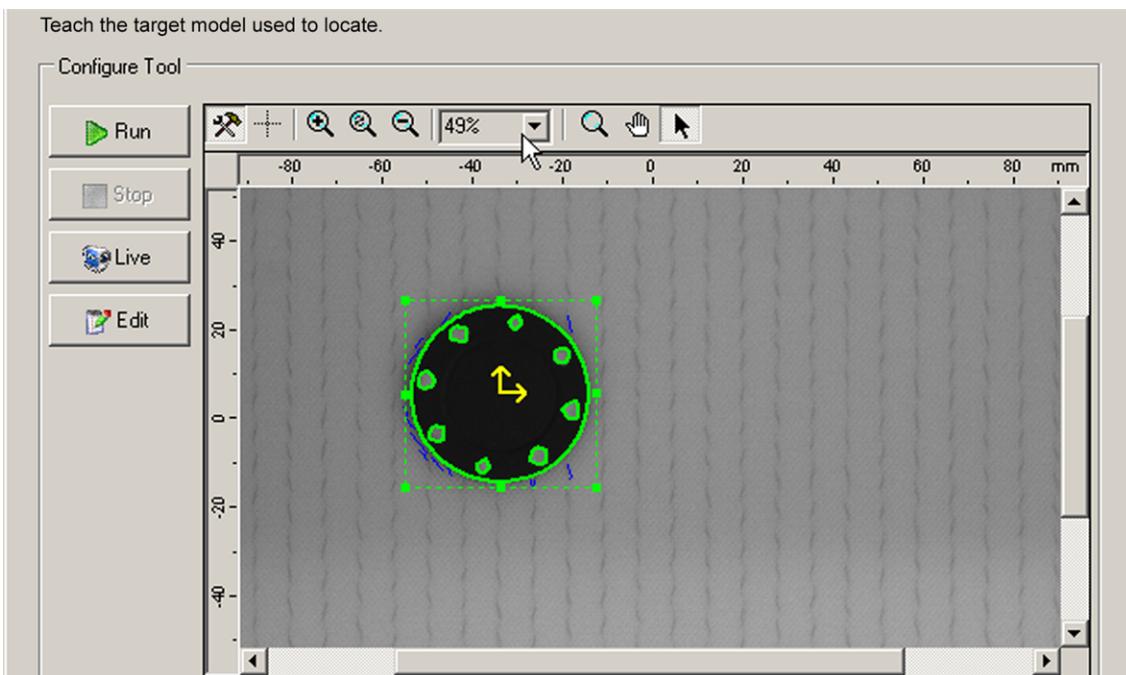
- Move the Robot to the Picture Position
  1. Move the robot out of the way of the camera.  
The robot should not be in the camera's field of view.
  2. Click Here.

- Teach the Vision Tool

The following figures show the model, origin, and bounding box before and after this procedure has been performed.



Vision Model Before Teaching



*Vision Model After Teaching*

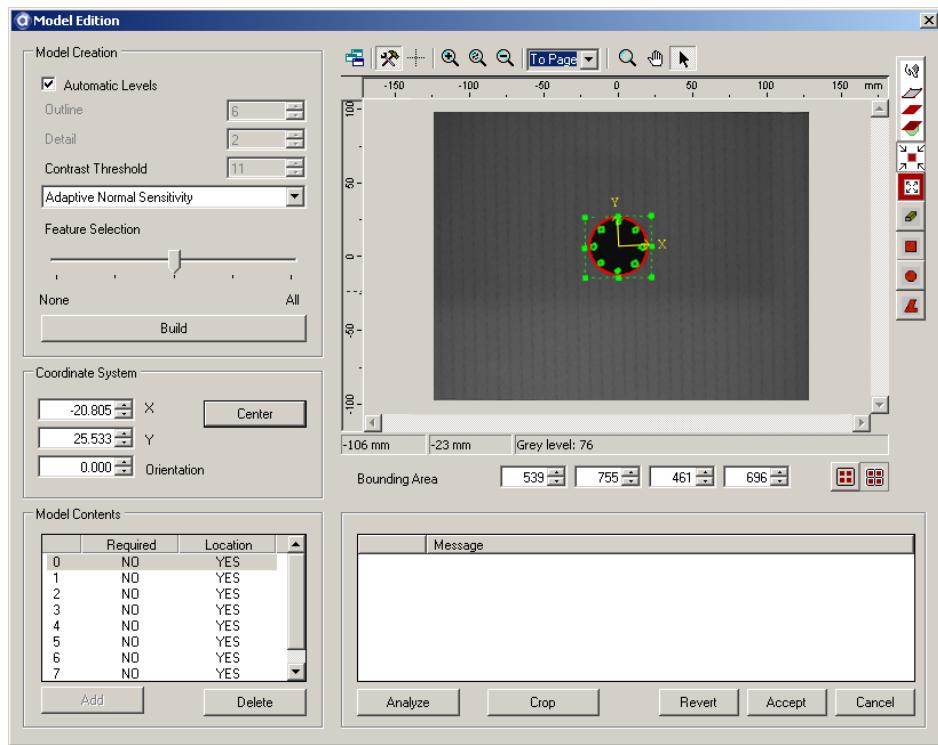
1. Place the target near the center of the work area.
2. Click Run to take an image.
3. Adjust the model bounding box to fully enclose the model.

Make it close to the perimeter of the model, as shown in the previous figure.

4. Move the origin marker to the center of the model.
  - Drag the origin marker to the center of the model with the mouse  
or
  - Click Edit, to open the Edit window (shown in the following figure).
    - Click Center to center the marker on the bounding box  
or
    - Click on the outline of the model, then click Center to center the marker on the found part

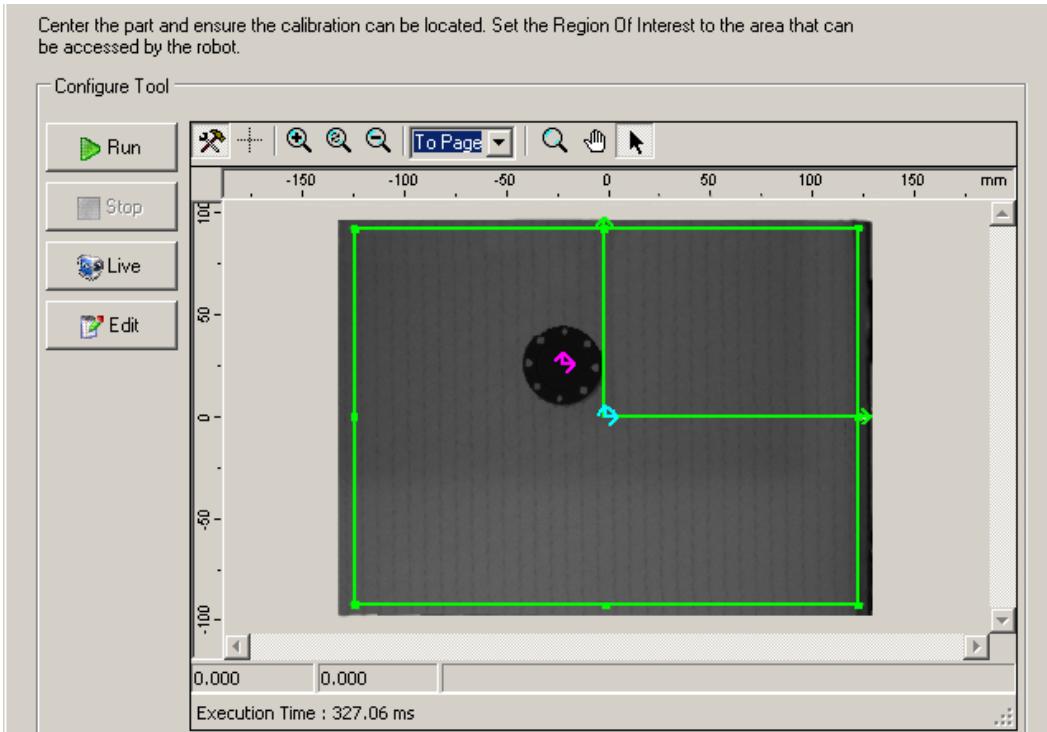
## ACE Sight Camera Calibration

---



*Edit Window*

- Take a Picture of the Calibration Target



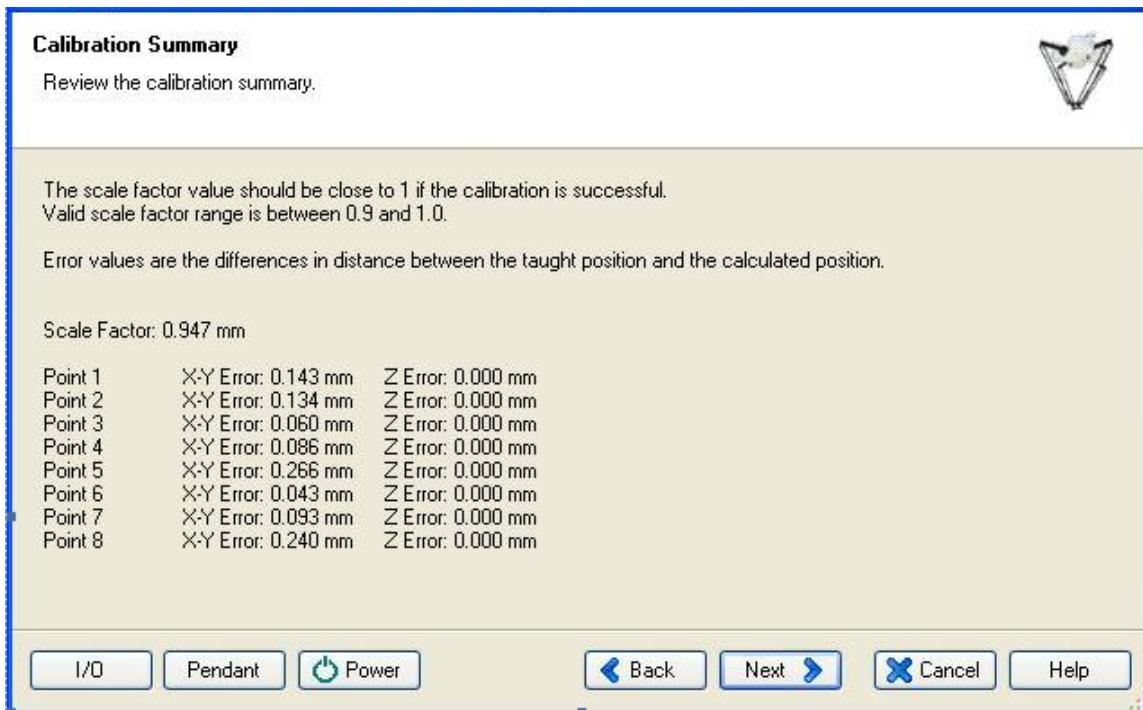
### *Located Calibration Target*

1. Adjust the bounding box to the work area. This must include the calibration target, as shown in the previous figure.
  2. If no magenta origin marker is displayed, click Run.  
If necessary, move the target so it can be located, then click Run.
  3. Click Next.
- Move the Robot to the Pick Position
    1. Move the robot to the target.  
Click Pendant to display the Pendant window. This enables you to move the robot, so that the end-effector is centered over the target.
    2. Click Here.
    3. Click Next.

- Start Automated Calibration
  1. Close the Pendant window by clicking Cancel.
  2. Click Start.

The robot will perform multiple picks and places, taking an image before each pick. The number of picks and places will vary, depending on the type of calibration being performed. It will usually be 10 or more.

When this step is complete, a Calibration Summary screen will be displayed.



*Calibration Summary Screen*

This screen shows the amount of XY and Z difference between the calculated calibration points and the actual points used to generate the calibration. The automated calibration obtains more points than are mathematically required for calibration. The calculated calibration represents an average calibration. When ACE Sight uses the calculated calibration to predict where each actual point should be located, the difference between the actual and calculated point is displayed as the error.

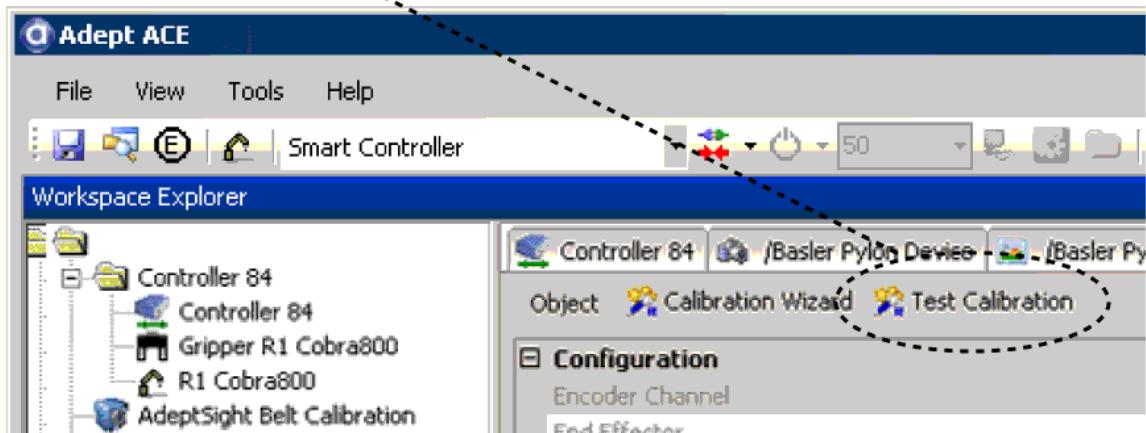
Your Calibration Summary may show a different number of points.

After clicking Next, the Task Manager pane displays "Calibration completed successfully".

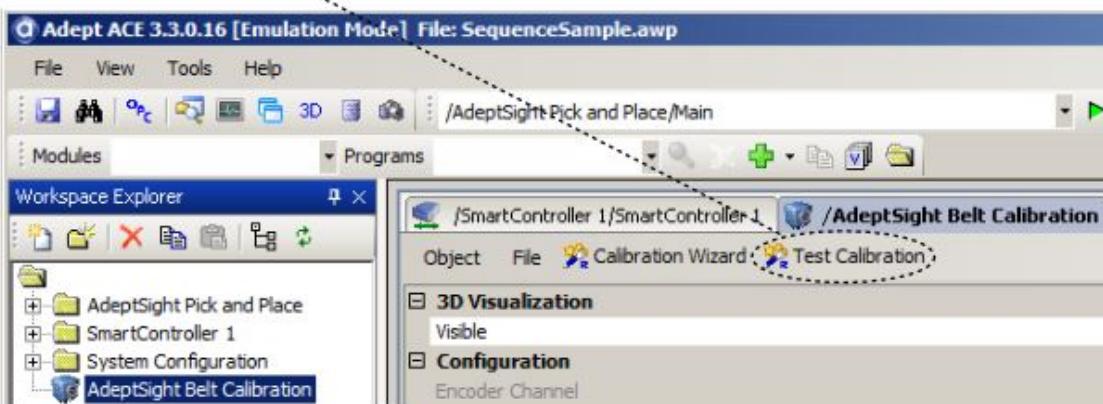
You may need to manually release the gripper at the end of this procedure.

## Test Procedure

Click on Test Calibration after running the Calibration Wizard.



Click on Test Calibration after running the Calibration Wizard



The screens in this procedure are:

- Select End-Effector Tip

This allows you to test a calibration with a different end-effector tip than the one that was used for calibration.

- Move the Robot to the Picture Position

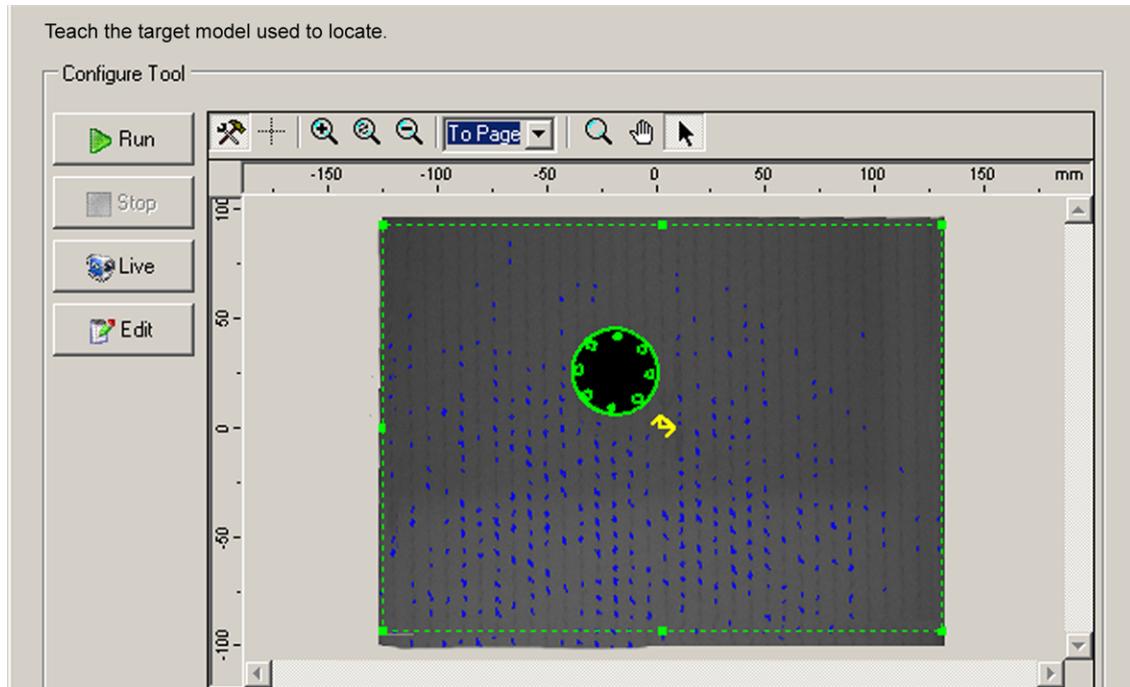
1. Move the robot out of the way of the camera.

The robot should not be in the camera's field of view.

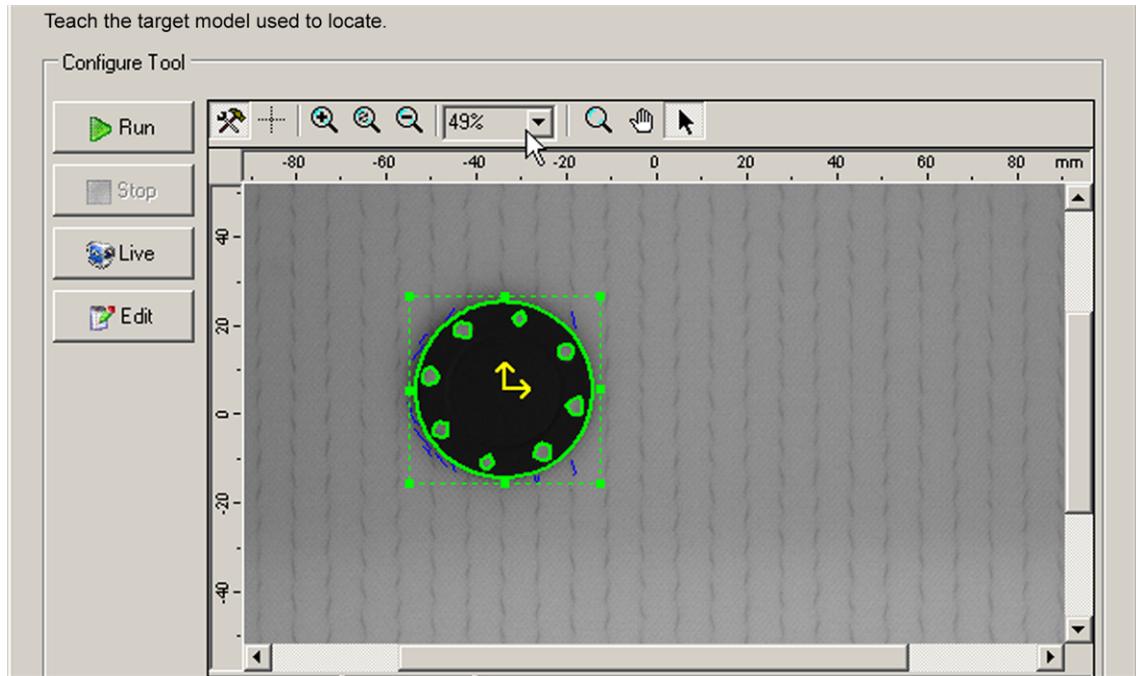
2. Click Here.

- Teach the Vision Tool

The following figures show the model, origin, and bounding box before and after this procedure has been performed.

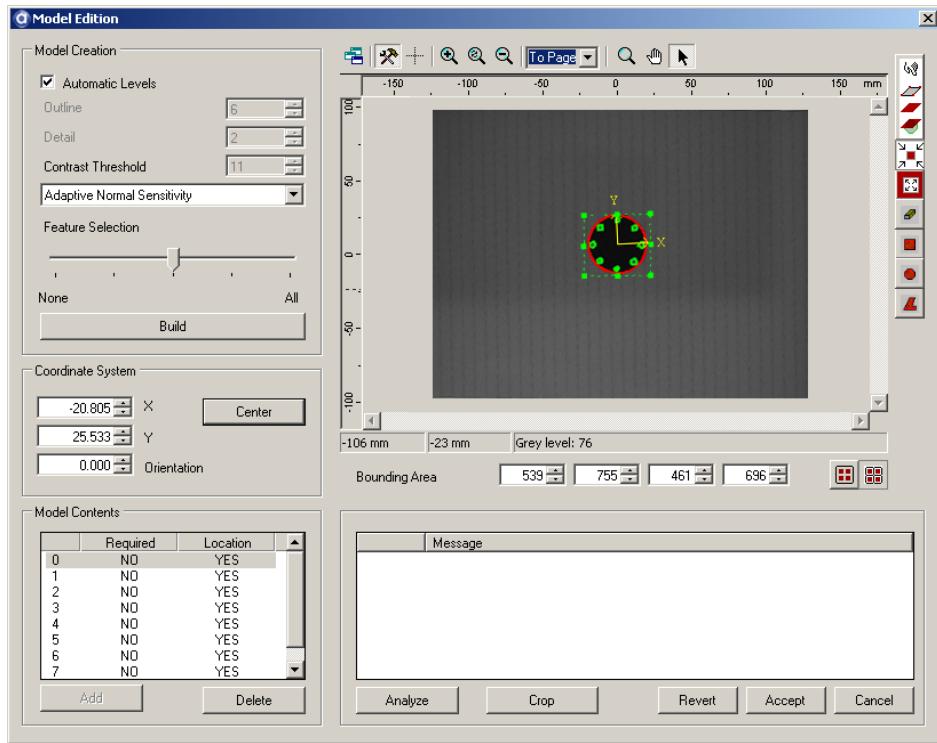


*Vision Model Before Teaching*



*Vision Model After Teaching*

1. Place the target near the center of the work area.
2. Click Run to take an image.
3. Adjust the model bounding box to fully enclose the model.  
Make it close to the perimeter of the model, as shown in the previous figure.
4. Move the origin marker to the center of the model.
  - Drag the origin marker to the center of the model with the mouse  
or
  - Click Edit, to open the Edit window (shown in the following figure).
    - Click Center to center the marker on the bounding box  
or
    - Click on the outline of the model, then click Center to center the marker on the found part



### Edit Window

- Locating the Target

See the figure *Located Calibration Target*.

- Adjust the bounding box to the work area. This must include the calibration target.

This will most likely be correctly adjusted from the calibration procedure.

- If no magenta origin marker is displayed, click Run.

If necessary, move the target so it can be located, then click Run.

- Click Next. The robot should move to the target.

- Continue testing the calibration

  - I wish to continue testing

    - Move the robot to the location where the picture is taken

    - Locating the calibration target

    - Do not continue testing

## Related Topics

[Calibration Overview](#)

[ACE Sight Camera Calibration](#)

[ACE Sight Camera Calibration Interview Wizard](#)

[ACE Sight Camera Calibration Wizard - Automated, w/ Belt](#)

[ACE Sight Camera Calibration Wizard - Automated Upward-Looking](#)

[ACE Sight Camera Calibration Wizard - Manual, no Belt](#)

[ACE Sight Camera Calibration Wizard - Manual w/Belt](#)

[ACE Sight Camera Calibration Wizard - Automated, Arm-Mount](#)

[ACE Sight Camera Calibration Wizard - Manual, Arm-Mount](#)

[ACE Sight Camera Calibration Wizard - Manual, Upward-looking](#)

### ***ACE Sight Camera Calibration Wizard - Automated, with Belt***

This topic covers calibrating an arm- or fixed-mounted camera to a robot, with a conveyor belt.

## Requirements

- The robot, belt, and camera must be correctly connected and functioning.
- The camera itself must be calibrated. See Standalone Camera Calibration on page 320.
- The ACE Sight Belt Calibration Wizard must have completed successfully.
- The ACE Sight Camera Calibration Interview Wizard must have completed successfully.

## What this Wizard Does

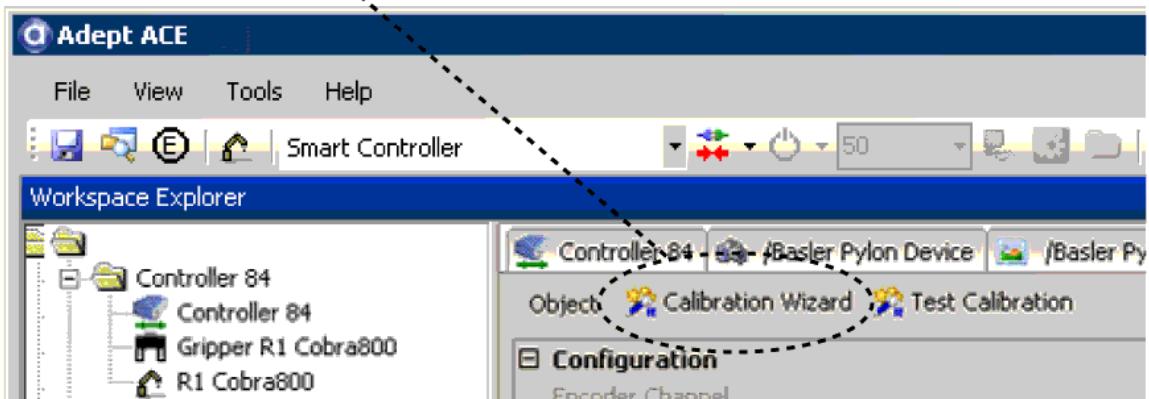
With this wizard you will:

- Teach the picture-taking position, the vision model, and the placement position
- Perform a robot-to-camera calibration

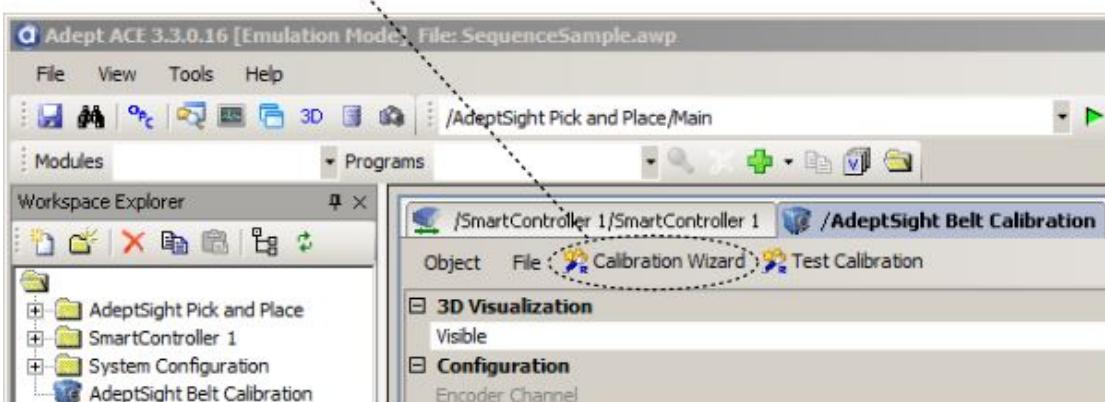
## Procedure

From the ACE Sight Camera Calibration Object window, click Calibration Wizard.

Click on Calibration Wizard after running the Calibration Interview Wizard.



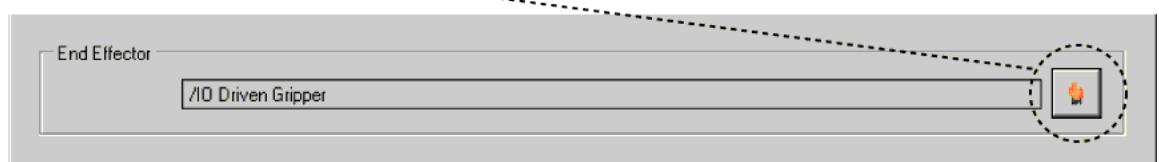
Click on Calibration Wizard after running the Calibration Interview Wizard



The screen titles will all say "Robot-to-Belt Camera Calibration Sequence".

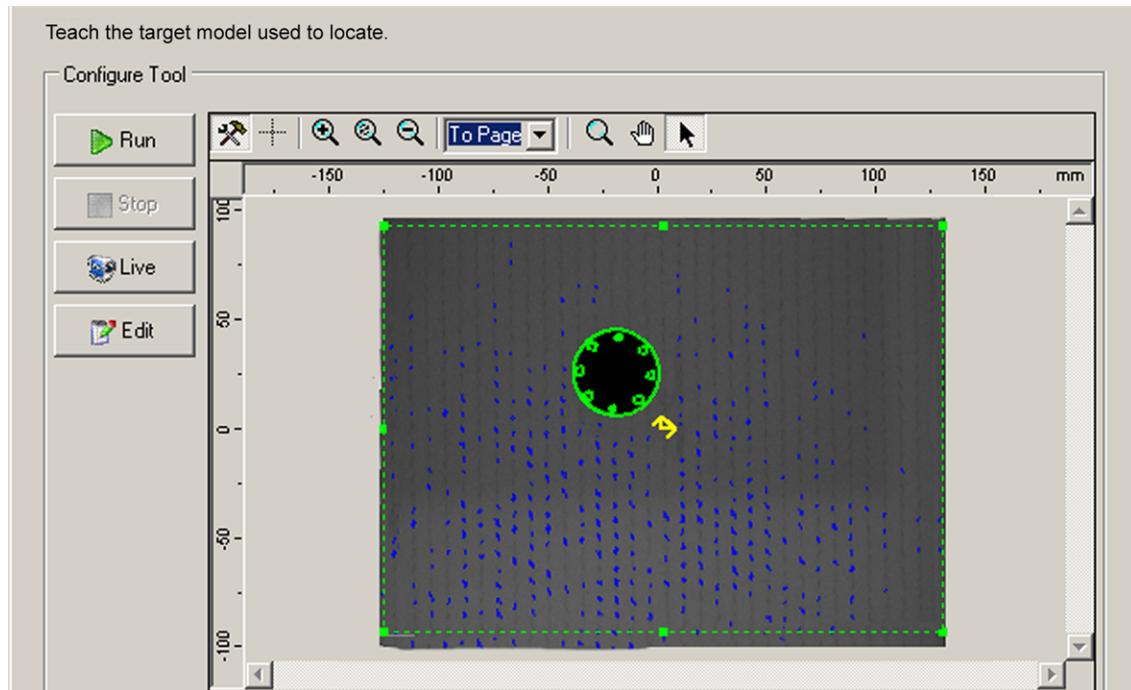
The screens in this procedure are:

- Select the End-Effector

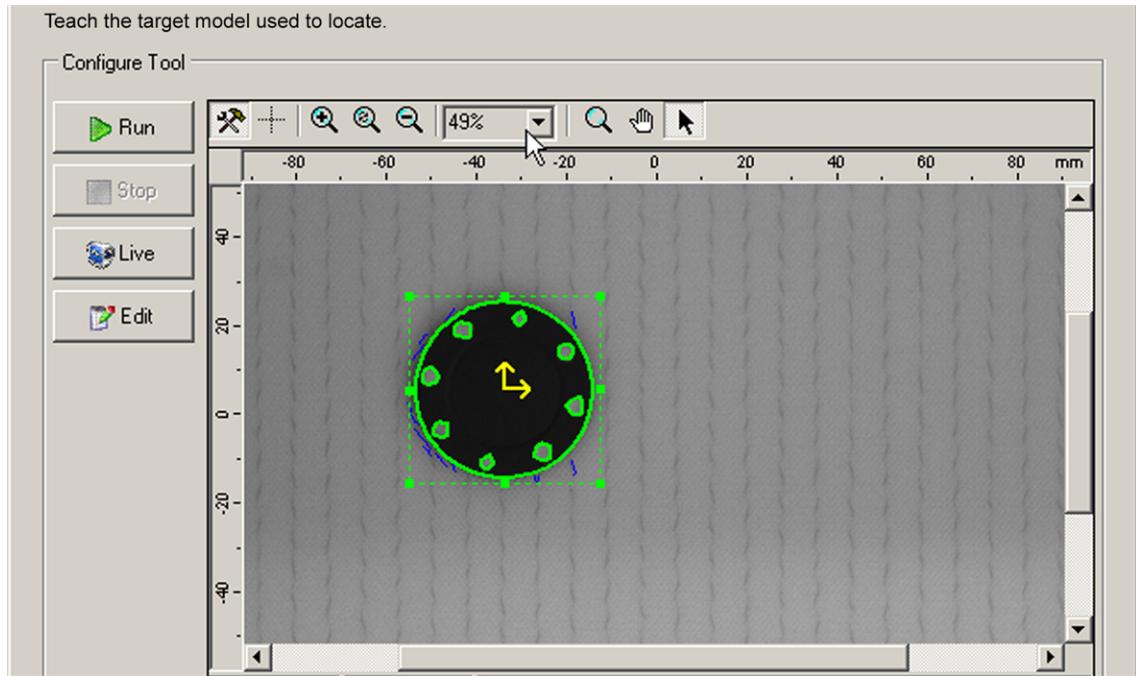


- Teach the Vision Tool

The following figures show the model, origin, and bounding box before and after this procedure has been performed.



*Vision Model Before Teaching*

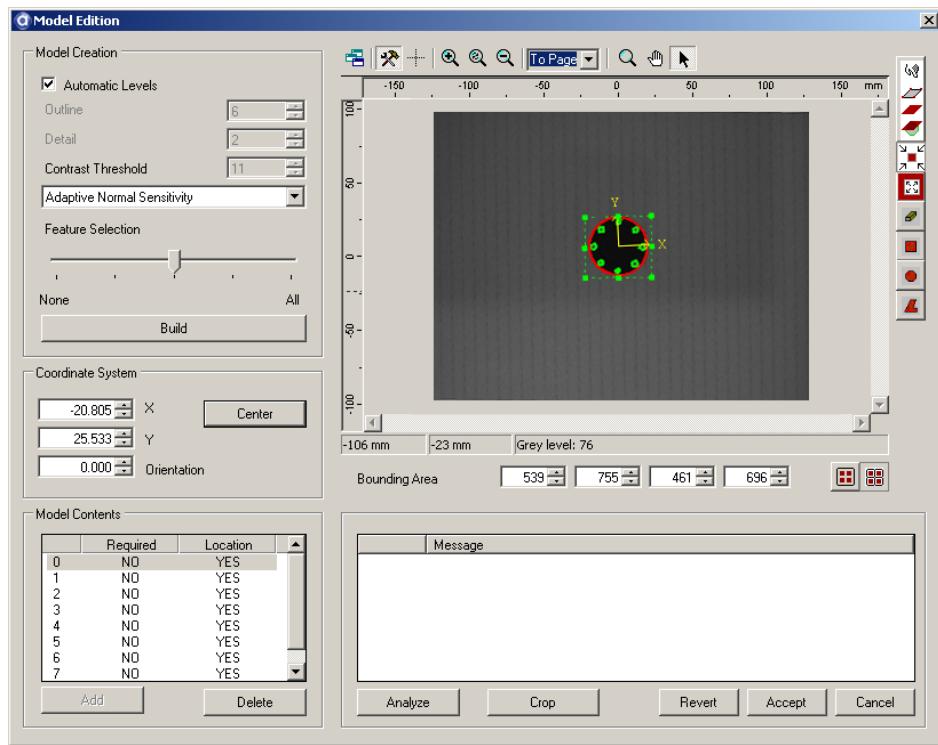


*Vision Model After Teaching*

1. Place the target near the center of the work area.
2. Click Run to take an image.
3. Adjust the model bounding box to fully enclose the model.  
Make it close to the perimeter of the model, as shown in the previous figure.
4. Move the origin marker to the center of the model.
  - Drag the origin marker to the center of the model with the mouse  
or
  - Click Edit, to open the Edit window (shown in the following figure).
    - Click Center to center the marker on the bounding box  
or
    - Click on the outline of the model, then click Center to center the marker on the found part

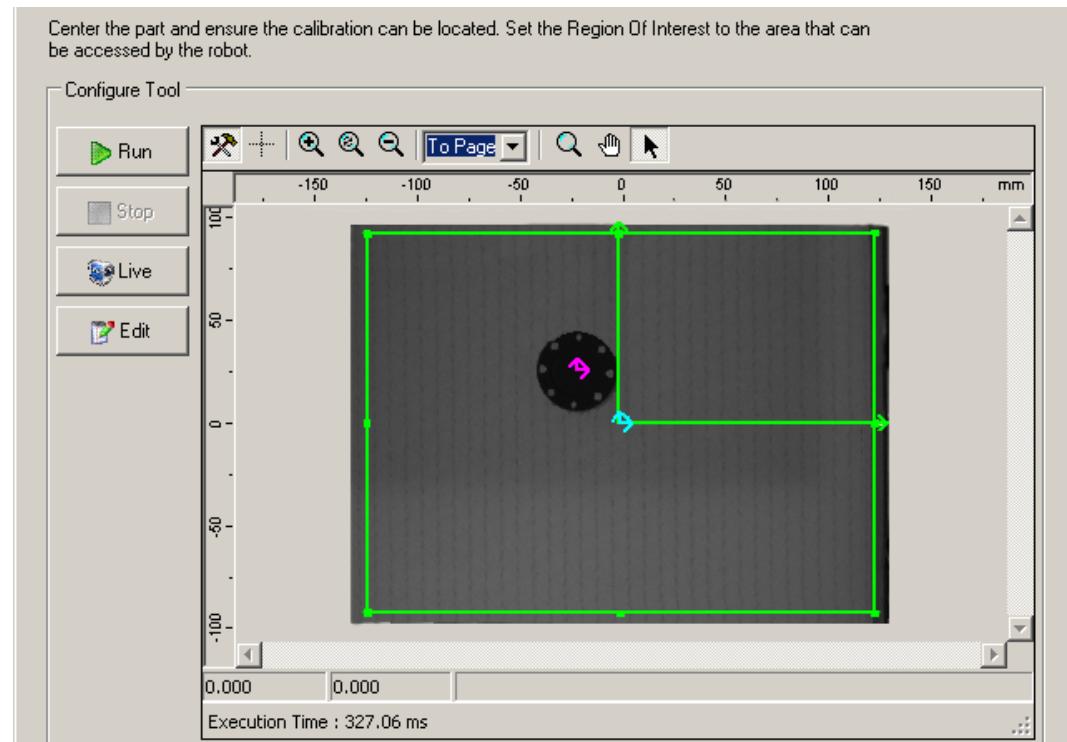
## ACE Sight Camera Calibration

---



*Edit Window*

- Take a Picture of the Calibration Target



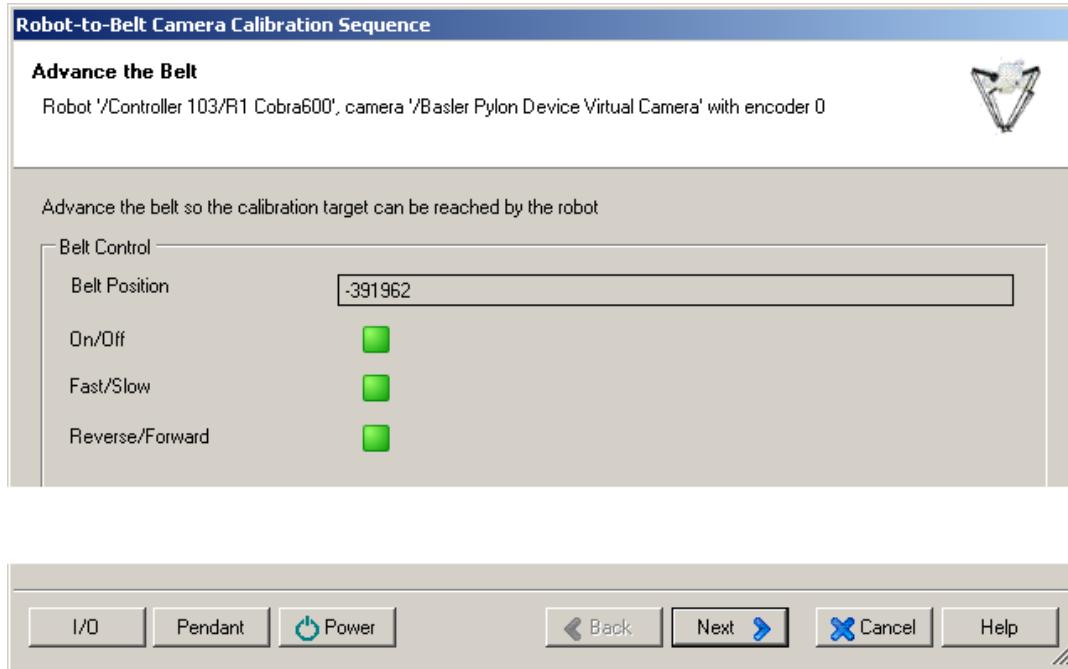
*Located Calibration Target*

1. Place the target near the upstream limit, within the work area.
2. Adjust the bounding box to the work area. This must include the calibration target.
3. If the target isn't marked with a magenta origin, click Run.

If necessary, move the target so it can be located, then click Run.

- Advance the Belt

**NOTE:** The belt will be advanced four times in this procedure. Advance the belt so that the target moves less than a quarter of the total distance from point 1 to the downstream limit.



Do not touch the target.

You can click I/O to control digital signals, if needed for moving the belt.

1. Align the robot with the target.
2. Click Here.

The following steps get repeated for  $n = 2$  through 4:

- Teach the robot position at point 1.
- Take a picture of the calibration target.
  1. Move the robot to the location where the picture is taken.
  2. Click Run.

The origin of the target should be displayed in magenta.

- Advance the belt.

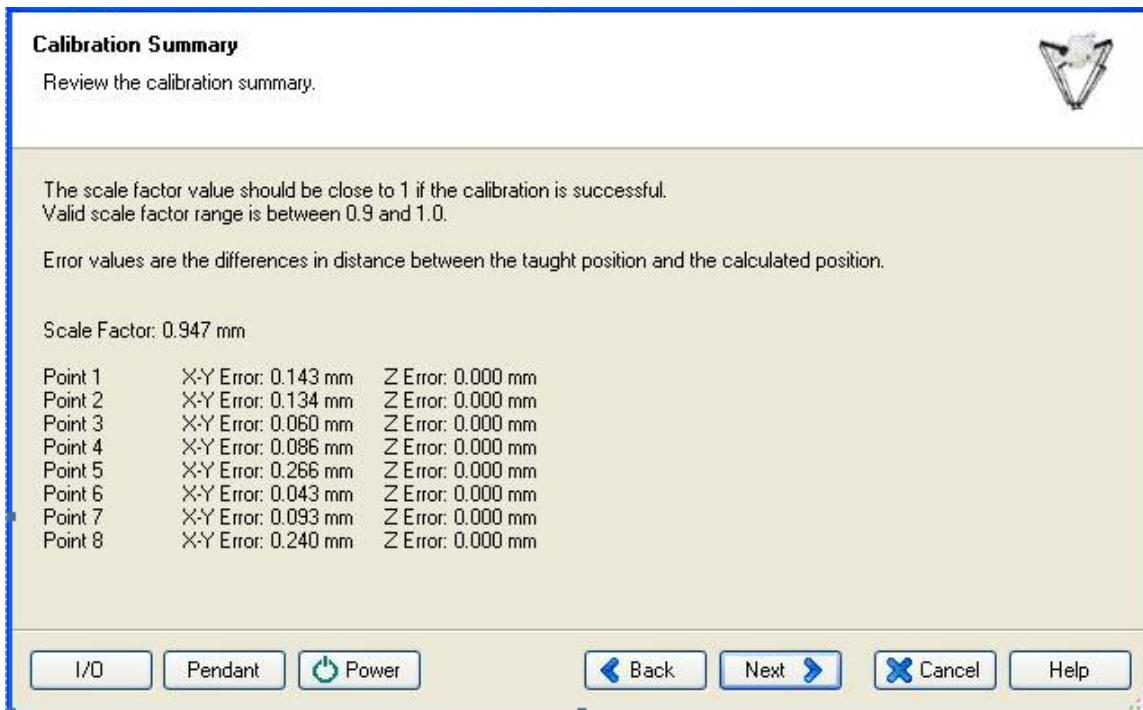
Do not touch the target.
- Move the robot above the target for point n.
  1. Align the robot with the target.
  2. Click Here.

After the four iterations are completed, the automated calibration can be performed.

- Start Automated Calibration
  1. Close the Pendant window by clicking Cancel.
  2. Click Start.

The robot will perform multiple picks and places, taking an image before each pick. The number of picks and places will vary, depending on the type of calibration being performed. It will usually be 10 or more.

When this step is complete, a Calibration Summary screen will be displayed.



*Calibration Summary Screen*

This screen shows the amount of XY and Z difference between the calculated calibration points and the actual points used to generate the calibration. The automated calibration obtains more points than are mathematically required for calibration. The calculated calibration represents an average calibration. When ACE Sight uses the calculated calibration to predict where each actual point should be located, the difference between the actual and calculated point is displayed as the error.

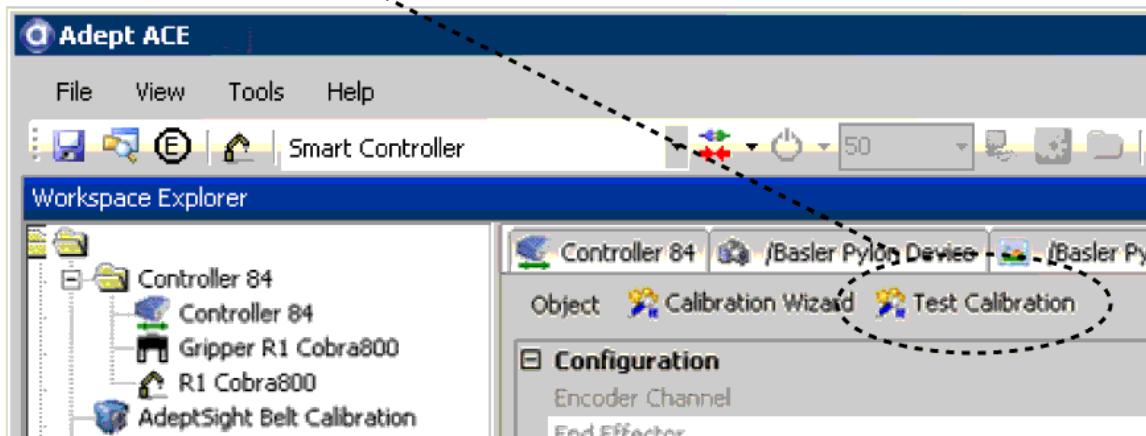
Your Calibration Summary may show a different number of points.

After clicking Next, the Task Manager pane displays "Calibration completed successfully".

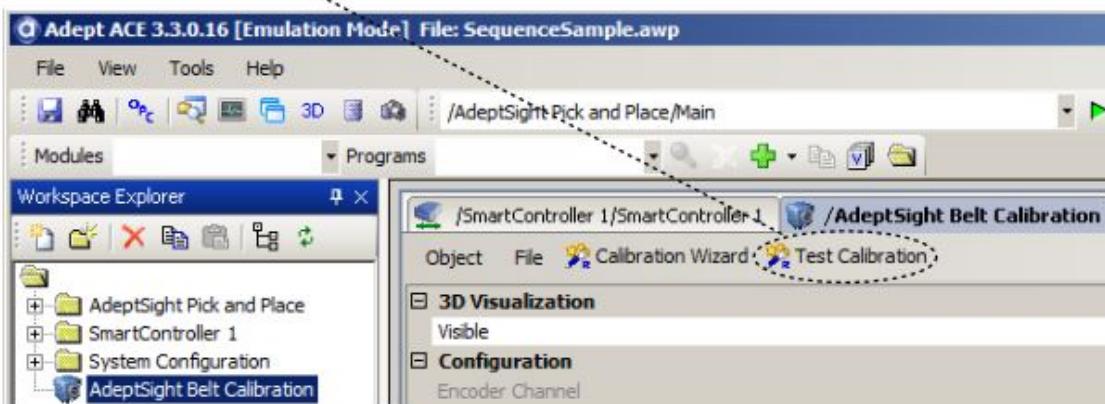
You may need to manually release the gripper at the end of this procedure.

## Test Procedure

Click on Test Calibration after running the Calibration Wizard.



Click on Test Calibration after running the Calibration Wizard



The screens in this procedure are:

- Select End-Effector Tip

This allows you to test a calibration with a different end-effector tip than the one that was used for calibration.

- Move the Robot to the Picture Position

1. Move the robot out of the way of the camera.

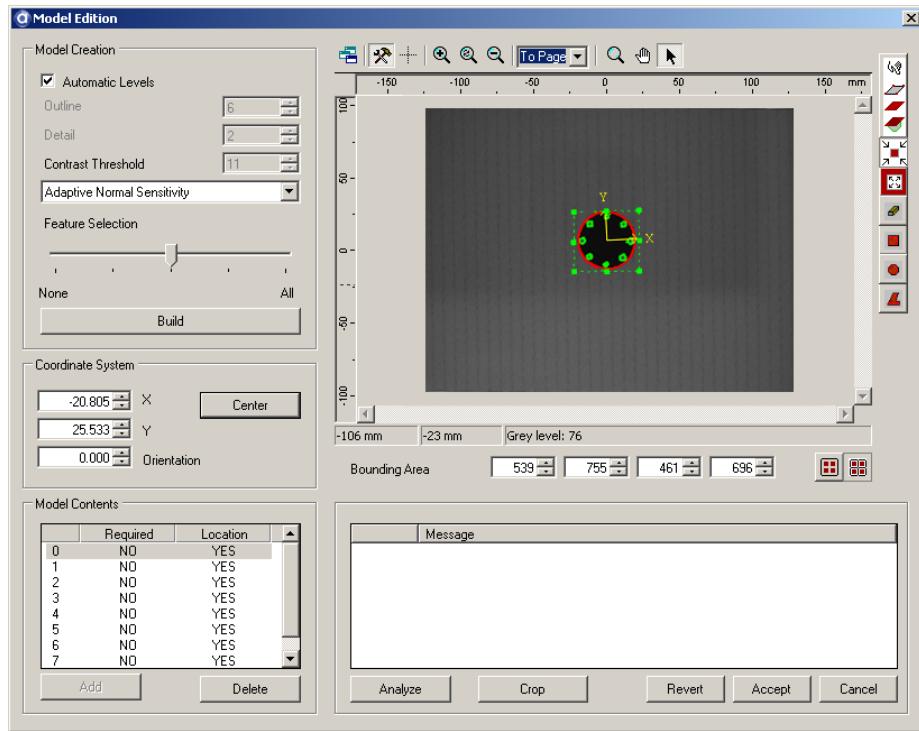
The robot should not be in the camera's field of view.

2. Click Here.
    - Teach the Vision Tool
- See the preceding figures, *Vision Model Before Teaching* and *Vision Model After Teaching*.
1. Place the target near the center of the work area.
  2. Click Run to take an image.
  3. Adjust the model bounding box to fully enclose the model.  
Make it close to the perimeter of the model, as shown in the following figure.
  4. Move the origin marker to the center of the model.
    - Drag the origin marker to the center of the model with the mouse  
or
    - Click Edit
      - Click Center to center the marker on the bounding box  
or
      - Click on the outline of the model, then click Center to center the marker on the found part

The Edit window is shown in the following figure.

## ACE Sight Camera Calibration

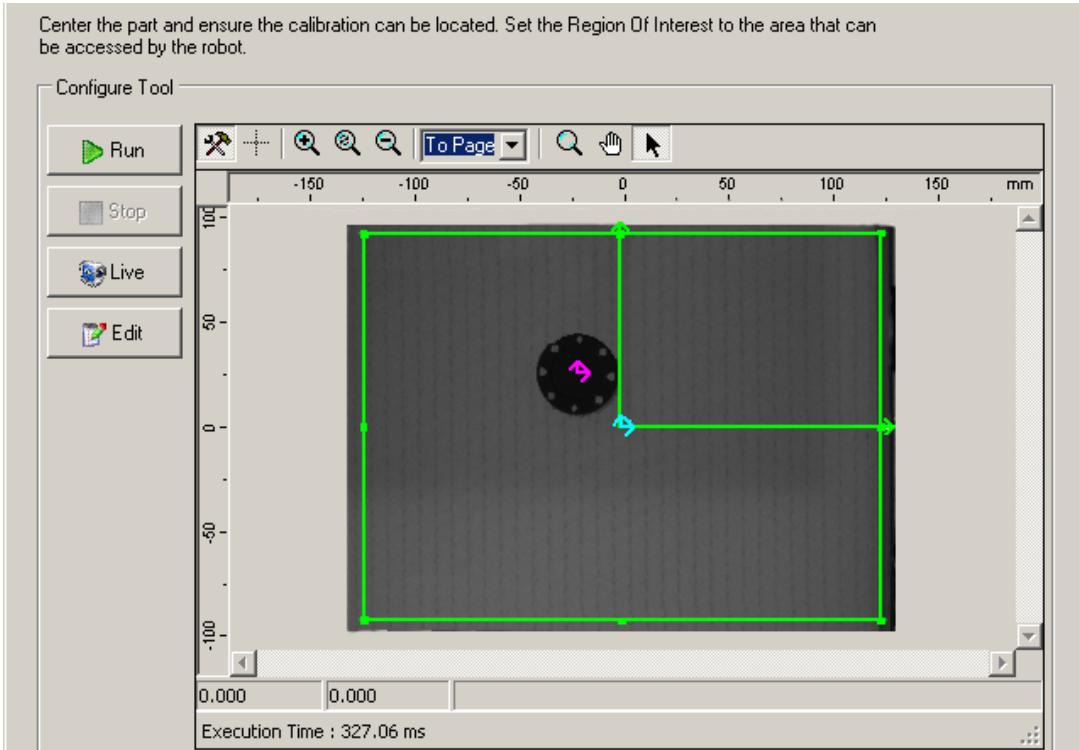
---



*Edit Window*

- Locate the Target

See the following figure.



### *Locating Calibration Target*

1. Place the target near the upstream limit, within the work area.
2. Adjust the bounding box to the work area. This must include the calibration target.
3. If the target isn't marked with a magenta origin, click Run.

If necessary, move the target so it can be located, then click Run.

### **Related Topics**

[Calibration Overview](#)

[ACE Sight Belt Calibration Interview Wizard](#)

[ACE Sight Belt Calibration Wizard](#)

[ACE Sight Camera Calibration Interview Wizard](#)

[ACE Sight Camera Calibration Wizard - Automated, no Belt](#)

[ACE Sight Camera Calibration Wizard - Automated, Arm-Mount](#)

[ACE Sight Camera Calibration Wizard - Automated Upward-Looking](#)

[ACE Sight Camera Calibration Wizard - Manual, no Belt](#)

[ACE Sight Camera Calibration Wizard - Manual w/Belt](#)

[ACE Sight Camera Calibration Wizard - Manual, Arm-Mount](#)

[ACE Sight Camera Calibration Wizard - Manual, Upward-looking](#)

### ***ACE Sight Camera Calibration Wizard - Automated, Arm-Mount***

This topic covers calibrating an arm-mounted camera to a robot, without a conveyor belt.

#### **Requirements**

- The robot and camera must be correctly connected and functioning.
- The camera itself must be calibrated. See Standalone Camera Calibration on page 320.
- The ACE Sight Camera Calibration Interview Wizard must have completed successfully.

#### **What this Wizard Does**

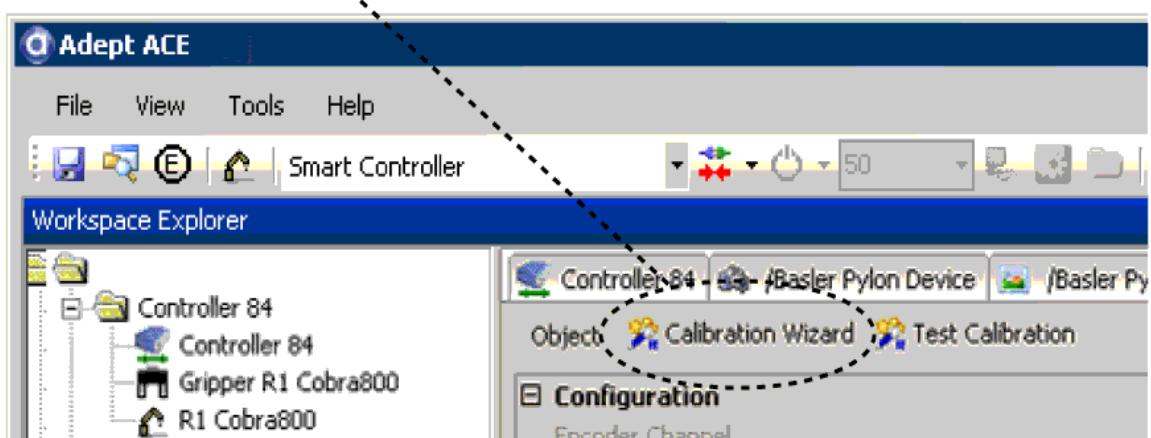
With this wizard you will:

- Teach the picture-taking position, the vision model, and the placement position
- Perform a robot-to-camera calibration

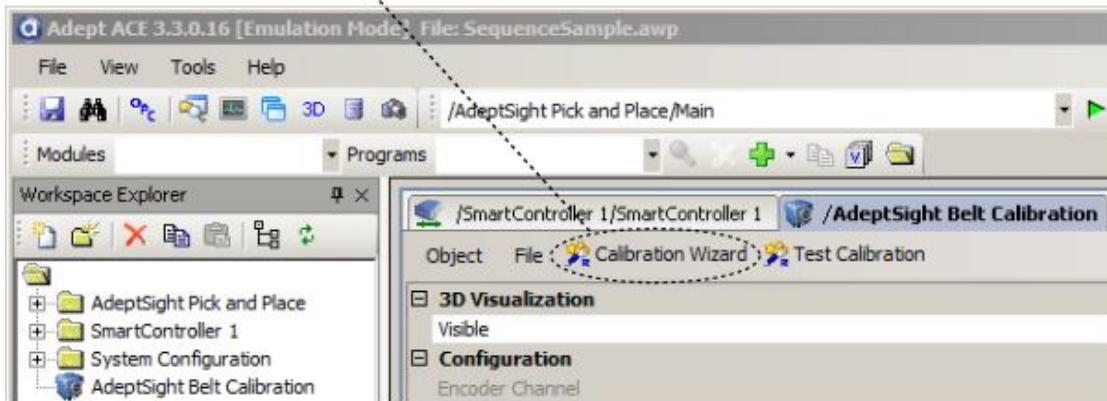
#### **Procedure**

From the Camera Calibration Object window, click Calibration Wizard.

Click on Calibration Wizard after running the Calibration Interview Wizard.



Click on Calibration Wizard after running the Calibration Interview Wizard

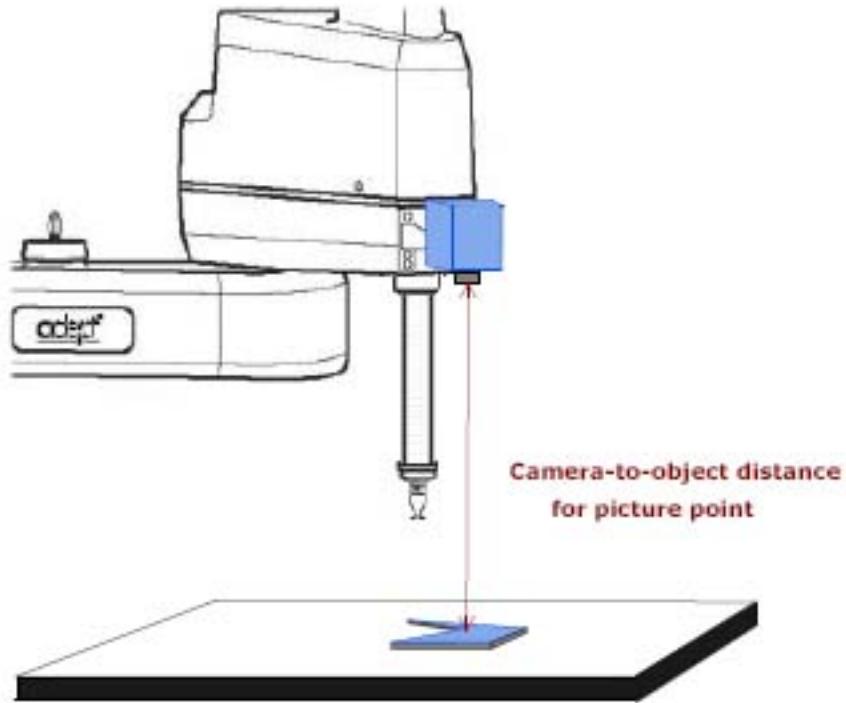


The screen titles will all say "Robot-to-Fixed or Arm-Camera Automated Calibration Sequence".

The screens in this procedure are:

- Select the End-Effector

- Move the Robot to the Picture Position



1. Move the robot so the camera is over the target.

The target must be in the camera's field of view.

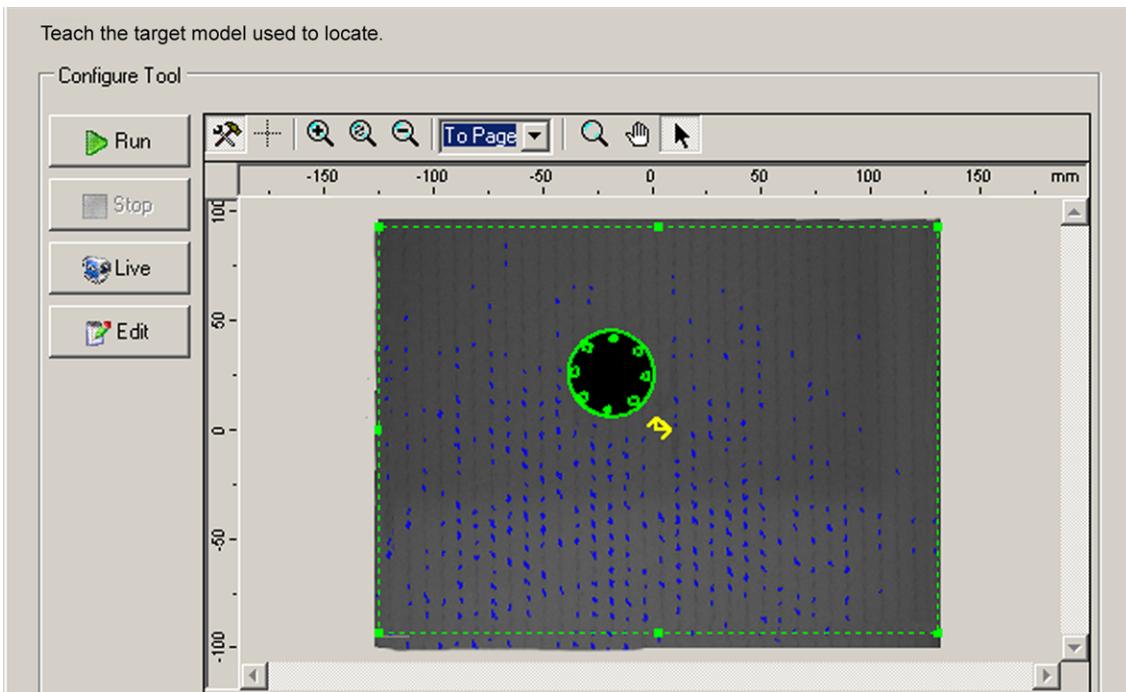
2. Click Here.

- Teach the Vision Tool

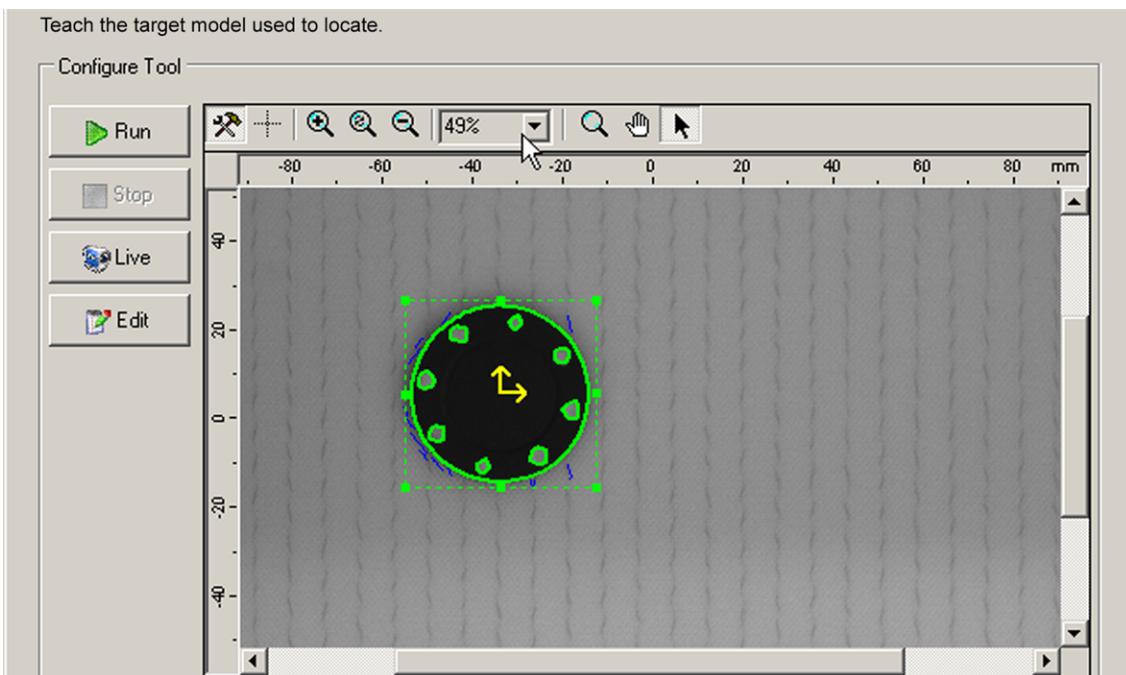
The following figures show the model, origin, and bounding box before and after this procedure has been performed.

## ACE Sight Camera Calibration

---



Vision Model Before Teaching



*Vision Model After Teaching*

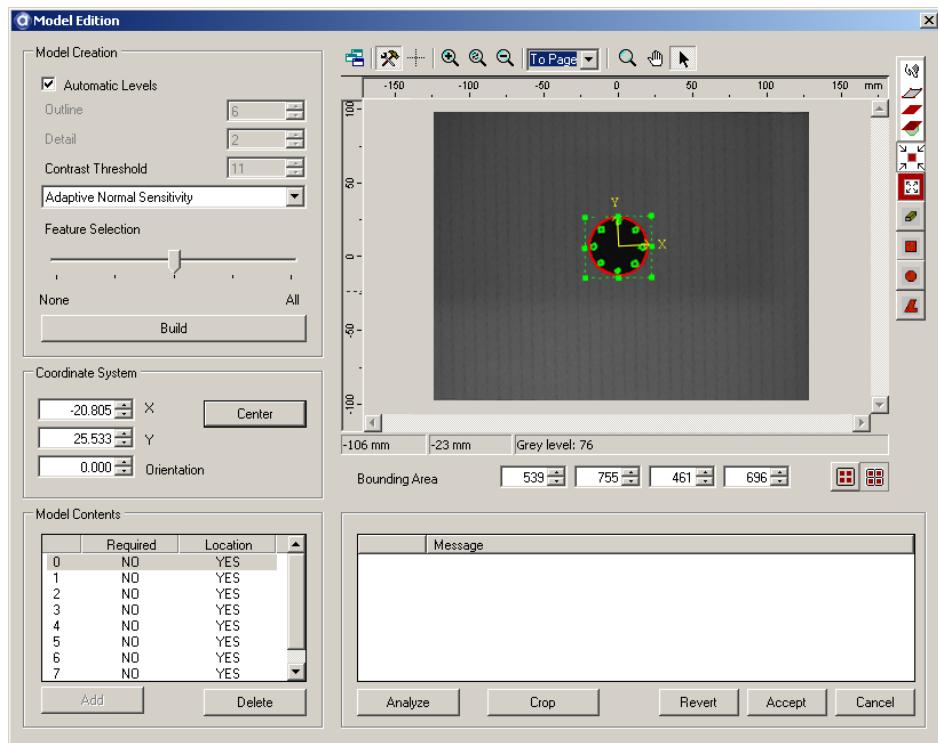
1. Place the target near the center of the work area.
2. Click Run to take an image.
3. Adjust the model bounding box to fully enclose the model.

Make it close to the perimeter of the model, as shown in the previous figure.

4. Move the origin marker to the center of the model.
  - Drag the origin marker to the center of the model with the mouse  
or
  - Click Edit, to open the Edit window (shown in the following figure).
    - Click Center to center the marker on the bounding box  
or
    - Click on the outline of the model, then click Center to center the marker on the found part

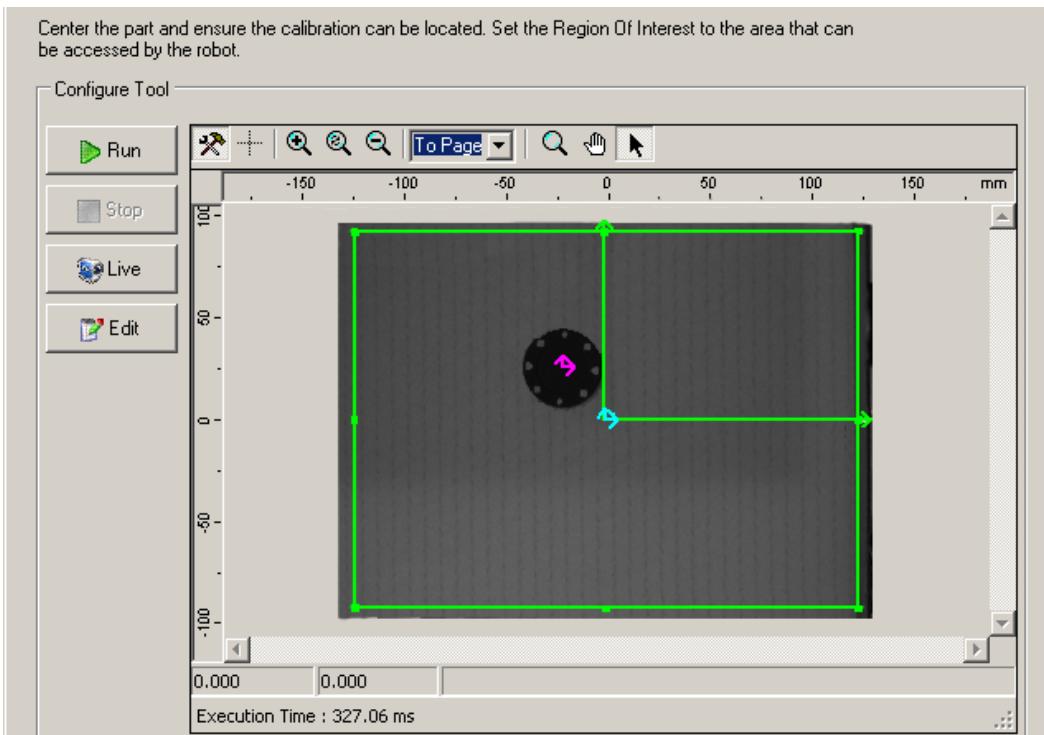
## ACE Sight Camera Calibration

---



*Edit Window*

- Take a Picture of the Calibration Target



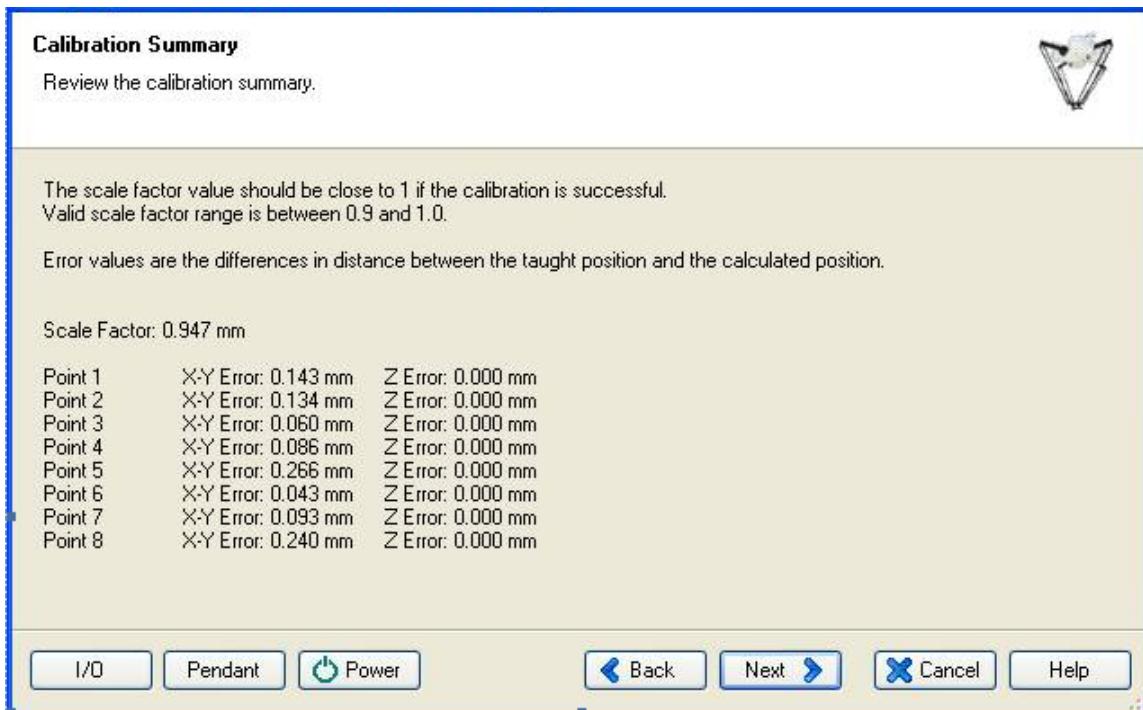
### *Located Calibration Target*

1. Adjust the bounding box to the work area. This must include the calibration target, as shown in the previous figure.
  2. If no magenta origin marker is displayed, click Run.  
If necessary, move the target so it can be located, then click Run.
  3. Click Next.
- Define Calibration Point
    1. Move the robot to the target.  
Click Pendant to display the Pendant window. This enables you to move the robot, so that the end-effector is centered over the target.
    2. Click Here.

- Start Automated Calibration
  1. Close the Pendant window by clicking Cancel.
  2. Click Start.

The robot will perform multiple picks and places, taking an image before each pick. The number of picks and places will vary, depending on the type of calibration being performed. It will usually be 10 or more.

When this step is complete, a Calibration Summary screen will be displayed.



*Calibration Summary Screen*

This screen shows the amount of XY and Z difference between the calculated calibration points and the actual points used to generate the calibration. The automated calibration obtains more points than are mathematically required for calibration. The calculated calibration represents an average calibration. When ACE Sight uses the calculated calibration to predict where each actual point should be located, the difference between the actual and calculated point is displayed as the error.

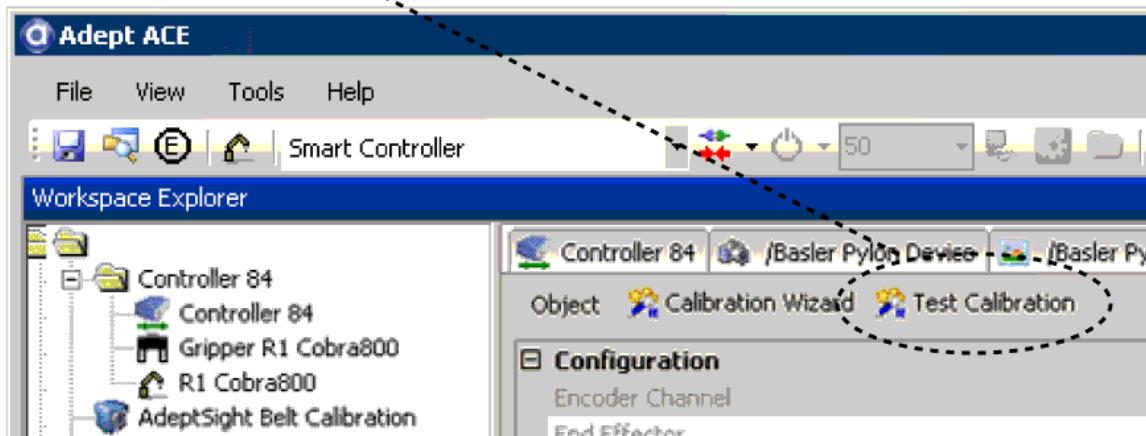
Your Calibration Summary may show a different number of points.

After clicking Next, the Task Manager pane displays "Calibration completed successfully".

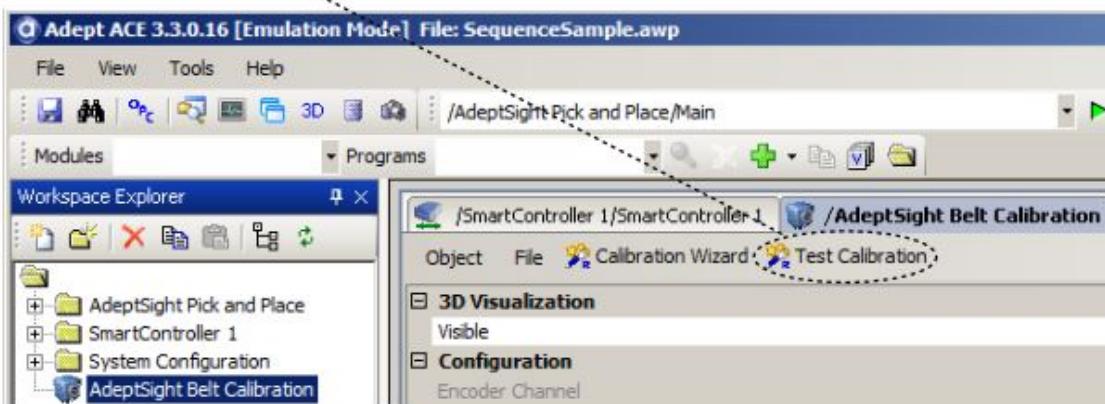
You may need to manually release the gripper at the end of this procedure.

### Test Procedure

Click on Test Calibration after running the Calibration Wizard.



Click on Test Calibration after running the Calibration Wizard

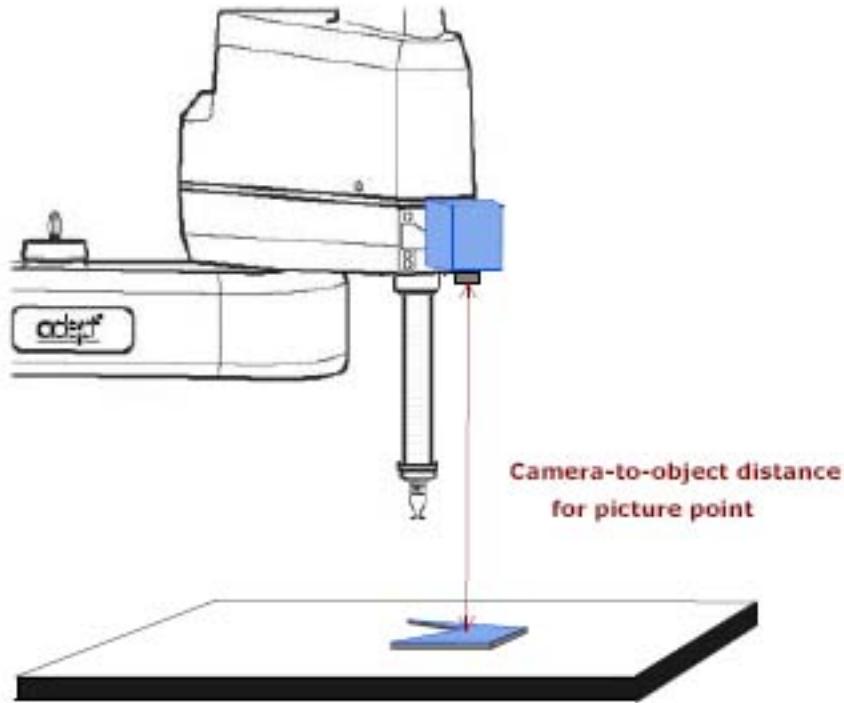


The screens in this procedure are:

- Select End-Effector Tip

This allows you to test a calibration with a different end-effector tip than the one that was used for calibration.

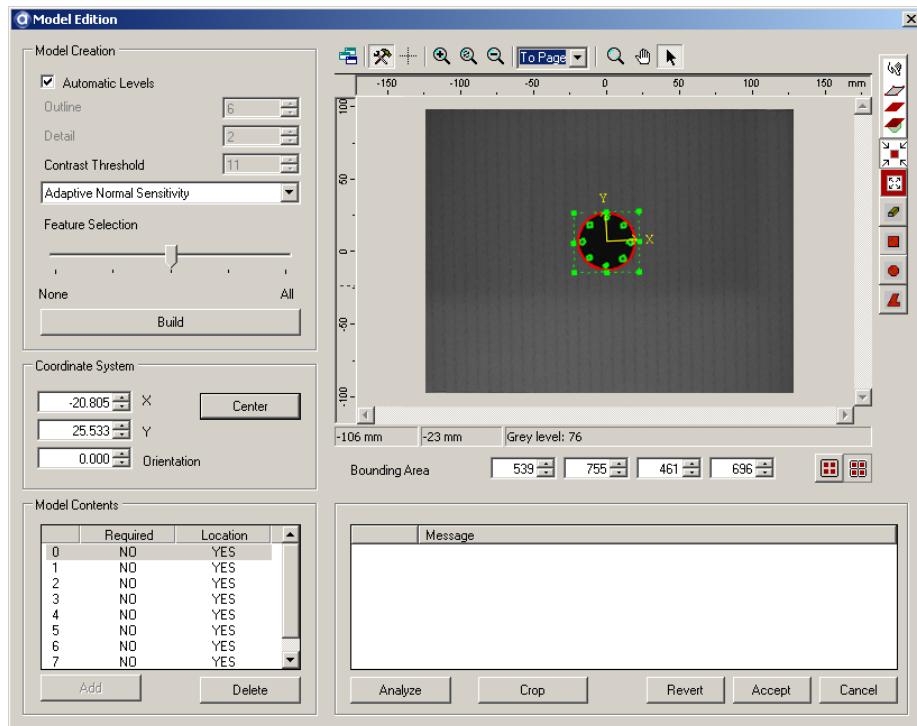
- Move the Robot to the Picture Position



1. Move the robot so the camera is over the target.  
The target must be in the camera's field of view.
  2. Click Here.
- Teach the Vision Tool  
See the preceding figures, *Vision Model Before Teaching* and *Vision Model After Teaching*.
    1. Place the target near the center of the work area.
    2. Click Run to take an image.
    3. Adjust the model bounding box to fully enclose the model.  
Make it close to the perimeter of the model, as shown in the following figure.
    4. Move the origin marker to the center of the model.
      - Drag the origin marker to the center of the model with the mouse
      - or

- Click Edit
  - Click Center to center the marker on the bounding box
  - or
  - Click on the outline of the model, then click Center to center the marker on the found part

The Edit window is shown in the following figure.



*Edit Window*

- Locate the Target
  1. Adjust the bounding box to the work area. This must include the calibration target.
 

**NOTE:** The calibration target should be correctly adjusted from the calibration procedure.
  2. If no magenta origin marker is displayed, click Run.  
If necessary, move the target so it can be located, then click Run.

3. Click Next. The robot should move to the target.
  - Continue testing the calibration
    - I wish to continue testing
      - Move the robot to the location where the picture is taken
      - Locating the calibration target
    - Do not continue testing

## Related Topics

[Calibration Overview](#)

[ACE Sight Belt Calibration Interview Wizard](#)

[ACE Sight Belt Calibration Wizard](#)

[ACE Sight Camera Calibration Interview Wizard](#)

[ACE Sight Camera Calibration Wizard - Automated, no Belt](#)

[ACE Sight Camera Calibration Wizard - Automated, w/ Belt](#)

[ACE Sight Camera Calibration Wizard - Automated Upward-Looking](#)

[ACE Sight Camera Calibration Wizard - Manual, no Belt](#)

[ACE Sight Camera Calibration Wizard - Manual w/Belt](#)

[ACE Sight Camera Calibration Wizard - Manual, Arm-Mount](#)

[ACE Sight Camera Calibration Wizard - Manual, Upward-looking](#)

### ***ACE Sight Camera Calibration Wizard - Automated Upward-Looking***

This topic covers calibrating an upward-looking camera to a robot.

## Requirements

- The robot and camera must be correctly connected and functioning.
- The camera itself must be calibrated. See Standalone Camera Calibration on page 320.
- The ACE Sight Camera Calibration Interview Wizard must have completed successfully.

## What this Wizard Does

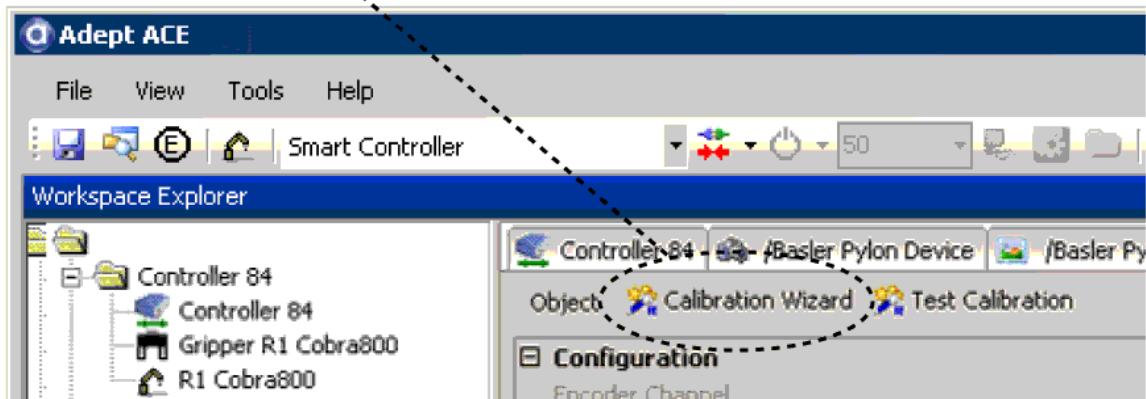
With this wizard you will:

- Teach the picture-taking position and the vision model
- Perform a robot-to-camera calibration

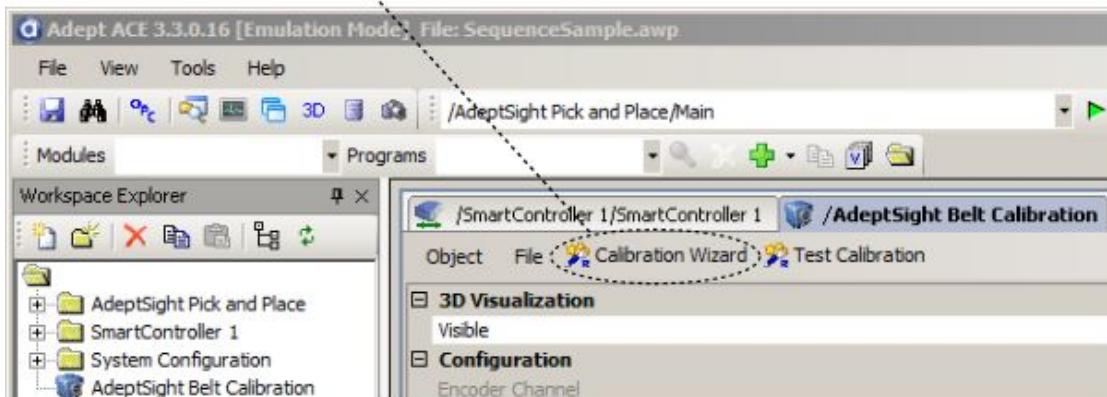
### Procedure

From the ACE Sight Camera Calibration Object window, click Calibration Wizard.

Click on Calibration Wizard after running the Calibration Interview Wizard.



Click on Calibration Wizard after running the Calibration Interview Wizard



The screen titles will all display "Robot-to-Refinement Camera Automated Calibration Sequence".

The screens in this procedure are:

- Select the End-Effector

- Move the Robot to the Picture Position

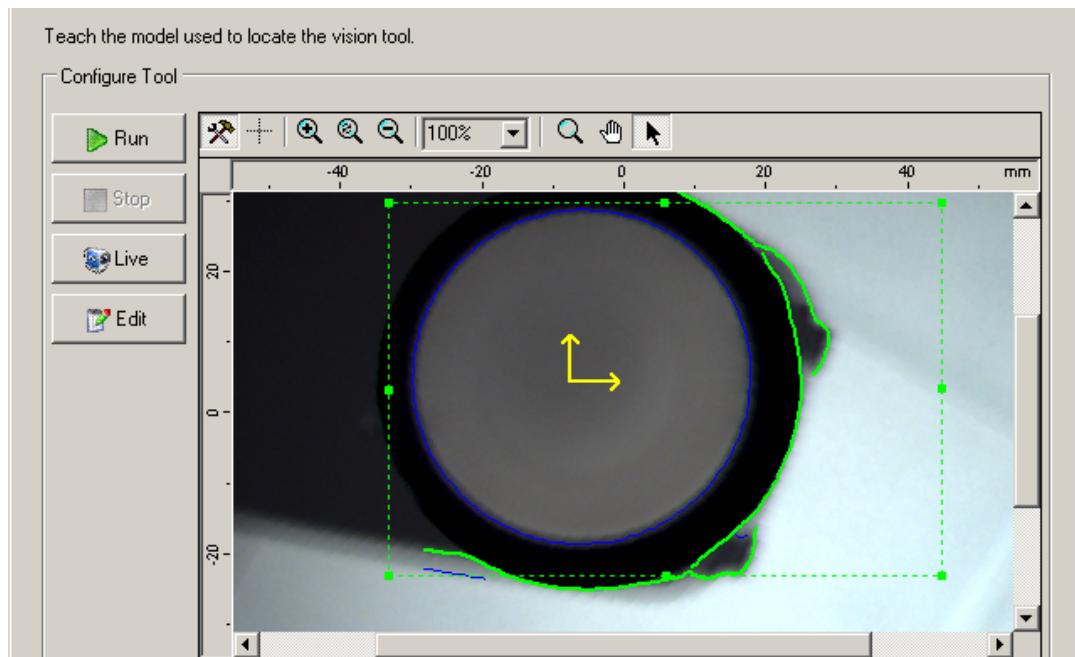
1. Move the robot over the camera.

The robot's end-effector must be in the camera's field of view.

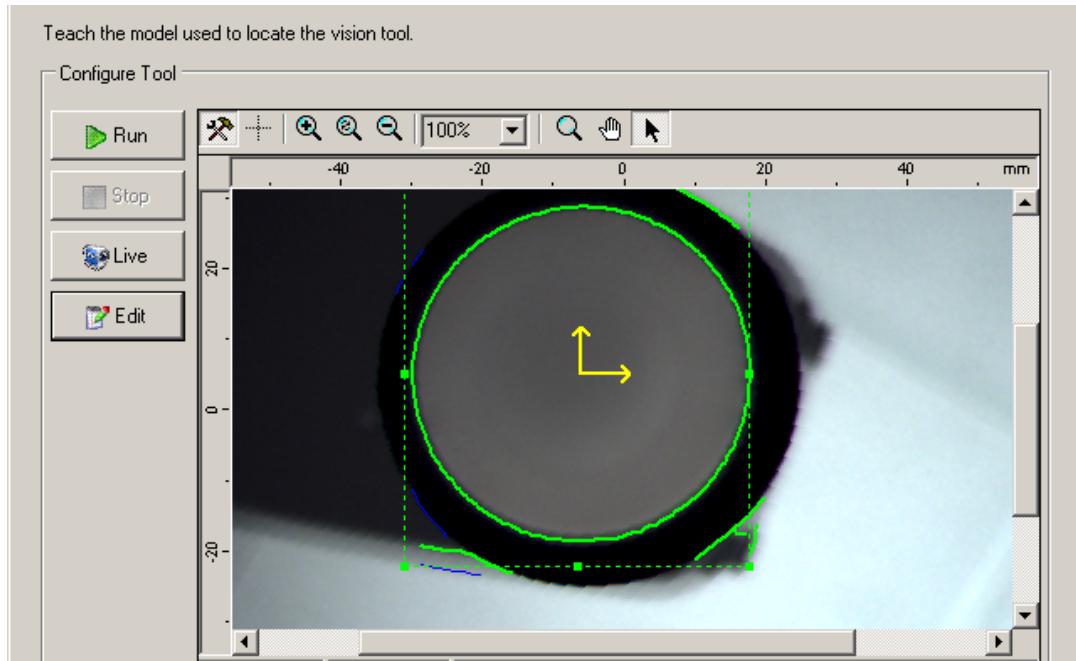
2. Click Here.

- Teach the Vision Tool

The following figures show the model, origin, and bounding box before and after this procedure has been performed.



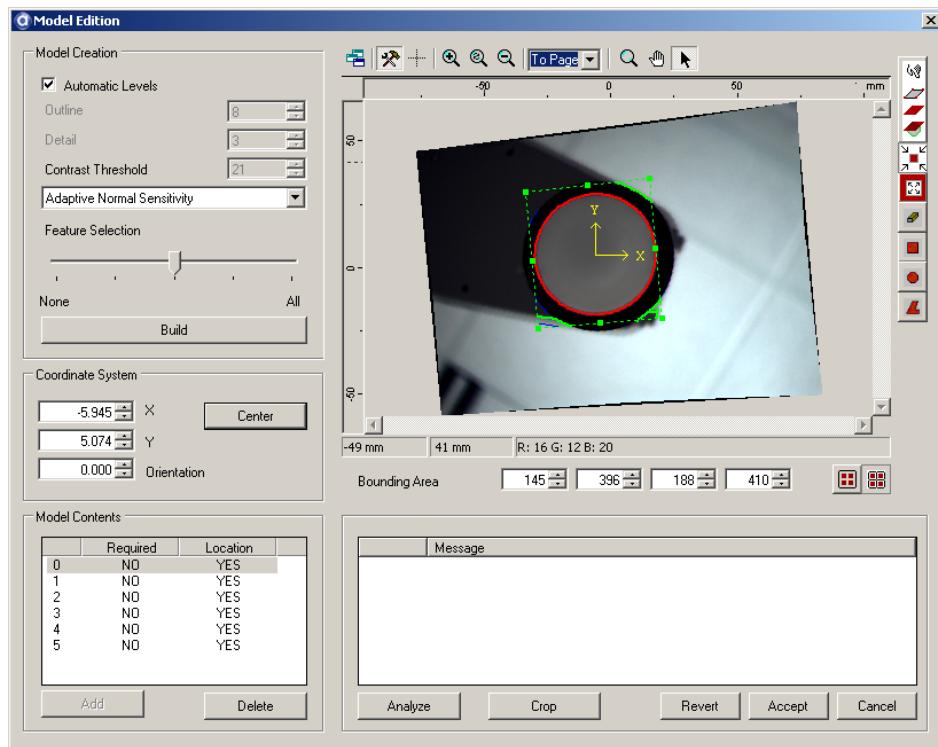
*Vision Model Before Teaching*



*Vision Model After Teaching*

1. Move the robot so that the model is near the center of the camera field of view.
2. Click Run to take an image.
3. Adjust the model bounding box to fully enclose the model.  
Make it close to the perimeter of the model, as shown in the previous figure.
4. Move the origin marker to the center of the model.
  - Drag the origin marker to the center of the model with the mouse  
or
  - Click Edit
    - Click Center to center the marker on the bounding box  
or
    - Click on the outline of the model, then click Center to center the marker on the found part

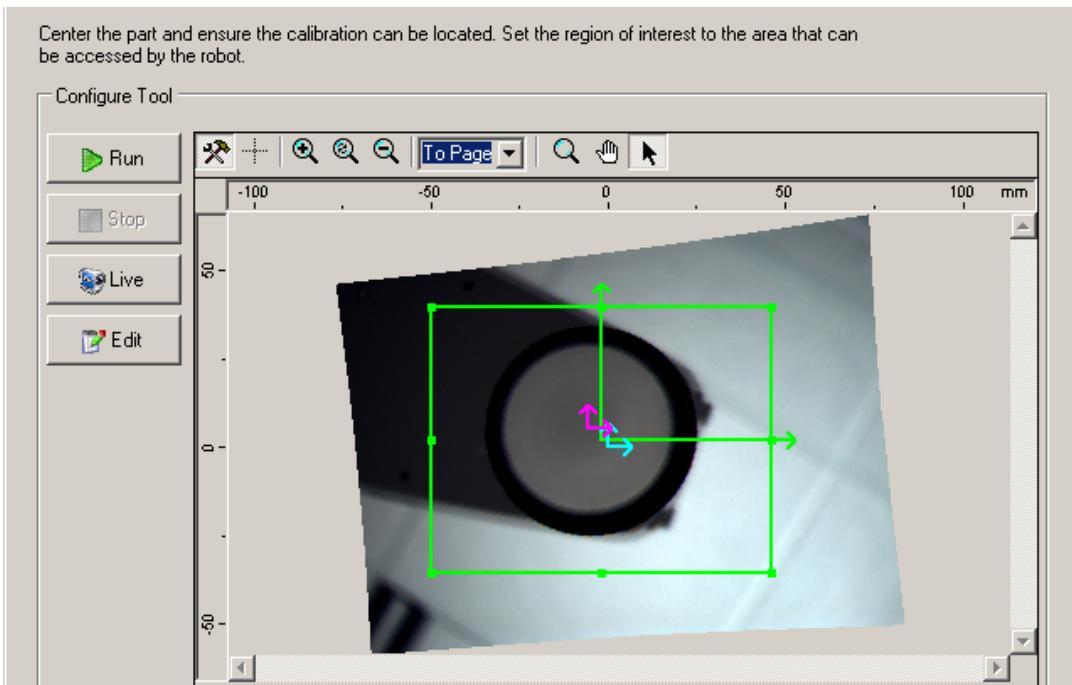
The Edit window is shown in the following figure.



### Edit Window

- Click Accept, then click Next.

- Define the Region of Interest



### *Defining the Region of Interest*

1. Adjust the bounding box to the camera field of view. This must include the calibration target.
2. If no magenta origin marker is displayed, click Run.

If necessary, move the robot so the target can be located, then click Run.

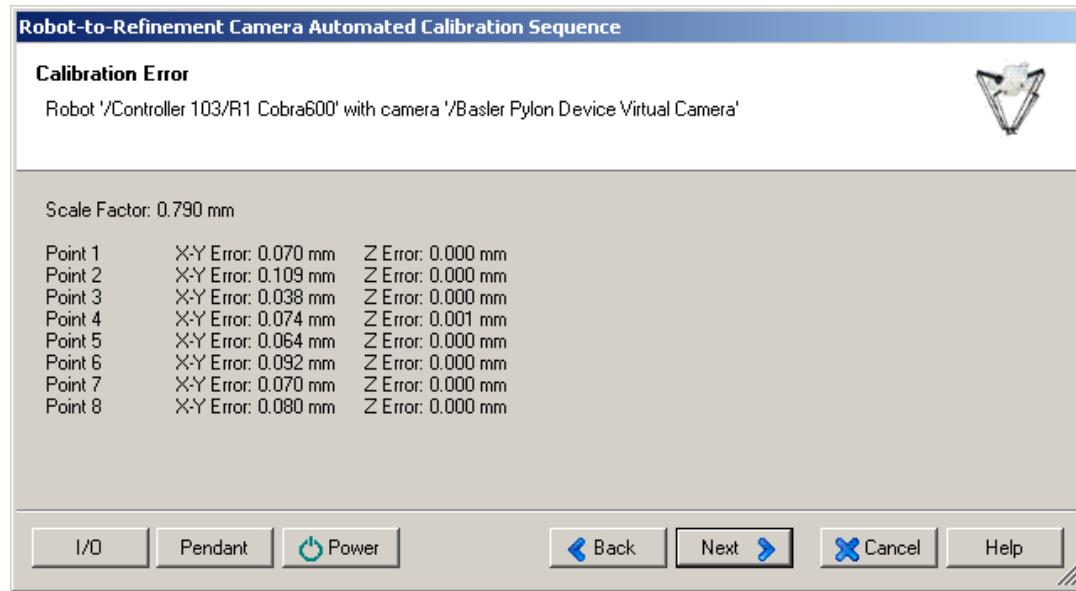
- Start the Calibration

1. Close the pendant window by clicking Cancel.
2. Click Start.

The robot will perform multiple moves, taking images after each move.

When this step is complete, the Task Manager pane displays "Calibration completed successfully".

The final screen displays the calibration error detected during the calibration. An example follows:



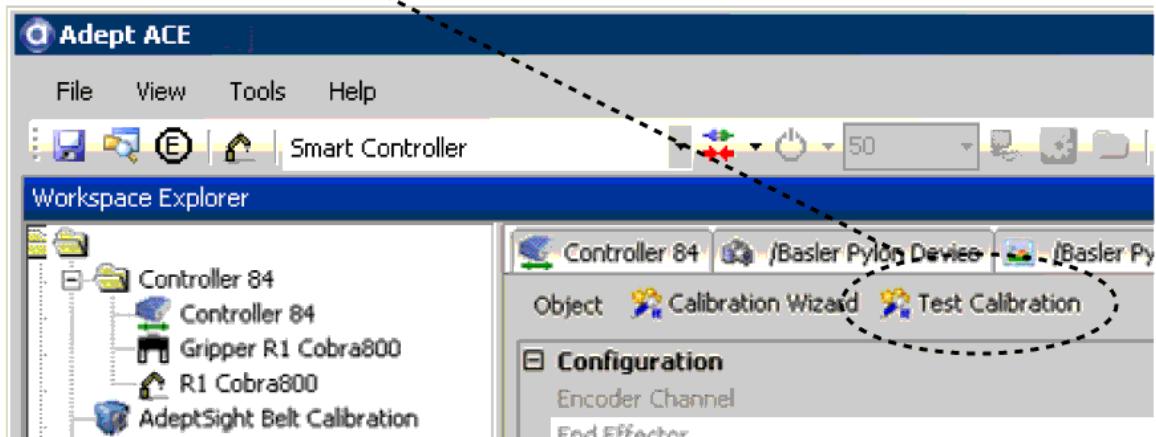
This screen shows the amount of XY and Z difference between the calculated calibration points and the actual points used to generate the calibration. The automated calibration obtains more points than are mathematically required for calibration. The calculated calibration represents an average calibration. When ACE Sight uses the calculated calibration to predict where each actual point should be located, the difference between the actual and calculated point is displayed as the error.

Your Calibration Summary may show a different number of points.

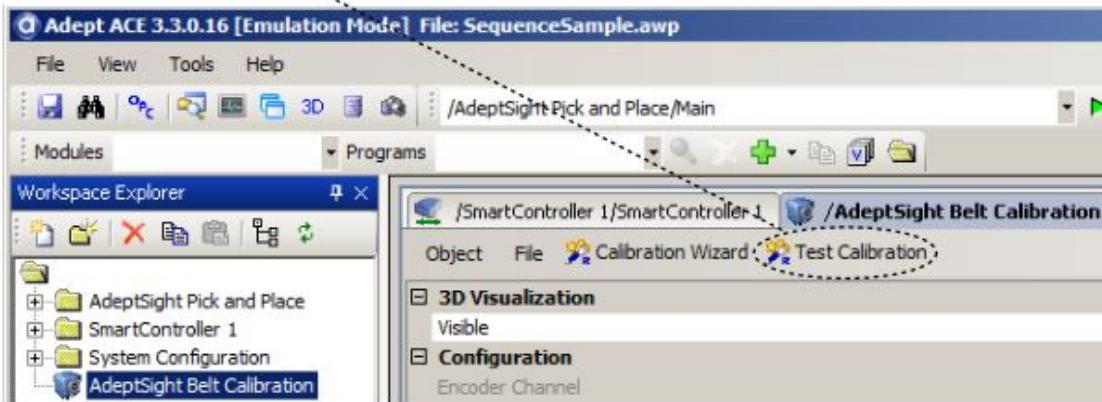
You may need to manually release the gripper at the end of this procedure.

## Test Procedure

Click on Test Calibration after running the Calibration Wizard.



Click on Test Calibration after running the Calibration Wizard



The screens in this procedure are:

- Select End-Effector Tip

This allows you to test a calibration with a different end-effector tip than the one that was used for calibration.

- Move the Robot to the Picture Position

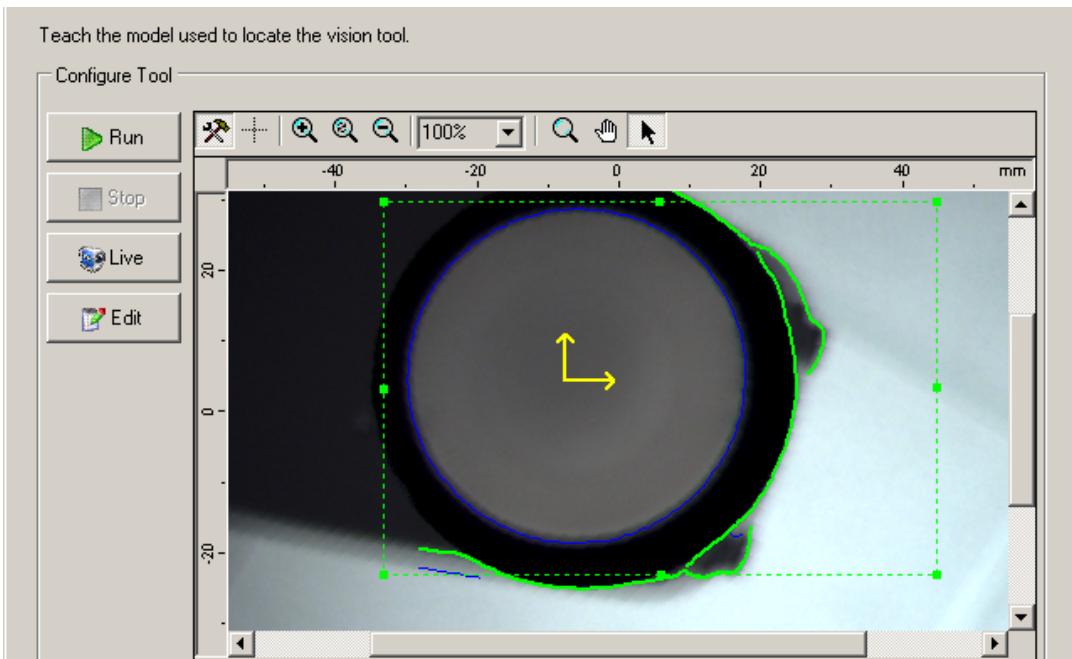
1. Move the robot over the camera.

The robot's end-effector must be in the camera's field of view.

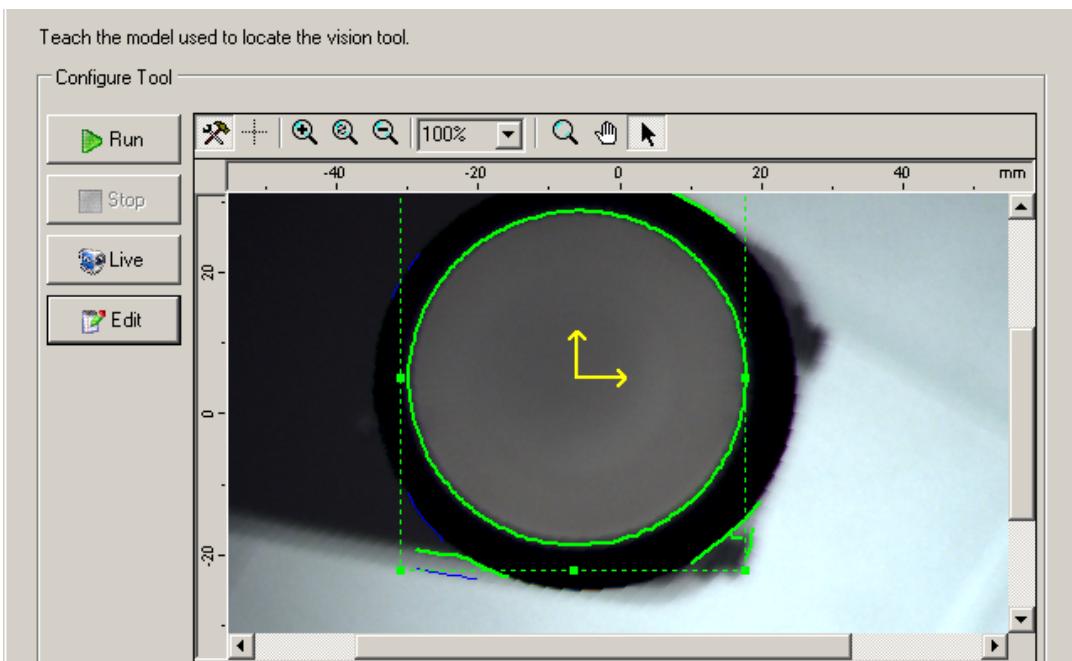
2. Click Here.

- Teach the Vision Tool

The following figures show the model, origin, and bounding box before and after this procedure has been performed.



*Vision Model Before Teaching*



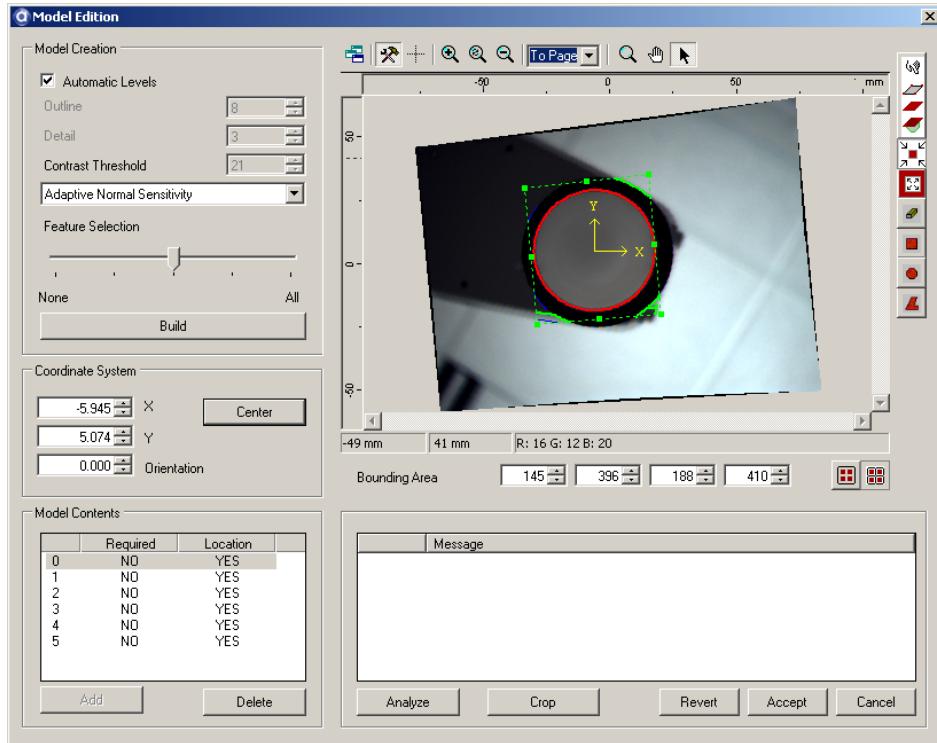
*Vision Model After Teaching*

1. Move the robot so that the model is near the center of the camera field of view.
2. Click Run to take an image.
3. Adjust the model bounding box to fully enclose the model.

Make it close to the perimeter of the model, as shown in the previous figure.

4. Move the origin marker to the center of the model.
  - Drag the origin marker to the center of the model with the mouse  
or
  - Click Edit
    - Click Center to center the marker on the bounding box  
or
    - Click on the outline of the model, then click Center to center the marker on the found part

The Edit window is shown in the following figure.



### *Edit Window*

- Click Accept, then click Next.
- Locate the Target
  1. Adjust the bounding box to the camera field of view. This must include the calibration target.
  2. If no magenta origin marker is displayed, click Run.  
If necessary, move the robot so the target can be located, then click Run.
- Align robot to target 1  
The test can calibrate to multiple targets, if desired.
- Continue testing the calibration
  - I wish to continue testing
    - Move the robot to the location where the picture is taken
    - Locating the calibration target
  - Do not continue testing

## Related Topics

[Calibration Overview](#)

[ACE Sight Belt Calibration Interview Wizard](#)

[ACE Sight Belt Calibration Wizard](#)

[ACE Sight Camera Calibration Interview Wizard](#)

[ACE Sight Camera Calibration Wizard - Automated, no Belt](#)

[ACE Sight Camera Calibration Wizard - Automated, w/ Belt](#)

[ACE Sight Camera Calibration Wizard - Automated, Arm-Mount](#)

[ACE Sight Camera Calibration Wizard - Manual, no Belt](#)

[ACE Sight Camera Calibration Wizard - Manual w/Belt](#)

[ACE Sight Camera Calibration Wizard - Manual, Arm-Mount](#)

[ACE Sight Camera Calibration Wizard - Manual, Upward-looking](#)

### ***ACE Sight Camera Calibration Wizard - Manual, no Belt***

This topic covers calibrating an arm- or fixed-mounted camera to a robot, without a conveyor belt.

## Requirements

- The robot and camera must be correctly connected and functioning.
- The camera itself must be calibrated. See Standalone Camera Calibration on page 320.
- The ACE Sight Camera Calibration Interview Wizard must have completed successfully. See ACE Sight Camera Calibration Interview Wizard on page 642.

## What this Wizard Does

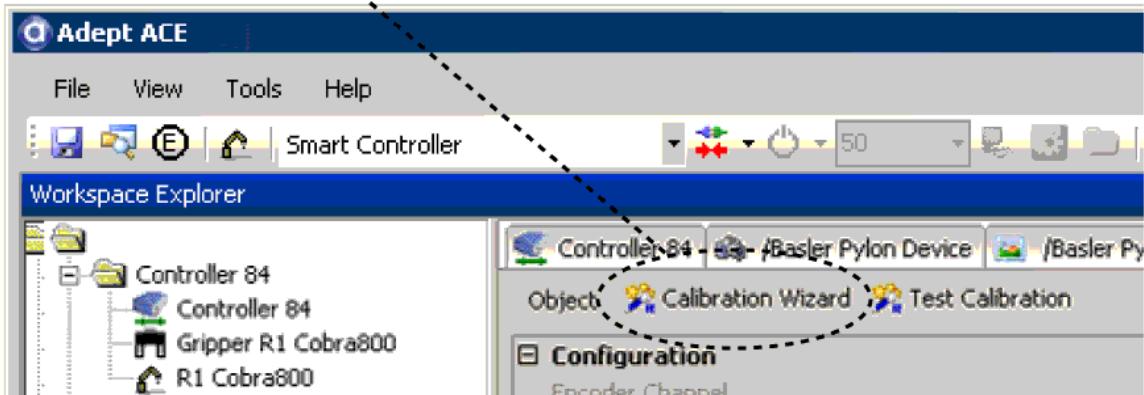
With this wizard you will:

- Teach the picture-taking position, the vision model, and the placement position
- Perform a robot-to-camera calibration

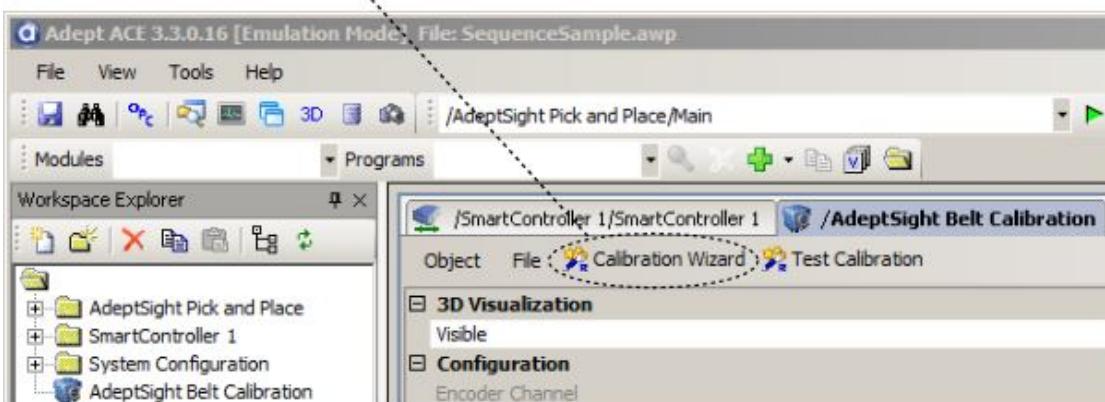
## Procedure

From the Camera Calibration Object window, click Calibration Wizard.

Click on Calibration Wizard after running the Calibration Interview Wizard.



Click on Calibration Wizard after running the Calibration Interview Wizard



The screen titles will all say "Robot-to-Fixed or Arm-Camera Manual Calibration Sequence".

The screens in this procedure are:

- Select the End-Effector
- Move the Robot to the Picture Position
  1. Move the robot out of the way of the camera.

The robot should not be in the camera's field of view.

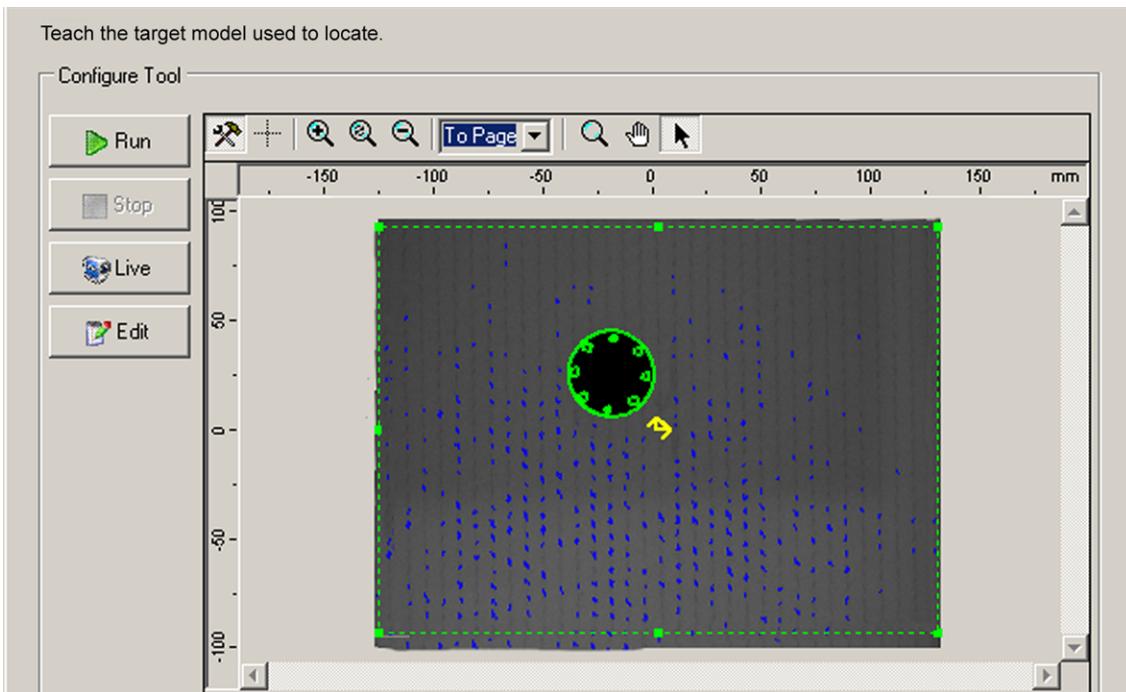
2. Click Here.

- Teach the Vision Tool

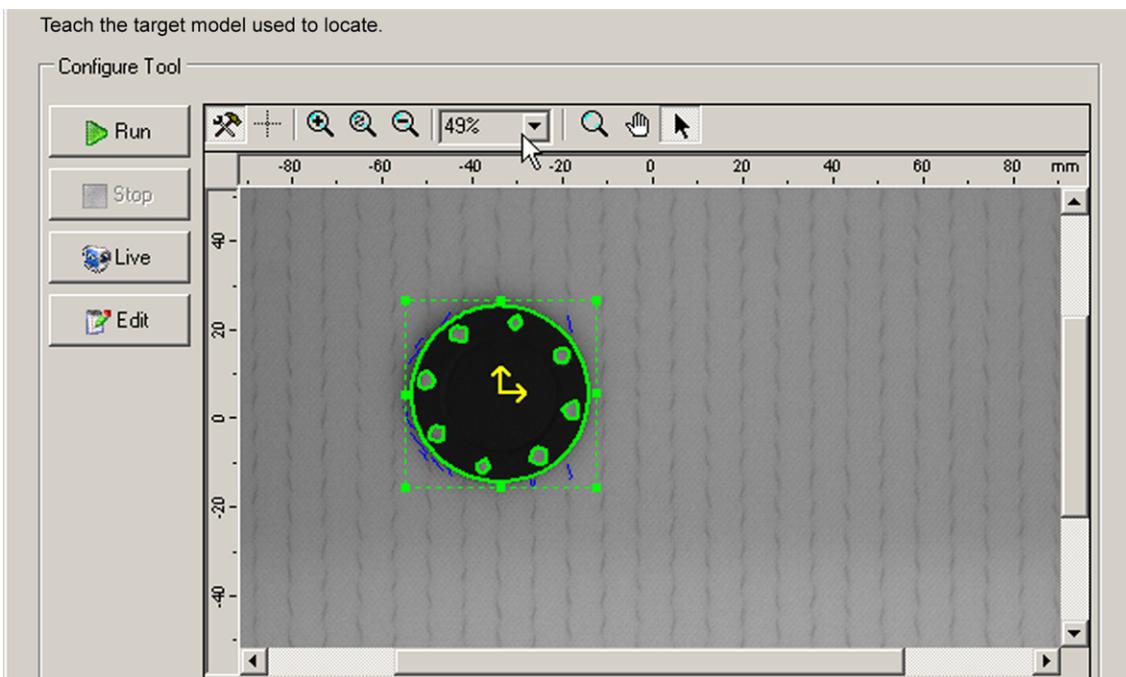
The following figures show the model, origin, and bounding box before and after this procedure has been performed.

## ACE Sight Camera Calibration

---



*Vision Model Before Teaching*



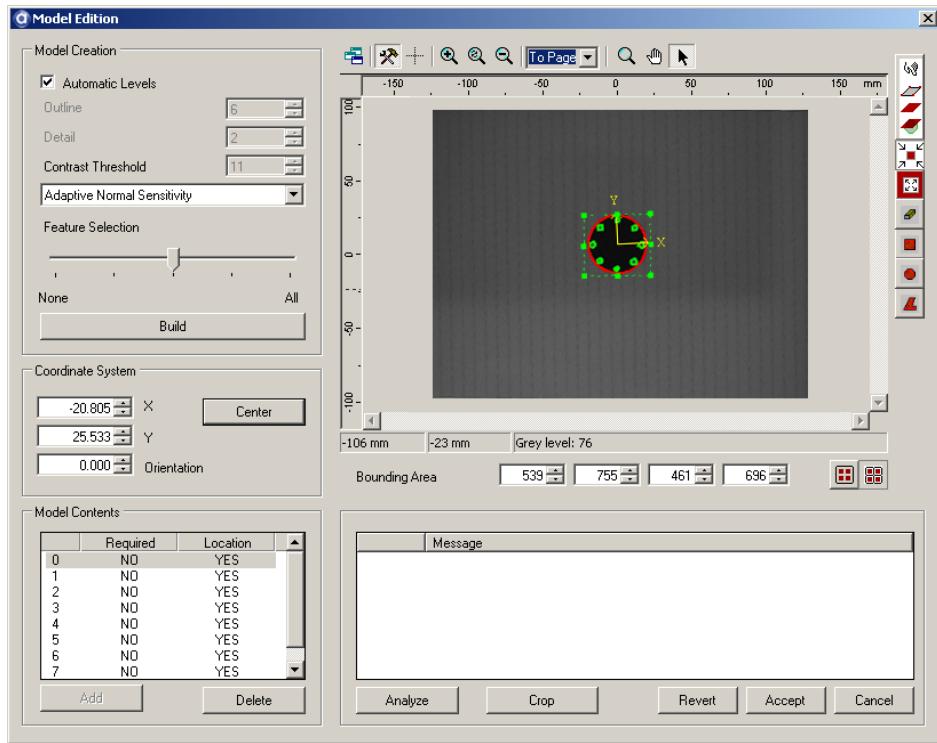
*Vision Model After Teaching*

1. Place the target near the center of the work area.
2. Click Run to take an image.
3. Adjust the model bounding box to fully enclose the model.

Make it close to the perimeter of the model, as shown in the previous figure.

4. Move the origin marker to the center of the model.
  - Drag the origin marker to the center of the model with the mouse  
or
  - Click Edit, to open the Edit window (shown in the following figure).
    - Click Center to center the marker on the bounding box  
or
    - Click on the outline of the model, then click Center to center the marker on the found part

## ACE Sight Camera Calibration



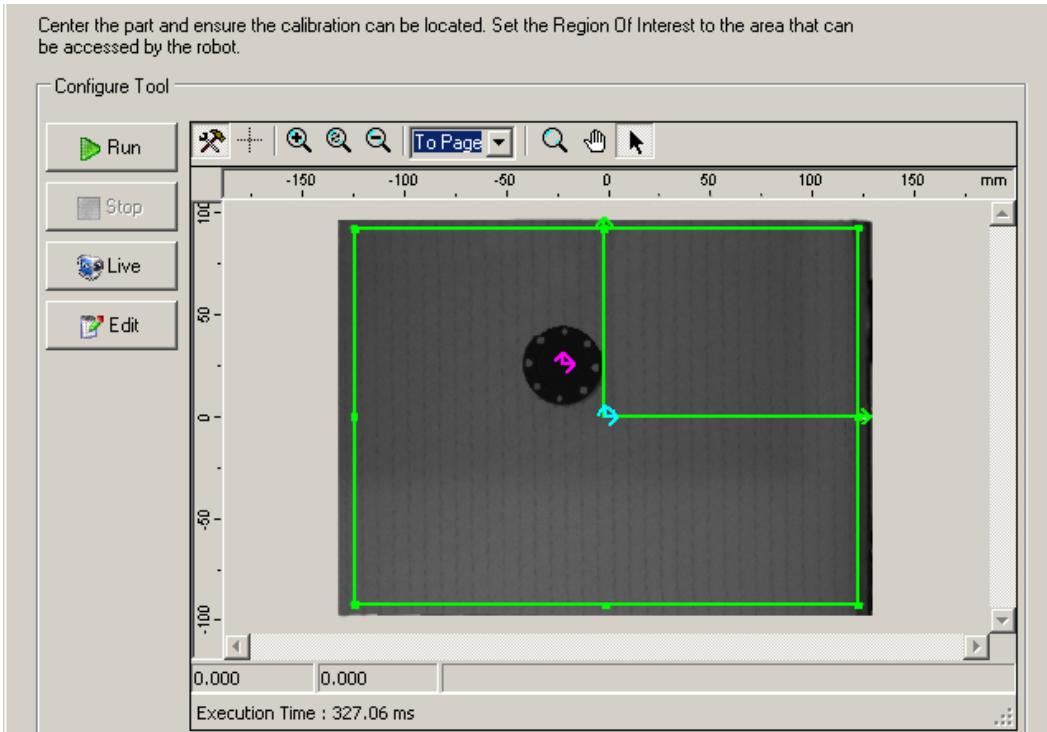
*Edit Window*

You will place targets at a minimum of four points. These should be close to the four corners of the workspace for best accuracy.

The following steps are repeated for  $n = 1$  through 4:

- Define Calibration Point n
  - 1. Place the target.
  - 2. Align the robot to the target.
  - 3. Click Here.
- Move the Robot to the Picture Position
  - 1. Move the robot out of the way of the camera.  
The robot should not be in the camera's field of view.
  - 2. Click Here.

- Take a Picture of the Calibration Target



*Located Calibration Target*

1. Adjust the bounding box to the work area. This must include the calibration target, as shown in the previous figure.
2. If no magenta origin marker is displayed, click Run.  
If necessary, move the target so it can be located, then click Run.
3. Click Next.

If n is less than 4, this goes back to Define Calibration Point n, for the next value of n.

- Minimum number of points

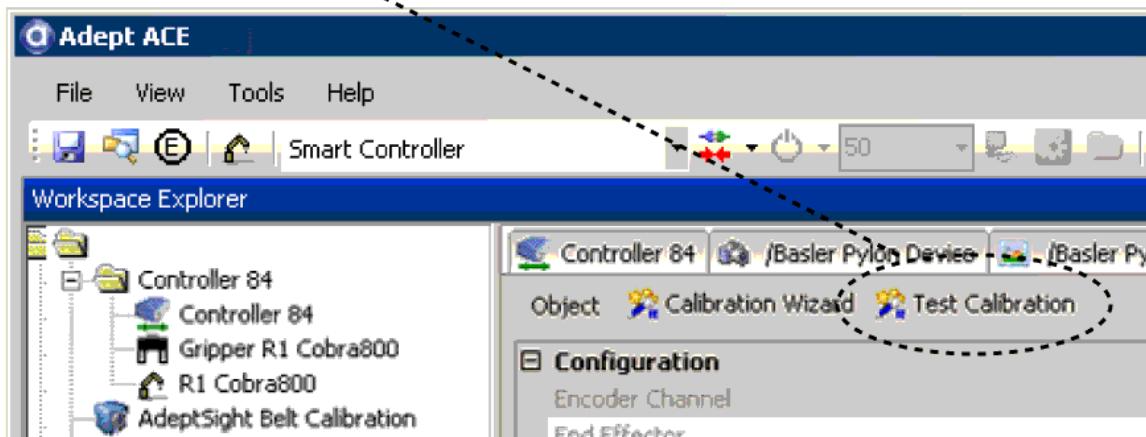
You have collected the minimum number of points needed to perform a calibration. You can choose to calibrate now, or continue collecting more points, possibly improving the accuracy of the calibration.

- Continue adding more points
  1. Define Calibration Point.
  2. Move the robot to the location where the picture is taken.

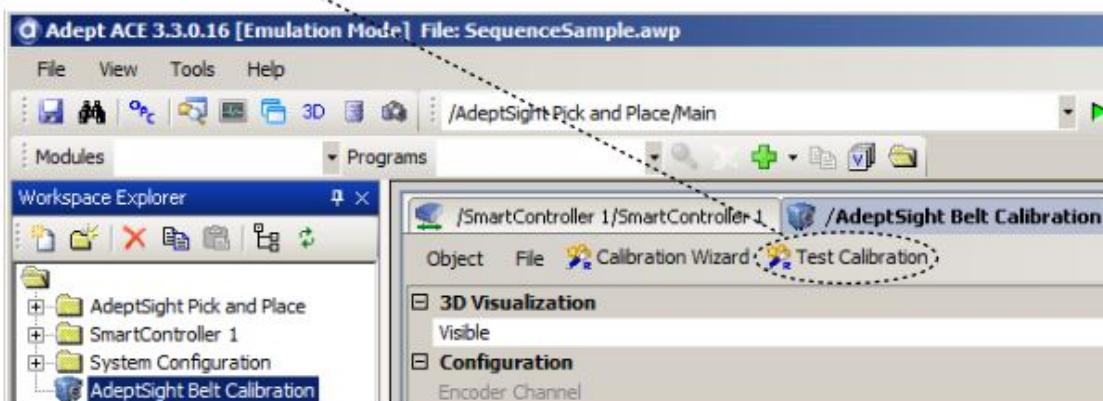
3. Continue adding more points.
  - Calibrate with current set of points

### Test Procedure

Click on Test Calibration after running the Calibration Wizard.



Click on Test Calibration after running the Calibration Wizard



The screens in this procedure are:

- Select End-Effector Tip

This allows you to test a calibration with a different end-effector tip than the one that was used for calibration.

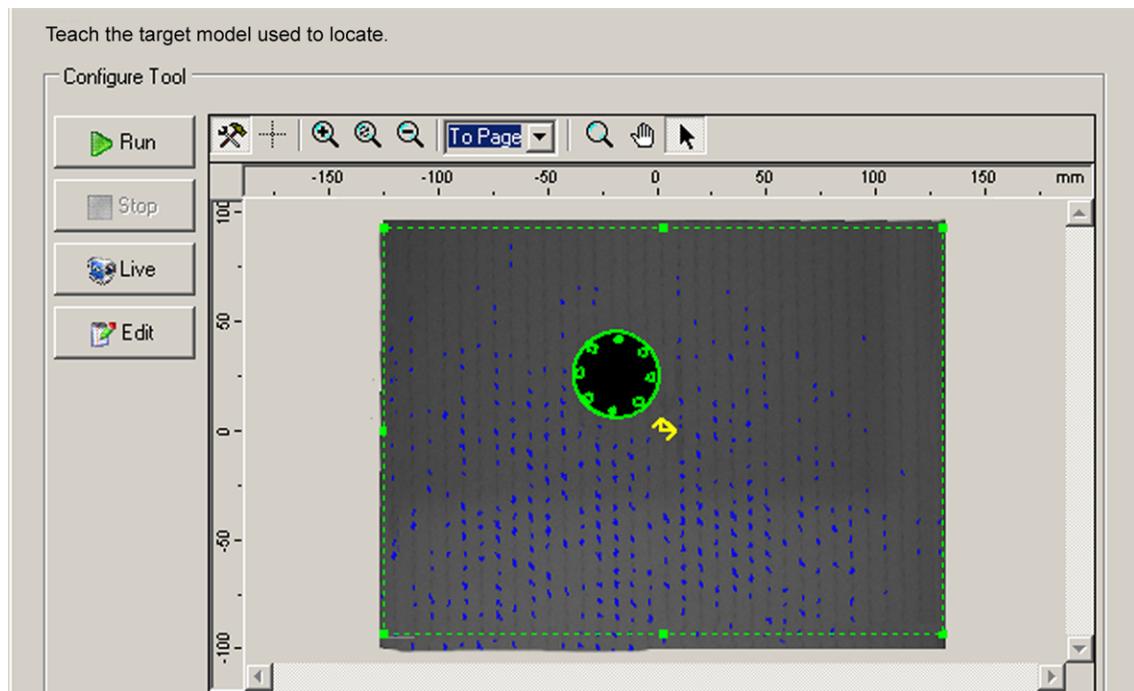
- Move the Robot to the Picture Position
  1. Move the robot out of the way of the camera.

The robot should not be in the camera's field of view.

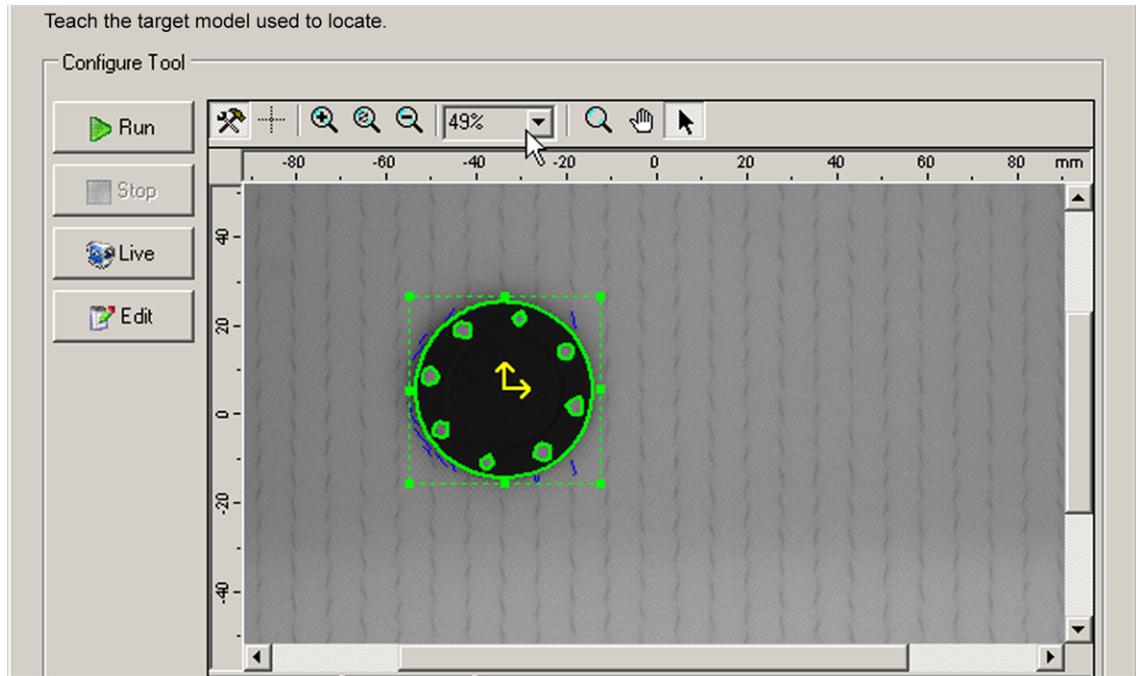
2. Click Here.

- Teach the Vision Tool

The following figures show the model, origin, and bounding box before and after this procedure has been performed.

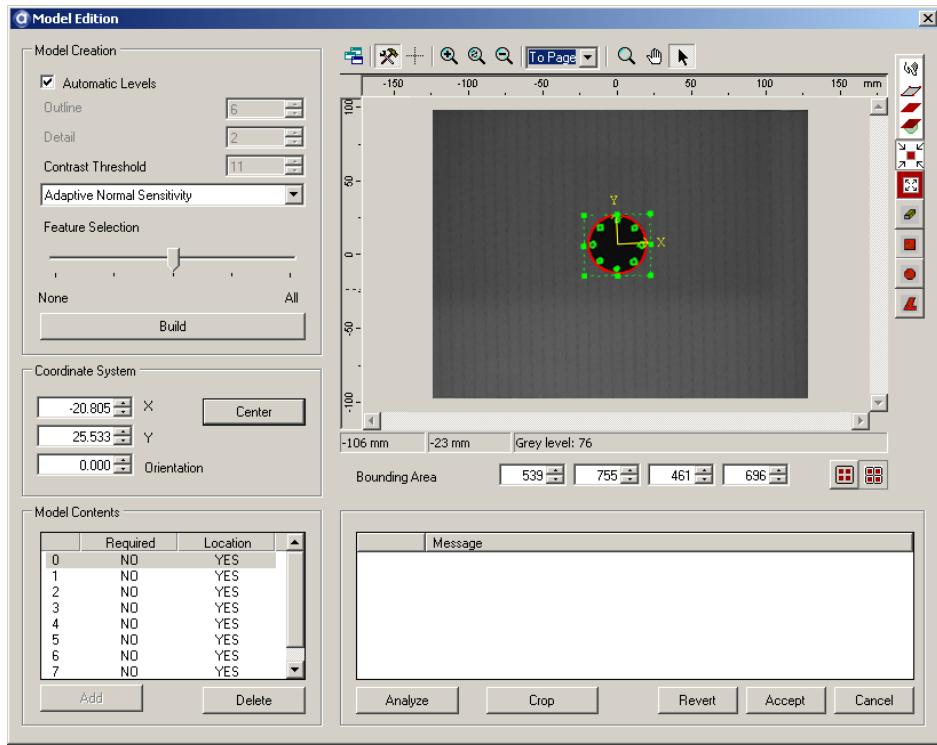


*Vision Model Before Teaching*



*Vision Model After Teaching*

1. Place the target near the center of the work area.
2. Click Run to take an image.
3. Adjust the model bounding box to fully enclose the model.  
Make it close to the perimeter of the model, as shown in the previous figure.
4. Move the origin marker to the center of the model.
  - Drag the origin marker to the center of the model with the mouse  
or
  - Click Edit, to open the Edit window (shown in the following figure).
    - Click Center to center the marker on the bounding box  
or
    - Click on the outline of the model, then click Center to center the marker on the found part



*Edit Window*

- Locate the Target

1. Adjust the bounding box to the work area. This must include the calibration target.

**NOTE:** The calibration target should be correctly adjusted from the calibration procedure.

2. If no magenta origin marker is displayed, click Run.

If necessary, move the target so it can be located, then click Run.

3. Click Next. The robot should move to the target.

- Continue testing the calibration

- I wish to continue testing

- Move the robot to the location where the picture is taken
- Locating the calibration target

- Continue testing the calibration
- Do not continue testing

## Related Topics

[Calibration Overview](#)

[ACE Sight Belt Calibration Interview Wizard](#)

[ACE Sight Belt Calibration Wizard](#)

[ACE Sight Camera Calibration Interview Wizard](#)

[ACE Sight Camera Calibration Wizard - Automated, no Belt](#)

[ACE Sight Camera Calibration Wizard - Automated, w/ Belt](#)

[ACE Sight Camera Calibration Wizard - Automated, Arm-Mount](#)

[ACE Sight Camera Calibration Wizard - Automated Upward-Looking](#)

[ACE Sight Camera Calibration Wizard - Manual w/Belt](#)

[ACE Sight Camera Calibration Wizard - Manual, Arm-Mount](#)

[ACE Sight Camera Calibration Wizard - Manual, Upward-looking](#)

### ***ACE Sight Camera Calibration Wizard - Manual with Belt***

This topic covers calibrating an arm- or fixed-mounted camera to a robot, with a conveyor belt.

## Requirements

- The robot, belt, and camera must be correctly connected and functioning.
- The camera itself must be calibrated. See Standalone Camera Calibration on page 320.
- The ACE Sight Belt Calibration Wizard must have completed successfully.
- The ACE Sight Camera Calibration Interview Wizard must have completed successfully.

## What this Wizard Does

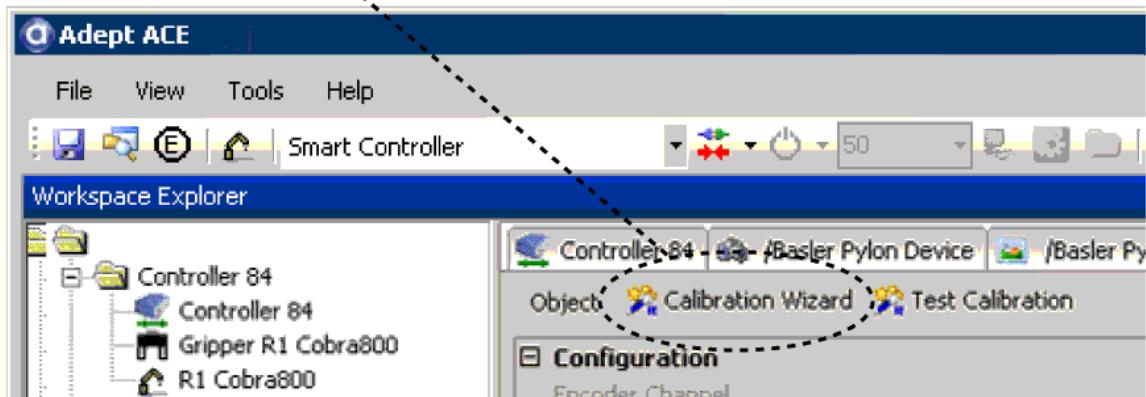
With this wizard you will:

- Teach the picture-taking position, the vision model, and the placement position
- Perform a robot-to-camera calibration

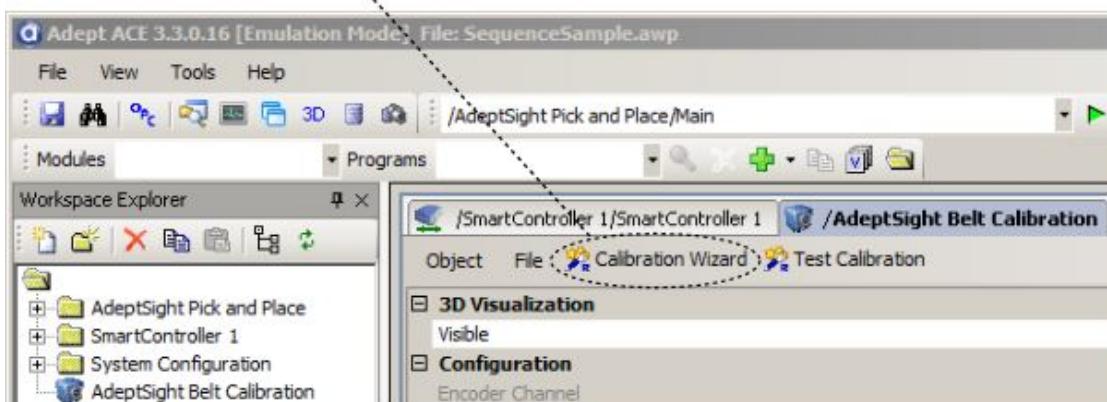
## Procedure

From the ACE Sight Camera Calibration Object window, click Calibration Wizard.

Click on Calibration Wizard after running the Calibration Interview Wizard.



Click on Calibration Wizard after running the Calibration Interview Wizard



The screen titles will all display "Robot-to-Belt Camera Calibration Sequence".

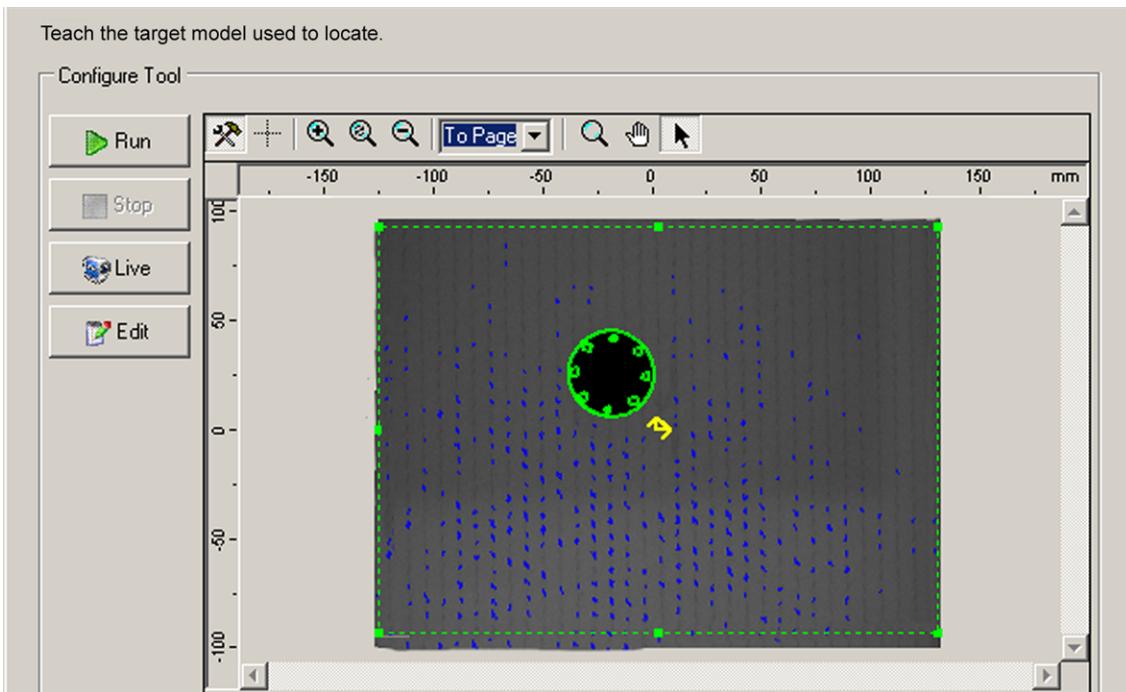
The screens in this procedure are:

- Select the End-Effector
- Teach the Vision Tool

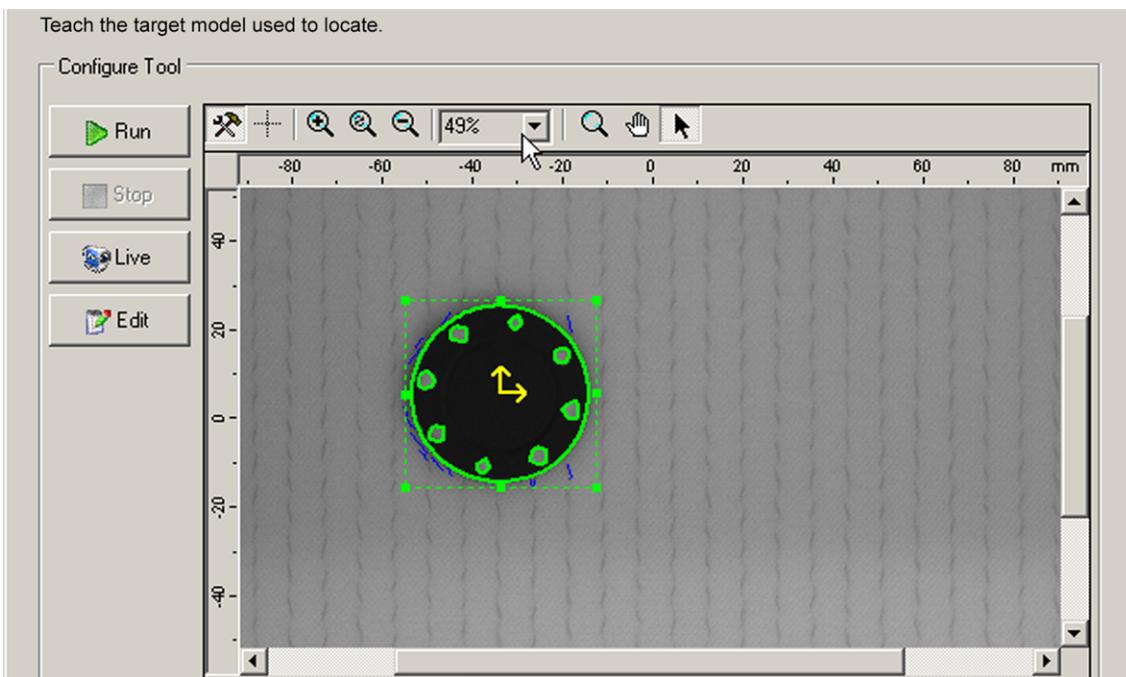
The following figures show the model, origin, and bounding box before and after this procedure has been performed.

## ACE Sight Camera Calibration

---



*Vision Model Before Teaching*



*Vision Model After Teaching*

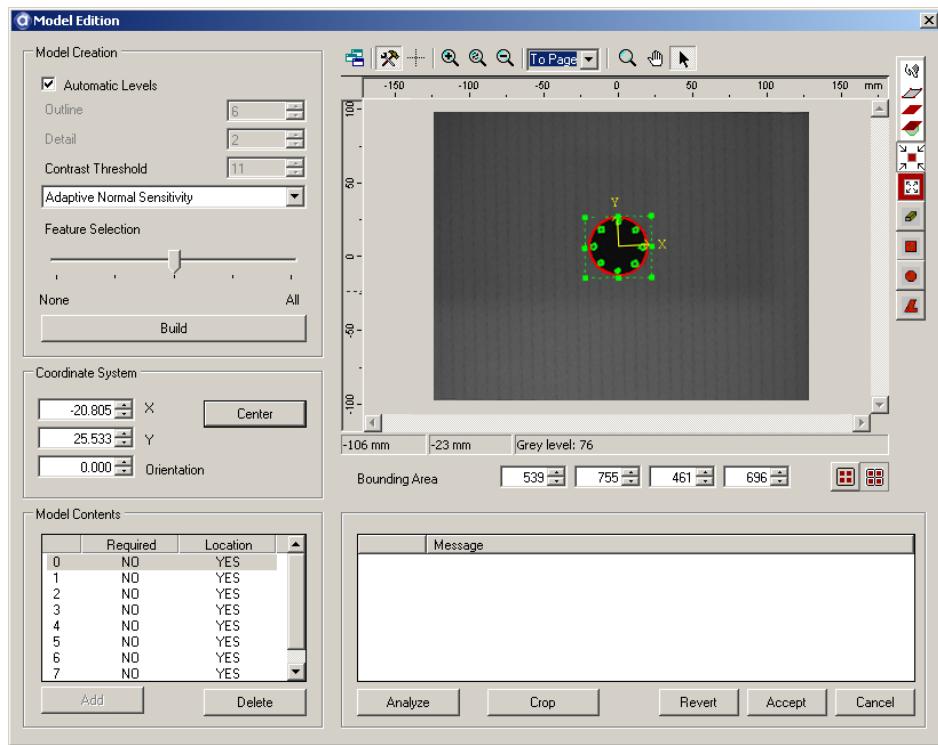
1. Place the target near the center of the work area.
2. Click Run to take an image.
3. Adjust the model bounding box to fully enclose the model.

Make it close to the perimeter of the model, as shown in the previous figure.

4. Move the origin marker to the center of the model.
  - Drag the origin marker to the center of the model with the mouse  
or
  - Click Edit, to open the Edit window (shown in the following figure).
    - Click Center to center the marker on the bounding box  
or
    - Click on the outline of the model, then click Center to center the marker on the found part

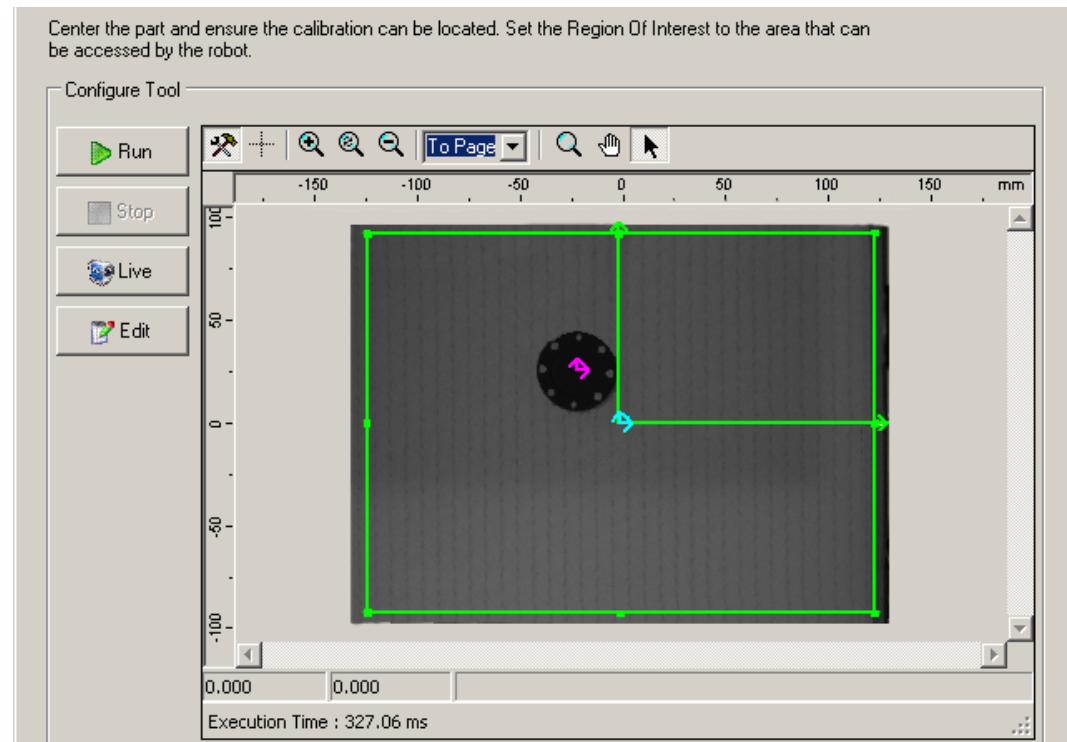
## ACE Sight Camera Calibration

---



*Edit Window*

- Take a Picture of the Calibration Target



*Located Calibration Target*

1. Place the target near the upstream limit, within the work area.
2. Adjust the bounding box to the work area. This must include the calibration target.
3. If the target isn't marked with a magenta origin, click Run.

If necessary, move the target so it can be located, then click Run.

- Advance the Belt

**NOTE:** The belt will be advanced four times in this procedure. Advance the belt so that the target moves less than a quarter of the total distance from point 1 to the downstream limit.

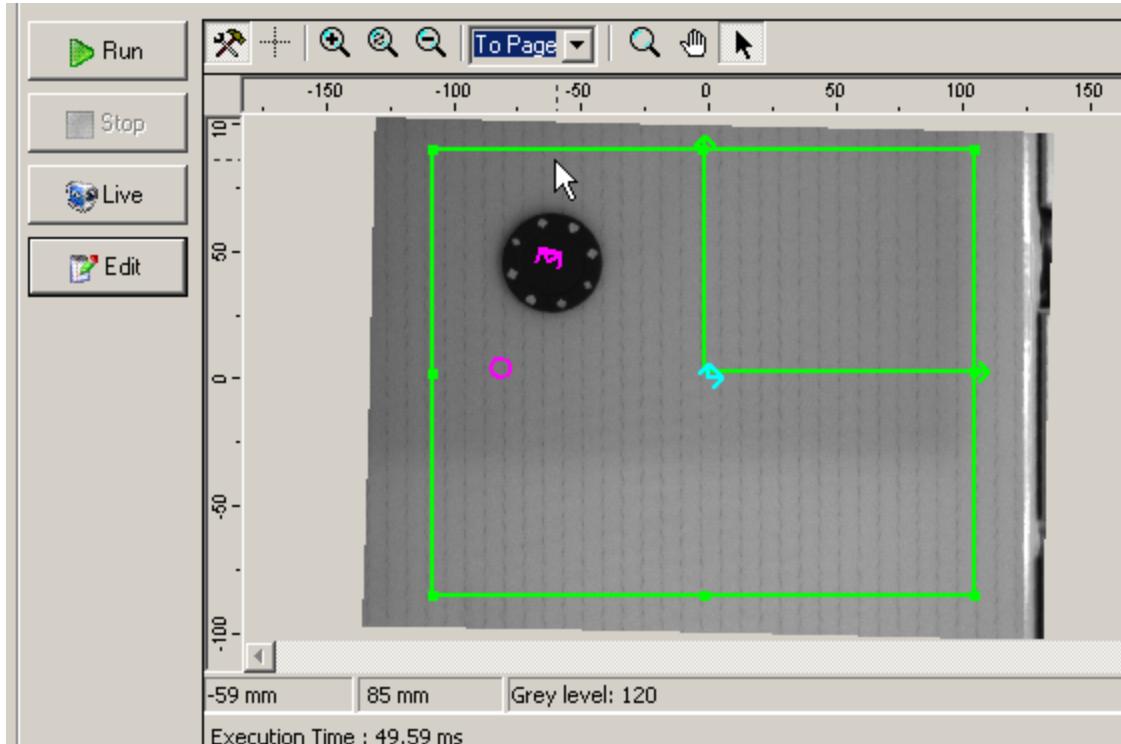
Do not touch the target.

- Teach the Robot Position at Object 1

1. Align the robot with the target.
2. Click Here.

The following steps are repeated for  $n = 2$  through 4.

A magenta circle will remain for each target instance that was found, so, at the end, you will see four circles.



#### *Example Magenta Circle*

- Take a Picture of the Calibration Target
  1. Move the robot to the location where the picture is taken.
  2. Click Run.

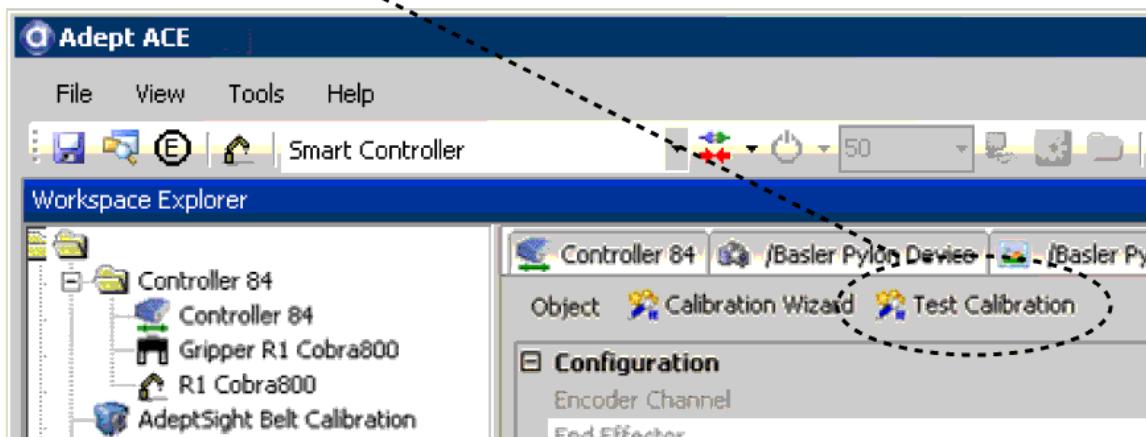
The located target should be identified with a magenta origin marker. If this isn't displayed after clicking Run, this calibration will fail. This will occur if the belt advances the target past the downstream limit.

- Advance the Belt  
Do not touch the target.
- Teach the Robot Position at Object n

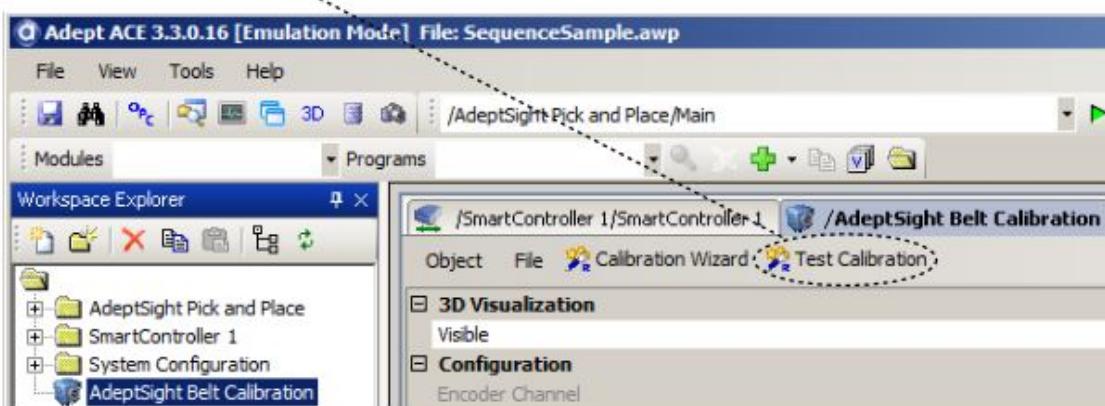
1. Align the robot with the target.
  2. Click Here.
    - Minimum Number of Required Calibration Points Attained
- Choose either:
- Continue adding more points
  - Calibrate with current set of points

### Test Procedure

Click on Test Calibration after running the Calibration Wizard.



Click on Test Calibration after running the Calibration Wizard



The screens in this procedure are:

- Select End-Effector Tip

This allows you to test a calibration with a different end-effector tip than the one that was used for calibration.

- Move the Robot to the Picture Position

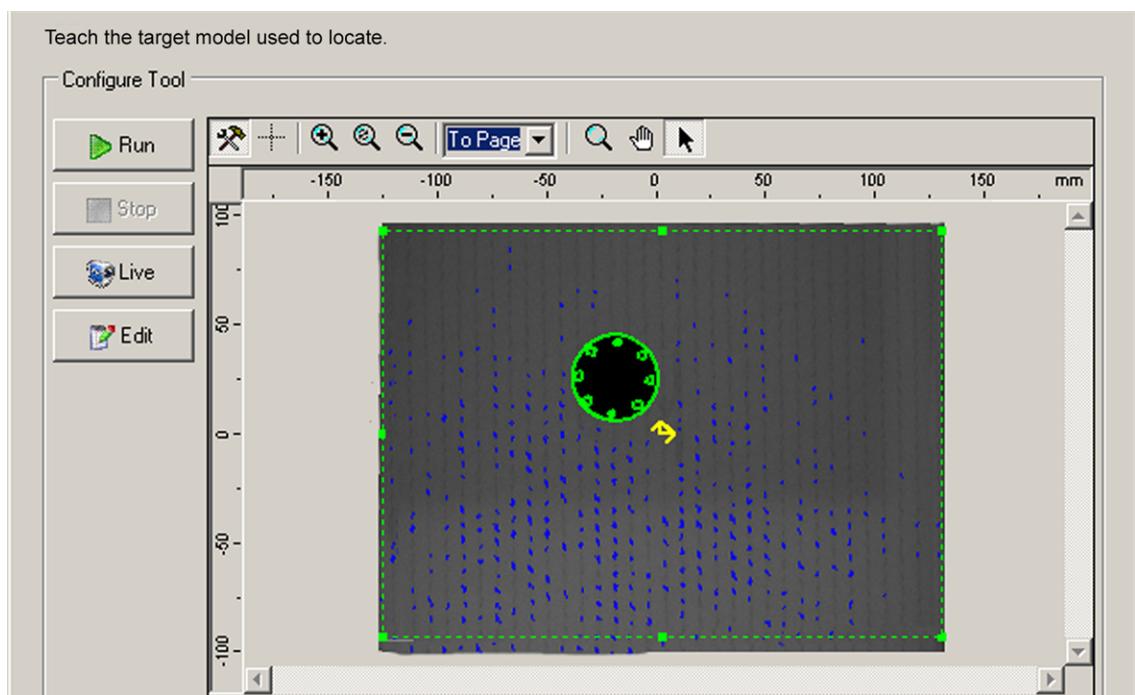
1. Move the robot out of the way of the camera.

The robot should not be in the camera's field of view.

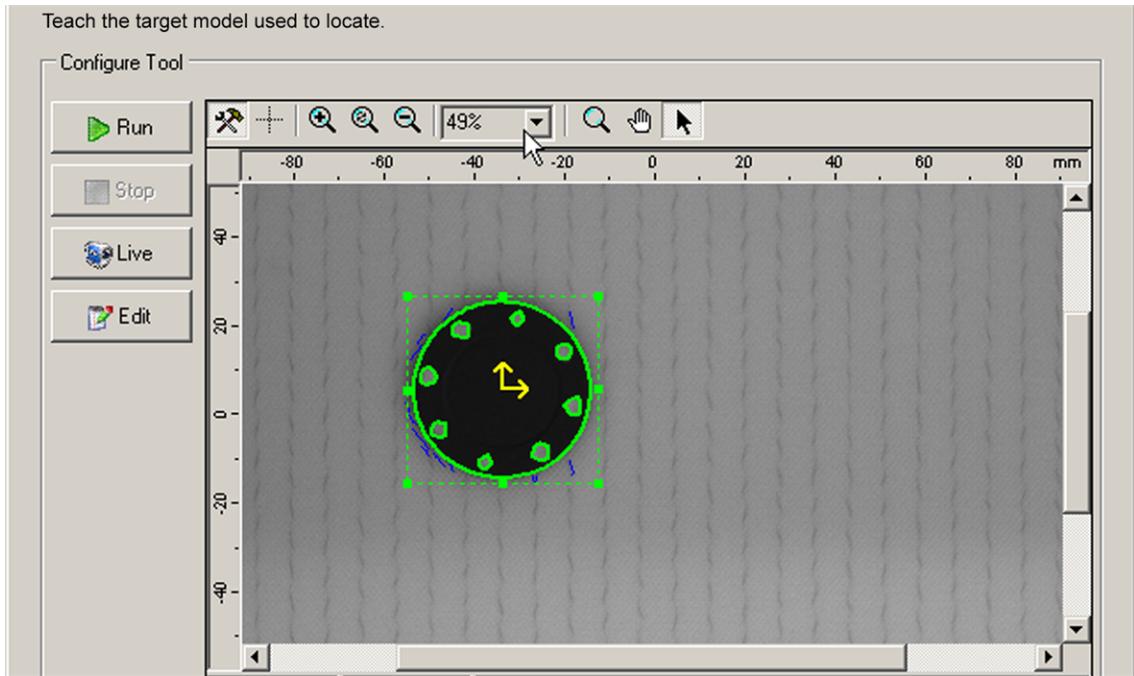
2. Click Here.

- Teach the Vision Tool

The following figures show the model, origin, and bounding box before and after this procedure has been performed.

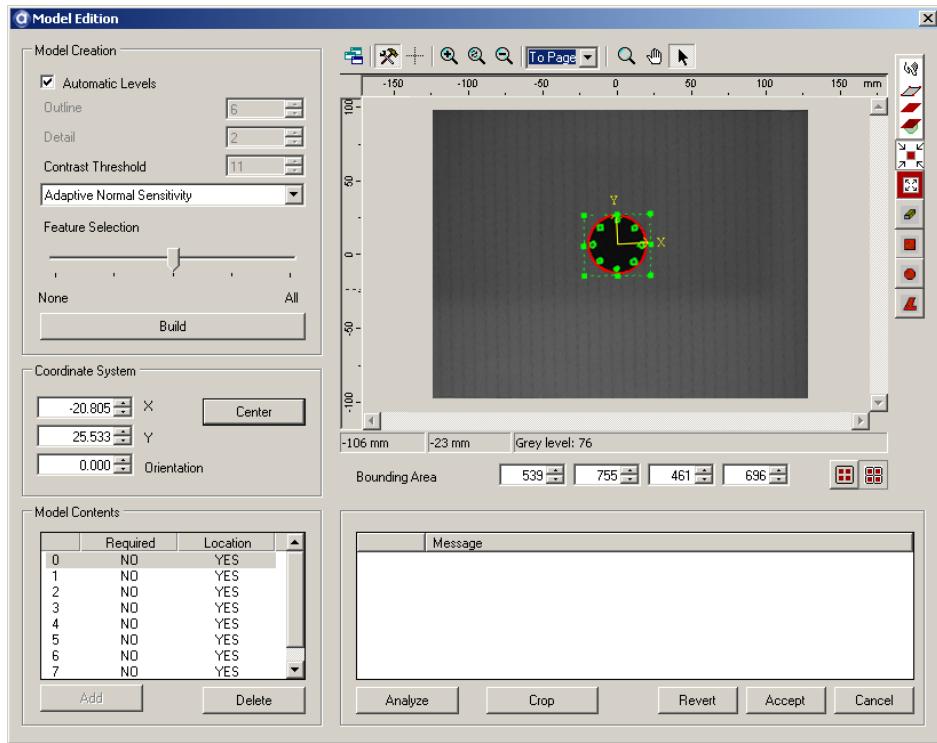


*Vision Model Before Teaching*



*Vision Model After Teaching*

1. Place the target near the center of the work area.
2. Click Run to take an image.
3. Adjust the model bounding box to fully enclose the model.  
Make it close to the perimeter of the model, as shown in the previous figure.
4. Move the origin marker to the center of the model.
  - Drag the origin marker to the center of the model with the mouse  
or
  - Click Edit, to open the Edit window (shown in the following figure).
    - Click Center to center the marker on the bounding box  
or
    - Click on the outline of the model, then click Center to center the marker on the found part



### *Edit Window*

- Locating the Target

- Close the Pendant window, if open (click Cancel).
- Click Run.

The target should be marked with a magenta origin marker. If not, move the target, and click Run again.

- Advance the Belt

The robot should go to the target when you click Next.

- Continue testing the calibration

- I wish to continue testing
  - Move the robot to the location where the picture is taken
  - Locating the calibration target
  - Advance the belt

- Continue testing the calibration
- Do not continue testing

This can be repeated, from "Locating the calibration target", as many times as desired.

## **Related Topics**

[Calibration Overview](#)

[ACE Sight Belt Calibration Interview Wizard](#)

[ACE Sight Belt Calibration Wizard](#)

[ACE Sight Camera Calibration Interview Wizard](#)

[ACE Sight Camera Calibration Wizard - Automated, no Belt](#)

[ACE Sight Camera Calibration Wizard - Automated, w/ Belt](#)

[ACE Sight Camera Calibration Wizard - Automated, Arm-Mount](#)

[ACE Sight Camera Calibration Wizard - Automated Upward-Looking](#)

[ACE Sight Camera Calibration Wizard - Manual, no Belt](#)

[ACE Sight Camera Calibration Wizard - Manual, Arm-Mount](#)

[ACE Sight Camera Calibration Wizard - Manual, Upward-looking](#)

## ***ACE Sight Camera Calibration Wizard - Manual, Arm-Mount***

### **Requirements**

- The robot and camera must be correctly connected and functioning.
- The camera itself must be calibrated. See Standalone Camera Calibration on page 320.
- The ACE Sight Camera Calibration Interview Wizard must have completed successfully.

### **What this Wizard Does**

With this wizard you will:

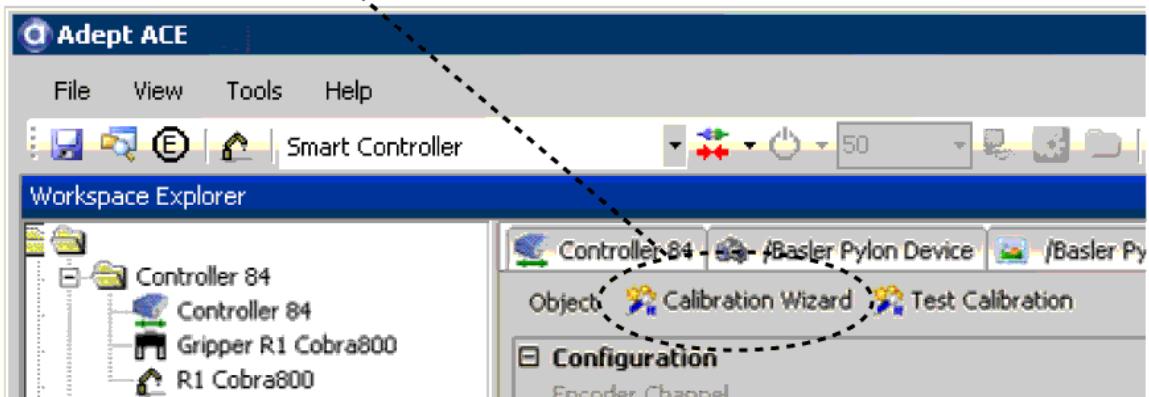
- Teach the picture-taking position, the vision model, and the placement position
- Perform a robot-to-camera calibration

### **Procedure**

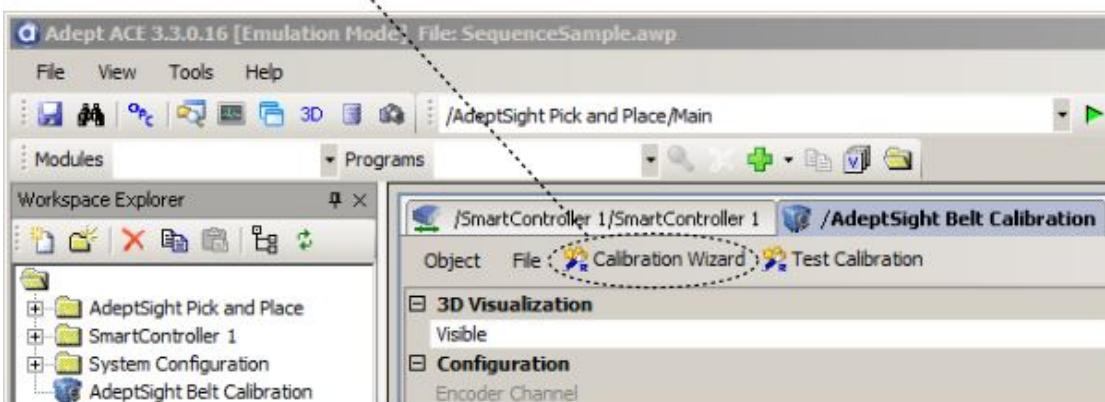
From the Camera Calibration Object window, click Calibration Wizard.

---

Click on Calibration Wizard after running the Calibration Interview Wizard.



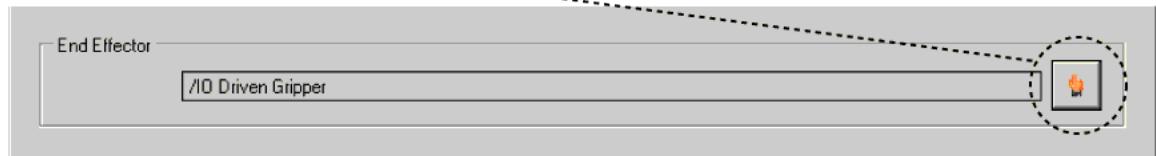
Click on Calibration Wizard after running the Calibration Interview Wizard



The screen titles will all say "Robot-to-Fixed or Arm-Camera Manual Calibration Sequence".  
The screens in this procedure are:

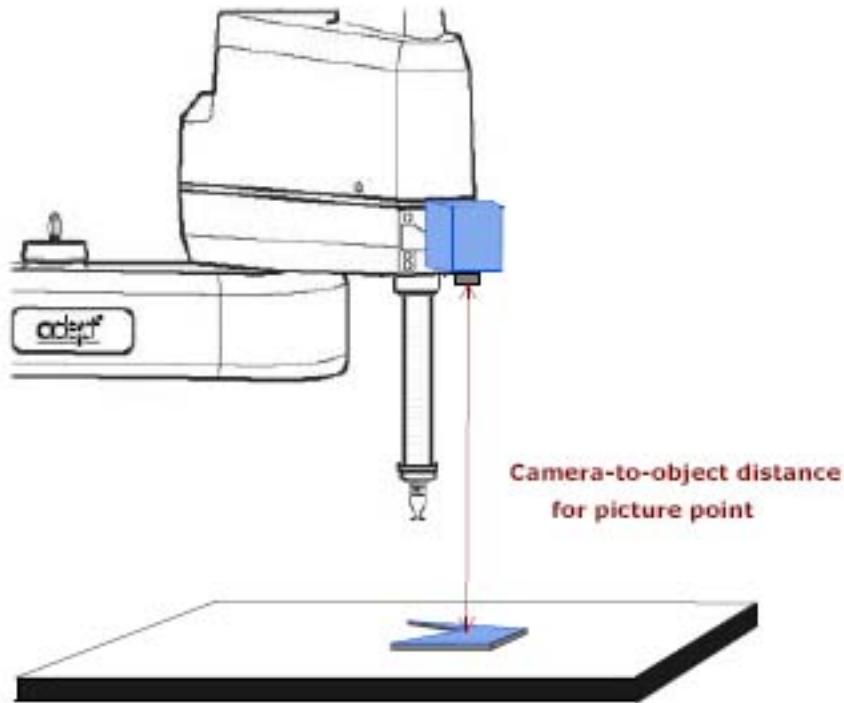
- Select the End-Effector

1. Click on the browse icon to display a list of available end-effectors.



2. Select the end-effector that you wish to use.

3. Click OK.
  4. Click Next.
- Move the Robot to the Picture Position

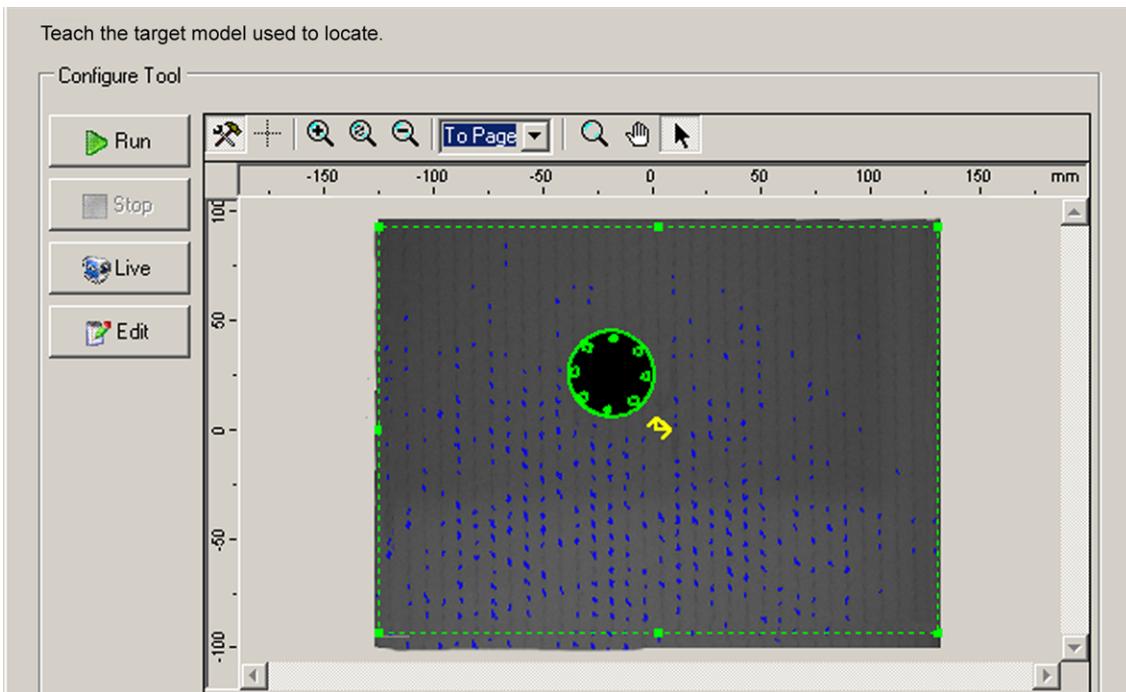


1. Move the robot so the camera is over the target.  
The target must be in the camera's field of view.
  2. Click Here.
- Teach the Vision Tool

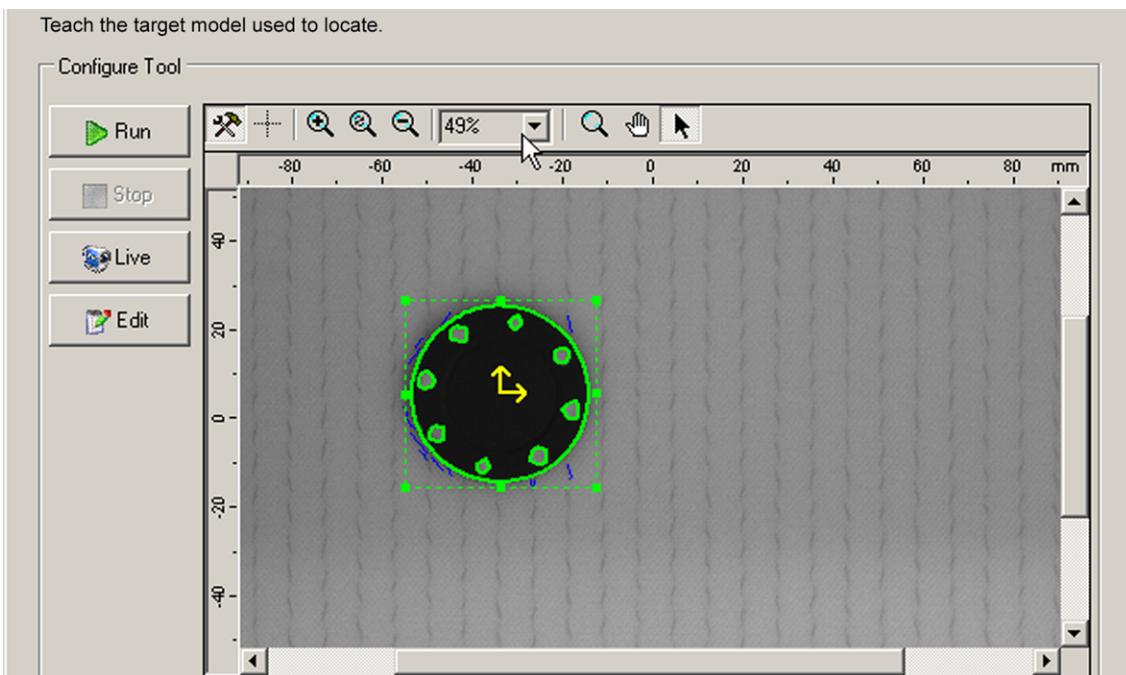
The following figures show the model, origin, and bounding box before and after this procedure has been performed.

## ACE Sight Camera Calibration

---



*Vision Model Before Teaching*

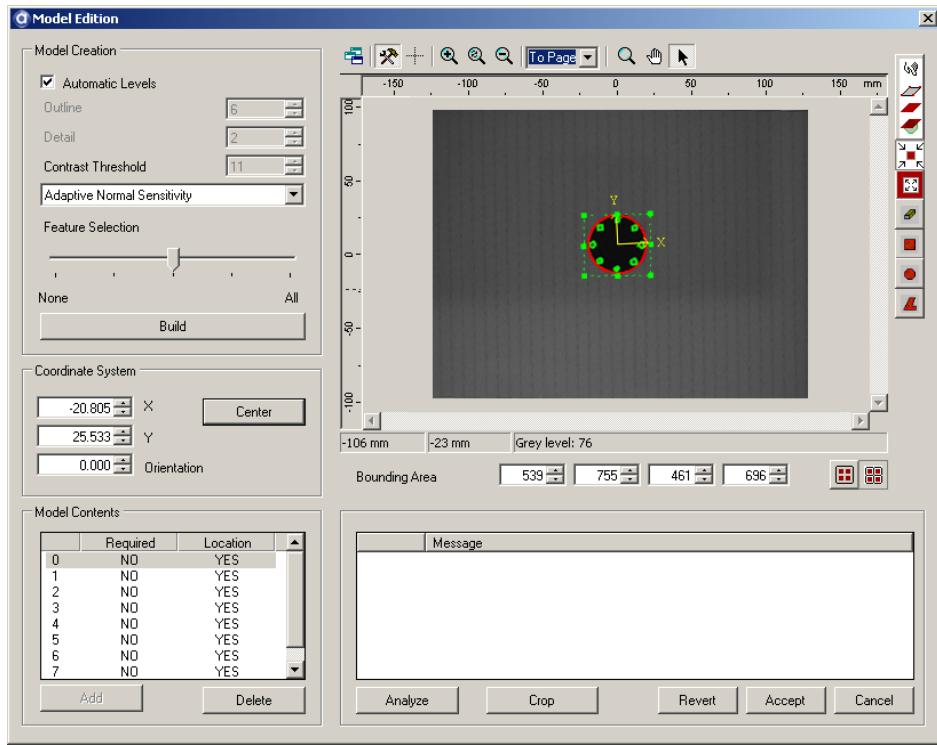


*Vision Model After Teaching*

1. Place the target near the center of the work area.
2. Click Run to take an image.
3. Adjust the model bounding box to fully enclose the model.

Make it close to the perimeter of the model, as shown in the previous figure.

4. Move the origin marker to the center of the model.
  - Drag the origin marker to the center of the model with the mouse  
or
  - Click Edit, to open the Edit window (shown in the following figure).
    - Click Center to center the marker on the bounding box  
or
    - Click on the outline of the model, then click Center to center the marker on the found part



### *Edit Window*

You will place targets at a minimum of four points. These should be close to the four corners of the workspace for best accuracy.

The following steps are repeated for  $n = 1$  through 4.

- Define Calibration Point n

Place the target in the camera field of view.

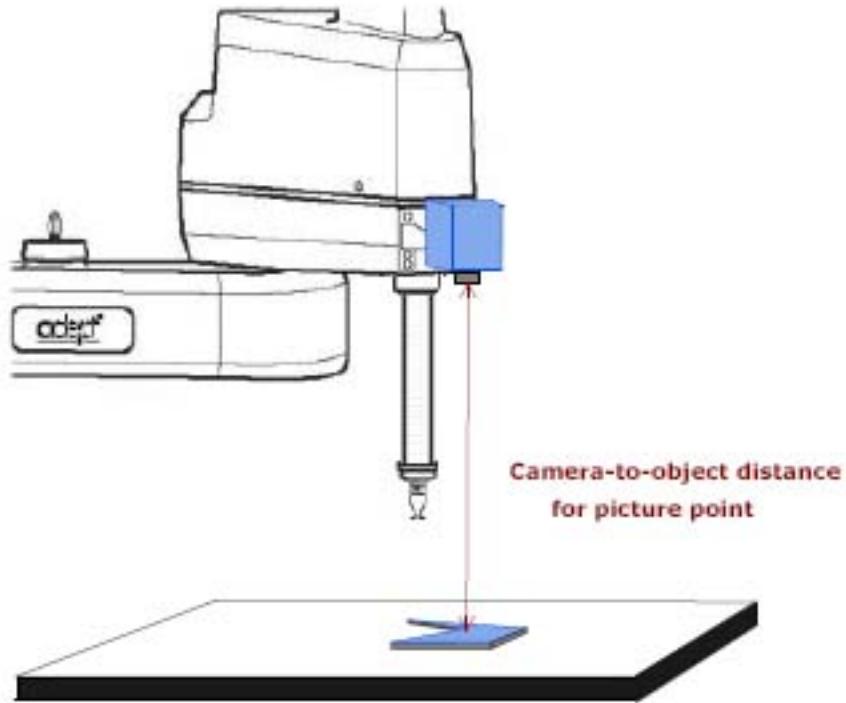
1. Place the target.

Each target placement should be in a different quadrant of the vision window from the last target placement.

At this point, the display is "live", so you can see what the camera sees.

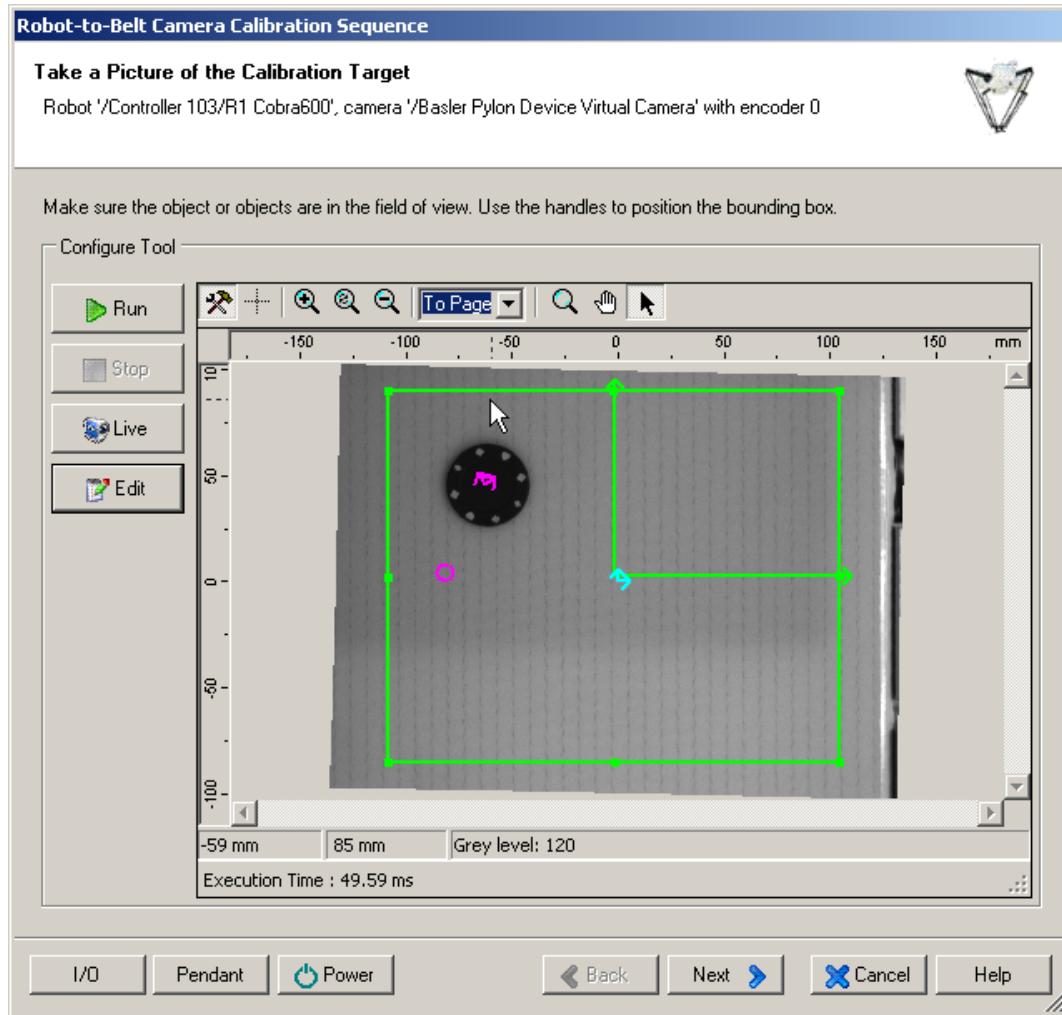
2. Align the robot to the target.
3. Click Here.

- Move the Robot to the Picture Position



1. Move the robot so the camera is over the target.  
The target must be in the camera's field of view.
2. Click Here.

- Take a Picture of the Calibration Target



*Picture of Calibration Target*

1. Ensure the bounding box includes the calibration target, as in the previous figure. Adjust if necessary.
2. If no magenta origin marker is displayed, click Run.

After acquiring the four points, the following will be displayed:

- Minimum number of points

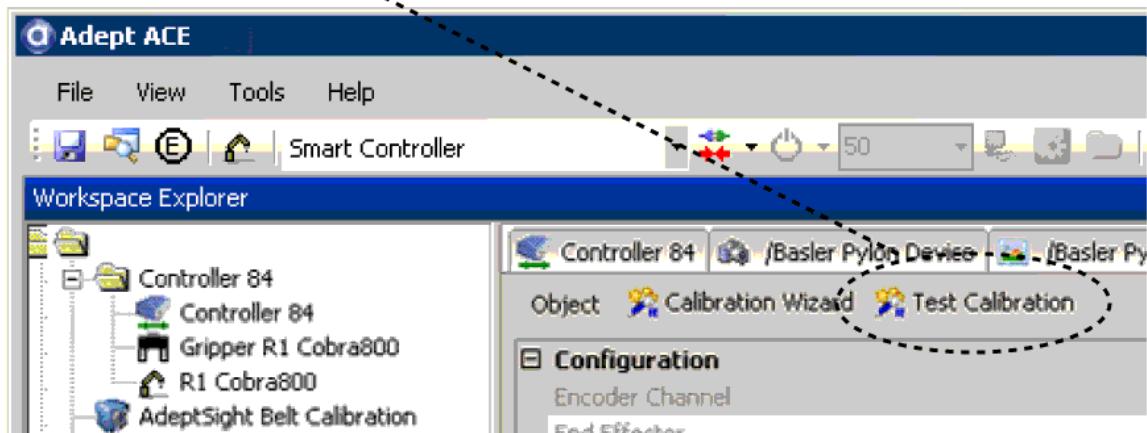
You have collected the minimum number of points needed to perform a calibration. You can choose to calibrate now, or continue collecting more points, possibly improving the accuracy of the calibration.

- Continue adding more points
- Define Calibration Point n
  - The value of n will be incremented with each additional point.
- Move the robot to the location where the picture is taken
- Locate the calibration target
- Minimum number of points
- Calibrate with current set of points

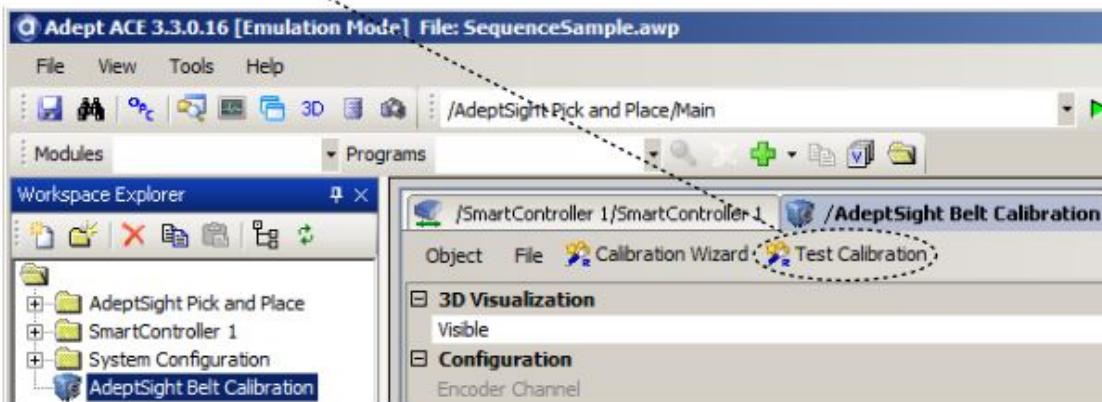
If you choose Continue, this repeats defining another point, aligning the robot to the target, locating the target, and choosing whether to continue.

## Test Procedure

Click on Test Calibration after running the Calibration Wizard.



Click on Test Calibration after running the Calibration Wizard



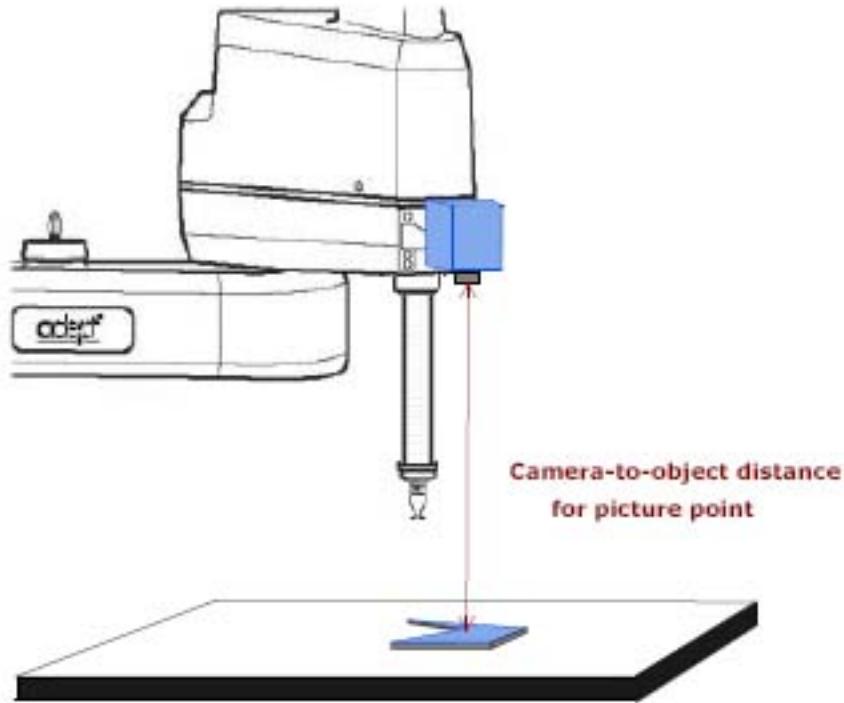
Select Test Calibration

The screens in this procedure are:

- Select End-Effector Tip

This allows you to test a calibration with a different end-effector tip than the one that was used for calibration.

- Move the Robot to the Picture Position



1. Move the robot so the camera is over the target.

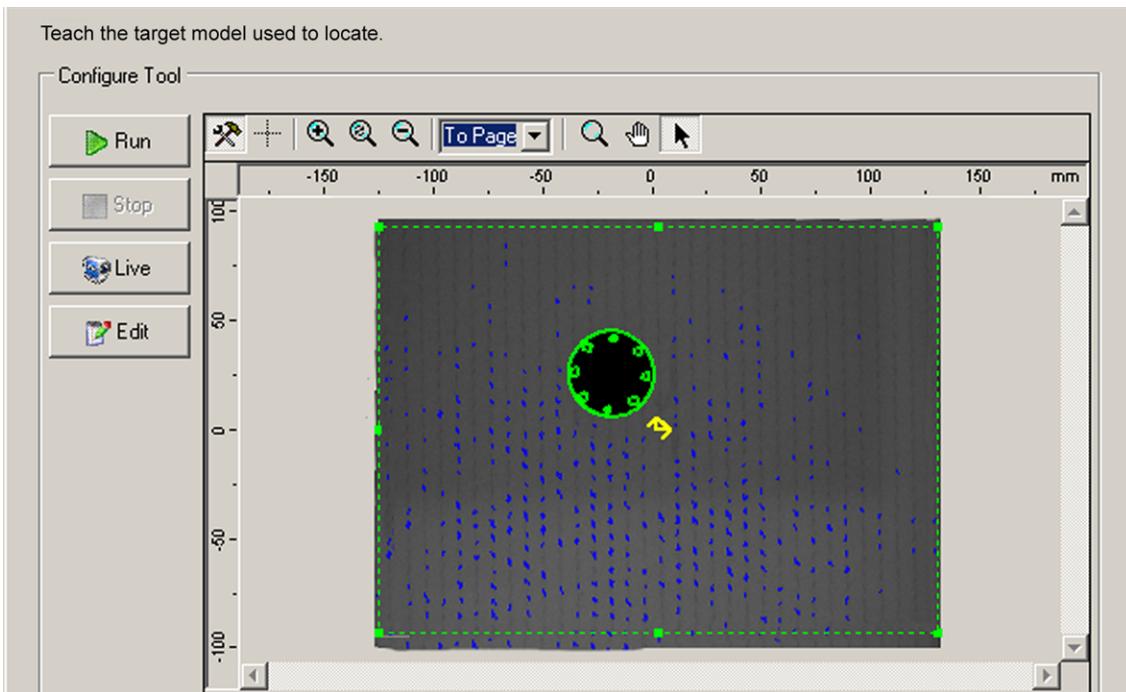
The target must be in the camera's field of view.

2. Click Here.

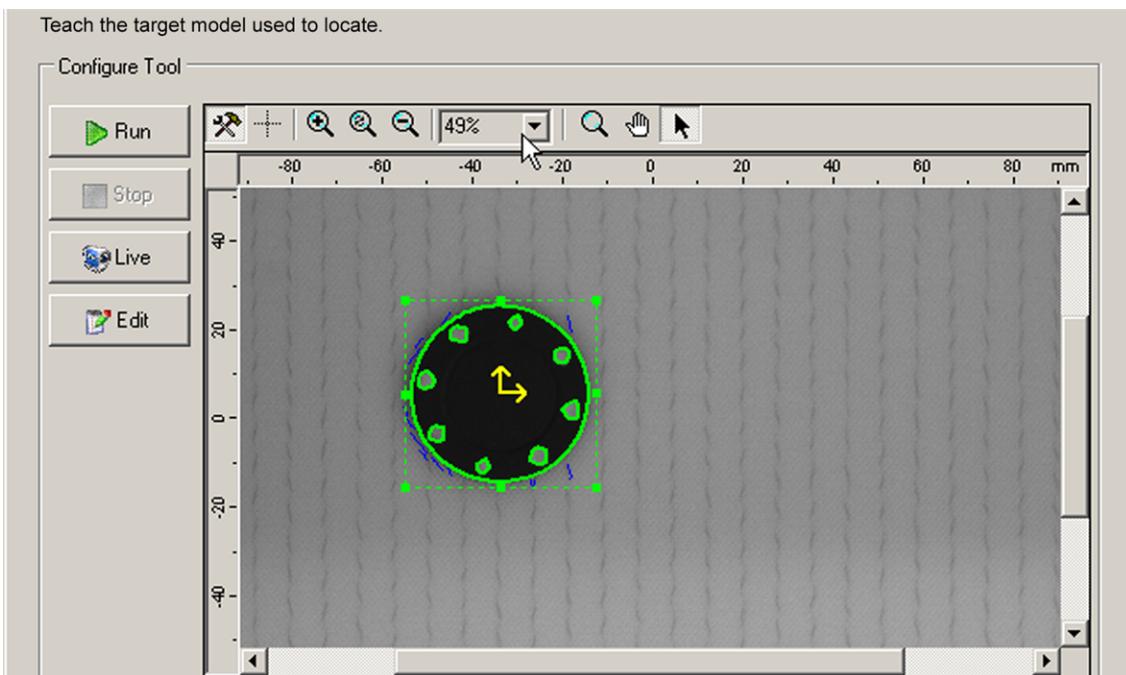
**NOTE:** The Vision Model for the test may be different than the model used to calibrate. You will get best results if they are similar.

- Teach the Vision Tool

The following figures show the model, origin, and bounding box before and after this procedure has been performed.



*Vision Model Before Teaching*

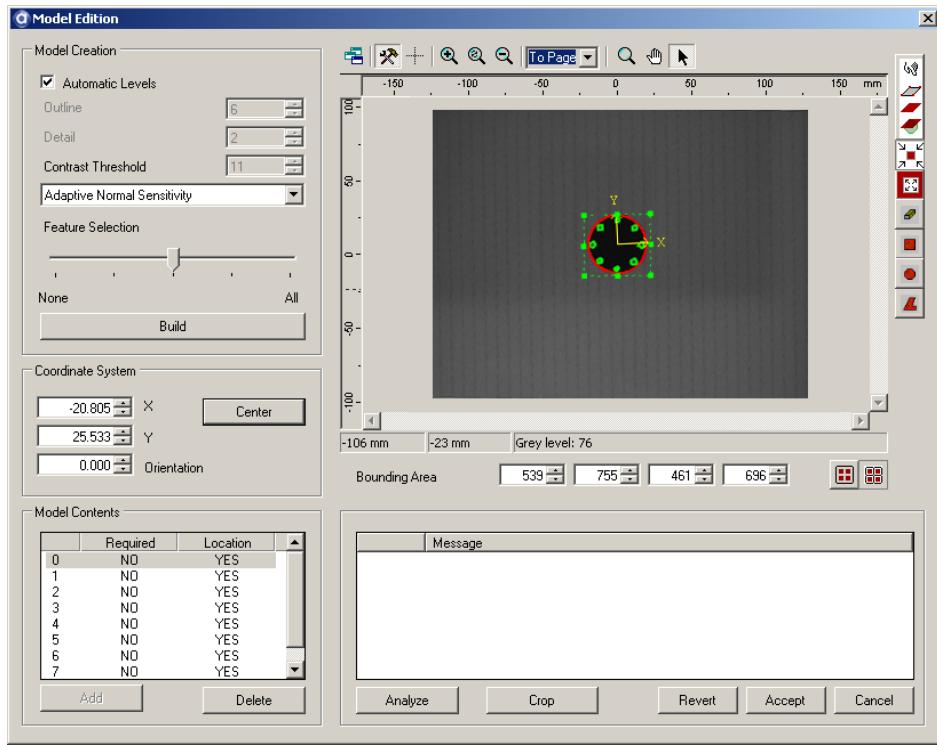


*Vision Model After Teaching*

1. Place the target near the center of the work area.
2. Click Run to take an image.
3. Adjust the model bounding box to fully enclose the model.

Make it close to the perimeter of the model, as shown in the previous figure.

4. Move the origin marker to the center of the model.
  - Drag the origin marker to the center of the model with the mouse  
or
  - Click Edit, to open the Edit window (shown in the following figure).
    - Click Center to center the marker on the bounding box  
or
    - Click on the outline of the model, then click Center to center the marker on the found part



*Edit Window*

- Locate the Target

1. Adjust the bounding box to the work area. This must include the calibration target.

**NOTE:** The calibration target should be correctly adjusted from the calibration procedure.

2. If no magenta origin marker is displayed, click Run.

If necessary, move the target so it can be located, then click Run.

3. Click Next. The robot should move to the target.

- Continue testing the calibration

- I wish to continue testing

- Move the robot to the location where the picture is taken
- Locating the target

- Continue testing the calibration
  - Do not continue testing

## Related Topics

[Calibration Overview](#)

[ACE Sight Belt Calibration Interview Wizard](#)

[ACE Sight Belt Calibration Wizard](#)

[ACE Sight Camera Calibration Interview Wizard](#)

[ACE Sight Camera Calibration Wizard - Automated, no Belt](#)

[ACE Sight Camera Calibration Wizard - Automated, w/ Belt](#)

[ACE Sight Camera Calibration Wizard - Automated, Arm-Mount](#)

[ACE Sight Camera Calibration Wizard - Automated Upward-Looking](#)

[ACE Sight Camera Calibration Wizard - Manual, no Belt](#)

[ACE Sight Camera Calibration Wizard - Manual w/Belt](#)

[ACE Sight Camera Calibration Wizard - Manual, Upward-looking](#)

### ***ACE Sight Camera Calibration Wizard - Manual, Upward-looking***

This topic covers calibrating an upward-looking camera to a robot.

## Requirements

- The robot and camera must be correctly connected and functioning.
- The camera must be calibrated. See Standalone Camera Calibration on page 320.
- The ACE Sight Camera Calibration Interview Wizard must have completed successfully.

## What this Wizard Does

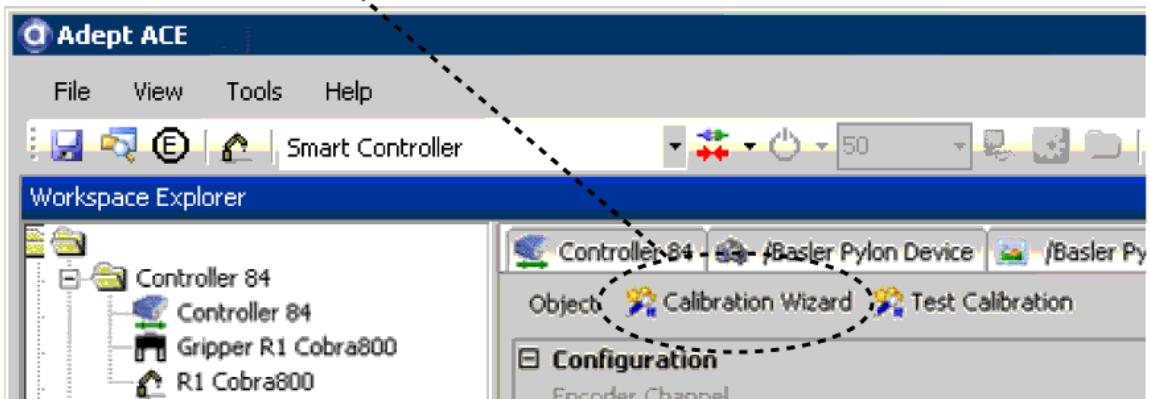
With this wizard you will:

- Teach the picture-taking position, the vision model, and the placement position
- Perform a robot-to-camera calibration

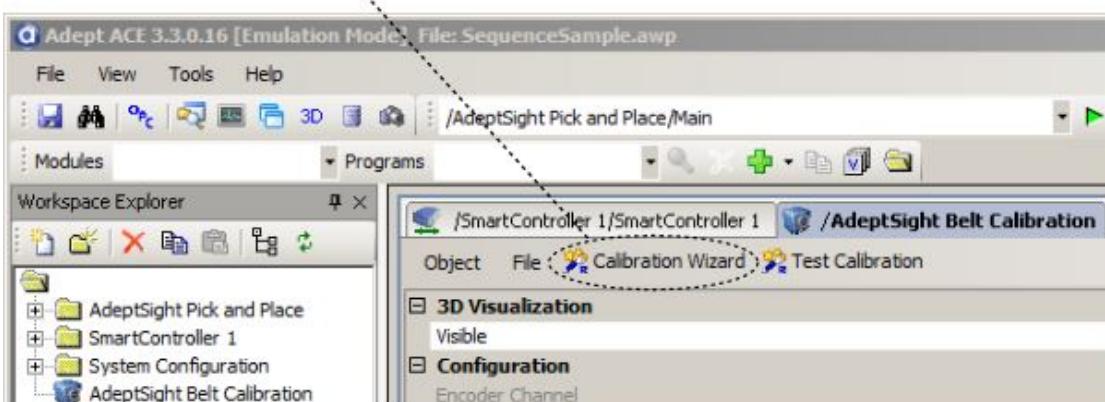
## Procedure

From the Camera Calibration Object window, click Calibration Wizard.

Click on Calibration Wizard after running the Calibration Interview Wizard.



Click on Calibration Wizard after running the Calibration Interview Wizard



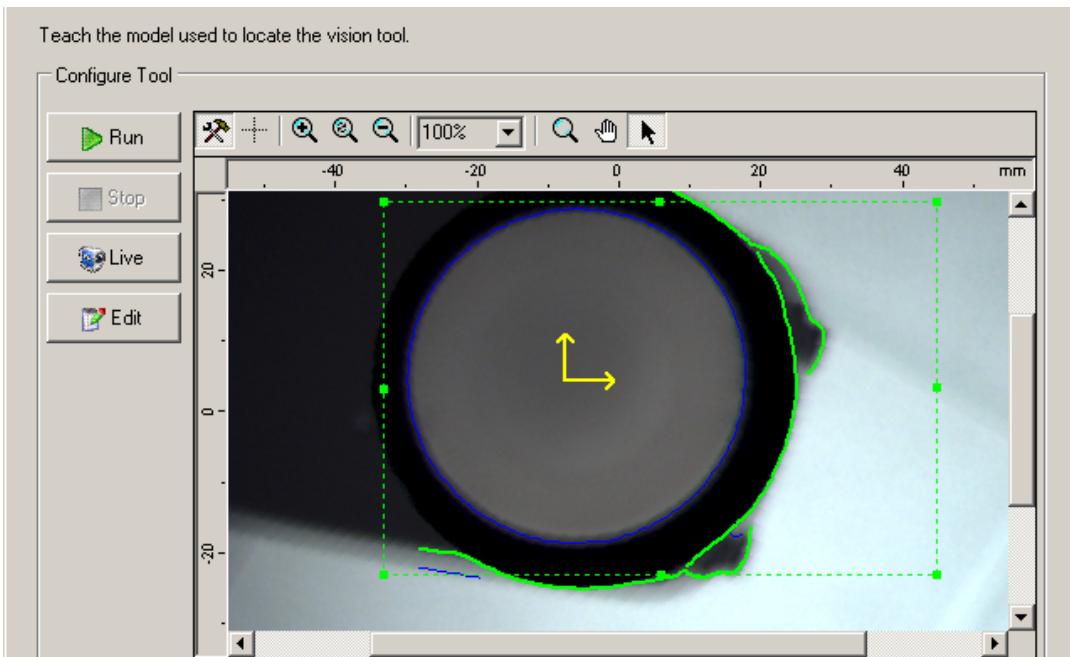
The screen titles will all display "Robot-to-Refinement Camera Automated Calibration Sequence".

The screens in this procedure are:

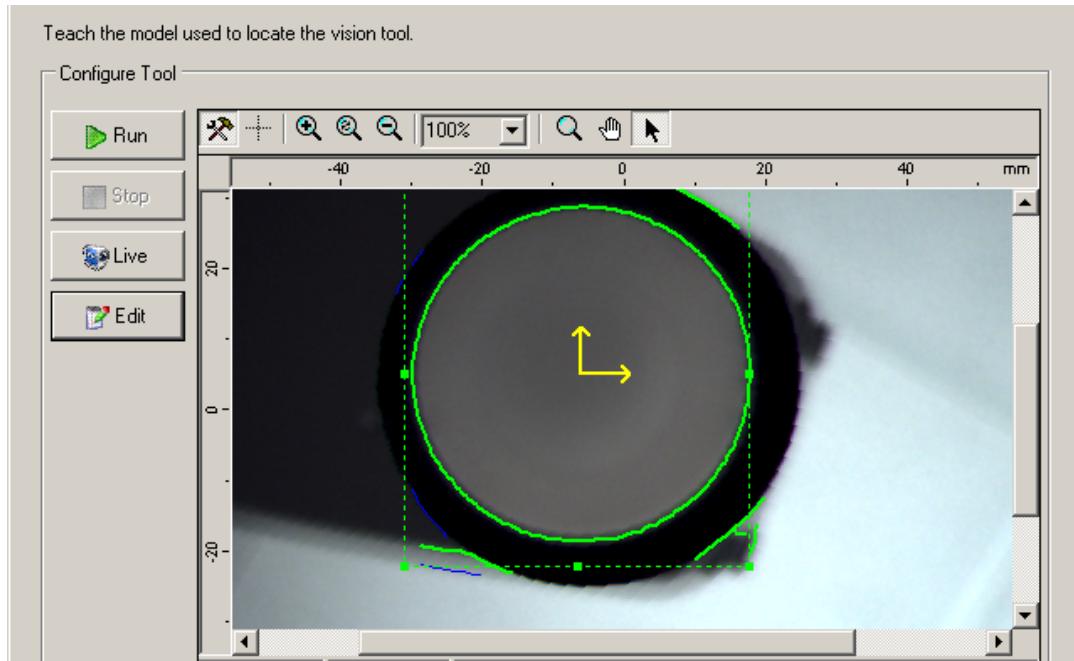
- Select the End-Effector
- Move the Robot to the Picture Position
  1. Move the robot over the camera.  
The robot's end-effector must be in the camera's field of view.
  2. Click Here.
- Teach the Vision Tool

The following figures show the model, origin, and bounding box before and after this

procedure has been performed.



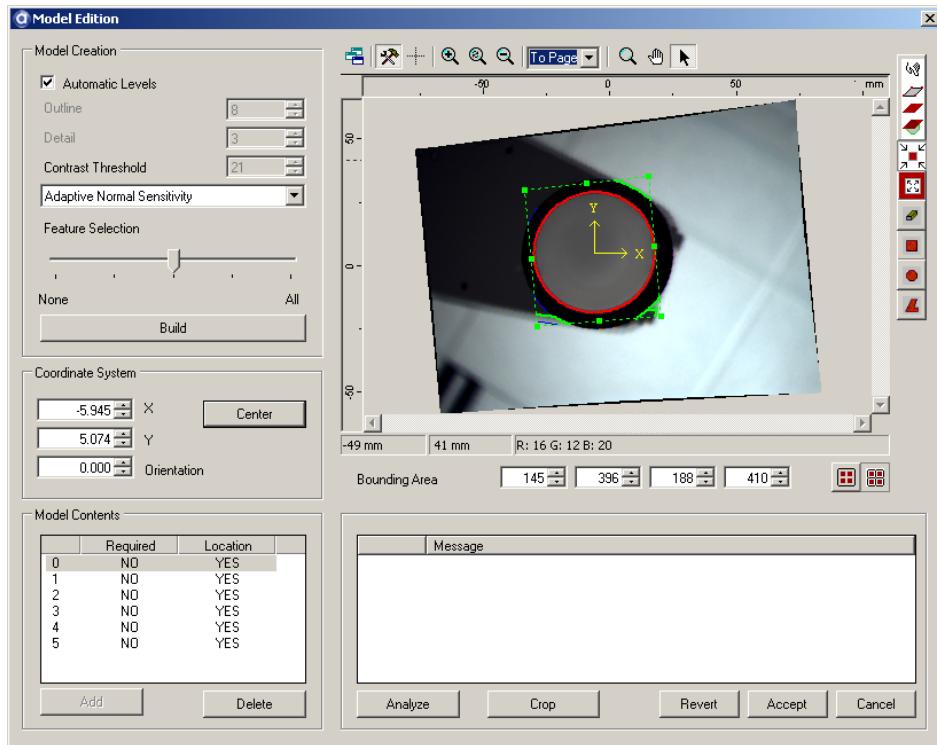
*Vision Model Before Teaching*



*Vision Model After Teaching*

1. Move the robot so that the model is near the center of the camera field of view.
2. Click Run to take an image.
3. Adjust the model bounding box to fully enclose the model.  
Make it close to the perimeter of the model, as shown in the previous figure.
4. Move the origin marker to the center of the model.
  - Drag the origin marker to the center of the model with the mouse  
or
  - Click Edit
    - Click Center to center the marker on the bounding box  
or
    - Click on the outline of the model, then click Center to center the marker on the found part

The Edit window is shown in the following figure.



### *Edit Window*

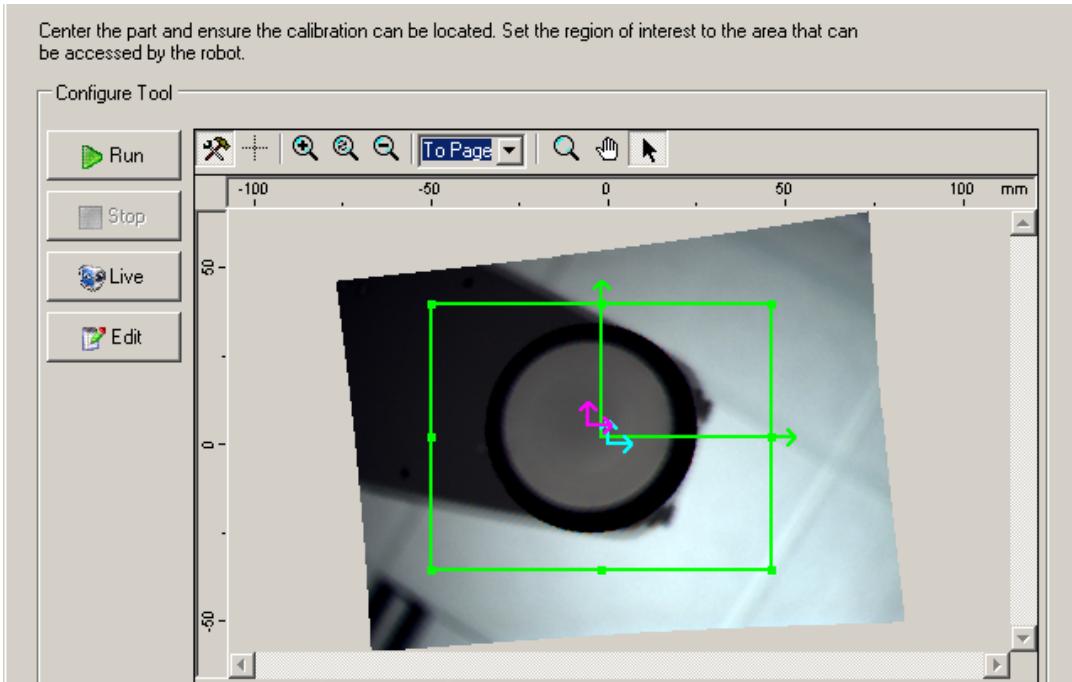
- Click Accept, then click Next.

You will move the robot (and target) to a minimum of four points. These should be close to the four corners of the camera field of view for best accuracy.

The following steps are repeated for  $n = 1$  through 4:

- Define Calibration Point n
  1. Move the robot in X and Y only, so the part is moved in the field of view.
  2. Click Here.

- Define the Region of Interest



### *Defining the Region of Interest*

1. Adjust the bounding box to the camera field of view. This must include the calibration target.
2. If no magenta origin marker is displayed, click Run.

If necessary, move the robot so the target can be located, then click Run.

If n is less than 4, this goes back to Define Calibration Point n.

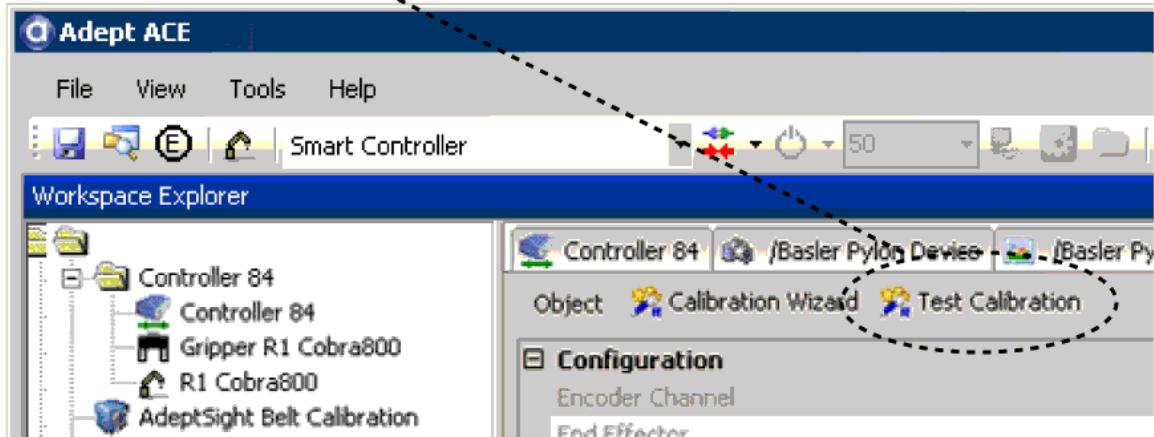
- Minimum number of points

You have added the minimum number of points required to compute the calibration. You can choose to calibrate now, or continue adding more points, possibly improving the accuracy of the calibration.

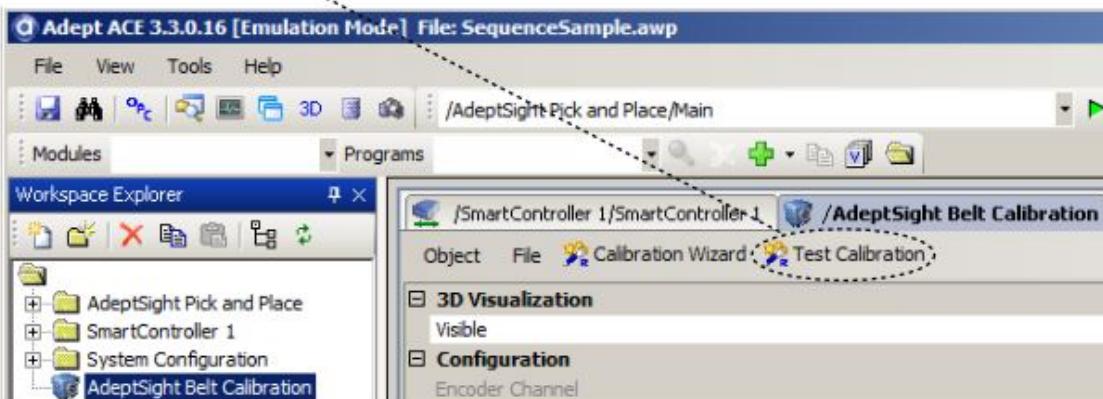
- Continue adding more points
  - Calibration Point n, where n is incremented for each point
  - Locating the calibration target
  - Minimum number of points
- Calibrate with current set of points

## Test Procedure

Click on Test Calibration after running the Calibration Wizard.



Click on Test Calibration after running the Calibration Wizard



The screens in this procedure are:

- Select End-Effector Tip

This allows you to test a calibration with a different end-effector tip than the one that was used for calibration.

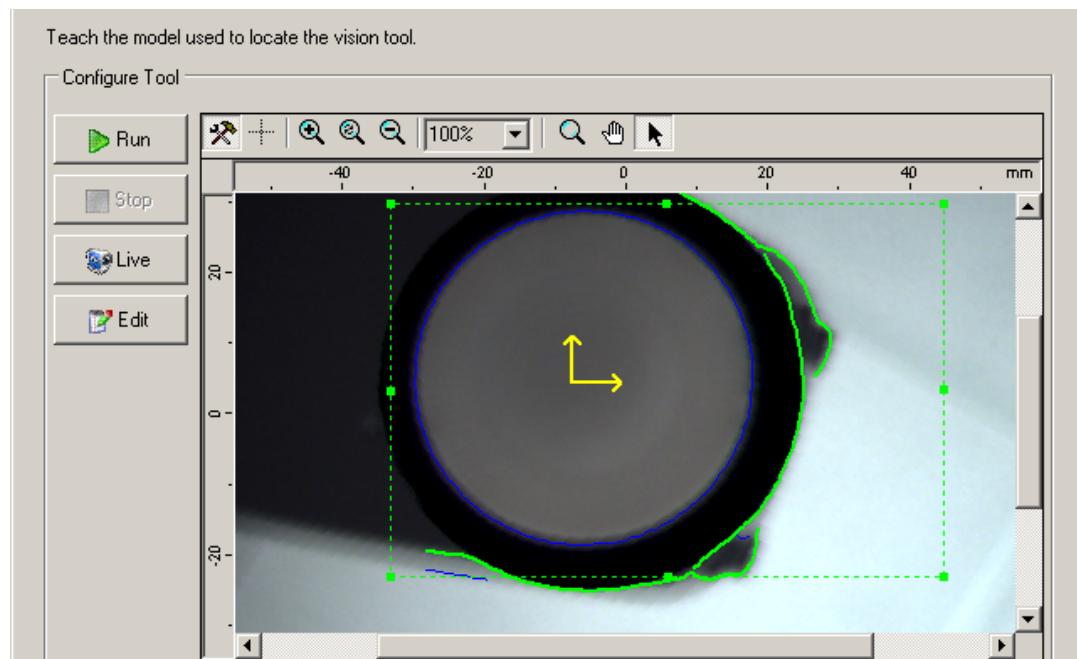
- Move the Robot to the Picture Position

1. Move the robot over the camera.

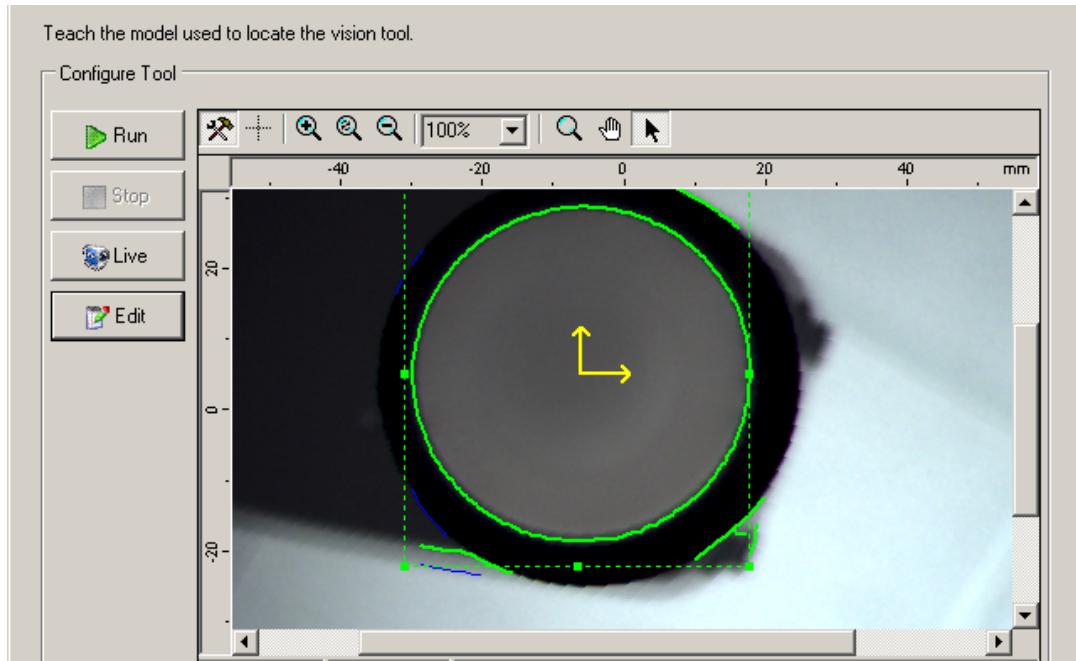
The robot's end-effector must be in the camera's field of view.

2. Click Here.
- Teach the Vision Tool

The following figures show the model, origin, and bounding box before and after this procedure has been performed.



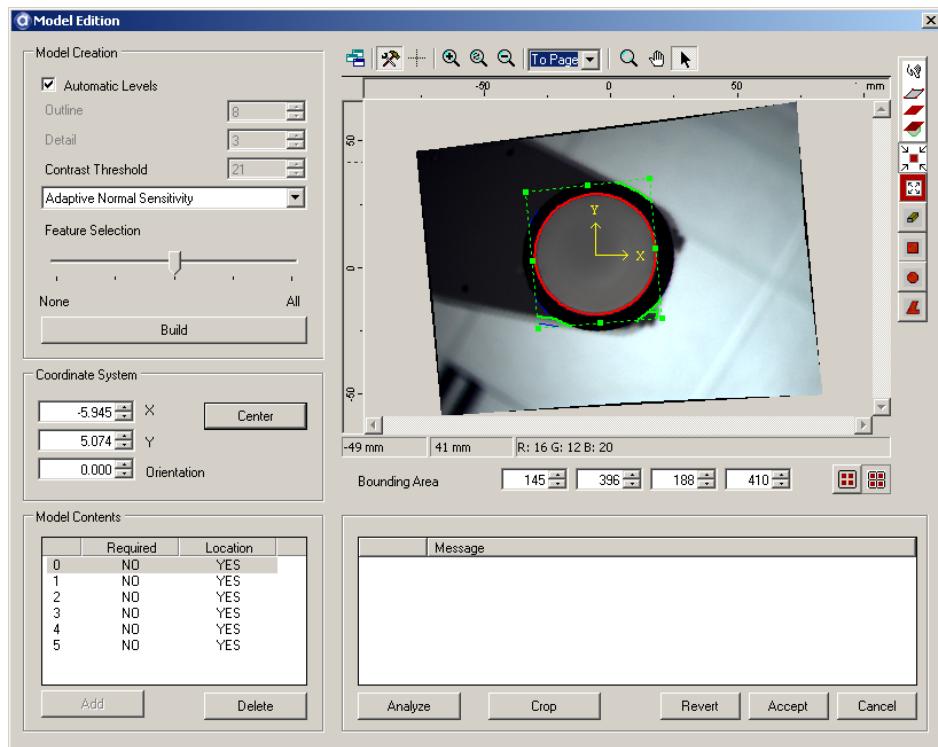
*Vision Model Before Teaching*



*Vision Model After Teaching*

1. Move the robot so that the model is near the center of the camera field of view.
2. Click Run to take an image.
3. Adjust the model bounding box to fully enclose the model.  
Make it close to the perimeter of the model, as shown in the previous figure.
4. Move the origin marker to the center of the model.
  - Drag the origin marker to the center of the model with the mouse  
or
  - Click Edit
    - Click Center to center the marker on the bounding box  
or
    - Click on the outline of the model, then click Center to center the marker on the found part

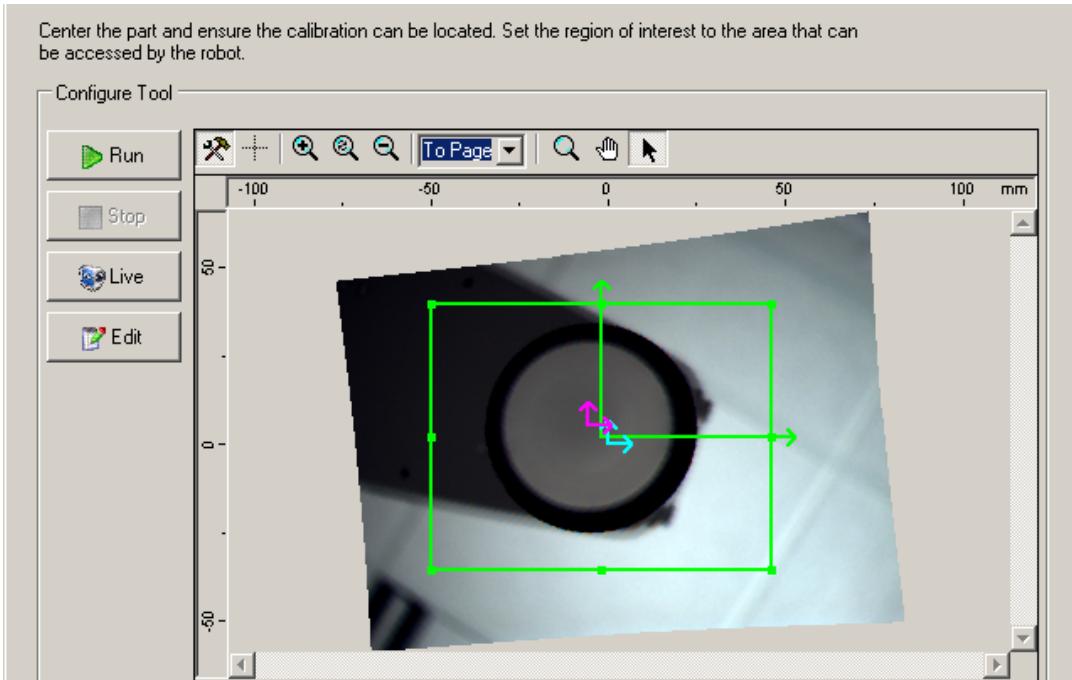
The Edit window is shown in the following figure.



### Edit Window

- Click Accept, then click Next.

- Define the Region of Interest



### *Defining the Region of Interest*

1. Adjust the bounding box to the camera field of view. This must include the calibration target.
2. If no magenta origin marker is displayed, click Run.

If necessary, move the robot so the target can be located, then click Run.

- Continue testing the calibration

- I wish to continue testing

- Move the robot to the location where the picture is taken
    - Locating the calibration target
    - Continue testing the calibration

- Do not continue testing

### **Related Topics**

[Calibration Overview](#)

[ACE Sight Belt Calibration Interview Wizard](#)

[ACE Sight Belt Calibration Wizard](#)

[ACE Sight Camera Calibration Interview Wizard](#)

[ACE Sight Camera Calibration Wizard - Automated, no Belt](#)

[ACE Sight Camera Calibration Wizard - Automated, w/ Belt](#)

[ACE Sight Camera Calibration Wizard - Manual w/Belt](#)

[ACE Sight Camera Calibration Wizard - Automated, Arm-Mount](#)

[ACE Sight Camera Calibration Wizard - Manual, Arm-Mount](#)

## **ACE Sight Latch Calibration**

ACE Sight Latch Calibration calibrates a robot to a latch.

This calibration uses a user-supplied sensor to generate a latch signal when an object, such as a pallet, reaches a specific point, typically on a conveyor belt. This synchronizes the rest of the system with the position of the object.

This calibration is available in Emulation Mode.

The end result of this calibration is an ACE Sight Latch Calibration object.

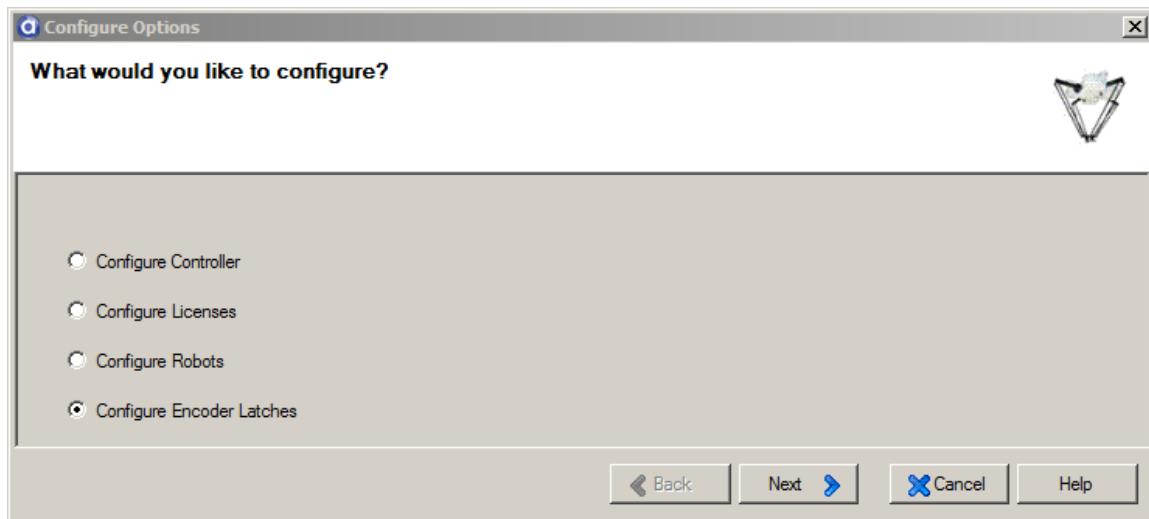
### **Requirements**

- The PC running the ACE Sight software must be connected to the controller for the robot.
- A belt calibration must exist in the Workspace.
- The robot, controller, and belt must be correctly connected and functioning.
- The signal to be used for the latch must be set in the SmartController. This is covered in the following section.

**NOTE:** In Emulation Mode, you will not be asked for a latch signal number, as this is not needed. You can then skip the following section.

To set the latch signal in the SmartController:

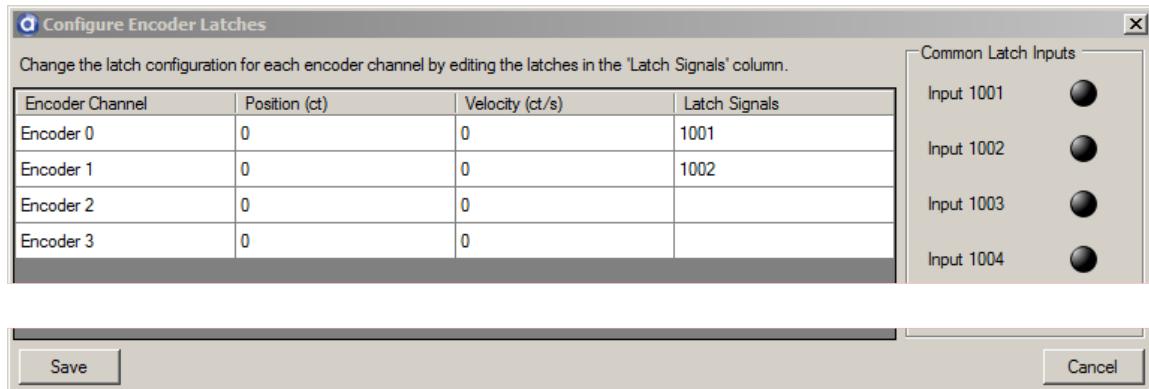
1. Double-click the SmartController object in the Tree structure to open the object editor.
2. Click Configure.
3. Check the Configure Encoder Latches.



*Configure Encoder Latches Selection*

4. Click Next.
5. Select the encoder number you want to use, and set the signal number for the latch you are using.

This encoder number will be used in the latch calibration object as a Latch Number.



*Setting Latch Signal*

To add a latch calibration to the Workspace, right-click in the Tree structure of the Workspace Explorer and select:

**New > Vision > ACE Sight > ACE Sight Latch Calibration.**

1. Click Next on the Welcome screen.
2. Select a robot, then click Next.  
Click on the hand icon to display the available robots.
3. Select an end-effector for the robot, then click Next.  
Click on the hand icon to display the available end-effectors.
4. Select a belt calibration.  
Click on the hand icon to display the available belt calibrations.
5. Click Finish.

A Latch Calibration object will have been created in the Tree structure. If you double-click on that object, it will open the object editor. From there, you can verify that the Encoder Number you selected for the SmartController is set correctly.

### **Related Topics**

[Calibration Overview](#)

[ACE Sight Belt Calibration](#)

[ACE Sight Camera Calibration](#)

## Programming ACE

The topics in this chapter describe how to use the programming features in the ACE software.

Programming Overview .....	746
Task Status Control .....	747
C# Language Programming .....	750
Profiler .....	755
V+ Programs and Variables .....	760
V+ Editor Tool .....	765
V+ Debugger Tool .....	769
V+ Task Manager .....	772
Monitor Window .....	777
Error Messages .....	780
Find Dialog .....	782
OPC Data Access and Process Control .....	785
Watch Variable Tool .....	794

## Programming Overview

There are several ways of programming ACE applications. Each method is summarized below. The method you choose depends on your programming experience and the type of application you want to develop. Also, note that the different methods of programming can be combined.

- **PackXpert Process Manager:** The Process Manager provides a point-and-click interface for configuring and programming the workcell. The PackXpert Process Manager is the recommended method for programming sophisticated packaging applications. For details, see Process Control on page 799.
- **C# Language Programming:** The ACE software provides a simple C# program editor, which can be used to create and edit programs. For details, see C# Language Programming on page 750.
- **V+ Programming:** The ACE software provides several V+ programming tools, such as a V+ Editor/Debugger, V+ Module Manager, and V+ Watch Variable tool, which can be used to create V+ programs for your application. For details, see V+ Editor Tool on page 765 and Watch Variable Tool on page 794.

## **Task Status Control**

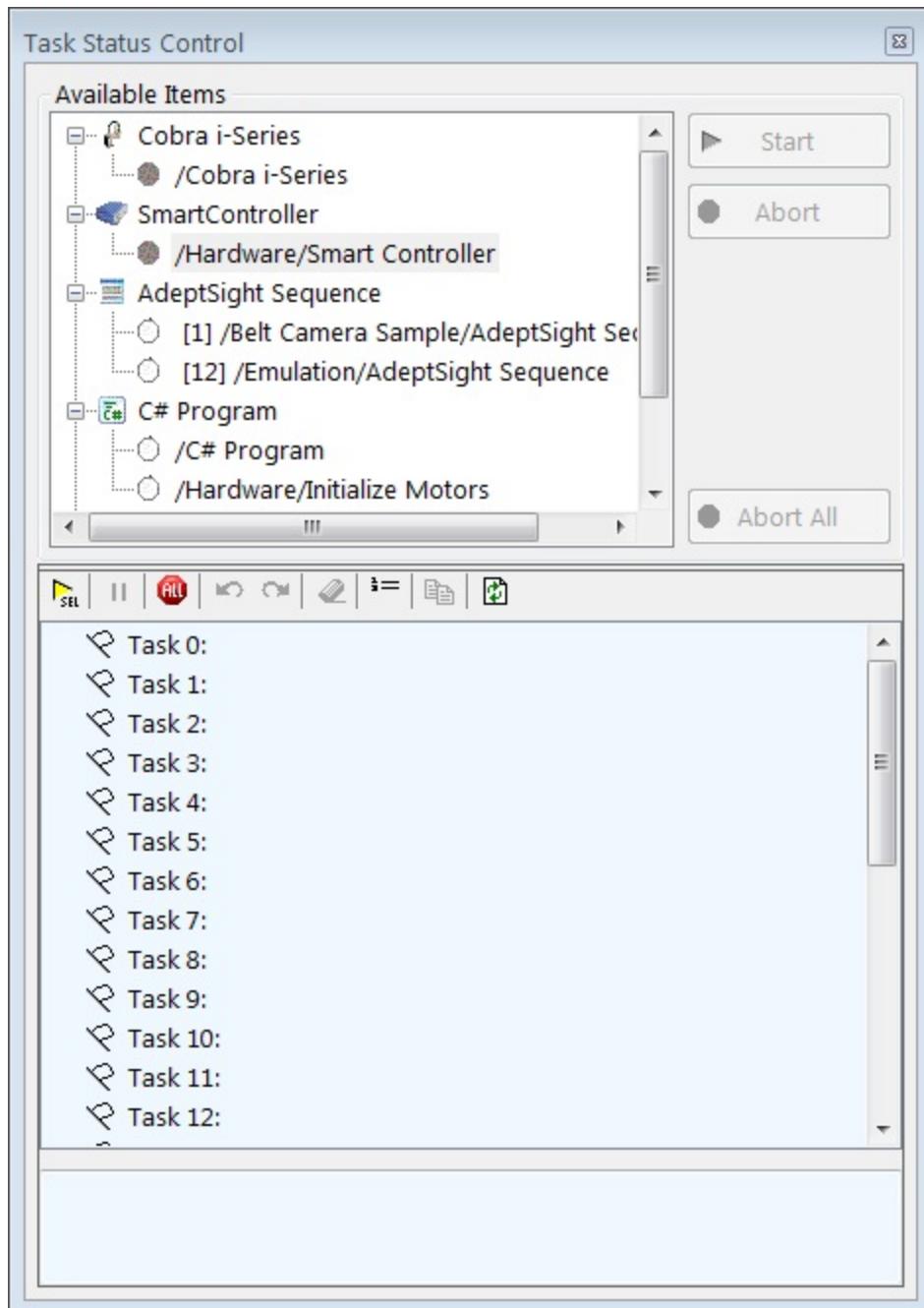
The Task Status Control provides an interface for controlling and monitoring the execution of one or more programs (such as C# script objects) that run concurrently on the PC or controller.

To open the Task Status Control from the ACE menu, select:

**View > Task Status Control**

## Task Status Control

---



Task Status Control

## Task Status Control

---

The Available Items group is used to select the item for display. The available items are arranged in a "tree view", which is grouped by task type (C# Program, Process Manager, etc.). Programs that are running or waiting for user interaction are marked as follows:

Icon	Description
	The program is running
	The program is paused
	The program is stopped (aborted)

Selecting a particular task in the list enables or disables buttons on the display according to the allowed recovery for the current state. The buttons on the Task Status Control have the following functions:

Item	Description
Abort	Enabled during execution, pauses or exceptions, this button causes the execution to stop for the selected task.
Abort All	Enabled during execution, pauses or exceptions, this button causes the execution to stop for all tasks.
Start	Starts (executes) the selected task.

## C# Language Programming

The tool described in this section is based on the C# programming language. The C# programming language is used for writing PC-based applications. If you are new to the C# programming language, there is documentation, tutorials, and online classes available through the following URL:

<http://msdn.microsoft.com/>

The ACE software provides a simple C# program editor, which can be used to create and edit C# programs.

In addition to the C# program editor, the ACE software provides:

- a Custom Allocation Script editor, which allows you to create and edit custom C# part-allocation programs for use with the Process Manager. For details, see Custom Allocation Script on page 825.
- a Custom Vision tool, which allows you to create C# programs that run in the context of a vision operation. From within a Custom Vision tool, other tools can be executed and return a set of results which are used as the output of the tool. For details, see the *ACE Sight User's Guide*.

### Using the C# Program Editor

The C# program editor can be used to create and edit programs for performing various tasks and "automation" within the ACE environment. These programs can execute from the Task Status Control. They can also be executed from V+ using the ace\_srvc.v2 library.

A C# program could be used to:

- handle product changes for a workcell or production line
- enable/disable processes in a packaging application
- load different vision models and associate them with a locator tool

To use the C# program editor:

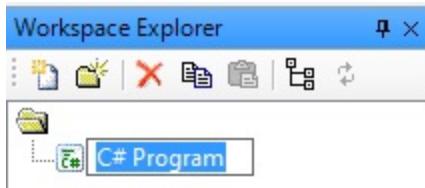
1. Create a C# Program object. Right-click on the workspace explorer folder view and select **New > Program > C# Program**. This will create a C# Program object and open the editor, as shown in the following figure.

The screenshot shows the C# Program Editor interface. The main window displays a C# script with syntax highlighting. The code includes several 'using' statements and a 'namespace' declaration. The 'Main' method is defined, and its body contains a 'Trace.WriteLine' call and a multi-line comment. Below the editor is a status bar with tabs for 'Compile Errors', 'Trace Messages', and 'Browse Variables'. The 'Compile Errors' tab is selected, showing a list of three errors:

Line	Col	Message
14	39	Statement expected.
18	49	; expected.
22	2	} expected.

C# Program Editor

2. When the C# Program object is created, it will be selected in the folder tree so that you can assign a name to your C# program.

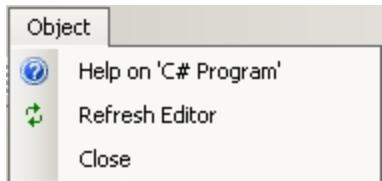


*Renaming the C# Program Object*

3. Use the toolbar items and code editor, described below, to edit and compile your program.
4. Use the Task Status Control to execute the program. For details, see Task Status Control on page 747.

## Toolbar Items

This section describes the selections available from the C# program editor toolbar.



### Help

Displays the online help for the C# program editor.

### Refresh Editor

Refreshes the contents of the editor window.

### Close

Closes the C# program editor.

Item	Description
	Deletes the selected text and stores a copy in the Windows clipboard.
	Copies the selected text to the Windows clipboard.
	Pastes the contents of the Windows clipboard at the current cursor position.
	Clears the messages in the Trace Messages tab.
	Compiles the program.
	Compiles and runs the program.
	(Active when program is running) Stops the program.
Release Mode ▾	Selects the editor mode: Release Mode is the normal execution mode; Debug Mode is used to debug the program.

Item	Description
	(Debug Mode only) Used to step over a call.
	(Debug Mode only) Executes the next line of the program (single step through the program).
	(Debug Mode only) Runs the program.

## Tab Items

The following tabs are available in the editor.

Item	Description
Compile Errors	Displays a list of compile errors. Each error provides the line number, column number, and error message.
Trace Messages	Displays the trace messages (text included in any Trace.WriteLine() call in the program).
Browse Variables	Allows you to add variables to the window; the listed variables are updated after every step.

## Code Editor

The main part of the C# program editor contains a code editor. This is a simple editor that is used to create and edit your C# programs.

1. Use the code editor section of the C# program editor to create your C# program. A sample program is provided, to get you started.

While creating your program, if there are any error, they are shown at the bottom portion of the code editor. The line number, character position (column) and message is provided, as shown in the following figure.

Line	Col	Message
12	11	; expected.

*Error Message*

When you double-click on a line, it will highlight the line in the code editor. When you hover over the error message and it is too long, it will display as a tool tip.

2. You can use the toolbar items, described above, to access editing functions, such as cut, copy, and

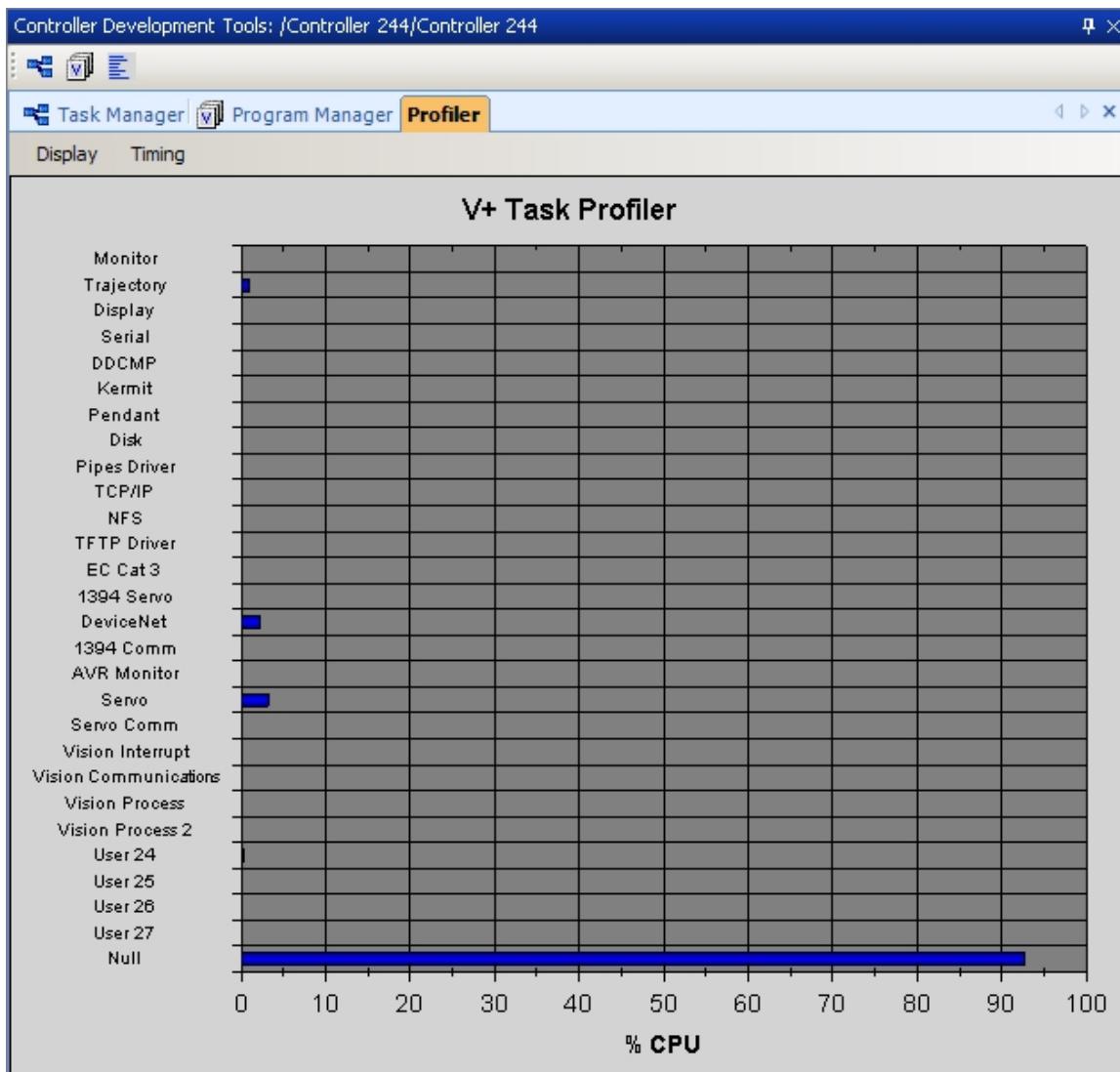
paste.

3. After you have finished editing your program, click the compile icon to compile your program.

## Profiler

The Profiler tool is used to provide a graphical display of CPU time being used by executing system and user tasks. It is useful for developing and debugging applications.

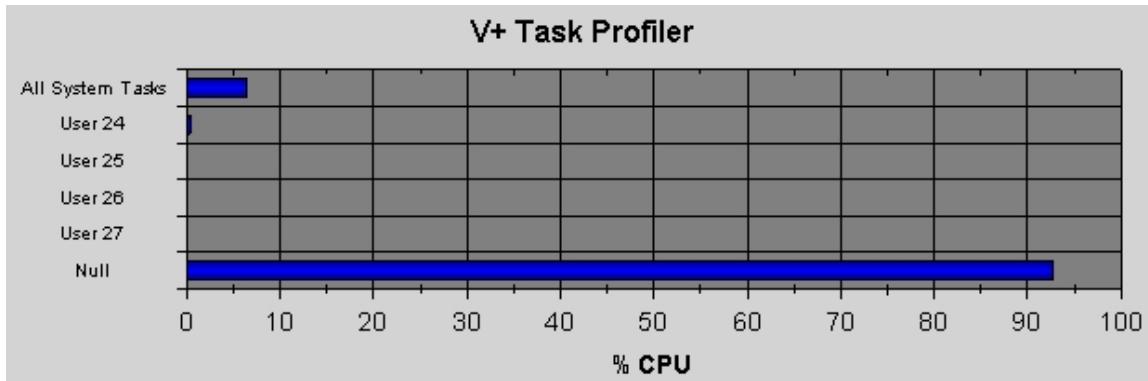
To access the Profiler tool Click the Profiler (  ) icon on the Controller toolbar. The Profiler tool opens, as shown in the following figure.



*V+ Task Profiler*

## Operation

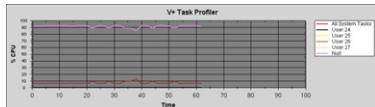
The profiler displays CPU usage on the % CPU Time graph. The % CPU Time graph is a bar graph that shows the percent of time used by the system and user tasks (see the following example graph).



*Percent (%) CPU Time Graph*

The example graph shows that the running system is allocating approximately 7 percent of its time to system tasks, one percent to user task 24, zero percent to user tasks 25 - 27, and 93 percent to the Null category. (The Null category shows the % CPU time not used by any other tasks.)

Additionally, the graph can be switched to "history" mode, as shown in the following figure. In this mode, the tasks are displayed as line plots (a different colored line represents each task, as shown in the color key). This allows you to track changes in CPU load over time for each task.



*History Graph*

## Choosing Displayed Tasks

The Display menu allows you to specify which tasks are displayed. Choosing "All System Tasks" shows all the individual V+ system tasks on the bar graph. If "All System Tasks" is not chosen, all system task usage is displayed on one line marked System Tasks (as shown in the previous figure).

**NOTE:** Selected options are indicated by a checkmark.

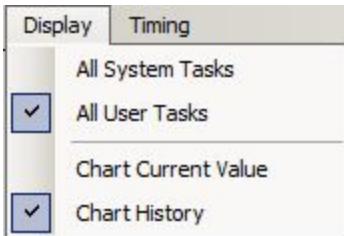
Choosing "All User Tasks" displays all the user tasks available to your system. If "All User Tasks" is not chosen, only tasks with a program on the execution stack are displayed.

## Profiler Menus

The Profiler menus allow you to determine which display is presented and to determine how often the displayed graph is updated.

### Display Menu

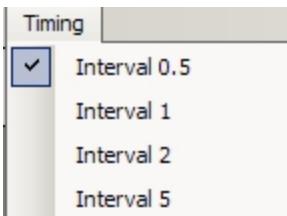
The Display menu determines which display is presented and how many lines of information are graphed.



Menu Item	Description
All System Tasks	When chosen, each system task is displayed on its own line. When not chosen, all system task time is accumulated into one line labeled System Tasks.
All User Tasks	When chosen, all user tasks are graphed whether or not they are in use. When not chosen, only the tasks that have a program on the execution stack are displayed.
Chart Current Value	When chosen, the display is a bar graph that shows the current value of each task.
Chart History	When chosen, the display is a line graph that provides a historical plot of each task.

### Timing Menu

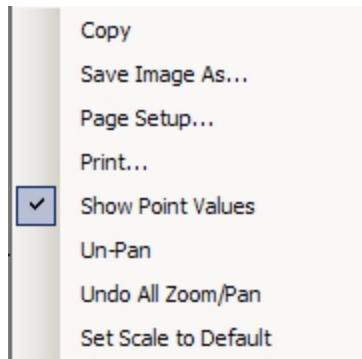
The Timing menu determines how often the graphs are updated.



Menu Item	Description
Snap Shot	Put the display in freeze mode and update the graphs.
Interval 0.5	Update graphs at 0.5-second intervals.
Interval 1	Update graphs at 1-second intervals.
Interval 2	Update the graphs every 2 seconds.
Interval 5	Update the graphs every 5 seconds.

**Shortcut Menu**

The shortcut menu is available when you right-click in the graph area of the Profiler window. This menu contains options for copying, saving and printing the data, as well as some pan and zoom options.



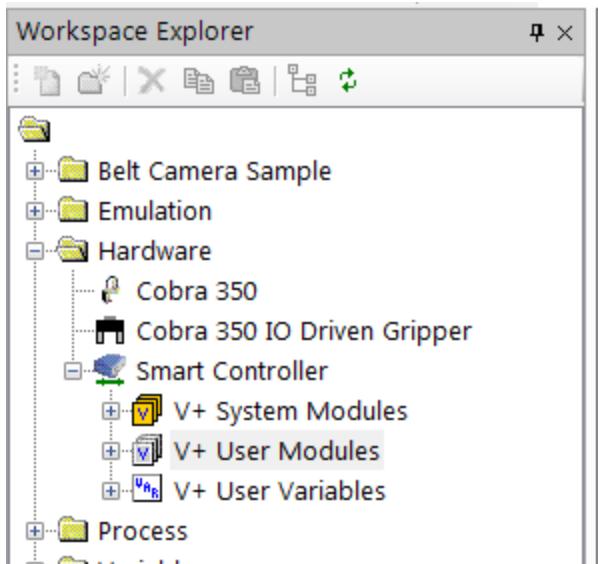
Menu Item	Description
Copy	Copies the image to the Windows clipboard.
Save Image As...	Saves the image to a file.
Page Setup...	Opens the Page Setup dialog, which is used to set the page and print parameters for printing the image.
Print...	Opens the Windows Print dialog, which is used to print the image.
Show Point Values	Adds the point values to the graph.

---

<b>Menu Item</b>	<b>Description</b>
Un-Pan	Removes (undo) the last pan operation.
Undo All Zoom/Pan	Removes all zoom and pan operations.
Set Scale to Default	Returns the scale to the default value.

## V+ Programs and Variables

All V+ programs and global variables are displayed in the Workspace Explorer under the controller to which the items belong:



### V+ Global Memory Display

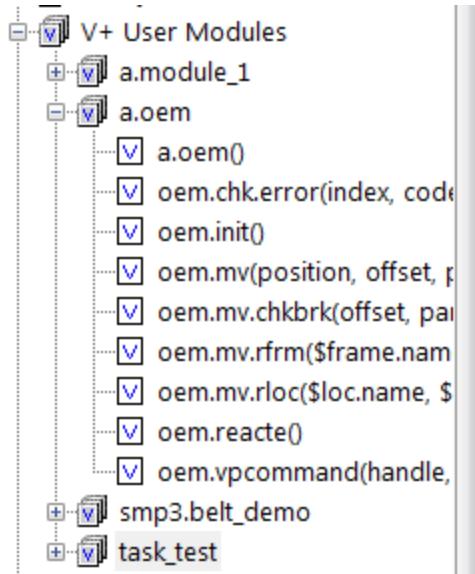
When a workspace is saved, all V+ User modules and V+ User Variables are saved with the workspace.

## V+ System Modules

Any V+ modules that are considered part of the ACE software are listed under the V+ System Modules tab. Most of these modules will be protected. None of these programs are saved with the workspace. Typically, a user will never interact with these programs.

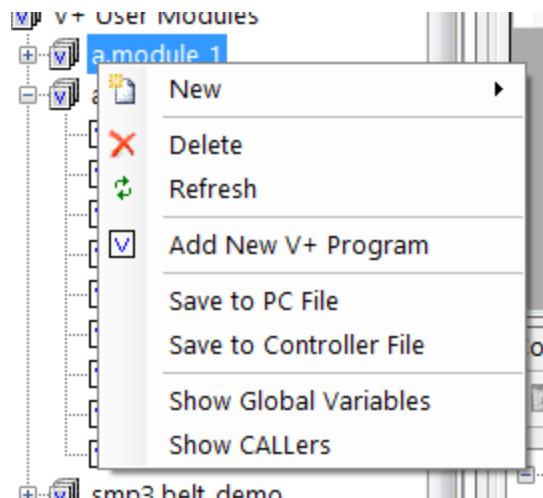
## V+ User Modules

Any V+ programs created as part of an application will be displayed under the V+ User Modules section.

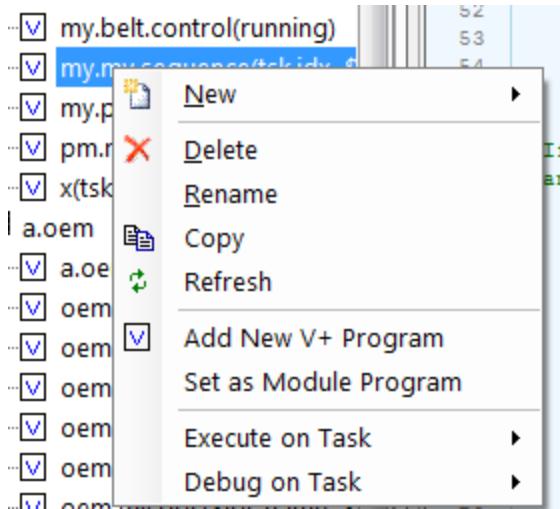


V+ User Modules

Users can double-click on a program to open the V+ program editor. See See "V+ Editor Tool" for more information. Many other common operations can be performed using the context menu when the module or program is selected:



V+ User Modules Context Menu

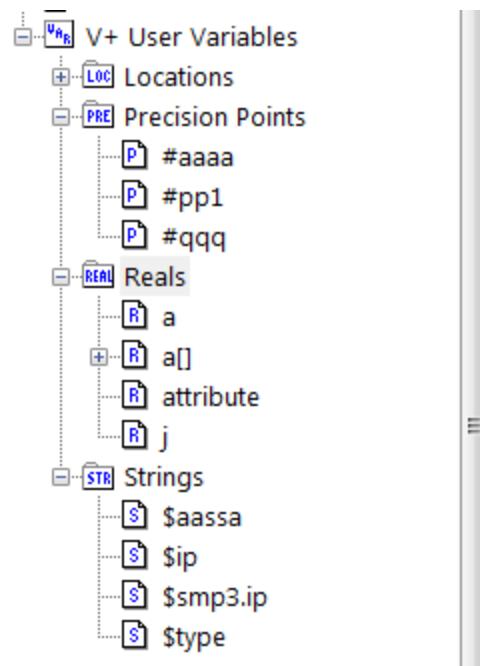


V+ User Program Context Menu

Option	Description
Add New V+ Program	Create a new V+ program
Save to PC File	Save the V+ module to a file on the PC
Save to Controller File	Save the V+ module to a file on the Controller
Show Global Variables	Show all V+ global variables referenced by the V+ program or module
Show CALLers	Show all programs that call V+ programs within the module.
Set as Module Program	Assigns the V+ program as the module program
Execute on Task	Executes the program on the selected task
Debug on Task	Primes the program and puts it in debug mode on the selected task

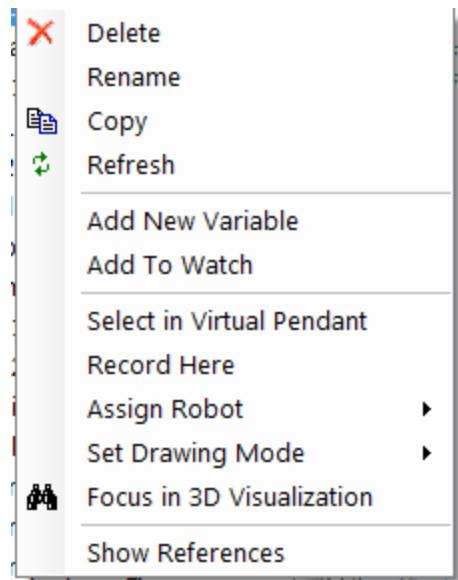
## V+ User Variables

Any V+ global variables created as part of an application will be displayed under the V+ User Variables section. The variables are organized based on type:



#### *V+ User Variables*

Users can double-click on a variable to edit the properties of a variable. Many other common operations can be performed using the context menu when the variable is selected:



*V+ User Variables Context Menu*

<b>Option</b>	<b>Description</b>
Add New Variable	Add a new variable
Add to Watch	Add the variable to the Watch Window. See "Watch Variable Tool"
Select in Virtual Pendant	Only displayed for Location or Precision Point Variables, selects the variable in the virtual pendant. See "Robot Jog Control"
Record Here	Only displayed for Location or Precision Point Variables, records the current location of the selected variable.
Assign Robot	Only displayed for Location or Precision Point Variables, assigns a robot to the variable.
Set Drawing Mode	Only displayed for Location or Precision Point Variables, identifies if the variable position is displayed in the 3D virtual display.
Focus in 3D Visualization	Only displayed for Location or Precision Point Variables, moves the camera of the 3D virtual display to focus on the location or precision point.
Show References	Shows all V+ programs that reference the V+ variable.

## V+ Editor Tool

The ACE V+ Editor is an online, interactive editor. The editor performs syntax checking and formatting while you are programming. You can open up as many editor sessions as needed. These can be arranged as a group of tabs, or split into horizontal or vertical windows, which allows you to compare two or more routines. For information on debugging V+ programs, see the V+ Debugger Tool on page 769.

**NOTE:** When ACE is connected to a controller and you open a program, the program will be read from V+ memory. If V+ memory and the editor ever get out of sync, you can simply close the editor and re-open it, and the content will be automatically updated.

To access the V+ Editor Tool, double click on a V+ program in the Workspace Explorer under the controller to which the program belongs.

**NOTE:**

1. If a program is not executable, ACE software will display the problem (for example, "Ambiguous AUTO invalid (-477)" or "Control structure error (-472)") when it the program is loaded into the V+ Editor.
2. You are not permitted to open/edit protected V+ programs. If you attempt to open a protected V+ program, the following message is displayed:



## Editor Tool

**NOTE:** You can set the font (type face and size) for the editor in the ACE options dialog. For details, see System Options on page 104.

When the Editor tool opens, the module and programs are listed on the left and the selected program is displayed in the editor on the right, as shown in the following figure.

The screenshot shows the V+ Editor Tool interface. The window title is "test2" and the tab is "Object". The toolbar contains various icons for file operations like Open, Save, Print, and a magnifying glass. Below the toolbar is the code editor area. The code is as follows:

```
.PROGRAM test2()
; this is another test

    TYPE "this is another test"

    WHILE TRUE DO
        WHILE condition DO
            Initiates processing of a WHILE structure
            if the condition is TRUE, or skipping of
            the WHILE structure if the condition is
            initially FALSE.
        TYPE "this test has completed"

.END
```

A tooltip is displayed over the "WHILE condition DO" line, explaining its function. The status bar at the bottom shows "Ln: 9 of 14 Col: 39" and "INS".

## V+ Editor Tool

**NOTE:** As shown in the previous figure, a V+ program must begin with a ".PROGRAM" instruction and end with a ".END" instruction. Therefore, these lines cannot be commented out. For more details on V+ program format, see the [V+ Language User's Guide](#).

1. These icons are used for program access and editing functions. A description of each icon is provided below.

**NOTE:** The debug icons, which are shown as dimmed in the previous figure, are used when the

program is in debug mode. For details, see V+ Debugger Tool on page 769.

Icon	Description
	Deletes the selected text and moves it to the Windows clipboard
	Copies the selected text to the Windows clipboard
	Pastes the contents of the Windows clipboard at the cursor position
	Undoes the last action performed
	Redoes the last undo action
	Toggles (sets or clears) a breakpoint at the current line
	Clears all breakpoints in the current program

2. The program display and editing area. Features of this area include:

Feature	Description
Copy / Paste	Allows you to copy and paste ASCII text from other programs or other sources (such as Microsoft WordPad).
Auto Complete	When you type the significant letters of a V+ keyword and then press Enter, the editor attempts to complete the keyword. <ul style="list-style-type: none"> <li>• If the keyword is successfully completed, it is displayed in blue text.</li> <li>• If there is an error, it is displayed in red text.</li> </ul>
Tool Tip syntax	If the you hover the mouse cursor over a keyword, the syntax and short description for that keyword is displayed in a tool tip (a pop-up message that displays next to the cursor).
Formatting and Syntax checking	As each line of program is entered, it is processed by ACE, which performs the formatting and checking, reports back the resulting format, and the editor is updated to reflect this. If there is a problem with the entry, the text is displayed in red. The bottom right status panel displays the current error. Additionally, you can hover over the red text to display a status message. This usually provides information on what the problem was with the entry.
Drag and Drop	The editor supports "drag and drop" of ASCII text from another Windows program onto an open area of the editor. You can also use this feature to copy and

Feature	Description
	move lines of code within the editor, as follows: <ul style="list-style-type: none"><li>• Copy: select text, drag cursor to new position and release</li><li>• Move: select text, hold Shift key while dragging cursor to new position, and release</li></ul>
Colorization	The code lines are colored to help visually distinguish between actual code, comments, and errors. The color are assigned as follows: <ul style="list-style-type: none"><li>• Green: comments.</li><li>• Blue: V+ code.</li><li>• Red: an error</li></ul>

Additionally, you can right-click in this area to display a shortcut menu with the following choices:

Item	Description
<b>Cut</b>	Delete the selected text and store a copy in the Windows clipboard.
<b>Copy</b>	Copy the selected text to the Windows clipboard.
<b>Paste</b>	Paste the contents of the Windows clipboard at the current cursor position.
<b>Edit</b>	Open the selected program or V+ global variable editor.
<b>Keyword Help</b>	Display help information for the selected keyword.
<b>Run to Cursor</b>	Execute the program up to the current cursor position.
<b>Comment Line</b>	Add a ";" character to the beginning of the line.
<b>Uncomment Line</b>	Remove the ";" character from the beginning of the line.

3. Displays the line number being edited and, if the cursor is on a keyword, the syntax of that keyword.
-

## V+ Debugger Tool

The ACE V+ Debugger is an online, interactive debugger. The debugger allows interactive program stepping while simultaneously displaying code variables and states. If a program in one module steps into a program in another module, the debugger will automatically step you into that program. Break points in the code can be added or removed while debugging. You can have as many active debugging sessions as there are tasks. This feature allows you to debug multiple tasks. For information on editing V+ programs, see [V+ Editor Tool on page 765](#).

**NOTE:** When ACE is connected to a controller and you open a program, the program will be read from V+ memory. If V+ memory and the editor ever get out of sync, you can simply close the editor and re-open it, and the content will be automatically updated.

To open a program in the V+ Debugger Tool, right-click on the program in the Task Manger window and select **Debug Task**. For details on the Task Manager window, see [V+ Task Manager on page 772](#).

A screenshot of the V+ Debugger Tool interface. The window title is "test2". The code editor displays the following V+ assembly-like code:

```
.PROGRAM test2()
; this is another test

    TYPE "this is another test"

    WHILE TRUE DO
        WHILE condition DO
            Initiates processing of a WHILE structure
            if the condition is TRUE, or skipping of
            the WHILE structure if the condition is
            initially FALSE.
        TYPE "this test has completed"

    END
```

The cursor is positioned at the start of the second line of the code. A tooltip box appears over the "WHILE condition DO" line, containing the following text:

WHILE condition DO  
Initiates processing of a WHILE structure  
if the condition is TRUE, or skipping of  
the WHILE structure if the condition is  
initially FALSE.

The status bar at the bottom shows "Ln: 4 of 14 Col: 36" and "INS".

### Tabs

The tabs are used to select a program for debugging. To add a program (tab), simply double-click the program name in the left pane (V+ associated programs pane).

### Toolbar Icons

The toolbar icons are used for inserting program control structures and statements, and for controlling program execution.

Icon	Description
	Deletes the selected text and moves it to the Windows clipboard.
	Copies the selected text to the Windows clipboard.
	Pastes the contents of the Windows clipboard at the cursor position.
	Undoes the last action performed.
	Redoes the last undo action.
	Toggles (sets or clears) a breakpoint at the current line.
	Clears all breakpoints in the current program.
	Executes the next step of the program (single step through the program). When the cursor reaches a subroutine call, the debugger steps into the subroutine. Identical to Step Into (F11) described in the following table.
	Single step through the program and step over the next CALL statement. Identical to Step Over (F10) described in the following table.
	Moves the execution pointer to the current cursor position.
	Retries failed step and continue execution.
	Continues program execution.

### Code Editor

The code editor displays the program currently being debugged. It includes the following features:

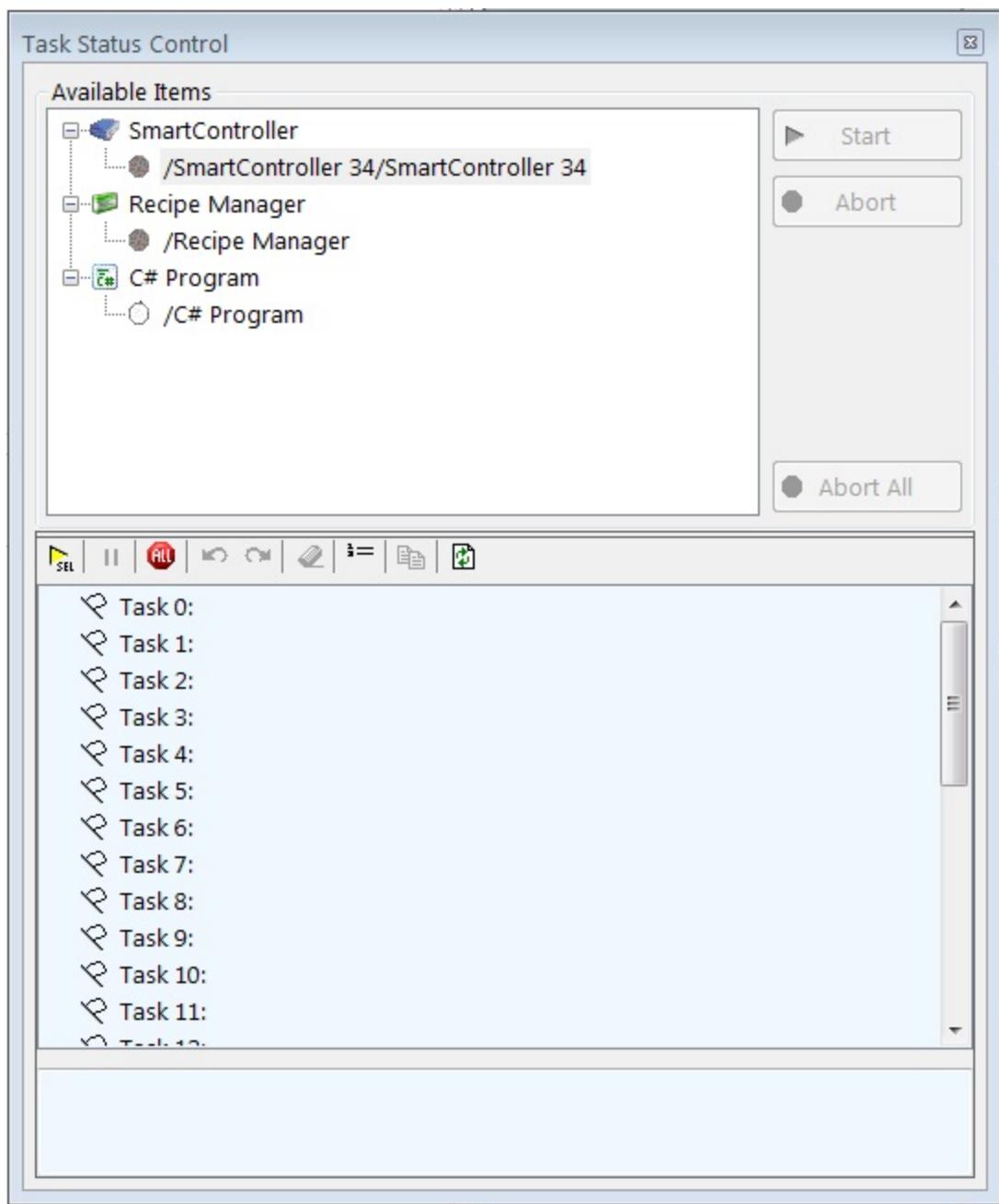
Feature	Description
Toggle Break Point (F9)	Turns on/off a break point at the current cursor line. The program will automatically pause when it reaches that point.
Step Over (F10)	Single step operation that skips a subroutine call. When the execution pointer is positioned at a CALL or CALLS instruction, typing F10 will cause the entire subroutine to be executed, and execution pauses at the step following the subroutine call. (F10 behaves exactly as F11 does when the current instruction is not a subroutine call.)
Step Into (F11)	Single step operation that will enter a subroutine call and single-step through the subroutine. After the last line of the subroutine has been executed, it returns to the step following the subroutine call.
Goto (Ctrl+G)	Moves the execution pointer to the specified line number.
Pause	Pauses the program task execution.
Retry	Retries the current line.
Proceed (F5)	Continues execution of the task until the next break point or the program terminates.
Tool Tips	Displays brief help for the item, as follows: <ul style="list-style-type: none"><li>If you hover the mouse pointer over a V+ keyword, the debugger displays the syntax and short description for that keyword.</li><li>If you hover the mouse pointer over a variable (while the task is paused) the debugger displays the variable value in a tool tip.</li></ul>

## The Program Variable Shortcut Menu

When you right-click on a V+ variable (or expression), a shortcut menu is displayed. There is an additional item, called "Add to Watch", which allows you to add the current variable to the [Watch Variable](#) list.

## **V+ Task Manager**

The Task Manager is used to display a list of the tasks that are currently running on the controller, and to start, stop, pause, and clear a selected task. The task manager can be accessed through the ACE Task Status Control.



Task Manager Window

## Toolbar

Use the toolbar icons to execute, pause, retry, etc., the selected program/module. Each icon is described below:

Icon	Description
	Executes the selected task. If a task is selected in the tree view pane this button will execute that task. If no task is selected, this button is dim. If a task is selected and it is not primed, the normal play button will change to . When you click this, you will be prompted for the program name to execute in the selected task. This makes it easier for you to launch a program if they choose not to drag and drop a program from the Program Manager.
	Pauses execution. The selected task execution is paused at the next instruction.
	Stops the execution of all running tasks.
	Retries the failed step and continues execution. If the selected task was paused or stopped due to an error, this button attempts to re-execute the current step and continue execution.
	Continues execution. If the selected task was paused or stopped due to an error, this button attempts to proceed execution of the task. This button is dimmed if there is no program for the given task or no task selected.
	Clears the selected task of any programs. A program must be cleared from the stack before it can be fully edited. Note that AUTO variables or calling arguments cannot be changed while a program is in a task stack.
	Toggles the display between occupied or all tasks
	Copies the contents of the selected task stack to the Windows clipboard. If a program terminates with an error, this allows you to copy and paste the stack contents and send it to the proper support person. Note that the robot ID is also recorded in this operation.
	Refreshes the contents of the window.

Note that you can also right-click in the task list area (described below) to display a shortcut menu that matches the functions of the Toolbar buttons described above. The menu contains the following items:

## Task List Area

---

Item	Description
Execute Task	Executes the selected task.
Execute Using ...	Prompts for the name of the program for executing the selected task (instead of drag and dropping).
Debug Task	Opens the Program Debugger for the selected task.
Debug Using ...	Prompts for the name of a program. Primes the specified program and opens the Program Debugger for the specified program.
Reset and Debug	Resets the program and open the Program Debugger for the selected task.
Pause	Pauses execution. The selected tasks execution is paused at the next instruction.
Stop All Tasks	Stops the execution of all running tasks.
Retry Failed Step	Retries the failed step and continues executing the task. If the selected task was paused or stopped due to an error, attempts to re-execute the current step and continue execution.
Proceed	Continues execution of a task. If the selected task was paused or stopped due to an error, this button attempts to continue the execution of the task.
Kill Task	Clear the selected task of any programs. A program must be cleared from the stack before it can be fully edited. Note that AUTO variables or calling arguments cannot be changed while a program is in a task stack.
Copy Stack to Clipboard	Copies the contents of the selected task stack to the Windows clipboard. If a program terminates with an error, this allows you to copy and paste the stack contents and send it to the proper support person. Note that the robot ID is also recorded in this operation.
Help	Opens the online documentation.

### Task List Area

The task list area shows the occupied tasks. Double-click any program in a task to load it into the editor. A node may be expanded, if there are programs on the stack to support this. Each called program from within a parent routine is an additional item under the node. The possible conditions for the flag icon are:

---

## Task Status Area

---

Icon	Description
	Task is idle or primed
	Task is executing
	Execution paused or at breakpoint
	Execution error or program execution was manually aborted
	Execution has completed

### Task Status Area

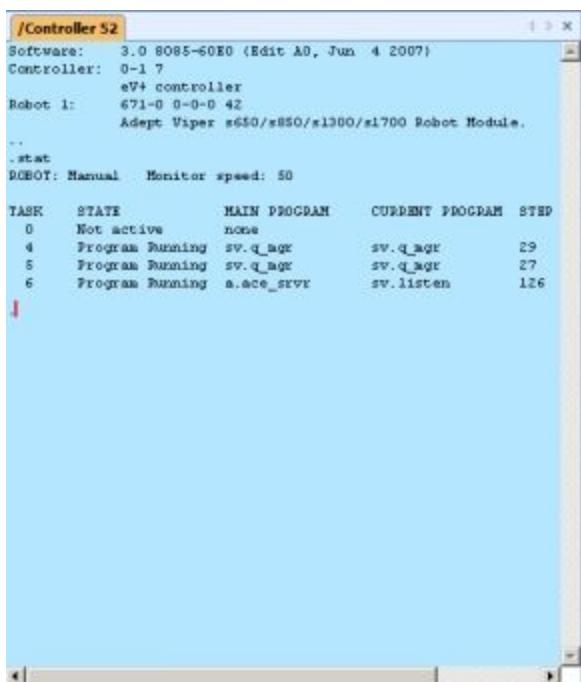
The task status area displays the status of the selected task. Highlighting a node displays the last state of the task. Double-clicking a node with a primed program will open up the debugger with that program name.

## Monitor Window

The Monitor Window is used to input V+ monitor commands to the corresponding controller. Additionally, user program output from the TYPE keyword is displayed within this window.

To open the Monitor window, click the Monitor window icon (  ) on the ACE toolbar.

**NOTE:** To run controller utility programs, use the [Configuration Tools](#).



Monitor Window

## Executing a V+ Monitor Command

To execute a V+ monitor command, type the command at the dot prompt. The results of the command are displayed in the same window. For more information on V+ monitor commands, see the *V+ Operating System User's Guide* and *V+ Operating System Reference Guide*.

**NOTE:** You cannot abort any ACE server tasks from the Monitor Window.

## V+ Startup Messages

During system startup, any V+ startup messages are displayed in the Monitor Window. For example, the following figure shows a V+ configuration error displayed in the Monitor Window.

```
Adept V+  
Copyright (c) 1984-2006 by Adept Technology, Inc.  
*Invalid software configuration*  
/POS_LATCH 1002  
  
Adept External Encoder Module.  
Adept Cobra s350/s600/s800 Robot Module.  
  
Configured as Category 3 Robot System per ISO 102  
  
Software: 16.3 87-7100 (Edit B8, 27-Feb-2006,  
Controller: 2000-2760 105 0  
Security ID: A405-9978-73B8  
Processor 1: 0.52 7-5 32MB  
Robot 1: 580-126 0-41-0 6  
          Adept Cobra s350/s600/s800 Robot Mod  
Servo: 16.0 (Edit A1)  
  
05-Apr-2008 17:03:41  
.
```

#### *Startup Messages*

### **Opening a V+ Program**

To open an existing V+ program, at the dot prompt, type the "see" command followed by the program name. The program opens in the ACE program editor.

**NOTE:** The program does not need to be part of the current ACE workspace.

For example, assume the program "myprogram" exists. When you type the command:

```
.see myprogram
```

the specified program opens in the V+ Editor tool. For more details, see V+ Editor Tool on page 765.

### **Creating a New V+ Program**

To create a new V+ program, at the dot prompt, type the "see" command followed by the new program name. The New Program dialog opens and, after confirmation, the program opens in the ACE program editor.

For example, assume the program "mynewprogram" does not exist. When you type the command:

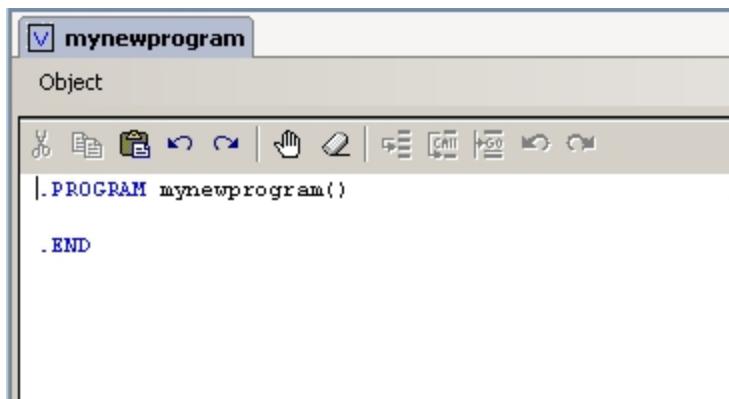
```
.see mynewprogram
```

the specified program is entered into the New Program dialog, as follows.



*New Program Dialog*

You must confirm the program and module name, and then click **OK**. The new program opens in the V+ Editor tool.

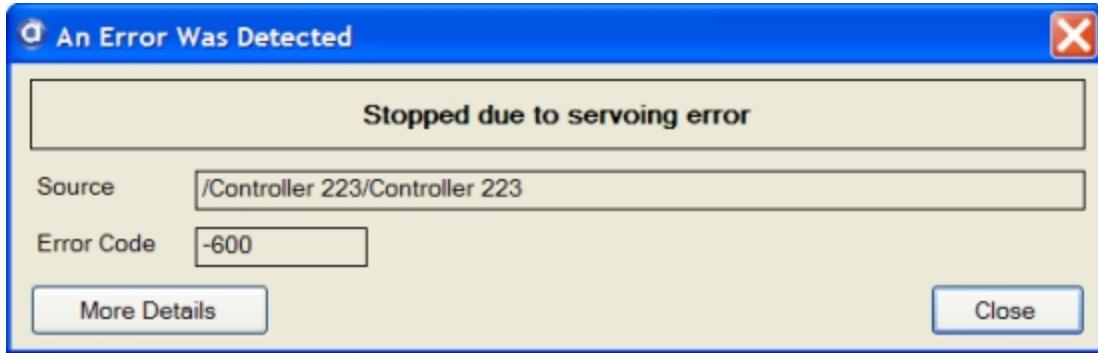


*V+ Editor Tool*

For more details, see V+ Editor Tool on page 765.

## Error Messages

The Error was Detected dialog provides information on any ACE or V+ errors that you may encounter when programming or operating your ACE system, as shown in the following figure.



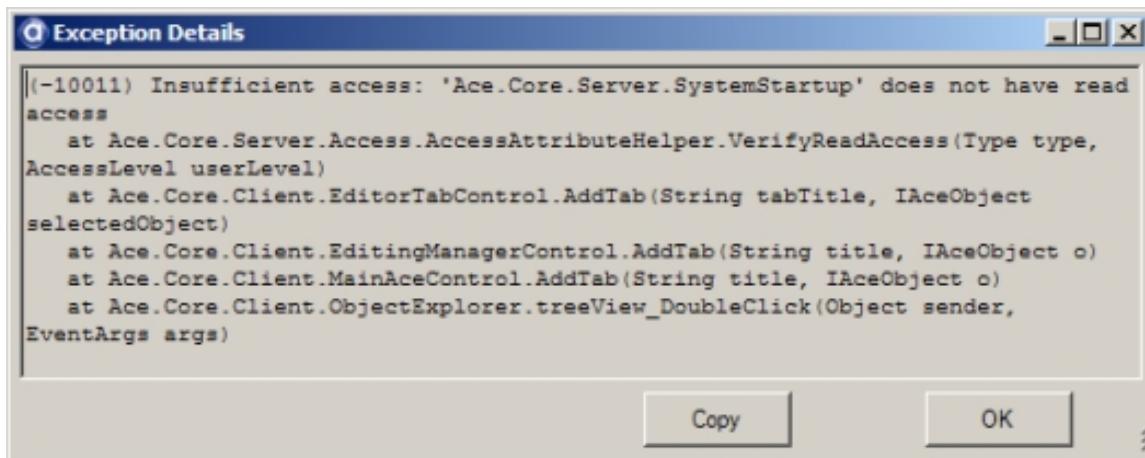
*Error was Detected Dialog*

The dialog displays the specific error message text associated with the error.

The Error Code field displays the error code number for the error, if a code number is available. All V+ errors will have a specific error code number associated with the error.

The Source field provides the information on the cause of the error. For example, the source could be a controller in the system, or a general system error.

The More Details button is used to display the ACE details of the message, which can be copied and sent to support personnel, if needed.



*Error Details*

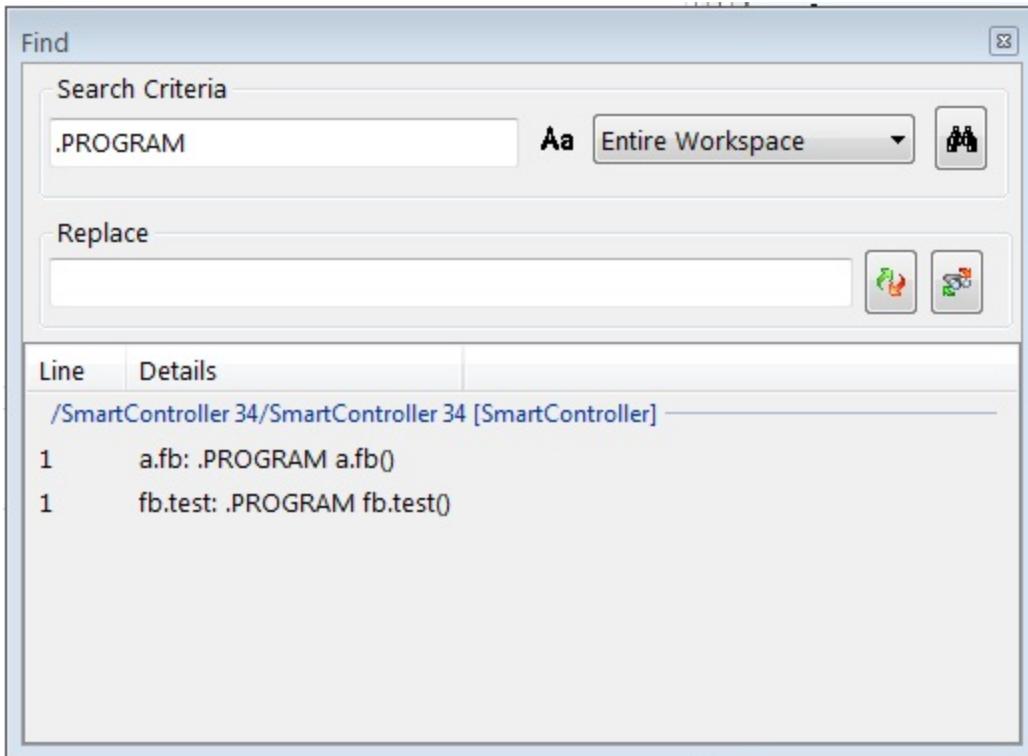
**NOTE:** For more details on V+ error messages, see the section titled System Messages in the [\*V+ Language Reference Guide\*](#).

## Find Dialog

The ACE software contains a Find dialog, which is useful for finding (and replacing) character strings in a text file or program within your workspace. This dialog is a global (workspace-wide) search tool.

The Find dialog can be accessed from the ACE toolbar, as follows:

- Click the Search ( ) icon on the toolbar.
- OR
- Press Ctrl+F while viewing the ACE workspace.



*Find Dialog*

This dialog allows you to find (and replace) a string in a program or text file located in your ACE workspace.

## Finding a String

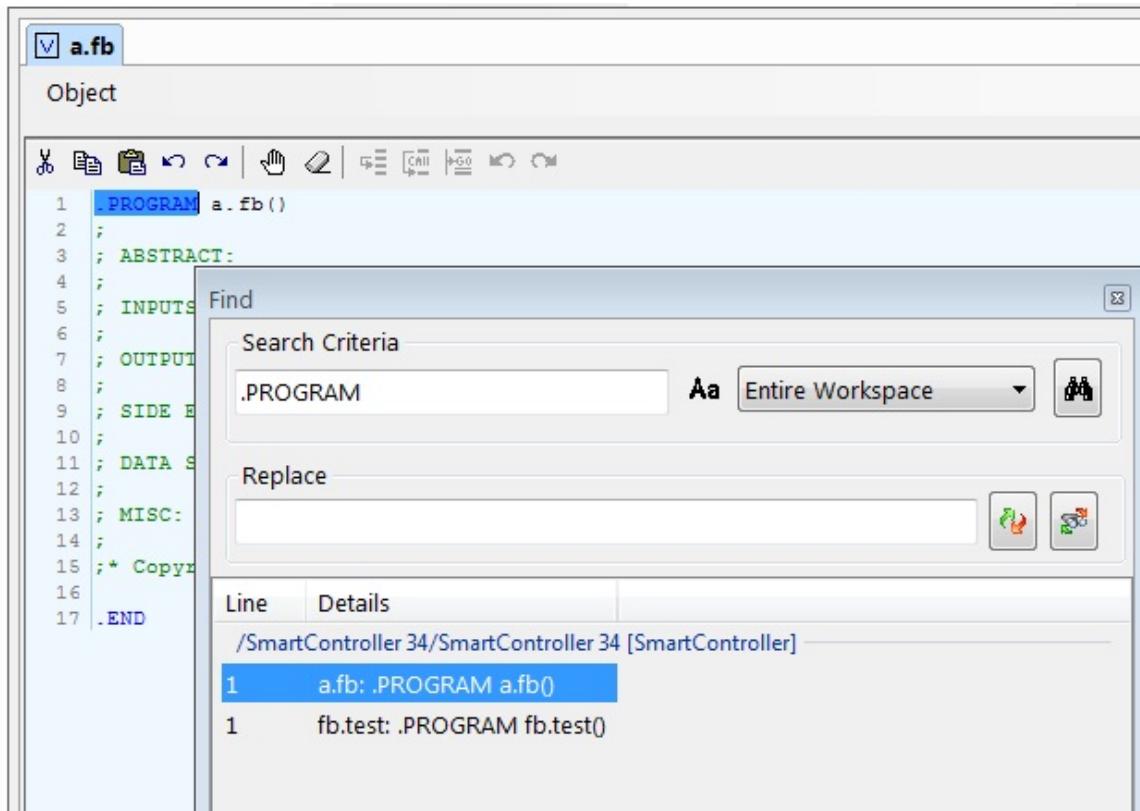
To find a string:

1. Type the desired string in the Search Criteria field.

Optionally, *before opening the Find dialog*, you can highlight the desired string in the C# Script Editor or the V+ Program Editor and then open the Find dialog. The highlighted string will be inserted in the Search Criteria field. This feature ensures that you capture the desired string correctly.

2. Use the drop-down list box to select from: Current Editor, All Open Editors, Entire Workspace, or All Content(default).
3. Optionally, select the Match Case option, if you want the search to be case-sensitive.
4. Click the Search icon (  ) to begin the find operation.

If matching strings are located, the file and string are displayed in the list area at the lower part of the dialog box. For example, the following figure shows the results of the find operation after searching for some test text that was found in a V+ program editor.



*Find Dialog with Results List*

## Replacing a String

To find and replace a string:

1. Type the desired string in the Search Criteria field.
2. Use the drop-down list box to select from: Current Editor, All Open Editors, or Entire Workspace (default).
3. Type the desired replacement string in the Replace field.
4. Click the Search icon ( ) to begin the find/replace operation.

If matching strings are located, the file and string are displayed in the list area at the lower part of the dialog box. For example, the previous figure shows the results of the find operation after searching for some test text that was typed into the C# Program editor tool.

5. Click the Replace Selected ( ) icon to replace only the selected string, or click the Replace All ( ) icon to replace all found strings.

## OPC Data Access and Process Control

**NOTE:** The OPC communications feature requires the USB hardware key (dongle) containing the OPC license. See Licensing Requirements on page 28 for details.

OPC stands for OLE for Process Control. It uses Microsoft's COM and DCOM technology to enable applications to exchange data on one or more computers using a client/server architecture.

OPC defines a common set of interfaces, so applications can retrieve data in exactly the same format regardless of whether the data source is a PLC, DCS, gauge, analyzer, software application or anything else. The data can be available through different connections, such as serial, Ethernet, or radio transmissions. Different operating systems, such as Windows, UNIX, DOS, and VMS are also used by many process control applications.

The purpose of OPC is to provide a standards-based infrastructure for the exchange of process control data that accommodates all of these different data sources, connections, and operating systems.

The OPC protocol consists of many separate specifications. OPC Data Access (DA) provides access to real-time process data. Using OPC DA you can ask an OPC server for the most recent values of anything that is being measured, such as flows, pressures, levels, temperatures, densities, and more. OPC support in ACE OPC is limited to the DA specification.

For more information on OPC, please see the OPC Foundation website at the following URL:

<http://www.opcfoundation.org>

## Installing and Enabling OPC

In ACE, OPC is used to:

- communicate the values of specified V+ global variables.
- control (remotely start or stop) an ACE application.
- control (remotely start or stop) an ACE process.

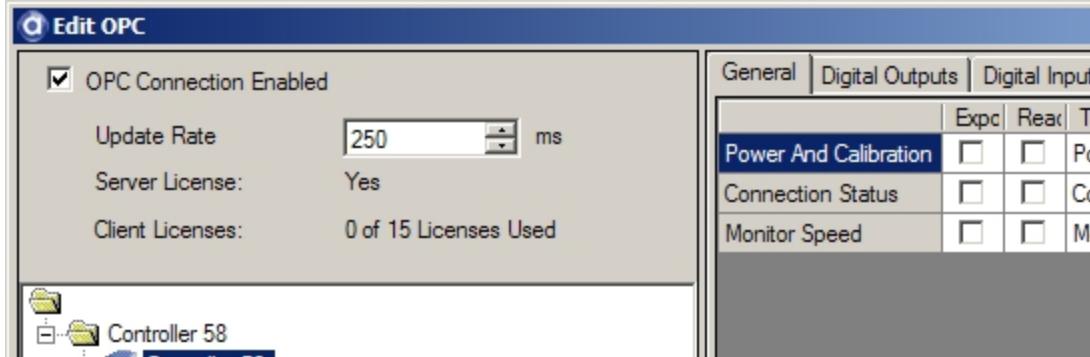
To enable the OPC Data Access and Process Control feature:

1. The OPC option license must be available on the USB hardware key (dongle). See PC Licenses on page 29 for details. When the OPC license is available, the Edit OPC window displays the number of licenses available and in use, as shown in the following figure.

If the OPC license is not detected, the interface:

- disables the Edit OPC window
  - changes the Workspace Explorer status bar OPC icon to "not connected" (when clicked on, the text indicates the license is not enabled)
2. Select the OPC Connection Enabled option in the Edit OPC window. This enables the ACE OPC server.

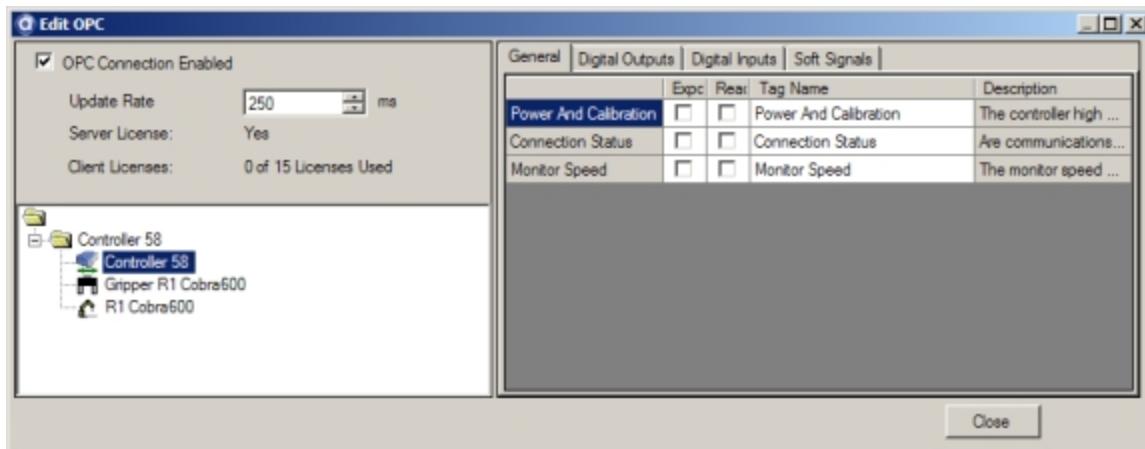
To open the Edit OPC window, select **View > Edit OPC** from the ACE main menu.



3. Setup the OPC feature you want to use in your ACE application:
  - To communicate the value of a variable, see [Communicating Variable Values](#) on page 787.
  - To control an application, see [Controlling Applications](#) on page 790.
  - To control a process, see [Controlling a Process](#) on page 793.
4. If the OPC Client is operating on a different PC than the Ace Server, the PC account and security settings for both PCs must be configured to allow OPC communications. For details, see [Overview](#) on page 1137.

## Exposing Object Data Elements

The Edit OPC tool is used to enable an OPC connection (for details, see [Installing and Enabling OPC](#) on page 785) and expose elements of objects in the ACE workspace. To open the tool, select **View > Edit OPC** from the ACE main menu.



*Edit OPC Window*

The Edit OPC window is divided into two panes: the left-hand pane shows a folder tree structure, which mirrors the tree view in the Workspace Explorer; the right-hand pane shows the tab groups and data items available in each for the selected object.

To expose a data item to OPC:

1. In the left-hand pane, use the tree view to locate and select the object that contains the desired data element.
2. In the right-hand pane, use the tabs to locate the desired data element.
3. Select the Expose check box to expose that element to OPC.
4. Optionally, select the Read Only check box, if you want to make the data element read only (it can be read but cannot be written to).
5. Optionally, rename the Tag Names for the exposed elements. Note the following:
  - Each tag name must be unique.
  - Every item must have a tag name (in other words, the Tag Name field cannot be empty).

## Communicating Variable Values

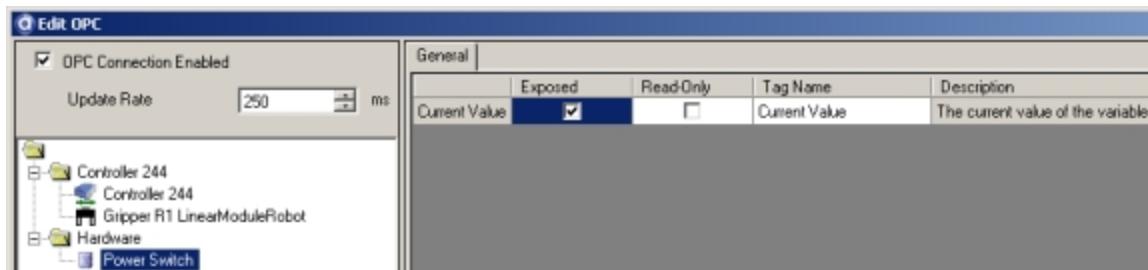
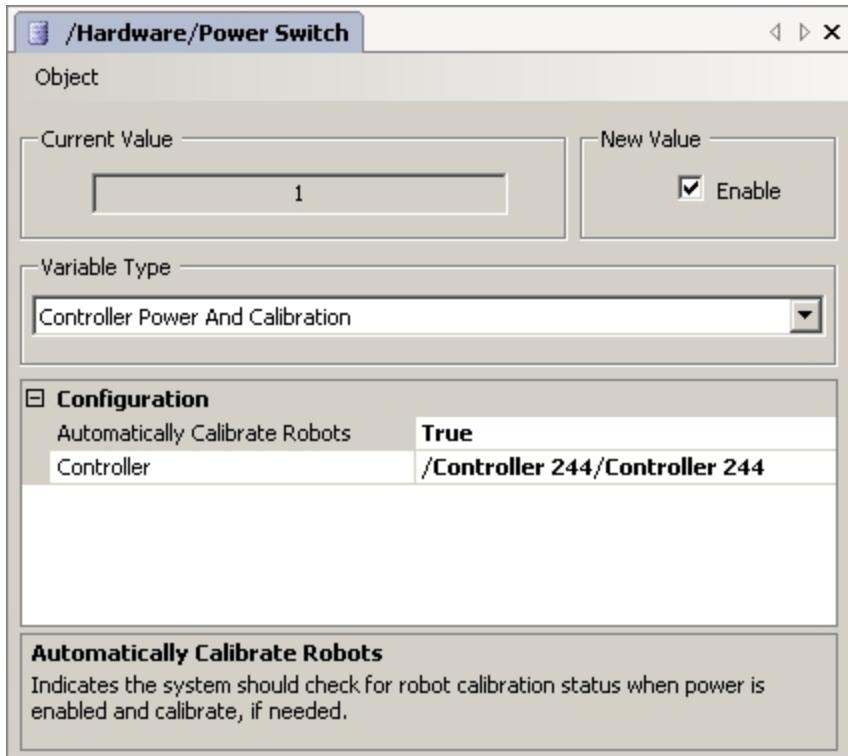
The OPC Data Access and Process Control feature can be used to communicate the values of specified V+ global and local variables.

To enable this feature:

1. Install and enable the OPC Data Access and Process Control feature. For details, see [Installing and Enabling OPC](#) on page 785.
2. Create a variable for the value that you want to export. For details, see [Variable Editor](#) on page 131.
3. Enable OPC Export for that variable. For details, see [Exposing Object Data Elements](#) on page 786.
4. In your OPC client application, create an OPC tool that reads the exported value. See your OPC client documentation for details.

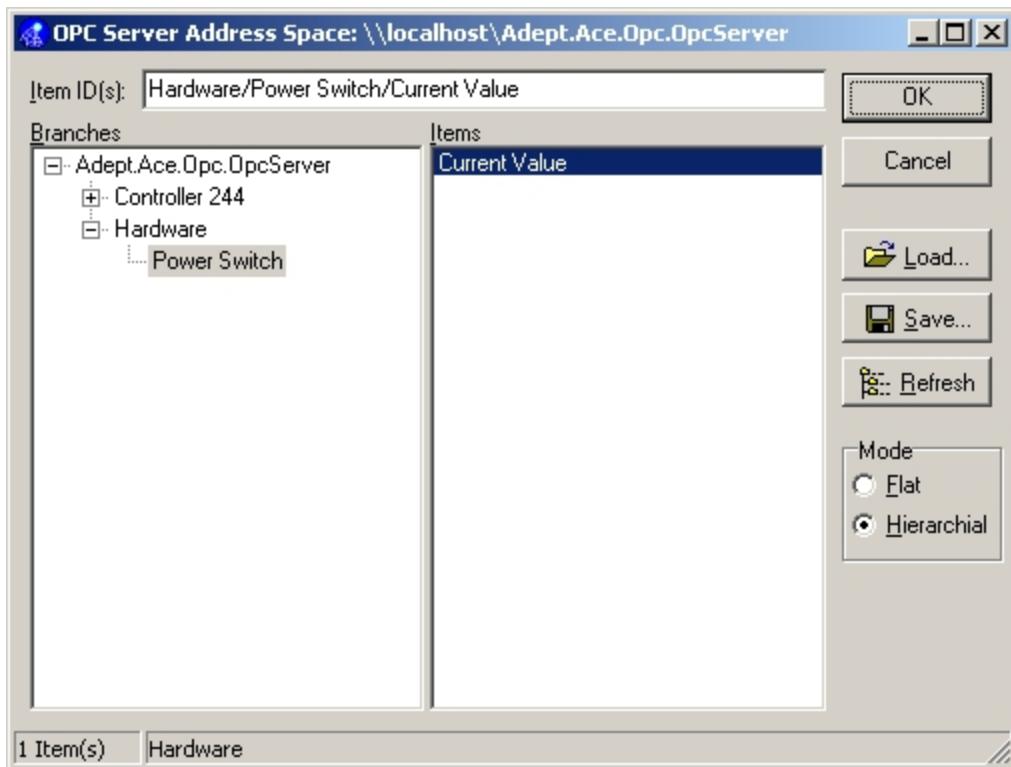
### Variable Export Details

Through the OPC server, a variable is exported to a namespace that matches the name of the variable in ACE.



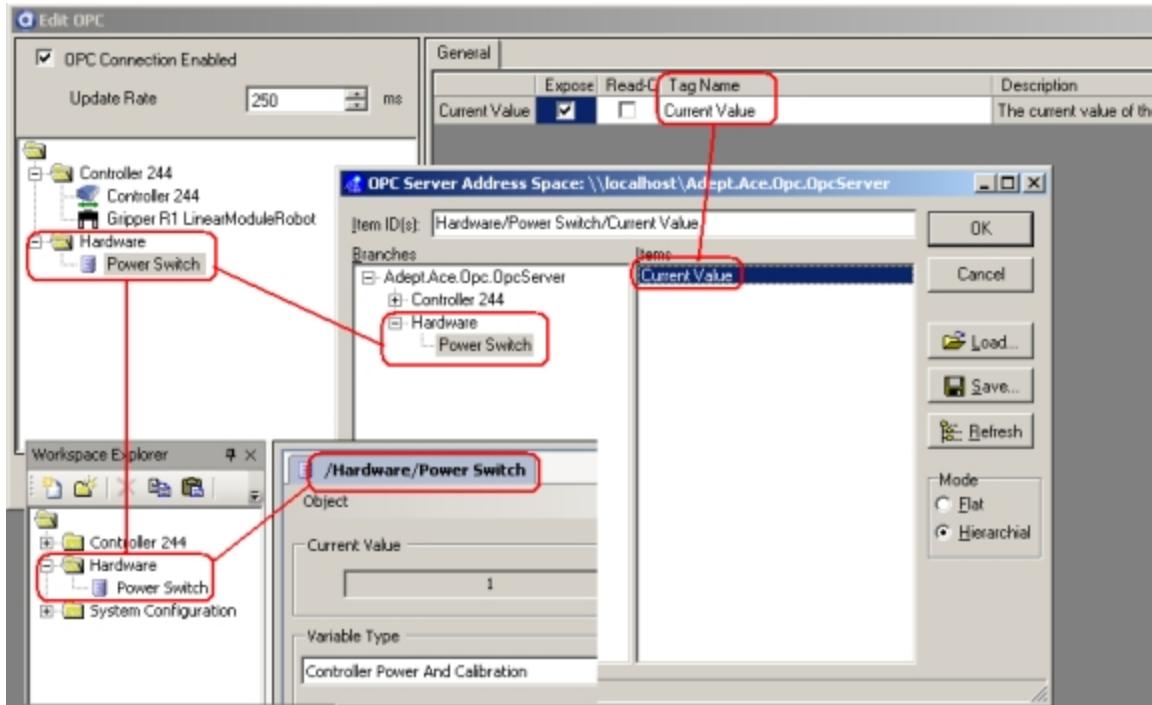
#### *Numeric Variable with OPC Export Enabled*

The following figure shows an OPC client browsing for the variable shown in the previous figure.



*OPC Client Browsing for Variable*

The following figure shows how the namespace in OPC matches the ACE directory structure.



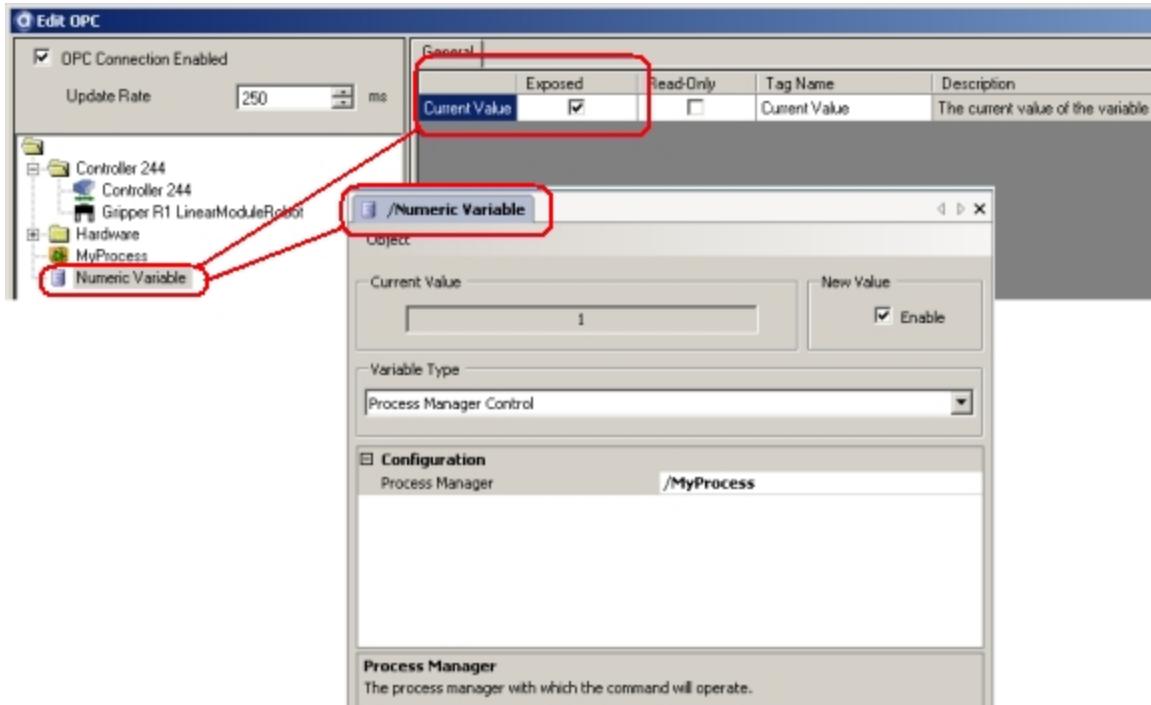
*Folder Structure Matched in Edit OPC Window and in OPC Client*

## Controlling Applications

The OPC Data Access and Process Control feature can be used to control (remotely start or stop) a specified application, such as an ACE PackXpert packaging application.

To enable this feature:

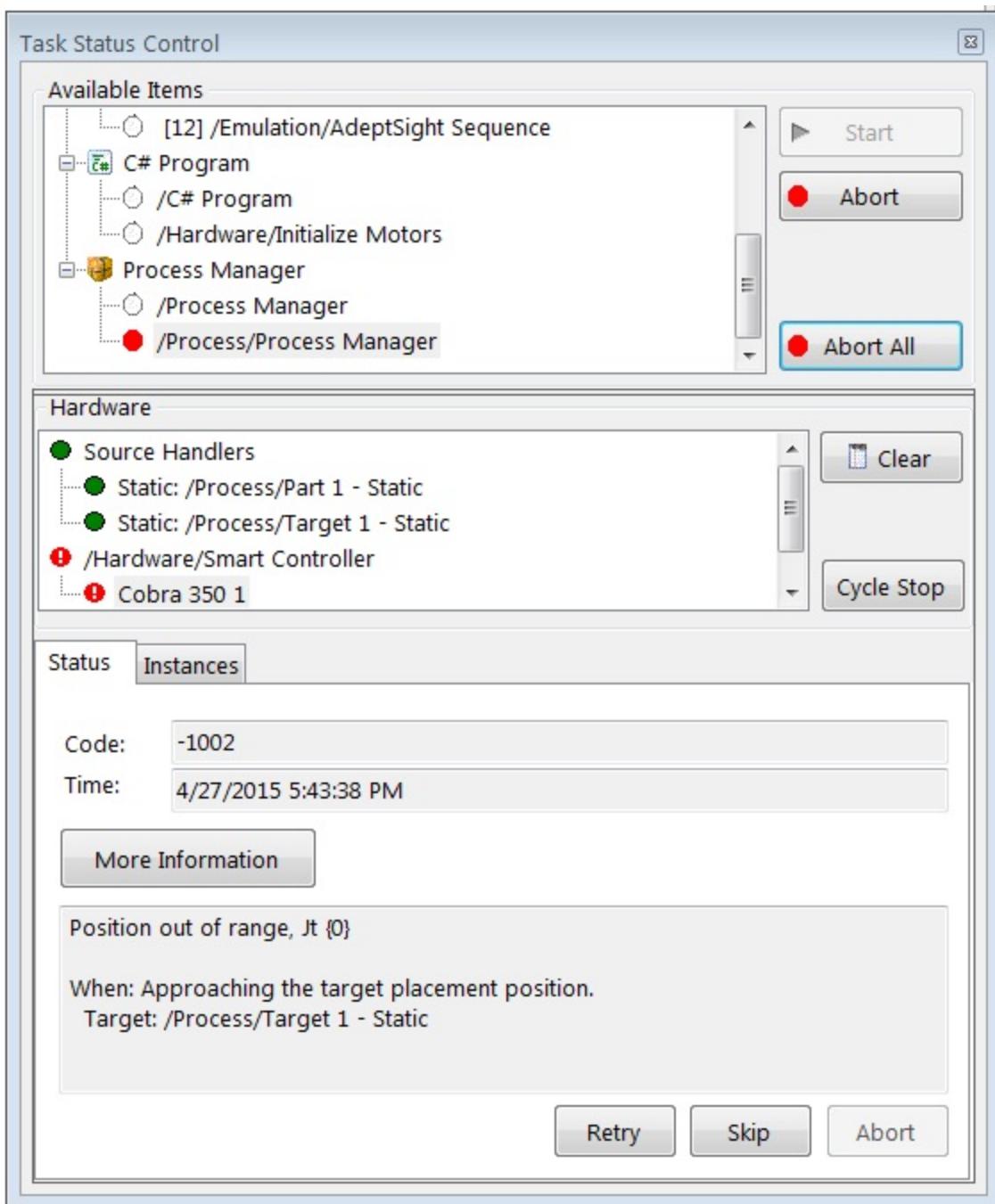
1. Install and enable the OPC Data Access and Process Control feature. For details, see [Installing and Enabling OPC](#) on page 785.
2. Create a Numeric Variable for the Process Manager control. For details, see [Variable Editor](#) on page 131.
3. Enable OPC Export for the Process Manager control. For details, see [Exposing Object Data Elements](#) on page 786.



#### *Process Manager Control Variable Type*

4. Set the Variable Type to Process Manager Control.
5. For the OPC options, select Export, and verify that Read-only is not selected.
6. In your OPC client application, create an OPC tool that writes to the exported value. This will be used to start/stop the specified application. See your OPC client documentation for details.

A Process Manager Control is available, which can also be used to start/stop the application, similar to using an OPC control. For details, see Process Manager Control on page 920.



Process Manager Control

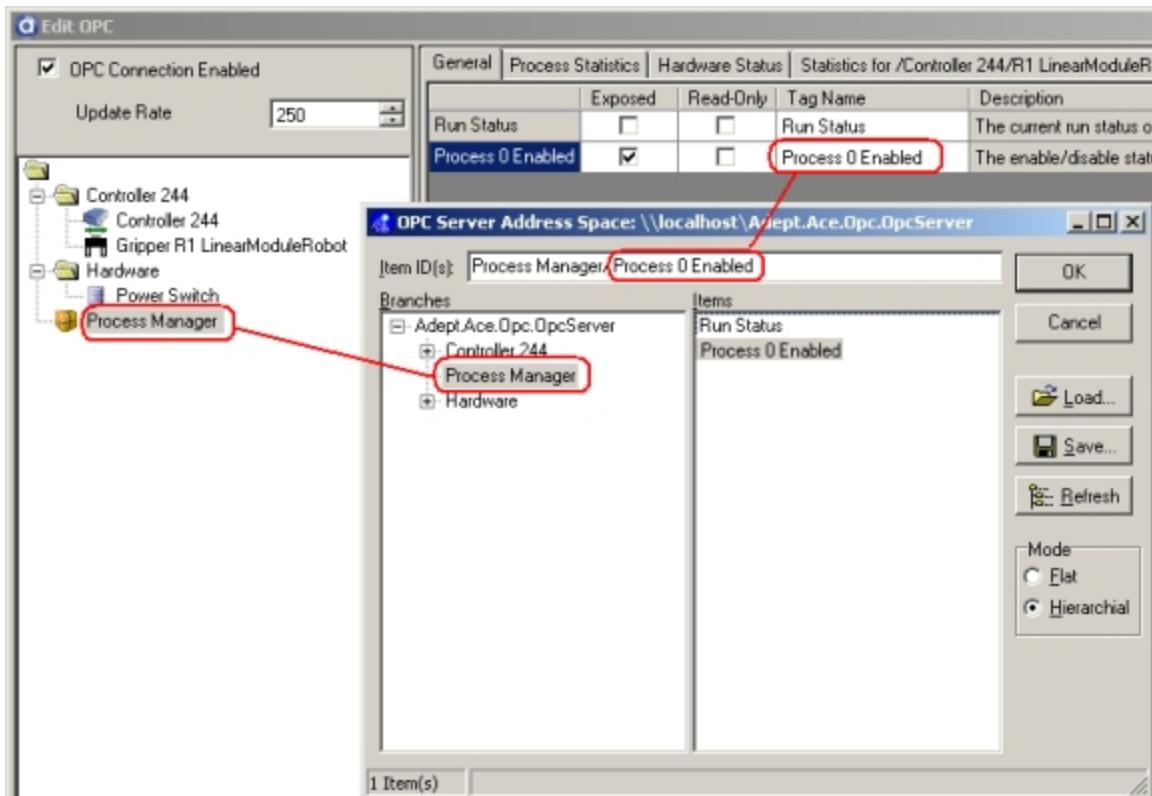
---

## Controlling a Process

For some applications, it may be necessary to stop certain processes (for example, stopping an upstream robot and allowing a downstream robot to clear the belt of all remaining product) before stopping the entire application. The OPC Data Access and Process Control feature can be used to control (remotely start or stop) a specified ACE process.

To enable this feature:

1. Install and enable the OPC Data Access and Process Control feature. For details, see [Installing and Enabling OPC](#) on page 785.
2. For the OPC options, select Export, and verify that Read-only is not selected.



Selecting the Process to Control

**NOTE:** The Process Index value represents a process on the Process Manager editor Processes list. Because ACE, like V+, uses a zero-based index, the first process on the list would be '0', the second would be '1', and so on.

3. In your OPC client application, create an OPC tool that writes to the exported value. This will be used to start/stop the specified process. See your OPC client documentation for details.

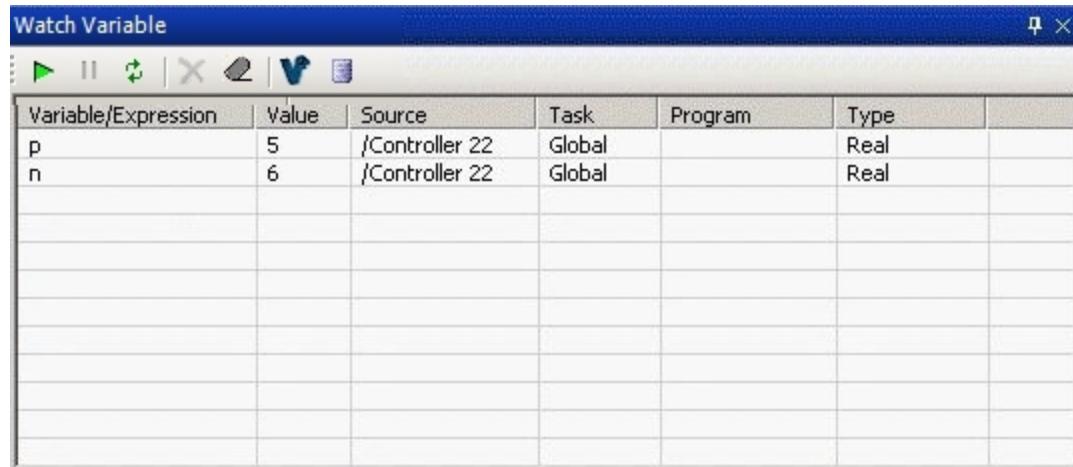
# Watch Variable Tool

The Watch Variable tool is used to monitor specified V+ global and local variables while debugging a V+ program or to monitor ACE variables.

The Watch Variable tool stores the last defined list to the connected controller. Then, when the tool is opened, it retrieves the stored list from the controller and displays it as described below.

To open the Watch Variable tool from the ACE menu, select:

## **View > Watch Variable**



## *Watch Variable Tool*

**NOTE:** The variables on the list are sorted by: Source, Type, then Name. This allows variables from different controllers to be grouped together.

ACE variables can be added to the watch list by dragging and dropping the variable from the Workspace Explorer to the Watch Variable list. V+ variables can be added by:

- right-clicking a variable in the V+ Editor/Debugger tool and selecting "Add to Watch"
  - using the [Variable Browser](#) window

In the Watch Variable tool:

- Use the toolbar icons to select, scan, and delete variables, clear the list, etc.

Each icon is described below:

Icon	Description
	Continuous Scan – updates all displayed variables continuously. The scan rate varies depending on controller type and number of variables displayed.

Icon	Description
	Stop Scan – terminates the continuous scan mode.
	Refreshes the contents of the window.
	Deletes the selected variable from the list.
	Clears the entire watch variables list.
	Opens the Variable Browser window, which is used to select variables for the Watch Variable list.
	Opens the Add Workspace Variable window, which is used to add an existing variable to the Watch Variable list.

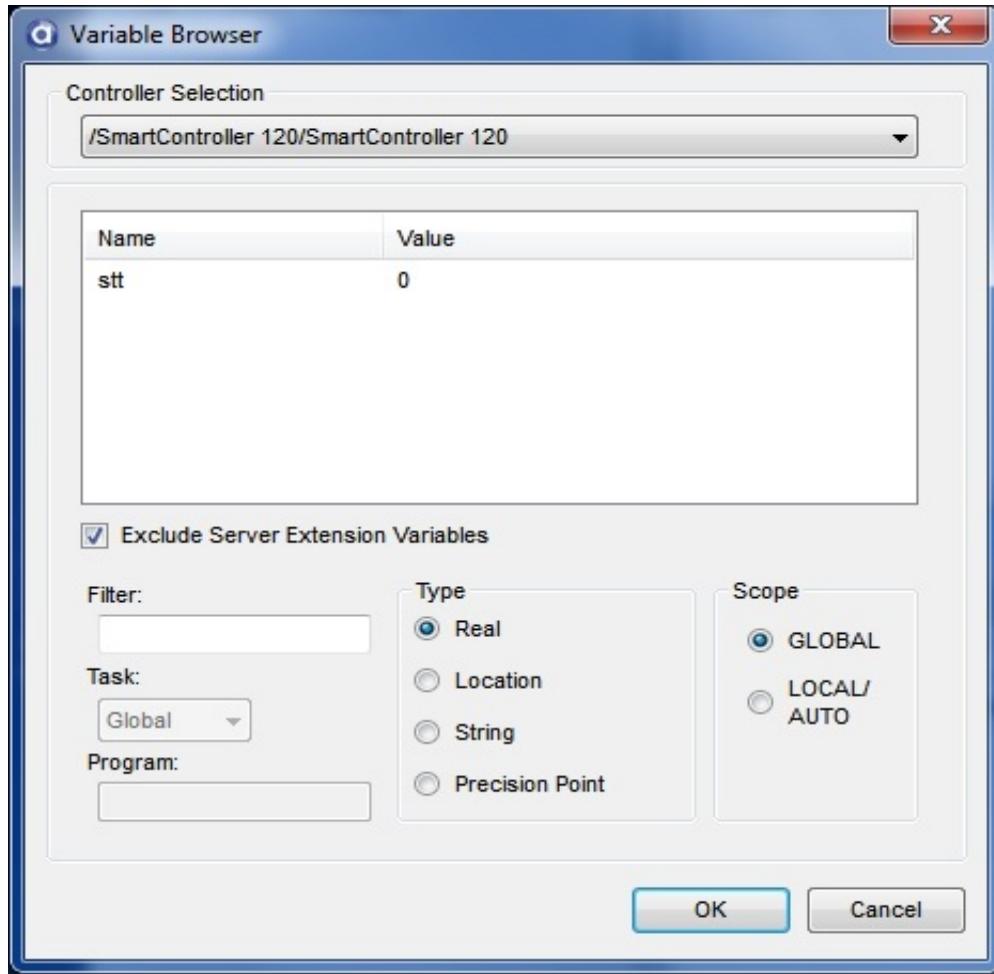
- The grid area displays a list of variables, which are assigned by you, to be watched. The variable, variable value, source, task, V+ program, and variable type are displayed in this list. Expressions are also allowed (i.e., "a\*random").

## Variable Browser Window

The Variable Browser window is used to select V+ variables for the Watch Variable list. To access the Variable Browser window, click the  icon on the Watch Variable toolbar.

## Variable Browser Window

---



*Variable Browser Window*

To add a variable to the Watch Variable list:

1. Controller Selection - if the ACE system is connected to multiple controllers, use this item to select the controller that contains the variables to watch.
2. Select the desired variable from the Name list. The corresponding value of each variable is displayed in the Value list. You can narrow the list using the following items:

Item	Description
Filter	Enter characters into this field to locate matching variable names. For example, type "sv." to list all variable names that begin with those characters, such as "sv.base_task", "sv.cal_state", etc.

## Add Workspace Variable Window

---

Item	Description
Task	Select the task that contains the desired variables.
Program	Specify the program that contains the desired variables.
Type	Select the variable type: Real, Location, String, or Precision Point.
Scope	Select the scope: - Global: displays all global variables - Local/Auto: displays all variables declared as local/auto

3. Click **OK** to add the selected variable to the Watch Variable list, or click **Cancel** to exit without adding the variable to the Watch Variable list.

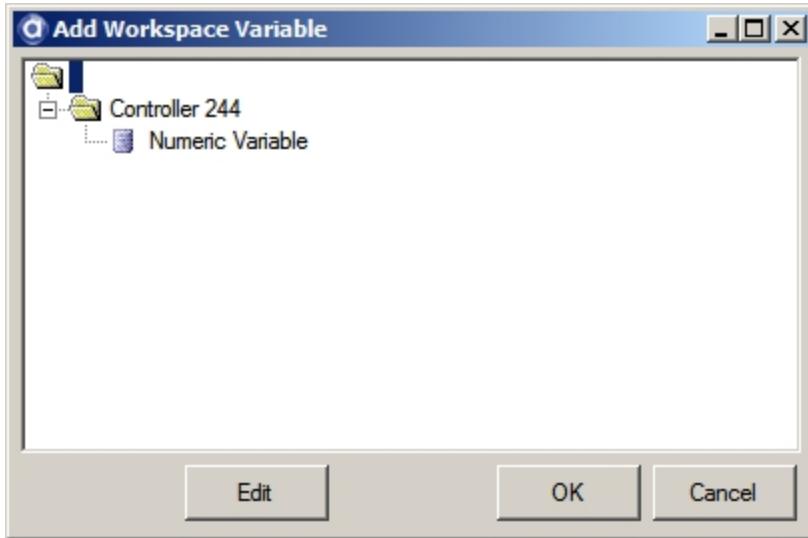
**NOTE:** If you rename the object associated with a variable (for example, rename the Controller object from "/Controller 22" to "/Adept Controller"), the variables will be updated and associated with the new object name.

### Add Workspace Variable Window

The Add Workspace Variable window is used to add an existing variable (in other words, a variable that is already in the workspace) to the Watch Variable list.

To add an existing workspace variable to the Watch Variable list:

1. Click the Add Workspace Variable (  ) icon. The Add Workspace Variable window opens.



*Add Workspace Variable Window*

2. The window shows the variables that are in the workspace. Select the desired variable from the list.

Optionally, you can click **Edit**, to open the Variable Editor. For details, see Variable Editor on page 131.

3. Click **OK** to add the selected variable to the Watch Variable list.

## Process Control

The topics in this chapter describe the Process Manager and ACE PackXpert features of the ACE software.

Overview .....	800
Process Menu and Features .....	807
Belt .....	809
Custom Allocation Script .....	825
Part .....	829
Part Buffer .....	836
Part Target .....	841
Process Manager .....	843
Process Manager Control .....	920
Process Pallet .....	927
Status/Error Codes .....	932
Custom V+ Programs .....	937
Vision Refinement Station .....	941
Workspace Positioning .....	945

## Overview

The ACE software allows you to build applications, such as ACE PackXpert packaging applications which can be basic pick-and-place cells or complex cells with multiple cameras, conveyors, and robots. You can create and "program" these cells without having to write any programming code. Further, the software optimizes the cell programming to maximize throughput, handle part flow and robot utilization (line balancing), and other cell parameters based on settings that you specify.

This is accomplished by dividing the work between the PC and the Adept controller, as follows:

Operating on the PC:

- Vision location/camera management
- Vision object filtering
- Hardware/line configuration and balancing scenarios
- Tracking part status as parts are processed by the robots

Operating on the controller:

- Queue managing instances that have been passed to the controller for processing. This includes notifying the PC concerning the status of parts being processed and not processed.
- Robot control

Operating assumptions:

- The grippers for the robot are wired and controlled at a V+ level. They can be defined in the software interface, but the low-level control is associated with the controller belonging to the robot.

For most applications (even complex conveyor-tracking or multi-robot applications), the entire application can be constructed and programmed through the software interface, without writing any program code. For applications that require greater control, you can override the default V+ code and make changes, as needed.

## Process Components

This section describes the Process components, which are accessed from the Process menu. The other application components, such as Adept robots, grippers (end-effectors), controllers, and vision tools, are described in other sections of this documentation.

### ***Process Pallets***

The Process Pallet object is used to define the layout and location of a pallet. The pallet can be in a static position or it can be located on a conveyor belt. The pallet can use a traditional row and column part layout or use a radial part layout.

### ***Belt and Belt Encoder***

The Belt object defines a conveyor belt used by the system. The Belt object maintains a list of encoders that are associated with the conveyor. The Belt Encoder defines the mm/count ratio of the encoder. The

Belt Encoder Controller Connection maintains a list of controllers that the encoder is wired into. The controller connection can also specify controller latching to a particular encoder. The Belt object also contains belt speed and start/stop controls.

### **Parts and Part Targets**

A Part object defines a part that is input for processing. The Part object has a Configuration drop-down list box that is used to specify, in general terms, how the part is input to the system.

A Part Target object defines a possible destination for a part. The possible configurations for a Part Target object are the same as for a Part object.

Depending on the selected configuration, additional information can be defined when configuring the part:

Part Configuration	Vision Properties	Belt Properties	Pallet
<b>Belt</b>	Optional	Required	Optional
<b>Static</b>	Not Used	Not Used	Optional
<b>Vision</b>	Required	Not Used	Optional

The following sections provide details on several of the Part Configuration options:

- Part Configuration

The Part Configuration defines the part pick or place requirements.

*Belt* - the part is picked from or placed onto a conveyor belt. It may use latching, a camera, or a spacing interval to determine the position. A pallet is optional.

*Static* - the part is picked from or placed onto a fixed location. Because it is a fixed location, no camera or belt is used. A pallet is optional.

*Vision* - the part pick or place process requires a fixed-location camera. There is no belt used. A pallet is optional.

- Vision Properties

Vision Properties are used when the Part Configuration is defined as Vision. A vision tool is specified that is used to locate the part—for example, this could be an inspection tool that filters instances based on some criteria. Additionally, the properties can be configured to filter the vision results based on a part name. This will most likely be associated with a named part returned from a locator model.

- Belt Properties

Belt Properties are used when the Part Configuration is defined as Belt. A belt and encoder must be specified for use with the product. Then a Belt Mode is defined, which describes how the part is related to the belt. For this item, additional information is required based on the selection:

Belt Mode	Vision Properties	Spacing
Vision	Required	Not Used
Latch	Not Used	Not Used
Spacing	Not Used	Required

**NOTE:** When the mode is Latch, the latch information is taken from the Belt object linked to the Belt Properties.

- Pallet

Pallet is an optional parameter that specifies the parts are acquired from a pallet or placed into a pallet. This optional parameter can be used in conjunction with a Belt, Static, or Vision configuration. It is important to note that when used with a Vision or Belt, the vision or belt is configured to locate the origin of the pallet, not the parts in the pallet.

- Part Process

The Process Manager is responsible for processing Parts input to the system and routing them to a Part Target. To do so, it maintains a Part Process list. A Part Process identifies an Adept Robot that can pick a Part or collection of Parts and place it at a Part Target or collection of Part Targets.

- If a Part or Part Target is a pallet, then the Part Process object allows for a Pallet Exclusion Configuration to be defined. The user can limit the pallet positions that can be accessed by the robot in this configuration.
- A relative priority can be associated with a given part process. This priority is used by the Process Manager when allocating parts to robots.
- The Part Process defines a gripper pick configuration and where the robot will place the parts.

**NOTE:** The Part Process defines a possible processing scenario for handling a Part. The Process Strategy is responsible for deciding which robots will process which parts. This is done using the list of Part Process objects as a guide to the valid combinations.

The Process Manager examines the list of Part and Part Targets associated with the Part Processes defined by the user. It will generate a list of Calibration objects, which are displayed to the user, as follows:

- Robot-to-Belt Calibration
- Robot-to-Belt-Camera Calibration
- Robot-to-Belt-Latch-Sensor Calibration
- Robot-to-Fixed-Camera Calibration
- Robot-to-Belt-Spacing-Reference Calibration

Each calibration object relates the robot to another hardware element in the application.

---

### **Motion Information**

After a collection of Part Processes is defined, the Process Manager scans the collection to determine what additional configuration data is needed to properly drive the process. The following are some examples:

- Each robot will need an idle position.
- For each part that is picked, motion parameters describing the approach height, depart height, motion configuration, and offset at the pick location must be defined.
- For each part that is to be delivered to a target, the approach height, depart height, motion configuration, and offset at the place location must be defined.

The Process Manager maintains a list of the required information that must be taught as part of the configuration of the system. The motion information is located in the Configuration Items group of the Process Manager editor. For details, see Configuration Items on page 856.

### **Part Sources**

The Process Manager analyzes the configuration of Part Processes in order to determine what Sources are needed to control the hardware. A Source is an object that interacts with the hardware and discovers Part Instances and Part Target Instances. A Source is allocated for each of these conditions:

- For each Static-defined Part or Part Target, a Source is created.
- For each Vision-defined Part or Part Target, one Source is created for each Virtual Camera referenced. This Source will process the collection of all Part or Part Targets referenced by the Virtual Camera.
- For each Belt-defined Part or Part Target, one Source is created for each Belt referenced. This Source will process the collection of all Part or Part Targets referenced by the Belt.

For each Source, the Process Manager allows you to modify certain parameters associated with the Source. As an example, the Vision and Static Source objects can be configured to interface with a feeder.

### **Part and Part Target Instances**

When an individual Part is located, it is represented by a Part Instance. When an individual Part Target point is identified, it is represented by a Part Target Instance. These objects identify the transformation and Part/Part Target information so the complete location can be resolved. If an individual Part Instance must be placed at a specific Part Target Instance, the Part Instance will have a link to the appropriate Part Target Instance.

Part Instance and Part Target Instance objects get allocated to a controller for processing by a robot. The Process Manager uses the Process Strategy to identify that allocation.

**NOTE:** The Process Manager knows if a Part/Part Target instance was processed, not processed, or if an error happened during processing because of a grip error. If a grip error occurs, that instance will not be transferred to next robot and will be counted as not processed in the statistics.

### **Process Handler**

When the Process Manager executes a process, it relies on a series of internal objects to manage the interaction with the hardware. The Process Manager is responsible for organizing and containing the information that is going to be processed. The Process Handler is responsible for using that information and managing the execution of the process.

In general, the runtime operation of the Process Manager will use the Part Process information to locate Part Instances and Part Target Instances.

Internally, the Process Handler maintains a collection of internal objects that are responsible for interacting with individual elements of the hardware. These objects fall into two categories: objects that generate Part Instances and Part Target Instances and objects that can process the instances.

### **Controller Queue**

The Controller Queue represents a controller with associated robots that can pick from Part Instances and place to Part Target Instances.

The Controller Queue communicates with the V+ Queue Manager task that manages the collection of parts to be processed by the robots on a given controller. The Controller Queue receives notification as the controller processes the instance information.

The Controller Queue also monitors for "robot down" or capacity issues with the robots connected to the controller. It notifies the Process Manager through an event, in the case that the controller is unable to process the items in its queue within a given time-frame. The "time-frame" is based on the belt speed and location of the parts on the belt given the upstream/downstream limits of the individual robots.

The Controller Queue maintains state information regarding its ability to accept parts. This information is used by the Process Strategy when determining how to allocate parts.

The Controller Queue also maintains statistics that are captured for a certain number of cycles, such as idle time, processing time, and parts/targets processed per minute. This information is available to the user and may be used in the allocation of Part Instances.

## **Line Balancing**

A Process Strategy is invoked to determine how to allocate the Part Instances and Part Target Instances identified by the Process Manager. It uses the list of Part Processes to allocate the instances to specific robots. The output of this process is passed to the Controller Queue object by the Process Manager.

Each Process Strategy operates under certain assumptions based on the process being monitored. Those assumptions determine which algorithms are used to perform the allocation.

### **Process Strategy**

This process strategy is predicated around certain assumptions on how robots will handle Parts and Part Targets. For part processing, the overflow from an upstream robot will be passed to the next downstream robot on the same controller. In other words, the first robot along the conveyor will pick all parts it is capable of picking. Any part it cannot pick will be picked by the next robot in the line. This pattern is repeated for all robots in the line.

---

For the processing of Part Targets, any targets configured as a *latched* Pallet will be passed from robot to robot, allowing each one to fill the slots with parts as defined by the Process Strategy.

There is no logic running that tries to optimize the allocations of parts or targets. The process strategy simply requests that each robot process as many Parts and Part Targets as possible, and remaining parts are passed to the next robot.

There are user-defined parameters that control this process strategy, namely:

- Robot Parameters - used to specify the queue size for the robot.
- Belt Window Parameters - used to set part-processing filters, which help to optimize cycle time.
- Belt Control Parameters - used to set conveyor belt on/off and speed controls, which can dynamically adjust the part flow to the robot.

These parameters are available in the Process Strategy editor.

### **Custom Process Strategy**

The system allows you to define your own process strategies, if needed, using C# within the application.

## **V+ Controller Software**

The V+ application is split into two sections:

- The first is a series of V+ programs that are responsible for picking and placing an instance.
- The second is a series of V+ programs responsible for managing the queue of parts and communicating with the PC.

### **Robot Control Code**

The V+ code is designed to run without any PC interaction. It is triggered by items arriving in the queue. Motion parameters are defined on the PC and then downloaded to variables on the controller. Multiple instances of this program are run, one for each robot in the configuration.

## **Process Flow**

The general data flow is as follows:

### **Step 1**

There is a Controller Queue that maintains a list of Robot Station objects. The Controller Queue communicates with programs running in V+. The Part Source and Target Source identify Part Instances and Part Target Instances for processing asynchronously to this data flow operation.

### **Step 2**

The Process Handler runs the part/target allocation algorithm. Part Instances and Part Target Instances are associated with a Robot Station.

**Step 3**

The Robot Station notifies the Controller Queue that the number of parts has changed. The Controller Queue checks to see if data needs to be sent to V+ and then sends the instance information, if needed.

**Step 4**

As Part Instance and Part Target Instance parts have been processed by V+, the V+ programs notify the Process Handler that they have been processed.

**Step 5**

The Process Handler removes the part or target information from the Part Source, Target Source, and Robot Station, as needed. This triggers the same Step 3 condition in the Controller Queue and the process continues from there.

## Process Menu and Features

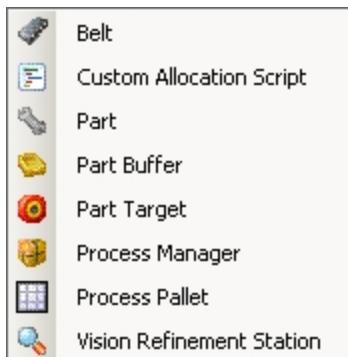
**NOTE:** The following software features require specific PC and Adept controller licenses. For details, see Licensing Requirements on page 28.

The Process menu options are used to define, configure, and manage the process elements of an application, such as an ACE PackXpert packaging application. The menu options include the process manager, conveyor belts, pallets, parts, part buffers, vision refinement stations, and part targets. The Process Manager editor provides a point-and-click interface for configuring, programming, and controlling these elements from a central location. Further, the Process Manager takes care of managing *multiple* controllers, robots, conveyors, parts, and pick-place locations in the application.

To access the Process menu options, right-click in the folder area of the ACE workspace explorer, and select:

### New > Process

The following menu choices are available:



#### Belt

The Belt object defines a physical conveyor belt used by the system. The Belt object maintains a list of encoders that are associated with the physical conveyor. For details, see Belt on page 809.

#### Custom Allocation Script

The Custom Allocation Script object is used to create and edit custom part-allocation programs for use with the Process Manager. For details, see Custom Allocation Script on page 825.

#### Process Manager

The Process Manager editor is the central point for creating and managing the application. For details, see Process Manager on page 843.

#### Part

The Part object defines an object that is input to the Process Manager editor for processing. For details, see Part on page 829.

#### Part Buffer

The Part Buffer object defines an object that is an “overflow” buffer where parts can be temporarily stored when an output conveyor belt (or feeder) is unavailable to accept more parts. For details, see Part Buffer on page 836.

#### Part Target

The Part Target object defines an object that is a possible destination for a part. For details, see Part Target on page 841.

#### Process Pallet

The Process Pallet object defines the layout of a pallet, which can

be used to pick parts from or place parts to. For details, see Process Pallet on page 927.

### **Vision Refinement Station**

The Vision Refinement Station object defines an object that is used to refine the part to gripper orientation for improved placement accuracy. For details, see Vision Refinement Station on page 941.

### **Wizards**

Many of the application components are configured using wizards (a series of screens that guide you through a process). For information on the wizard buttons and controls, see Wizards on page 181.

### **3D Visualization**

As items are added to the workspace, they can be shown in the 3D Visualization window, if the Simulation Visible option or Show in Visualization option is enabled in the object editor. For example, in addition to the robot, the 3D Visualization window shows:

- The robot belt windows
- The location of latch sensors
- The location of each camera field of view (FOV).

For more details on the 3D Visualization feature, see 3D Visualization on page 176.

## Belt

**NOTE:** When emulation mode is enabled, this item contains additional features. For more details on Emulation Mode, see Emulation Mode on page 1089.

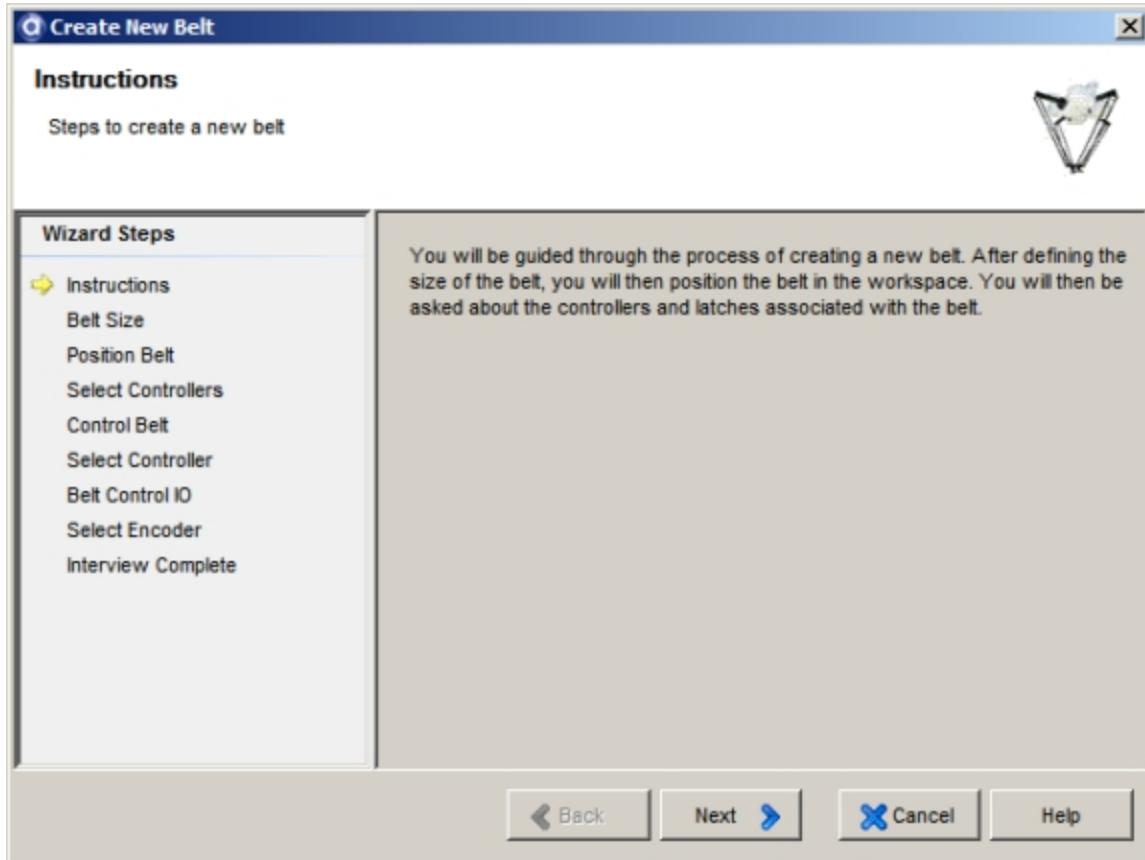
This section describes the Belt option on the Process menu.

The Belt object defines a conveyor belt used by the system. The Belt object maintains a list of encoders that are associated with the conveyor. The Encoders group Scale property defines the mm/count ratio for each encoder. The Encoders group Controller property contains a list of controllers to which the encoders are connected. The Encoders can also specify if a listed controller is configured to latch to a particular encoder.

### Creating a Belt Object

To create a Belt object, right-click in the folder area of the Workspace Explorer and select **New > Process > Belt** from the menu. The Create New Belt interview wizard opens, which will guide you through the process of creating a new Belt object for the workspace.

For more details on the buttons and indicators on the wizard pages, see Wizards on page 181.



### Create New Belt Wizard

Read the introduction and then click Next. Simply follow the interview wizard instructions in each step to complete the process of creating a new Belt object. During the process you will:

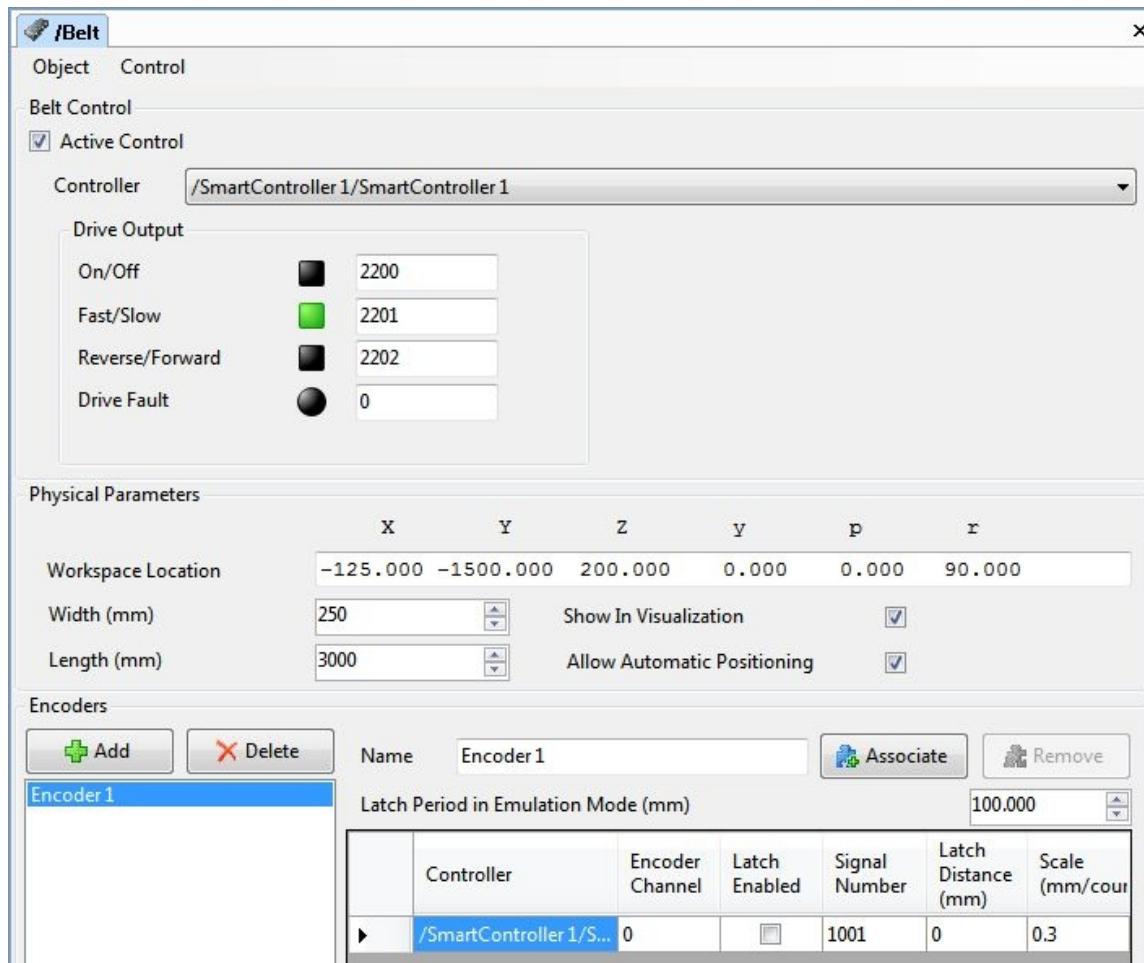
- Set the belt size
- Position the belt in the workspace
- Select a controller for the belt
- Set belt-control IO signals
- Select and test the encoder channel

When you have completed the wizard steps, a Belt object will be added to the workspace and the Belt editor will open, as shown in the following figure.

You can also open the Belt editor by double-clicking the Belt object in the folder area of the Workspace Explorer.

## Creating a Belt Object

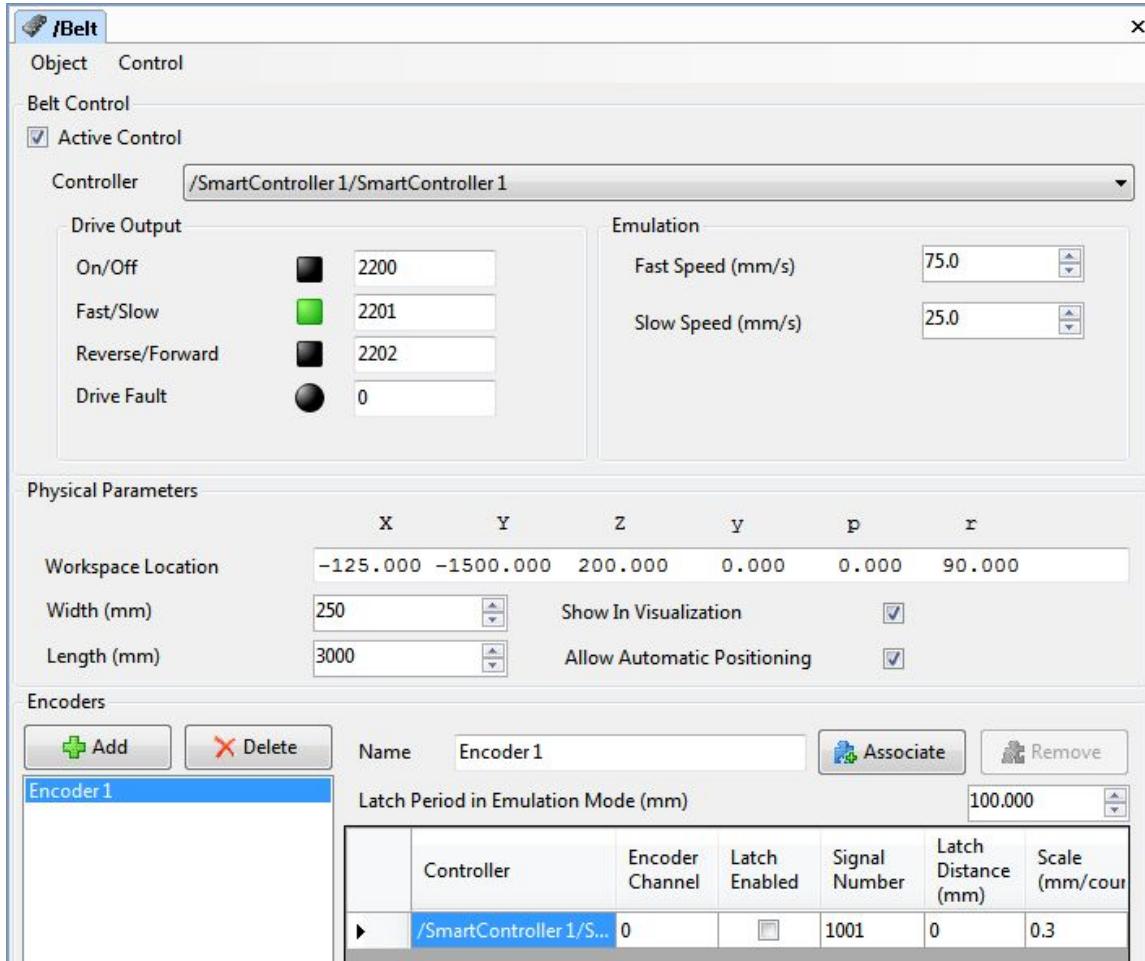
---



Belt Editor

## Menu Items

---

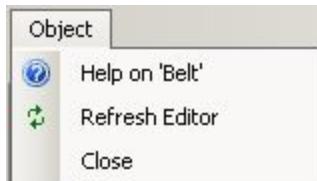


*Belt Editor in Emulation Mode*

## Menu Items

This section describes the selections available from the Belt Editor menu.

### Object Menu



#### Help

Displays the online help for the object editor.

#### Refresh Editor

Refreshes the contents of the object editor window.

#### Close

Closes the object editor.

**Control Menu****View Belt Encoders**

Displays a wizard used to view the operation of the belt encoder.

**Test Latch Signal**

Displays a wizard used to view the operation of the encoder latch signal.

For details on these menu items, see Viewing Belt Encoders and Testing Latches on page 818.

**Editor Parameters**

The middle portion of the Belt Editor contains the editor parameters. These are used to configure various settings on the selected conveyor belt. The Belt Editor parameters are described below.

**Belt Control Group**

The Belt Control group is used to specify various controls for a conveyor belt, as follows:

Item	Description
Active Control	<p>When selected, enables control of the conveyor belt from the specified Adept controller.</p> <p>When emulation mode is enabled:</p> <ul style="list-style-type: none"> <li>For the belts that do not have Active Control enabled:           <ul style="list-style-type: none"> <li>When the process starts, start all belts</li> <li>When the process stops, stop all belts</li> </ul> </li> <li>For the belts that have Active Control enabled:           <ul style="list-style-type: none"> <li>When the process stops, stop all belts that are in On/Off control mode. See the description of On/Off mode in this table.</li> </ul> </li> </ul> <p><b>NOTE:</b> For emulation-mode calibrations, the belt controls in the Calibration wizards will allow you to operate the belt, even when the Active Control option of the Belt object is not enabled.</p> <p>For more details on emulation mode, see Emulation Mode on page 1089.</p>
Controller	<p>Select an Adept controller to use for controlling the conveyor belt.</p> <p>Use the list icon ( ) to display the list of available controllers. If there is only one controller available in the ACE workspace, it will be automatically selected for you.</p>
On/Off	Enter the digital I/O signal number to use for starting and stopping the conveyor belt. When the signal is on, the conveyor belt is moving. Click the signal icon ( )

Item	Description
	 to toggle the signal state.
Fast/Slow	If your conveyor belt supports a fast and slow speed, enter the digital I/O signal number to use for toggling the conveyor belt between fast to slow speed. When the signal is on, the conveyor belt operates at its fast speed. Click the signal icon (  /  ) to toggle the signal state.
Reverse/Forward	If your conveyor belt supports switching between forward and reverse, enter the digital I/O signal number to use for toggling the conveyor belt between forward and reverse. When the signal is on, the conveyor belt operates in reverse direction. Click the signal icon (  /  ) to toggle the signal state.
Drive Fault	If the belt control fails, this indicator lights to show there is an error.
<i>The following items are added when emulation mode is enabled.</i>	
Fast Speed	Specifies the millimeters per second used when "fast" speed is enabled.  <b>NOTE:</b> This control is always available, even when "Active Control" is disabled.
Slow Speed	Specifies the millimeters per second used when "slow" speed is enabled.
Allow Automatic Positioning	When selected, enables automatic positioning of the belt object below the belt window in the workspace.  <b>NOTE:</b> This option ensures the belt is "below" the belt window. It only moves the belt in the "below" direction and never in the "above" direction. Therefore, if you position the belt window lower and the belt position is adjusted down, if you then move the belt window up, the belt object is not moved up.

### Physical Parameters Group

The Physical Parameters group is used to define the size of the conveyor belt and its location in the workspace, as follows:

**NOTE:** This information can be automatically generated by using the Workspace Referencing feature on the ACE Tools menu. For details, see Workspace Positioning on page 945.

Item	Description
Workspace Location	Enter the location of the conveyor belt in the workspace. The location is represented in World coordinates.

Item	Description
Width	Enter the width (in millimeters) for the conveyor belt.
Length	Enter the length (in millimeters) for the conveyor belt.
Show in Visualization	If selected, the conveyor belt will be shown in the 3D Visualization window. For details, see 3D Visualization on page 176.

The origin of the belt is defined as the farthest upstream point on the belt; the X-axis orientation defines the flow of the belt.

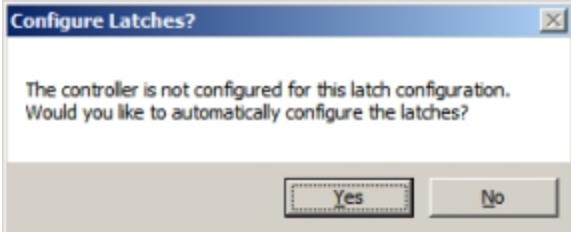
### Encoders Group

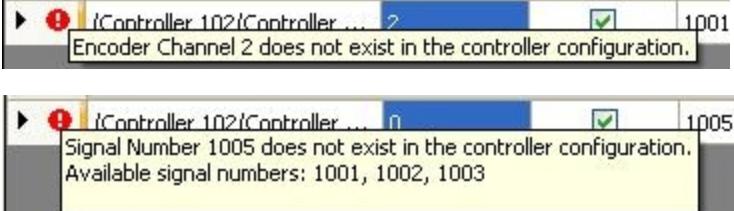
The Encoders group is used to specify various controls for a conveyor belt, as follows:

Item	Description
Add/Delete Buttons	Click <b>Add</b> to add a new encoder to the Encoder list. To remove an existing encoder from the list, select the encoder name and then click <b>Delete</b> .
Name	Enter a descriptive name for the encoder. A default encoder name is assigned when the new encoder is added to the Encoder list.
Associate/Remove Buttons	Click <b>Associate</b> to specify the controller to associate with the selected encoder. The Select a Controller dialog opens, which allows you to select an existing controller to associate with the encoder.  To remove an association, in the table area, click the row-selector (■) icon for the controller/encoder you wish to change, and then click <b>Remove</b> .
<i>The following items are added when emulation mode is enabled.</i>	
Latch Period in Emulation Mode	Specifies the latch period, in millimeters, that will be used when emulation mode is enabled.

The controller properties table provides the following controls for the encoder selected in the Encoder list. If there is more than one controller associated with an encoder, each association will be displayed in a separate row of the table. For this case, it is possible to have more than one entry in the table grid area.

Item	Description
Controller column	Displays the name of the controller assigned to the selected encoder.
Encoder Channel	Specifies the channel number to use for the selected encoder. Double-click the cell to edit the encoder channel number. This number is zero-based.
Latch Enabled	Select this check box to enable latching for the selected encoder.

Item	Description
	<p>When emulation mode is enabled and the latch is not configured, you are prompted to have the latches automatically configured.</p>  <p><i>Configure Latches Prompt</i></p> <p>Click Yes to configure the latches and open the Configure Encoder Latches tool. For more details on configuring latches, see Encoder Latches on page 198. For more details on Emulation Mode, see Emulation Mode on page 1089.</p> <p><b>NOTE:</b> If there is a configuration error and the alert icon (⚠) is displayed, you can double-click the icon to show the prompt in the previous figure and then click Yes to open the Configure Encoder Latches tool.</p>
Input Number	<p>Specifies the digital I/O latched signal input number for the selected encoder. Double-click the cell to edit the latch signal input number.</p> <p>You can enter more than one latch signal number, each one must be separated by a comma (1001, 1002, etc.). The available latch signals are defined in the Robots section of the V+ Configuration file. For details, see V+ System Configuration Editor on page 221. You can then configure the robot-to-belt calibrations to use a specific latch signal. By doing so, you will have multiple robot-to-camera calibrations sharing the same relative robot-to-belt calibration. For more details, see Defining Multiple Latch Signals for a Belt Encoder on page 822.</p> <p>When emulation mode is enabled, the latch input signal(s) for each encoder will be emulated. For more details on Emulation Mode, see Emulation Mode on page 1089.</p>
Latch Distance (mm)	Specifies the <i>minimum</i> latch distance for the selected encoder. Double-click the cell to specify the desired value.
Scale (mm/count)	<p>Specifies the millimeters of belt travel per encoder count for the selected encoder. Double-click the cell to specify the desired value.</p> <p>This information will be automatically generated when you calibrate the robot to the belt encoder. For details, see Belt Calibrations on page 848.</p>

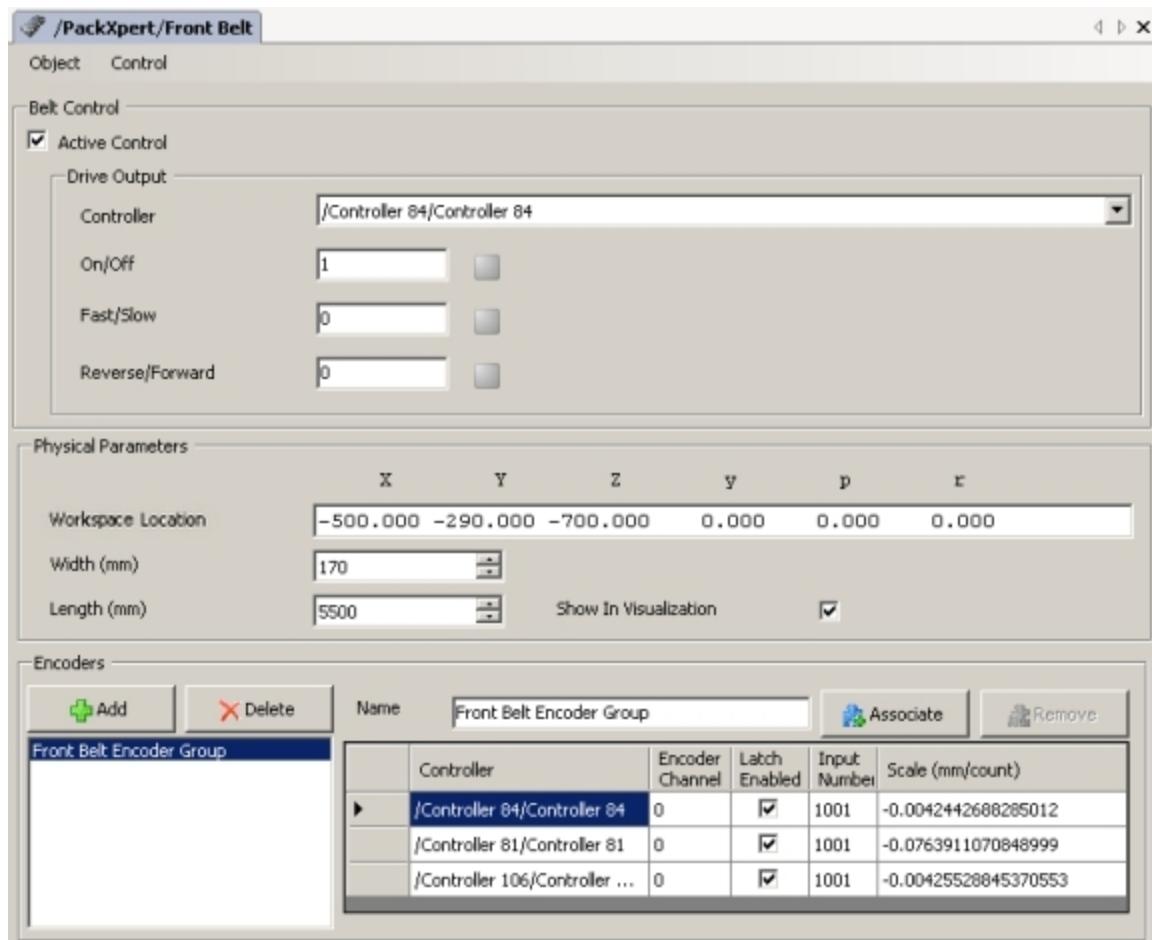
Item	Description
(Error message)	<p>If there is a configuration error, an alert icon (!) displays in the corresponding controller row. If you hover the mouse pointer over the icon, a message displays that describes the error. For example:</p>  <p>For more details on configuration messages and their associated codes, see Status/Error Codes on page 932.</p>

## Progressive Processing of Pallets

In a typical application, you will likely have only one controller associated with each encoder. However, in some applications, you will have multiple robots on different controllers to process parts or targets in pallets. In this configuration, each controller will have an encoder wired into the controller and a single latch signal wired into all controllers. The latch signal is used to detect the pallet; the encoders are used to monitor the belt movement.

As the pallet progresses along the belt, it gets "passed" between robots on all controllers. In this configuration, all controllers must have an entry under one encoder object (in the Encoders table of the Belt editor), as shown in the following figure.

**NOTE:** Not all latching applications require tracking like this. The "latch tracking" method is required only if multiple robots can fill the same pallet positions, and the system has to track the status of each pallet (i.e., what pallet positions are filled) across a controller boundary.



Belt Editor Showing Multiple Controllers/Encoders Tied to One Latch Signal

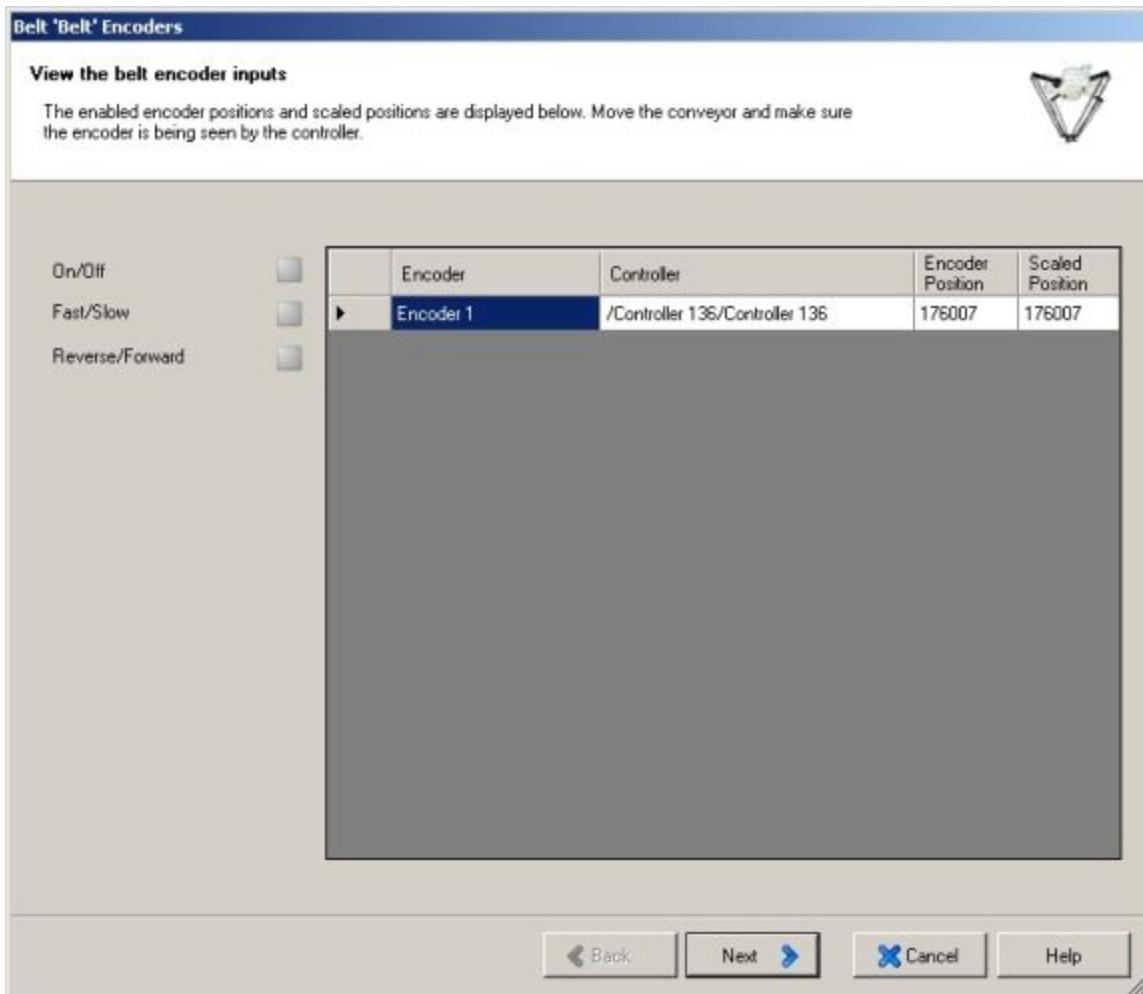
## Viewing Belt Encoders and Testing Latches

This section describes the View Belt Encoders and Test Belt Latches menu options in the Belt editor Control menu.

### ***Viewing Belt Encoders***

To view the encoder inputs for the selected belt:

1. From the Belt editor menu, select **Control > View Belt Encoders**. The Belt Encoders wizard opens.



#### Belt Encoders

2. Select a belt encoder from the list.
3. Activate the belt and watch the Encoder Position and Scaled Position values. These values should change when the belt moves.

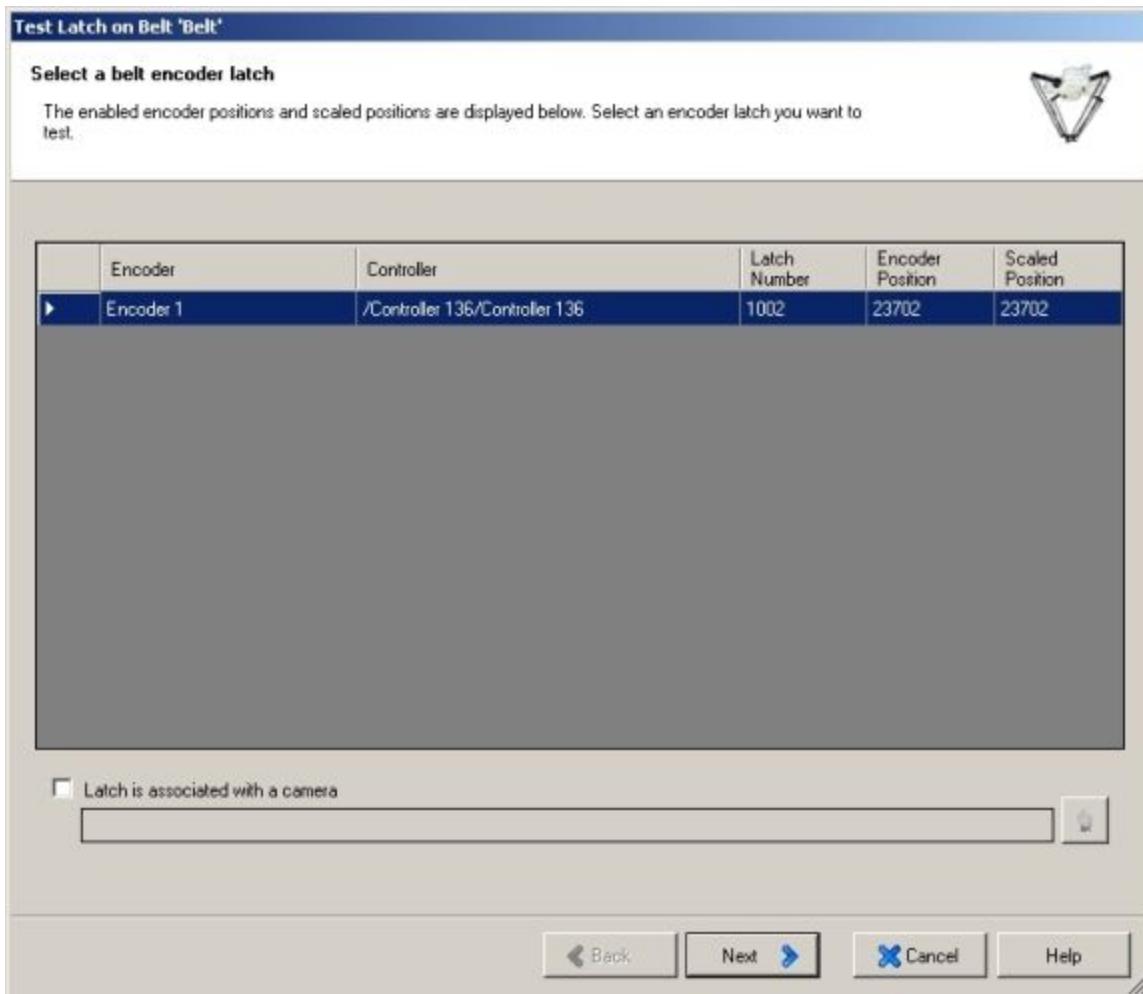
If the belt is under active control by the Adept controller, you can use the wizard belt controls to activate the belt. For details, see Wizards on page 181.

4. When finished, click **Next** to close the wizard.

#### Testing Latch Signals

To test the latch signal inputs for the selected belt:

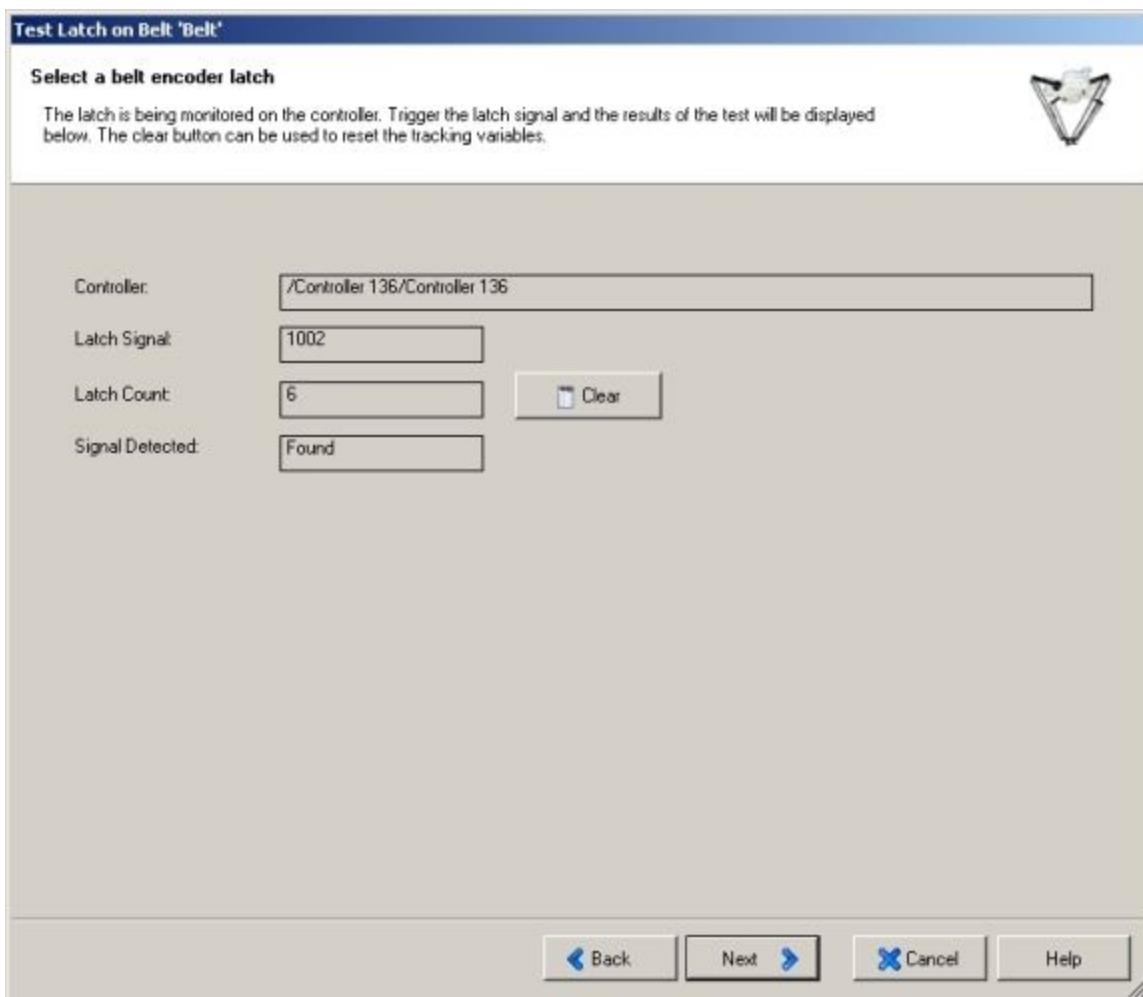
1. From the Belt editor menu, select **Control > Test Latch Signal**. The Belt Encoders wizard opens.



*Test Latch (Select a belt encoder latch)*

2. Select a belt encoder from the list. If the latch is associated with a camera, select that check box.

3. Click **Next** to proceed. The next page of the wizard opens.



#### *Test Latch (Test results)*

4. Activate the belt and watch the Latch Count field. This value should increment each time the latch sensor is activated.

If the latch is associated with a camera, a Picture button is displayed. When the Picture button is clicked, the latch count should increment.

5. When finished, click **Next** to close the wizard.

## Defining Multiple Latch Signals for a Belt Encoder

**NOTE:** This feature applies only to belt cameras. For example, multiple latch signals are not allowed if the configuration is used for a belt-latching configuration (such as locating multiple pallets with different latch signals). For that case, you will need to split the latches into separate encoders for the Belt object

The ACE software allows you to define multiple latch signals for a belt encoder controller connection. You can then configure the robot-to-belt calibrations to use a specific latch signal. By doing so, you will have multiple robot-to-camera calibrations sharing the same relative robot-to-belt calibration.

**NOTE:** By default, you do not need to set latch information. If you have one latch signal in the belt encoder controller connection, you do not need to do anything -- it will work, by default. Therefore, it is backward-compatible with the previous behavior.

To define multiple latch signals for a belt encoder:

1. Open the Belt editor. For details, see Belt on page 809.
2. Verify that you have associated a controller with the belt encoder, as shown in the following figure.

Encoders						
		Name	Associate		Remove	
Add		Delete				
Encoder 1						
			Controller	Encoder Channel	Latch Enabled	Input Number
			/Controller 244/Controller 244	0	<input checked="" type="checkbox"/>	1001
						Latch Distance (mm)
						50
						Scale (mm/count)
						1

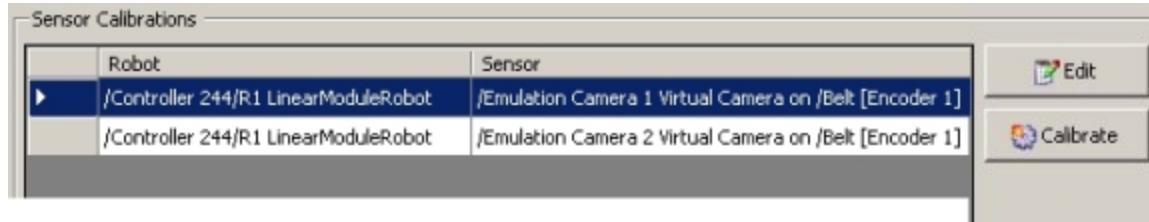
3. Specify the latch signals used by each camera (for example, camera 1 wires into 1001 and camera 2 wires into 1002). To specify the signal numbers, double click the Input Number cell and then enter the desired signal numbers—multiple signals must be separated by a comma (1001, 1002, etc.).

	Controller	Encoder Channel	Latch Enabled	Signal Number	Latch Distance (mm)	Scale (mm/count)
	/Controller 244/Controller 244	0	<input checked="" type="checkbox"/>	1001,1002	50	1

**NOTE:** The available latch signals are defined in the Robot section of the V+ System Configuration file. For details, see V+ System Configuration Editor on page 221.

4. Locate the robot-to-belt camera calibrations in the Sensor Calibration area of the Process Manager

editor, and click **Edit**.



5. Associate one camera calibration with latch signal number 1001 and the other with 1002, as follows:
  - a. Expand the Belt Latch Configuration group by clicking the + sign next to the item.
  - b. In the Latch Number field, enter the desired latch signal number.
  - c. Click the Use Default drop-down list and select False.
  - d. Click **Close** to close the editor.

For example, as shown in the figure below, the belt camera calibration is associated with latch signal number 1001.



6. Repeat the previous step for the other camera, using latch signal number 1002.

At runtime, the system will use the latch number sent from V+ when identifying the camera that the latch signal is relative to.

## Custom Allocation Script

The tool described in this section is based on the C# programming language. The C# programming language is used for writing PC-based applications. If you are new to the C# programming language, there is documentation, tutorials, and online classes available through the following URL:

<http://msdn.microsoft.com/>

The ACE software provides a Custom Allocation Script object, which contains a simple C# program editor. It is used to create and edit custom part-allocation programs for use with the Process Manager.

### Using the Custom Allocation Script Editor

The Custom Allocation Script provides two different entry points that are called: One is for allocating "non-belt" instances (parts and targets); the other is for "belt" instances. The program can manipulate the lists to indicate to the system what should be allocated to a given robot. This is called by the Process Manager when it needs to allocate parts.

To use the Custom Allocation Script:

1. Create a Custom Allocation Script object. Right-click on the workspace explorer folder view and select **New > Process > Custom Allocation Script**. This will create a Custom Allocation Script object and open the editor, as shown in the following figure.

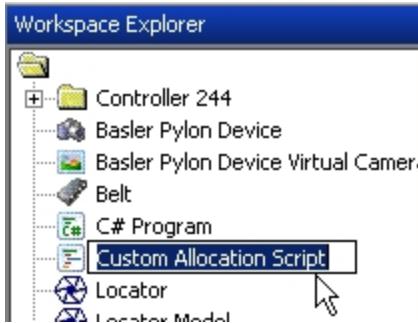
The screenshot shows a software interface titled "/Custom Allocation Script". The main area is a code editor with the following content:

```
1 using Ace.Core.Server;
2 using Ace.Process;
3 using Ace.Process.Server;
4 using Ace.Process.Server.Hardware;
5 using System;
6 using System.Collections.Generic;
7 using System.Diagnostics;
8
9 namespace Ace.Custom {
10
11     /// <summary>
12     /// Object defining a custom instance allocation algorithm
13     /// </summary>
14     public class Program {
15
16         public AceServer ace;
17
18         /// <summary>
19         /// Initializes a new instance of the <see cref="Program"/> class.
20         /// </summary>
21         public Program() {
22
23     }
24
25 }
```

Below the code editor is a toolbar with icons for Save, Open, and Run. The status bar at the bottom shows "Compile Errors | Trace Messages | Browse Variables". A table titled "Compile Errors" is present but empty.

*Custom Allocation Script Editor*

2. When the Custom Allocation Script object is created, it will be selected in the folder tree so that you can assign a name to your Custom Allocation Script program.

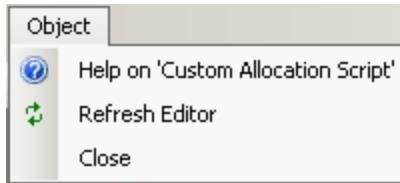


#### *Renaming the Custom Allocation Script Object*

3. Use the toolbar items and code editor, described below, to edit and compile your program.
4. Use the Process Strategy editor to associate the Custom Allocation Script with a process. For details, see Process Strategies on page 878.

## Toolbar Items

This section describes the selections available from the C# program editor toolbar.



### **Help**

Displays the online help for the C# program editor.

### **Refresh Editor**

Refreshes the contents of the editor window.

### **Close**

Closes the C# program editor.

Item	Description
	Deletes the selected text and stores a copy in the Windows clipboard.
	Copies the selected text to the Windows clipboard.
	Pastes the contents of the Windows clipboard at the current cursor position.
	Clears the messages in the Trace Messages tab.
	Compiles the program.
	Compiles and runs the program.

## Tab Items

---

Item	Description
	(Active when program is running) Stops the program.
Release Mode ▾	Selects the editor mode: Release Mode is the normal execution mode; Debug Mode is used to debug the program.
	(Debug Mode only) Used to step over a call.
	(Debug Mode only) Executes the next line of the program (single step through the program).
	(Debug Mode only) Runs the program.

## Tab Items

The following tabs are available in the editor.

Item	Description
Compile Errors	Displays a list of compile errors. Each error provides the line number, column number, and error message.
Trace Messages	Displays the trace messages (text included in any Trace.WriteLine() call in the program).
Browse Variables	Allows you to add variables to the window; the listed variables are updated after every step.

## Code Editor

The main part of the Custom Allocation Script editor contains a C# code editor. This is a simple editor that is used to create and edit your C# programs.

1. Use the code editor to create and edit your C# program. A sample program is provided, to get you started.
2. If needed, you can use the toolbar items, described above, to access cut, copy, paste, and find-and-replace functions.
3. After you have finished editing your program, click **Compile** to compile and save your program.

## Example

Custom allocation scripts are considered an advanced option. See the ACE Reference Guide for an example of an allocation script.

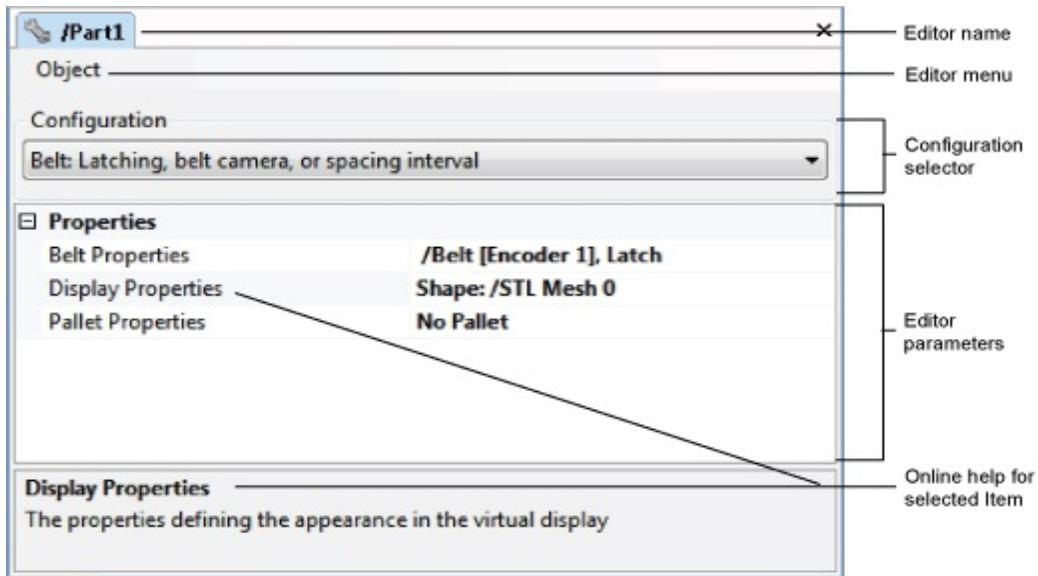
## Part

This section describes the Part option on the Process menu.

The Part object defines a physical object that is input to the application for processing. The Part has a Part Configuration property that specifies, in general terms, how the part is input to the application.

To create a Part object, right-click in the folder area of the Workspace Explorer and select **New > Process > Part** from the menu. The Part object is added to the Workspace Explorer and the Part object editor opens.

You can also open the Part editor by double-clicking the Part object in the folder area of the Workspace Explorer.



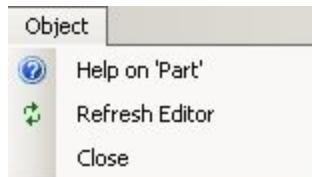
*Part Editor*

After you create the object, you can set the motion parameters (how the robot moves to and from that location) using the location editors. For details, see Location Editor Types on page 858.

## Menu Items

This section describes the selections available from the Part editor menu.

### Object Menu



#### Help

Displays the online help for the object editor.

#### Refresh Editor

Refreshes the contents of the object editor window.

### **Close**

Closes the object editor.

## **Configuration, Editor Parameters, and Online Help**

The middle portion of the Part editor contains the Configuration drop-down list box, Configuration editor items, and online help for the editor parameters.

### ***Configuration Drop-down List Box***

The Configuration drop-down list box is used to specify how the part is input to the system. The choices are:

- **Belt** - parts are located on a conveyor belt (using latching or fixed-spacing); optionally, vision and/or a pallet may be included in the part delivery system.
- **Static** - parts are acquired from a static location, such as a part feeder or a pallet.
- **Vision** - parts locations are acquired through the vision system. The camera is not located over a belt.

**NOTE:** If the part is supplied on a belt with a camera, the Belt option must be selected.

### ***Properties Editor***

The Properties editor is used to specify the configuration for the type of part configuration selected. Each item is described in the online help for that item. For details, see [Creating a Part Configuration](#).

### ***Online Help***

The bottom part of the Part object editor displays online help for the selected parameters.

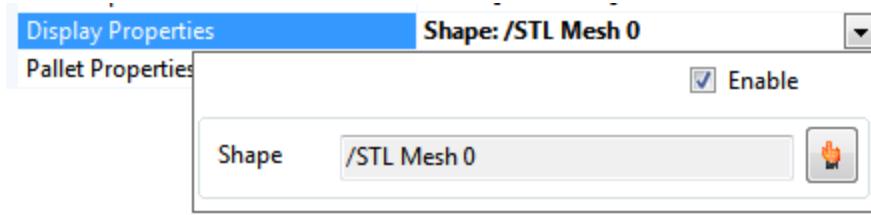
**NOTE:** Because online help for each parameter is provided directly in the editor, it is not duplicated in this documentation.

## **Creating a Part Configuration**

This section describes how to create a part configuration.

### ***Using a Custom Model for the 3D Display***

By default, ACE uses a "widget" to represent parts and part targets in the 3D visualization display. However, you have the option to use a custom model, which is imported from an STL mesh file. For details on the 3D visualization display, see [3D Visualization](#) on page 176.



To use a custom model:

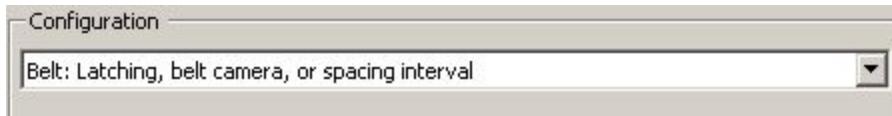
1. Click the Display Properties drop-down list box.
2. Check Enable to activate the Shape field.
3. Click the browse (Browse icon) icon to locate the .STL format file that you wish to use for the model.

### **Belt Configuration**

**NOTE:** When emulation mode is enabled, this item contains additional features. For more details on Emulation Mode, see Emulation Mode on page 1089.

To create a Belt configuration:

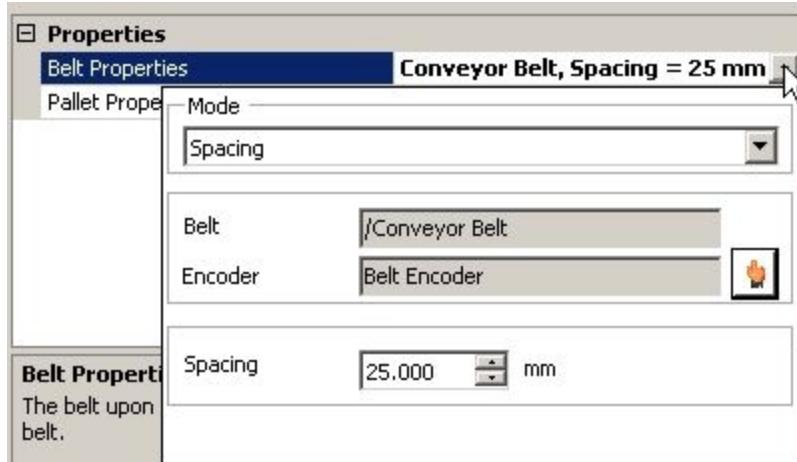
1. The Configuration drop-down list box is used to specify how the part is input to the system. Use the drop-down list box to specify "Belt".



The Properties area changes to show the properties that can be used with a Belt configuration.



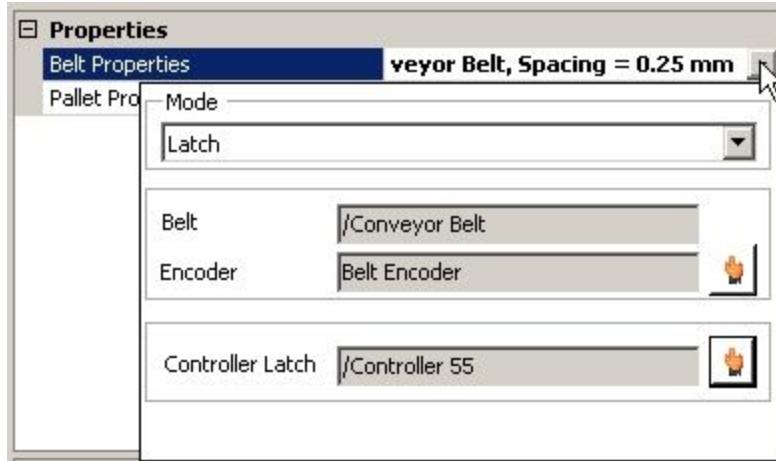
2. Use the Properties grid to enter the part-location information.
  - **Belt Properties** Required. Selects the belt that is used to handle the part and describes the mode of the belt. There are three modes available: Spacing, Latch, and Vision. Each mode is shown in the following figures.



*Belt Properties - Spacing Mode*

For this screen:

- **Belt** - the conveyor belt that will be used to handle the part.
- **Encoder** - the belt encoder used to control the conveyor belt.
- **Spacing** - the spacing, in millimeters, between parts on the conveyor belt.



*Belt Properties - Latch Mode*

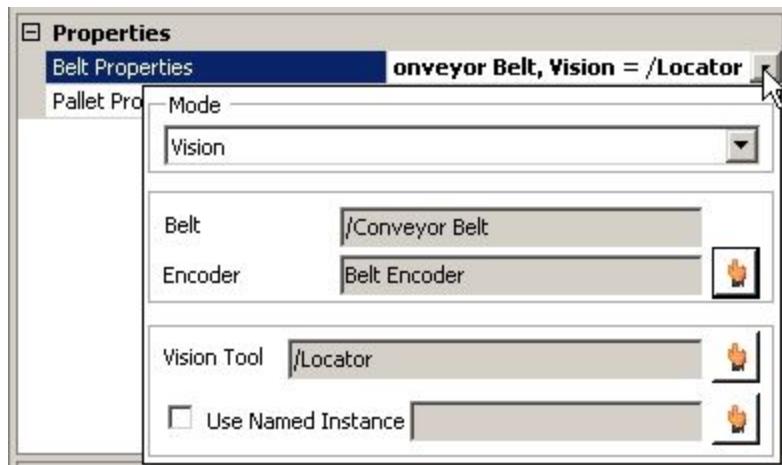
For this screen:

- **Belt** - the conveyor belt that will be used to handle the part.
- **Encoder** - the belt encoder used to control the conveyor belt.

- **Controller Latch** - the controller used for the conveyor belt latch signal.

In emulation mode, this item changes to:

**Latch Period in Emulation Mode** - the latch period, specified in millimeters.



*Belt Properties - Vision Mode*

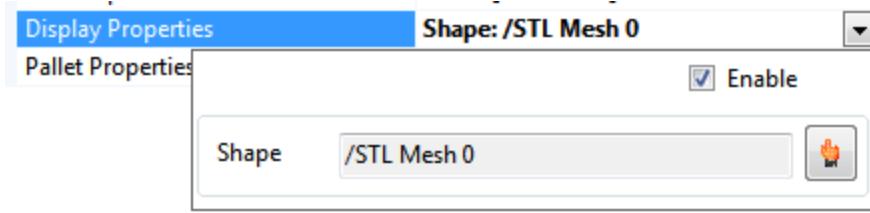
For this screen:

- **Belt** - the conveyor belt that will be used to handle the part.
- **Encoder** - the belt encoder used to control the conveyor belt.
- **Vision Tool** - the vision tool used to detect the part on the belt.
- **Use Named Instance** - (optional) the named instance of the part. Typically, this is the same as the Model name. For custom applications where a custom vision tool is used, this item would be used to specify custom names that had been associated with the different results returned from that tool.

- **Pallet Properties** Optional. Selects the pallet that is used to carry the part.

#### Using a Shape for the Pallet Display

By default, ACE uses an "invisible pallet" to represent pallets in the 3D visualization display. In other words, the parts and part targets are shown in the pallet arrangement, but the pallet itself is not visible. However, you have the option to use a shape (Box or Cylinder) or STL mesh custom model, which is imported from an STL mesh file. For details on the 3D visualization display, see 3D Visualization on page 176.



To use a shape:

1. Click the Pallet Properties drop-down list box.
2. Check Enabled to activate the Shape field.
3. Click the browse (Browse icon) icon to select a shape.
4. Use the shape object editor to specify the shape dimensions, or locate the .STL format file that you wish to use for the model.

### **Static Configuration**

To create a Static configuration:

1. The Configuration drop-down list box is used to specify how the part is input to the system. Use the drop-down list box to specify "Static".

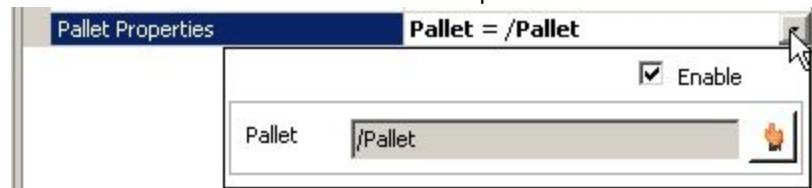


The Properties area changes to show the properties that can be used with a Static configuration.



2. Use the Properties grid to enter the part-location information.

- **Pallet Properties** Optional. Specifies the pallet that is used to carry the part. Select the Enabled check box and then choose the pallet.



### **Vision Configuration**

To create a Vision configuration:

---

1. The Configuration drop-down list box is used to specify how the part is input to the system. Use the drop-down list box to specify "Vision".

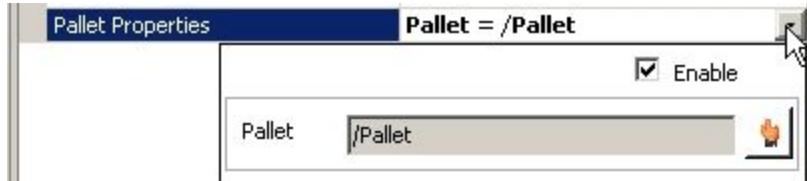


The Properties area changes to show the properties that can be used with a Vision configuration.

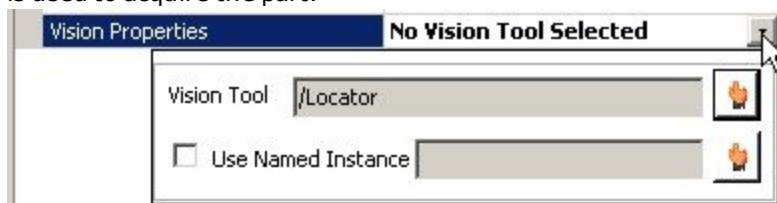


2. Use the Properties grid to enter the part-location information.

- **Pallet Properties** Optional. Selects the pallet that is used to carry the part.



- **Vision Properties** Required. Selects the vision tool and, optionally, the named instance that is used to acquire the part.



For this screen:

- **Vision Tool** - the vision tool used to detect the part on the belt.
- **Use Named Instance** - (optional) the named instance of the part. Typically, this is the same as the Model name. For custom applications where a custom vision tool is used, this item would be used to specify custom names that had been associated with the different results returned from that tool.

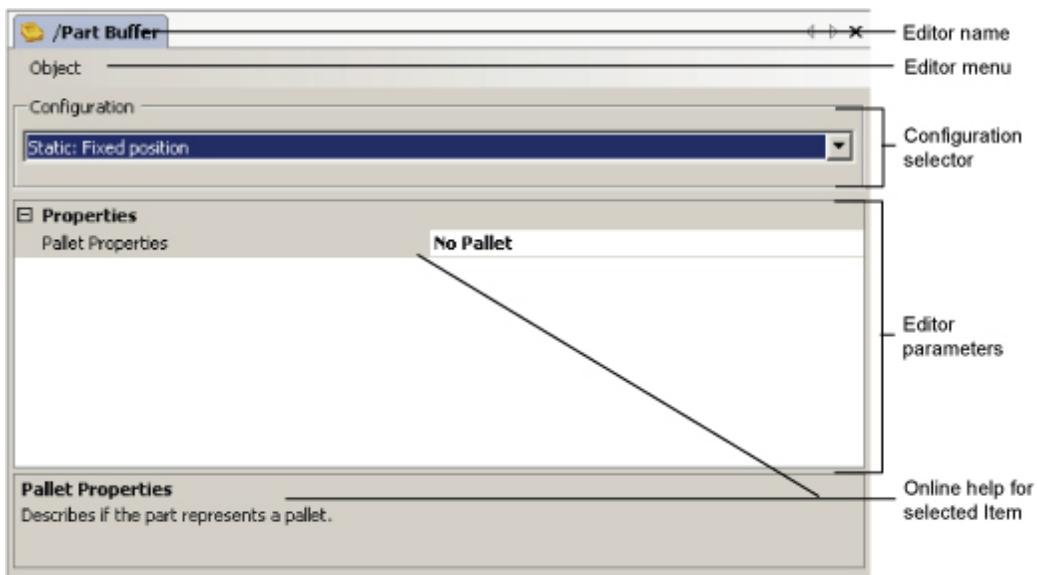
## Part Buffer

This section describes the Part Buffer option on the Process menu.

The Part Buffer object defines a physical object that is an “overflow” buffer where parts can be temporarily stored when an output conveyor belt (or feeder) is unavailable to accept the parts. A buffer can hold just a single part, or it can be a static pallet (or just a flat surface that acts as a pallet) that holds multiple parts. For pallets, you can also control the access method: Last In, First Out (LIFO) or First In, First Out (FIFO). The Process Pallet object editor is used to specify the pallet parameters. For details, see Process Pallet on page 927. The Process editor is used to specify the access method.

To create a Part Buffer object, right-click in the folder area of the Workspace Explorer and select **New > Process > Part Buffer** from the menu. The Part Buffer object is added to the Workspace Explorer and the Part Buffer object editor opens.

You can also open the Part Buffer editor by double-clicking the Part Buffer object in the folder area of the Workspace Explorer.



*Part Buffer Editor*

After you create the object, you can set the motion parameters (how the robot moves to and from that location) using the location editors. For details, see Location Editor Types on page 858.

## Menu Items

This section describes the selections available from the Part Buffer editor menu.

### Object Menu

Object	Help
 Help on 'Part Buffer'	Displays the online help for the object editor.
 Refresh Editor	Refreshes the contents of the object editor window.
 Close	Closes the object editor.

## Configuration, Editor Parameters, and Online Help

The middle portion of the Part Buffer editor contains the Configuration drop-down list box, Configuration editor items, and online help for the editor parameters.

### Configuration Drop-down List Box

The Configuration drop-down list box is used to specify how the part buffer is used by the system. The choices are:

- **Static** - parts are placed at a static location, such as a part feeder or a pallet.

### Properties Editor

The Properties editor is used to specify the configuration for the type of part configuration selected. Each item is described in the online help for that item. For details, see [Creating a Part Configuration](#).

### Online Help

The bottom part of the Part Buffer object editor displays online help for the selected parameters.

**NOTE:** Because online help for each parameter is provided directly in the editor, it is not duplicated in this documentation.

## Creating a Part Buffer Configuration

This section describes how to create a part buffer configuration.

### Static Configuration

To create a Static configuration:

1. The Configuration drop-down list box is used to specify how the part buffer is input to the system. Use the drop-down list box to specify "Static".

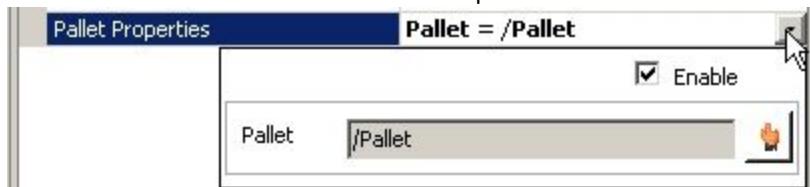


The Properties area changes to show the properties that can be used with a Static configuration.



2. Use the Properties grid to enter the buffer-location information.

- **Pallet Properties** Optional. Specifies the pallet that is used to hold the part(s). Select the Enabled check box and then choose the pallet.



After the pallet is specified, you can set the parameters for the pallet in the Process Pallet object editor. For details, see Process Pallet on page 927.

3. Now you can add the part buffer to a process, as described in the following section.

## Adding a Part Buffer to a Process

To add a part buffer to a process:

1. Add a Part Buffer object to your workspace, as described at the beginning of this topic.
2. Create a Part Buffer configuration. See the previous section for instructions.
3. Create a Process that includes a Robot, a Part and a Part Target. For details, see Processes on page 886.
4. Create a second process that includes the original Robot and Part, and select the Process Buffer as the Part Target.
5. Create a third process that includes the original Robot and Part Target, and select the Process Buffer as the Part.
6. Use the blue arrows in the Processes group to order the process by priority (the process at the top of the list gets the highest priority). For details, see Processes on page 886.

Processes			
Enabled	Part	Target	
<input checked="" type="checkbox"/> /Controller 244/R1			
<input checked="" type="checkbox"/>	/Part	/Part Target	
<input checked="" type="checkbox"/>	/Part	/Part Buffer	
<input checked="" type="checkbox"/>	/Part Buffer	/Part Target	

Add
Delete
Edit
Teach

*Part Buffers Added to the Processes Group*

- Set the part buffer access order, as described in the following section.

### Setting the Part Buffer Access Order

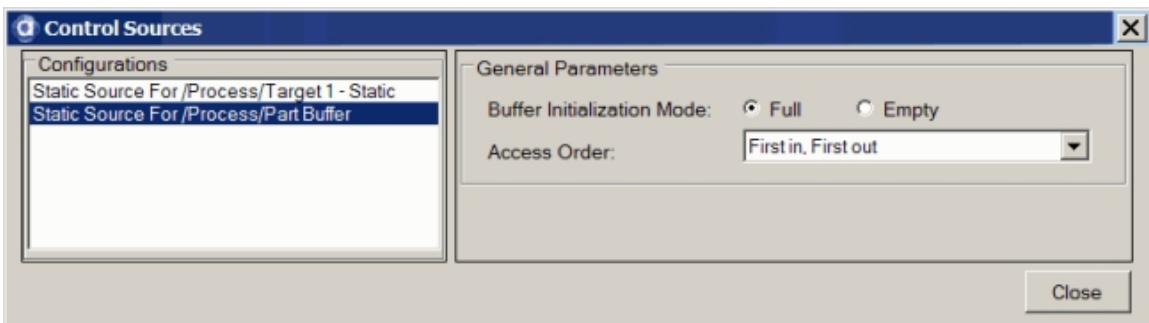
When a pallet is used for the part buffer, you need to specify how the parts will be accessed as the buffer is being emptied. There are two choices:

- First In, First Out (FIFO): The first part placed into the part buffer will be the first part removed.
- Last In, First Out (LIFO): The last part placed into the part buffer will be the first part removed.

**NOTE:** When parts are "stacked" (more than one layer is specified for the pallet), the access order must be set as LIFO.

To set the part buffer access order:

- In the Control Sources group, click **Edit**. The Control Sources editor opens, as shown in the following figure. For details, see Control Sources on page 869.



*Control Sources Editor with Part Buffer Selected*

- In the Configurations group, click Static Source For /Part Buffer, as shown in the previous figure.
- Optionally, select the desired Buffer Initialization Mode to indicate the state of the part buffer when

it is initialized. The default state is Empty, which means the buffer is empty when initialized.

4. Select the desired Access Order.
5. Click **Close** to close the Control Sources editor.

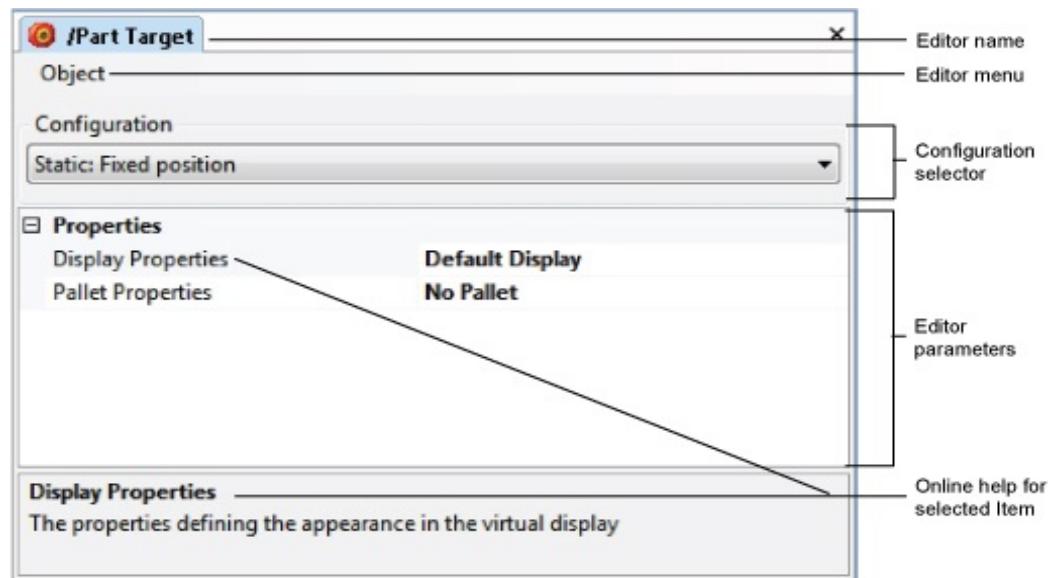
## Part Target

This section describes the Part Target option on the Process menu.

The Part Target object defines an object that is a possible destination for a part. The Part Target has a Part Configuration property that specifies, in general terms, how the part is handled by the application.

To create a Part Target object, right-click in the folder area of the Workspace Explorer and select **New > Process > Part Target** from the menu. The Part Target object is added to the Workspace Explorer and the Part Target object editor opens.

You can also open the Part Target editor by double-clicking the Part Target object in the folder area of the Workspace Explorer.



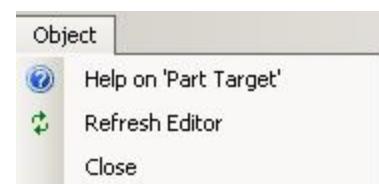
*Part Target Editor*

After you create the object, you can set the motion parameters (how the robot moves to and from that location) using the location editors. For details, see Location Editor Types on page 858.

## Menu Items

This section describes the selections available from the Part Target editor menu.

### Object Menu



#### Help

Displays the online help for the object editor.

#### Refresh Editor

Refreshes the contents of the object editor window.

**Close**

Closes the object editor.

## **Configuration, Editor Parameters, and Online Help**

The middle portion of the Part Target editor contains the Configuration drop-down list box, Configuration editor items and online help for the editor parameters. These are identical to those found in the Part object editor. For details, see the Part on page 829.

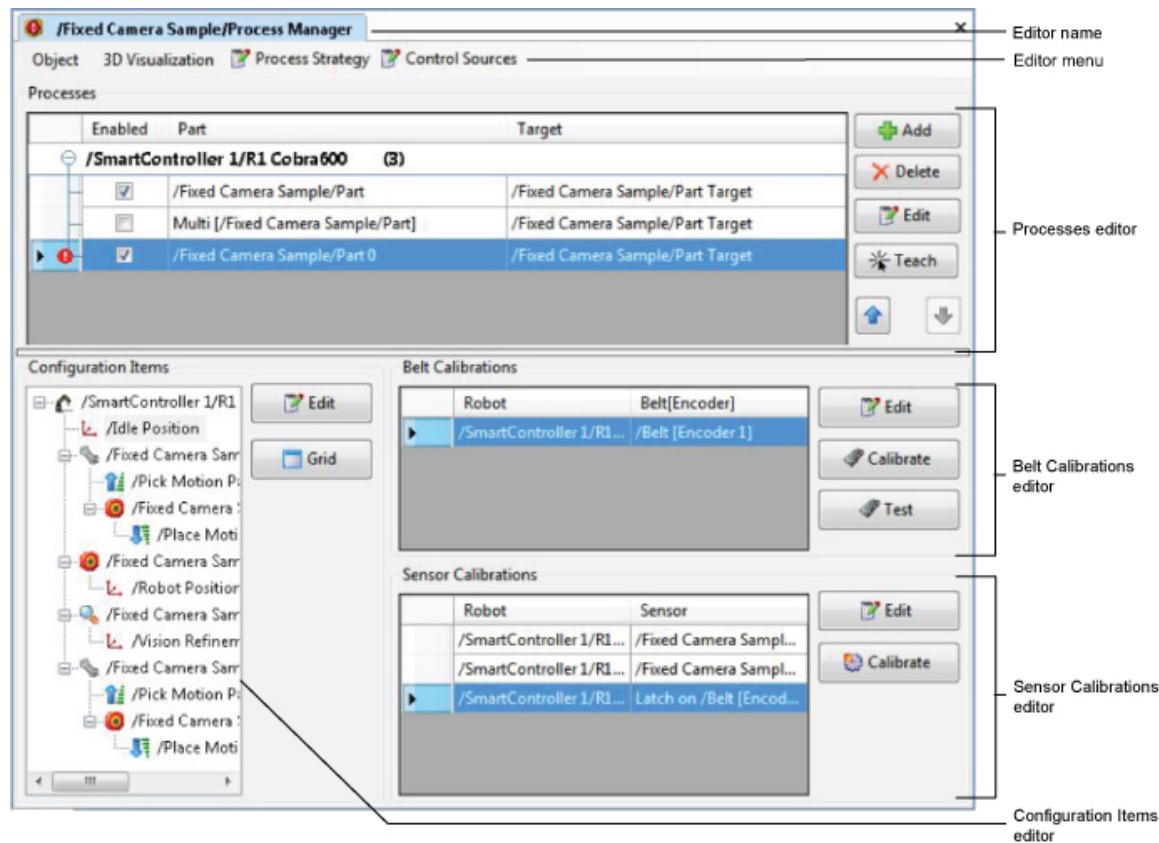
## Process Manager

The Process Manager is the central control point for developing a process-managed application, such as an ACE PackXpert packaging application. It allows you to create and "program" the application (even complex applications) without having to write any programming code. For advanced development, it provides access to "behind the scenes" V+ programming controls that allow you to tweak your application.

### Creating a Process Manager Object

To create a Process Manager object, right-click in the folder area of the Workspace Explorer and select **New > Process > Process Manager** from the menu. For example, to create an ACE PackXpert Process Manager object, you would select **New > Process > Process Manager** from the menu. The Process Manager object is added to the Workspace Explorer and the editor opens.

You can also open the editor by double-clicking the Process Manager object in the folder area of the Workspace Explorer.

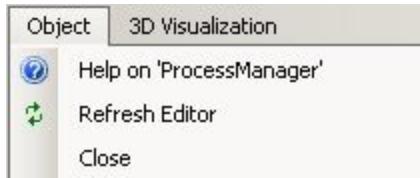


Process Manager Editor

## Menu Items

This section describes the selections available from the editor menu.

### **Object Menu**



#### **Help**

Displays the online help for the object editor.

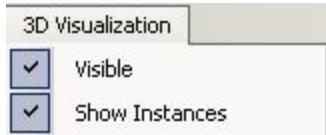
#### **Refresh Editor**

Refreshes the contents of the object editor window.

#### **Close**

Closes the object editor.

### **3D Visualization Menu**



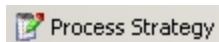
#### **Visible**

Displays the calibration objects in the 3D Visualization window.

#### **Show Instances**

Displays the Part and Part Target instances in the 3D Visualization window.

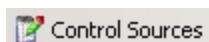
### **Process Strategy Menu**



#### **Process Strategy**

Displays the Process Strategy editor, which defines the pick-/place strategy for the Process Manager. For details, see [Process Strategy](#).

### **Control Sources Menu**



#### **Control Sources**

Displays the Control Sources editor, which defines the properties for part and target hardware sources, as required by the selected workcell configuration. For details, see [Control Sources](#).

## **Editor Parameters**

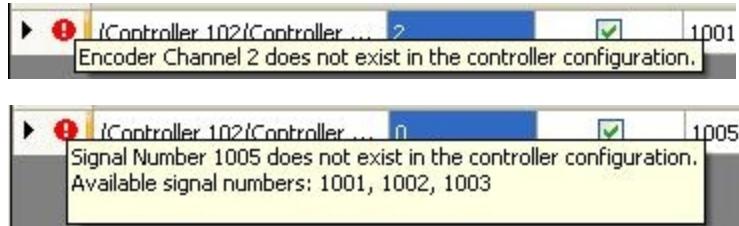
The middle portion of the editor contains the editor parameters. These are used to configure various items and settings related to the Process Manager object. The editor parameters are described below.

Item	Description
Belt Calibrations	Defines the robot-to-conveyor and vision-to-conveyor calibrations for the selected workcell process. For details, see <a href="#">Belt Calibrations</a> .

Item	Description
Configuration Items	Defines the workcell items, and the relationships between those items, that are associated with a particular workcell configuration. For details, see <a href="#">Configuration Items</a> .
Processes	Defines the elements (robot, pick location, place location) for a particular process. For details, see <a href="#">Process</a> .
Sensor Calibrations	Defines the robot-to-sensor calibrations for the selected workcell process. For details, see <a href="#">Sensor Calibrations</a> .

## Configuration Errors

If there is a configuration error, an alert icon (!) displays in the corresponding controller row. If you hover the mouse pointer over the icon, a message displays that describes the error. For example:

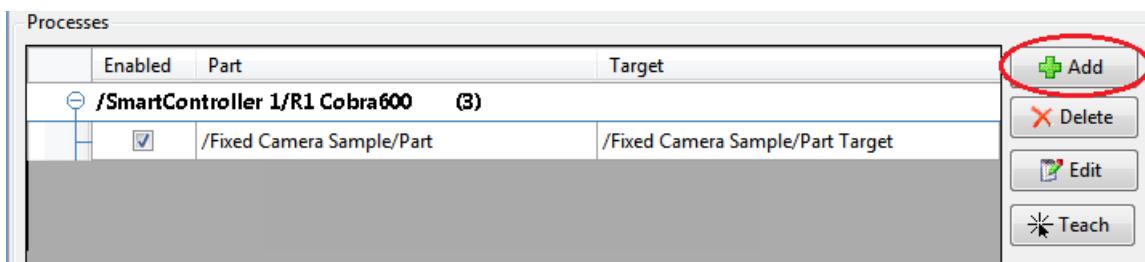


For more details on configuration messages and their associated codes, see Status/Error Codes on page 932.

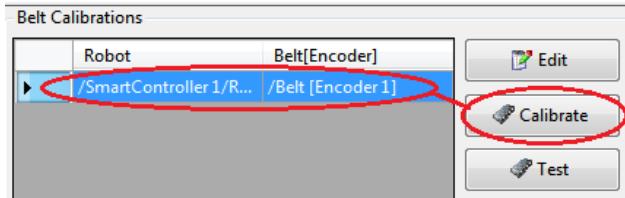
## Creating an Application

The following steps are used to create an application:

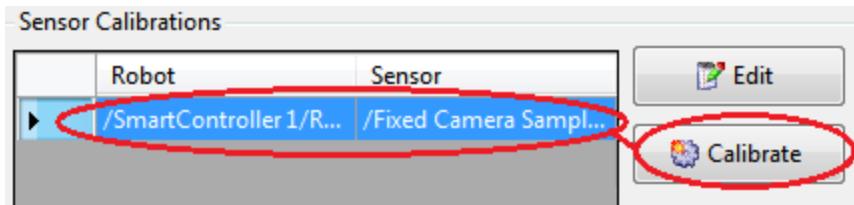
1. Create a Process. For details, see Creating a Process on page 888.



2. Perform the Belt Calibrations, as needed. For details, see Belt Calibrations on page 848.



3. Perform the Sensor Calibrations, as needed. For details, see Sensor Calibrations on page 909.



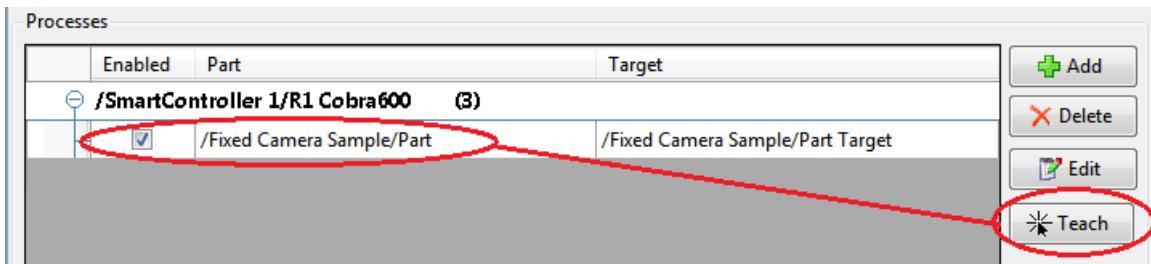
4. Edit the Process Strategy, as needed. For details, see Process Strategies on page 878.



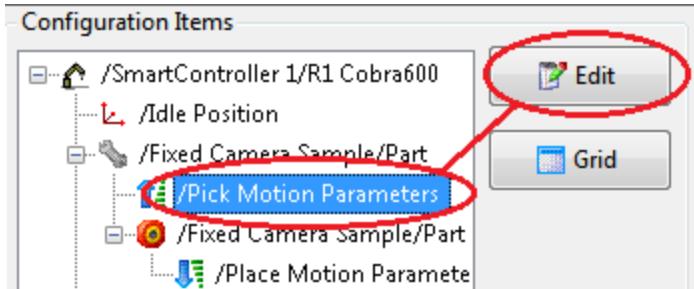
5. Select the Control Sources, as needed. For details, see Control Sources on page 869.



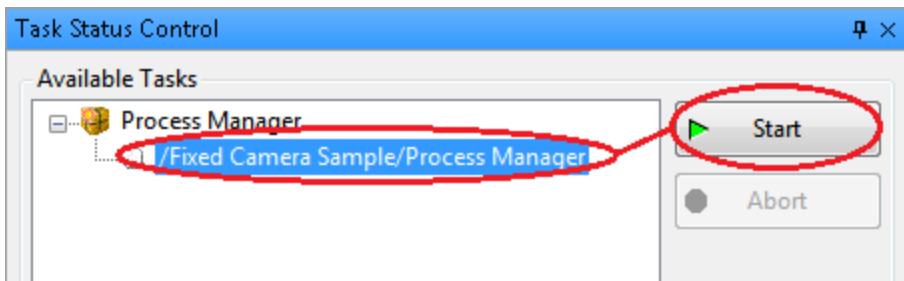
6. Teach the Process. For details, see Teaching a Process on page 894.



7. Edit the Configuration Items, as needed. For details, see Configuration Items on page 856.



- Finally, select and run the process using the Task Status Control. For details, see Task Status Control on page 747.



**NOTE:** For troubleshooting and support purposes, all Process Manager errors are recorded in the Event Log. For more details on the Event Log, see Event Log on page 1108.

## Creating Recipes

Beginning with ACE version 3.1, the ACE workspace can share robot-to-hardware calibration information between multiple Process Manager objects. This means that, if you use the same robot and hardware for each process, you can create multiple Process Manager objects in the ACE workspace without having to repeat the robot to hardware calibrations. This feature provides a form of "recipe management".

For example, let's assume you're setting up an ACE workspace to handle the packaging of various fruits. At this time, you have three fruits that you want to pack: apples, oranges, and peaches. All fruits will use the same robot, sensor, and infeed belt.

To create the packaging processes for each fruit:

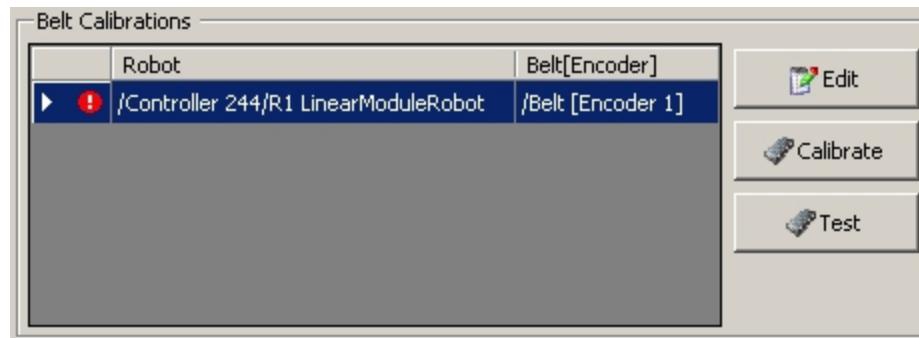
- Create a Process Manager object for apples. It could be named "Apple Packing".
- Add the Belt, Sensor, Part and Part Target objects. For details, see Objects in the Workspace on page 85 and Processes on page 886.
- Perform the Robot to Belt calibration. For details, see Belt Calibrations on page 848.
- Perform the Robot to Sensor calibration. For details, see Sensor Calibrations on page 909.

5. Optionally, edit the Process Strategy, as needed. For details, see Process Strategies on page 878.
  6. Optionally, edit the Control Sources, as needed. For details, see Control Sources on page 869.
  7. Teach the Process. For details, see Teaching a Process on page 894.
  8. Optionally, edit the Configuration Items, as needed. For details, see Configuration Items on page 856.
  9. Add a second Process manager object for oranges. It could be named "Orange Packing".
  10. Add a new Part and/or Part Target, if the pick or place requirements are different than those for apples. The Part and/or Pat Target must use the same robot, sensor and belt objects that were used with the apple-packaging process.
  11. Optionally, edit the Process Strategy and Control Sources, as needed.
  12. If a new Part and/or Part Target was added for oranges, teach the Process.
  13. Optionally, edit the Configuration Items, as needed.
  14. Repeat steps 9 - 13 for peaches.
- NOTE:** In the future, to add additional fruits, simply repeat steps 9 - 13 for each new fruit that you wish to add to the workspace.
15. Optionally, use the User Interface Designer to create an operator interface that provides easy selection of the desired fruit-packing process. For more details on the User Interface Builder, see User Interface Designer on page 1000.

## Belt Calibrations

This section describes the Belt Calibrations group in the Process Manager. For details, see Process Manager on page 843.

The Belt Calibrations group defines the robot-to-conveyor calibrations for the selected workcell process. For more details on defining a workcell process, see [Processes on page 886](#).

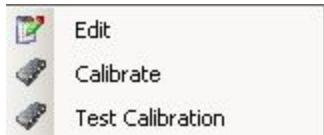


*Belt Calibrations Group*

Each of the items in the Belt Calibrations group is described in the following table:

Item	Description
Belt [Encoder]	The belt and encoder specified for the process. Double-click this item or click <b>Edit</b> to display the Belt Calibration Editor. For details, see Editing the Calibration Parameters on page 849.
Calibrate	Click to teach the selected process. For details, see Using the Calibration Wizard on page 852.
Edit	Click to edit the selected belt calibration. For details, see Editing the Calibration Parameters on page 849.
Robot	The robot specified for the belt calibration. Double-click this item or click <b>Edit</b> to display the Belt Calibration Editor. For details, see Editing the Calibration Parameters on page 849.
Test (Test Calibration)	Click to test the current belt calibration. For details, see Testing the Belt Calibration on page 854. This button is dimmed (not available) until the belt has been calibrated.

In addition to the items above, a shortcut menu is available, which duplicates the functions provided by the Edit, Calibrate, and Test Calibration buttons. To display the shortcut menu, right-click on a process in the Processes group. The following menu opens:



*Belt Calibrations Group Shortcut Menu*

For a description of the items on this menu, see the corresponding item in the previous table.

### ***Creating a Belt Calibration***

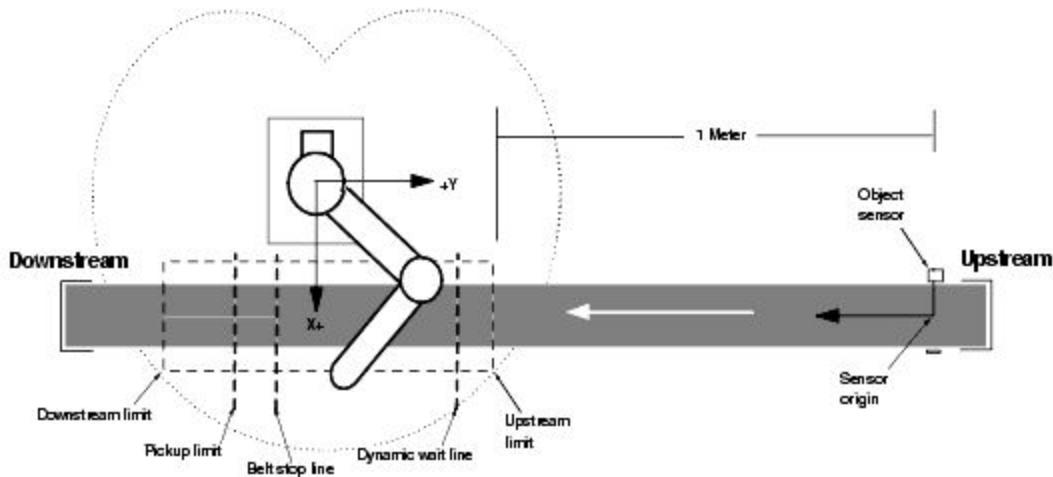
When a belt calibration is required, the Process Manager displays the Belt object name with an alert icon ( ! ) in the Belt Calibrations Group.

The belt is calibrated using the Belt Calibration wizard, which is accessed from the Calibrate button in the Belt Calibrations group. After the belt is calibrated using the wizard, the stored calibration values can be manually edited.

### ***Editing the Calibration Parameters***

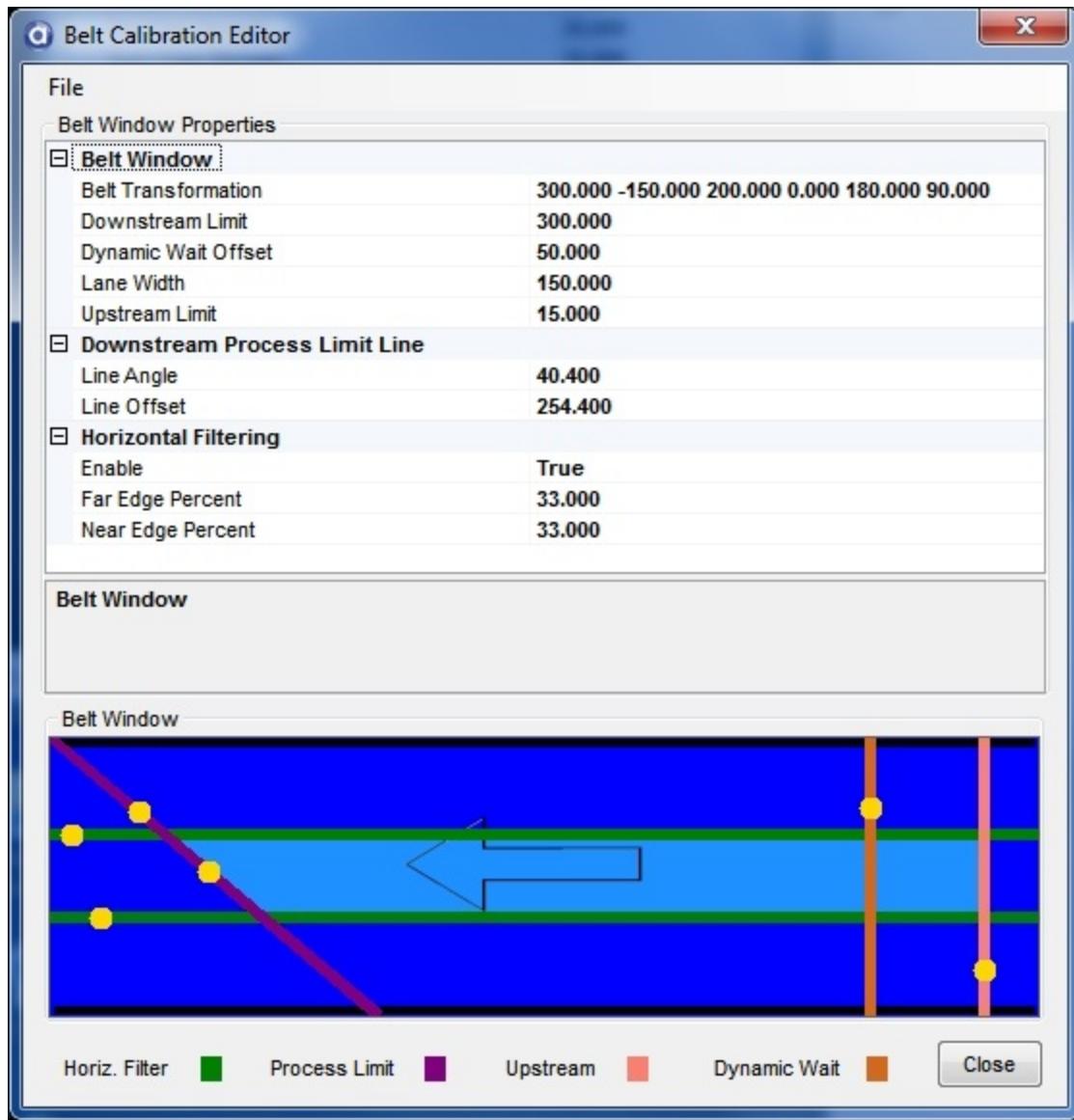
**NOTE:** The belt must be calibrated using the wizard, before the values can be manually edited. For details, see Using the Calibration Wizard in the next section.

After the belt has been calibrated using the Belt Calibration wizard, you can manually edit the stored belt-calibration parameters, such as upstream limit, downstream limit, downstream pick limit and the belt transformation. These parameters are edited using the Belt Calibration Editor. The following figure illustrates several of the Belt Calibration Editor items in a typical workcell.



*Belt Calibration Items in a Typical Workcell*

To access the Belt Calibration Editor, click **Edit** in the Belt Calibrations group. The Belt Calibration Editor opens.



*Belt Calibration Editor*

The top portion of the Belt Calibration Editor contains the Belt Properties parameters and the online help for each parameter. These are used to configure various settings on the selected conveyor.

**NOTE:** Because online help for each parameter is provided directly in the editor, it is not duplicated in this documentation.

The bottom area of the Belt Calibration Editor displays a graphical representation of the belt parameters. You can use the "handles" (blue dots) to position the lines in the graphical window. As the lines are moved, the values for the editor parameters are updated.

## Belt Filters

There are two areas of the Belt Calibration Editor parameters that control how parts are picked by the robot: the Downstream Pick Limit Line parameters and the Horizontal Filtering parameters. For more details on filtering, see Process Strategies on page 878

Item	Description
Downstream Limit Line	<p>The part positions will be compared against a line in the downstream part of the belt window (see the angled line in the previous figure).</p> <p>The parts closest to the line will be processed first. The position and angle of the line are specified using the line angle (degrees) and line offset (mm from conveyor origin) settings. You can also set the angle and position of the filter by dragging the blue dots (handles) to the desired positions.</p>
Horizontal Filtering	<p>If this is enabled (True), the pick area is limited to a subset of the width of the belt window.</p> <p>You can force different robots to pick in different horizontal regions (lanes) of the belt. For example, if you think of the conveyor belt as a three-lane highway (as shown in the previous figure), you may have robot 1 filtered to pick from the near one-third of the belt window, robot 2 filtered to pick from the middle one-third of the belt window, and robot 3 filtered to pick from the far one-third of the belt window.</p> <p>You can also set the angle and position of the filter by dragging the blue dots (handles) to the desired positions.</p>

## Saving and Loading a Calibration

After a calibration has been completed, the data can be saved by selecting **File > Save To** on the calibration editor menu. You can load a previously-saved calibration file by selecting **File > Load From** on the calibration editor menu.

## Using the Calibration Wizard

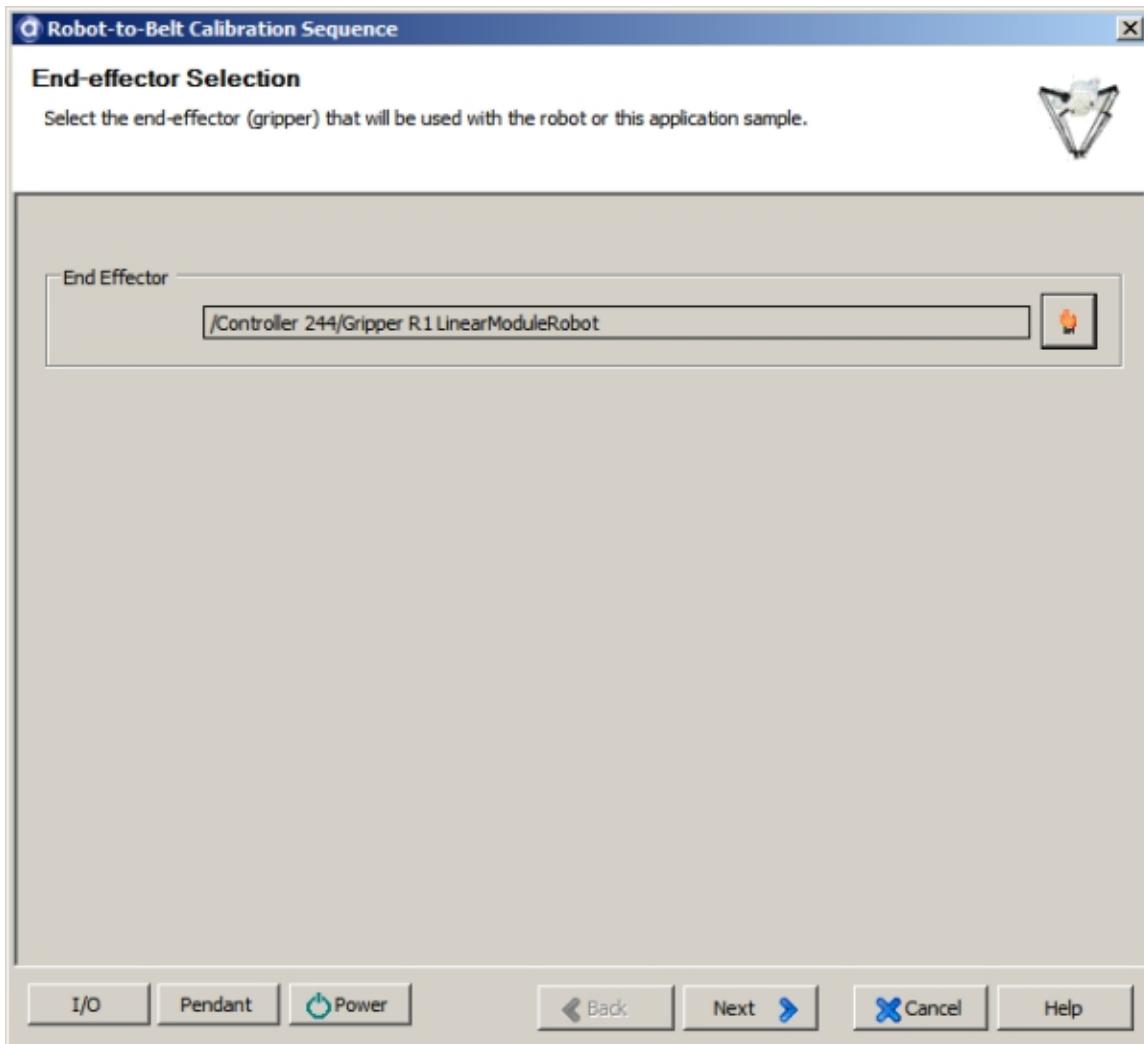
**NOTE:** This feature operates differently in emulation mode. For details on emulation mode wizards, see Emulation Mode Wizards on page 185.

The Robot-to-Belt Calibration wizard provides an interactive series of steps that guide you through the calibration process. Each step of the wizard contains graphical aids and text that describe the particular step being performed. As each step of the wizard is completed, the information from that step is collected and used to populate the fields in the Belt Calibration Editor, which is described in the previous section.

Beginning with ACE version 3.1.x, the wizards contain a "Wizard Steps" pane, on the left-hand side of the wizard screen, which shows the list of steps and the step you're currently on.

To access the Robot-to-Belt Calibration wizard:

1. Click **Calibrate** in the Belt Calibrations group. The Belt Calibration Wizard initializes. The Select the End-Effector page opens.



*Belt Calibration Wizard (Select the End-Effector)*

2. Continue by following the instructions on the wizard page.
3. When you have completed the current page, click **Next** to proceed to the next page of the wizard.

4. When you have completed the calibration procedure, the Test Belt Calibration page opens. For details, see Testing the Belt Calibration on page 854.

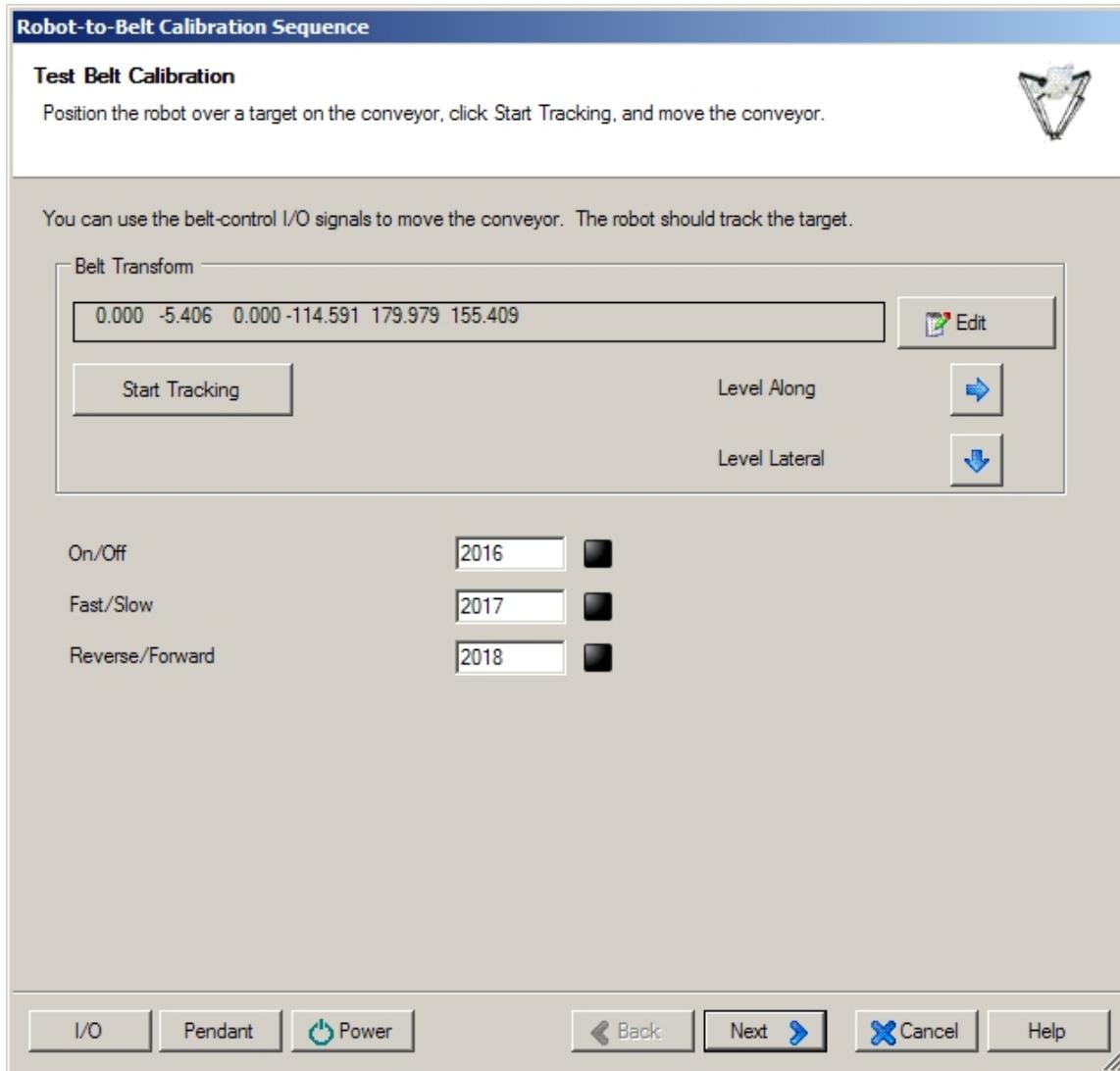
### **Testing the Belt Calibration**

The Test Belt Calibration page allows you to test the current robot-to-belt calibration.

**NOTE:** To test the calibration at a later time, you can get to this page directly by clicking **Test Current Calibration** on the shortcut menu of the Belt Calibrations group. For details, see Belt Calibrations Group Shortcut Menu on page 849.

## Belt Calibrations

---



*Belt Calibration Wizard (Test Calibration)*

**NOTE:** If the Robot Jog Control (Pendant) is open and is still in Manual (manual control) mode, the Wizard will recognize this and change the mode to "COMP" (computer control) mode.

Optionally, this page also allows you to remove the "yaw" (tilt) component from the belt transformation, as follows:

- Click the Level Along arrow to level the belt transformation along the length of the conveyor.
- Click the Level Lateral arrow to level the belt transformation along the width of the conveyor.

To test the robot-to-belt calibration:

1. Make sure the belt is turned off (so the belt is not moving).
2. Place a part on the belt.
3. Position the robot tool tip so that it is just above the center of the part.
4. On the Test Calibration page, click **Start Tracking**.
5. Start the conveyor (so the belt is moving). The robot should track the target location until it leaves the tracking region.
6. When you have finished, click **Stop Tracking** to stop the tracking.
7. Click **Next** to proceed. The Robot-to-Belt Calibration wizard closes.

The robot and belt conveyor are now calibrated.

## Configuration Items

This section describes the Configuration Items group in the Process Manager. For details, see Process Manager on page 843.

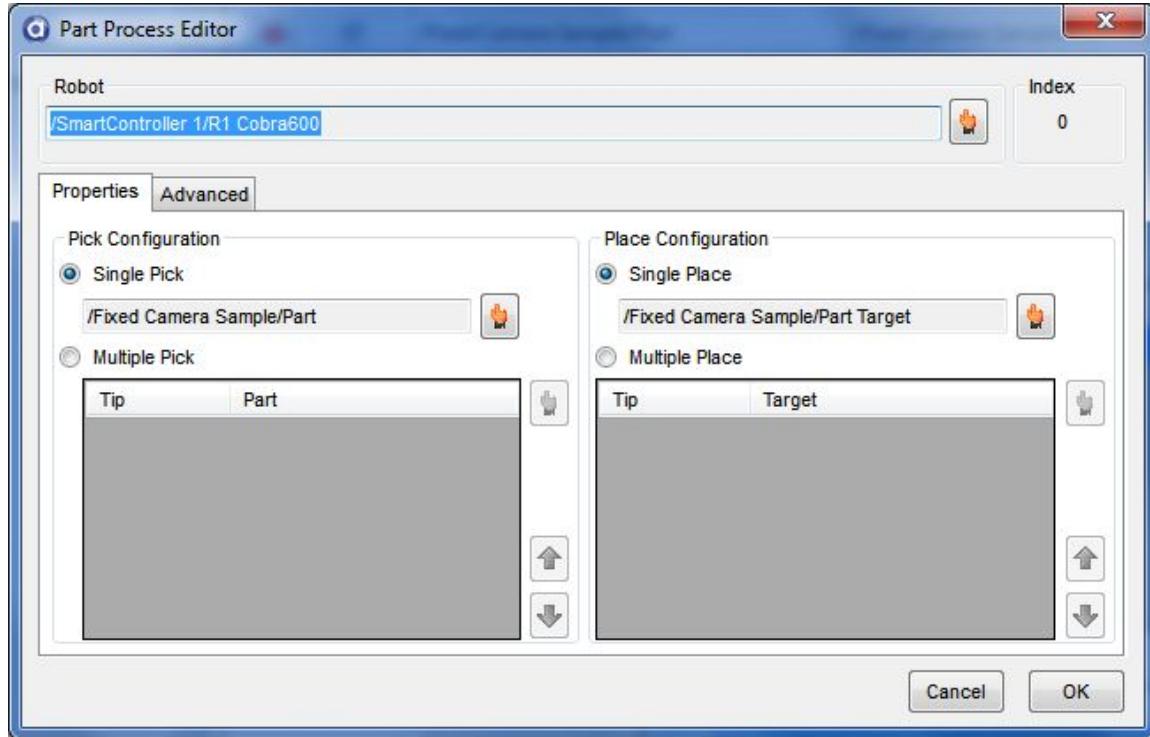
The Configuration Items group defines the workcell items, and the relationships between those items, that are associated with a particular workcell configuration. The Configuration Items group also allows quick access to the robot position (location) editors for a particular item, such as the idle, part pick, or part place location.

### *Creating the Configuration Items*

The Configuration Items are created automatically, as the workcell process is defined through the Part Process editor. As items are added/deleted in the Part Process editor, they are added/deleted in the Configuration Items group. For example, a basic pick and place application would look like this in the Part Process editor:

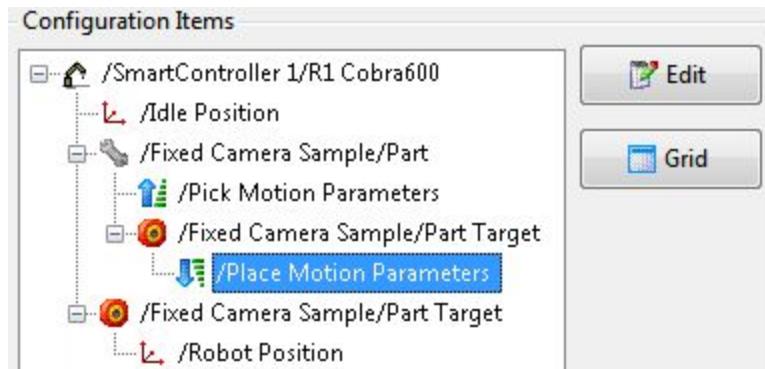
## Configuration Items

---



*Part Process Editor (Simple Pick-Place Configuration)*

The corresponding Configuration Items group looks like this:



*Configuration Items (Simple Pick-Place Configuration)*

The Configuration Items are arranged in a tree structure to show the relationships between the workcell items. The Configuration Items group contains the following features:

- You can expand or collapse a tree branch by clicking the '+' or '-' icons next to an item name.
- You can double-click any of the Position objects (idle or robot) or select the Position object and click **Edit** to open the location editor, which can be used to manually enter the object location. For details, see Location Editor Types on page 858.
- You can click **Grid** and use the Motion Sequence Grid Editor to edit the motion parameters and offset locations. For details, see Motion Sequence Grid Editor on page 866.

### ***Location Editor Types***

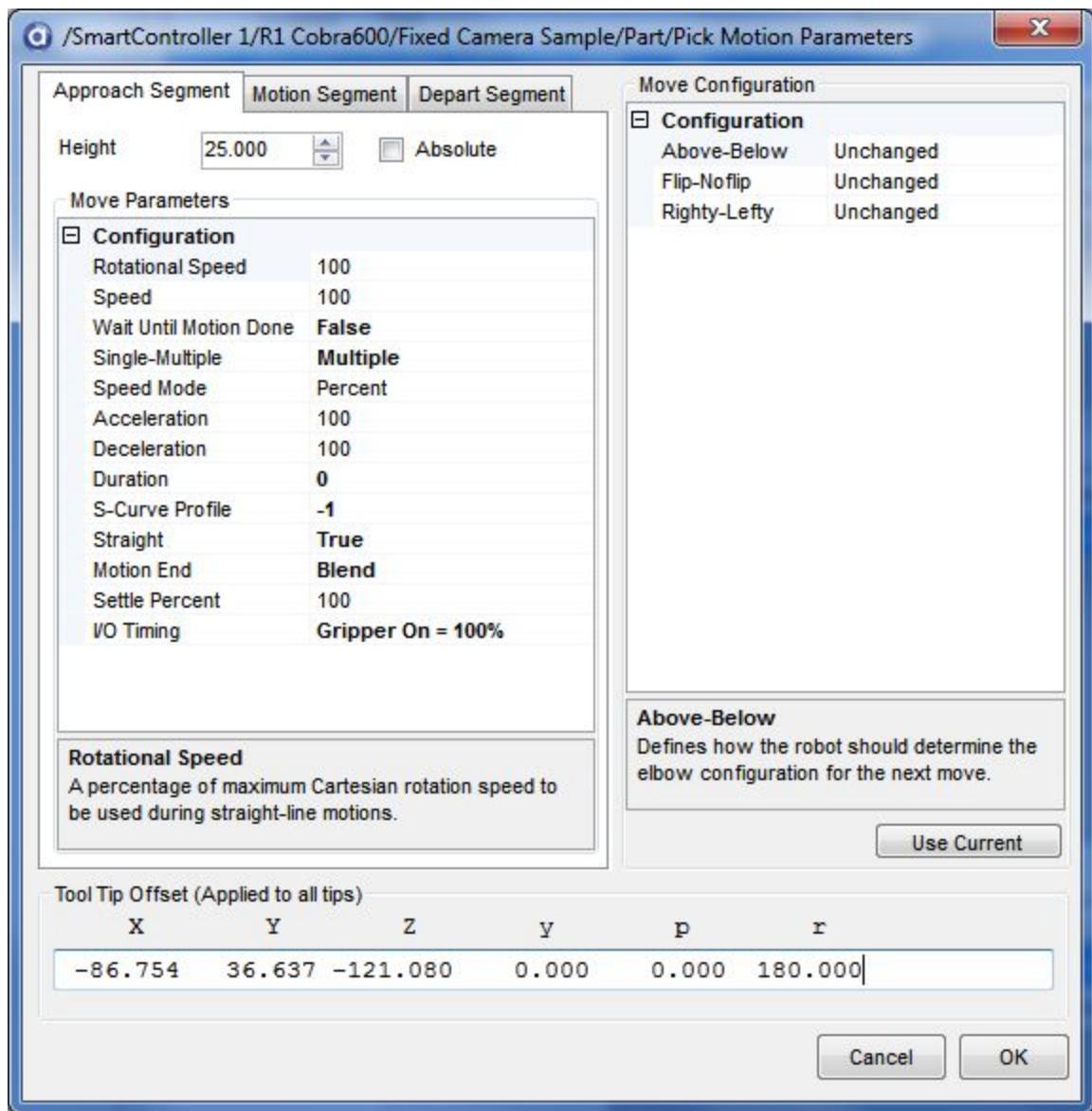
There are two types of location editors: a simple editor, which allows you to enter location information; and an enhanced position editor, which contains additional sections, such as Move Parameters, Move Configuration, Approach/Depart Parameters, etc.

For example, the Idle Position editor, shown in the following figure, is an enhanced position editor, which contains additional properties for Move Parameters and Move Configuration.

**NOTE:** The editor title bar indicates the type of parameters being edited: Idle Position, Pick Motion Parameters, or Place Motion Parameters.

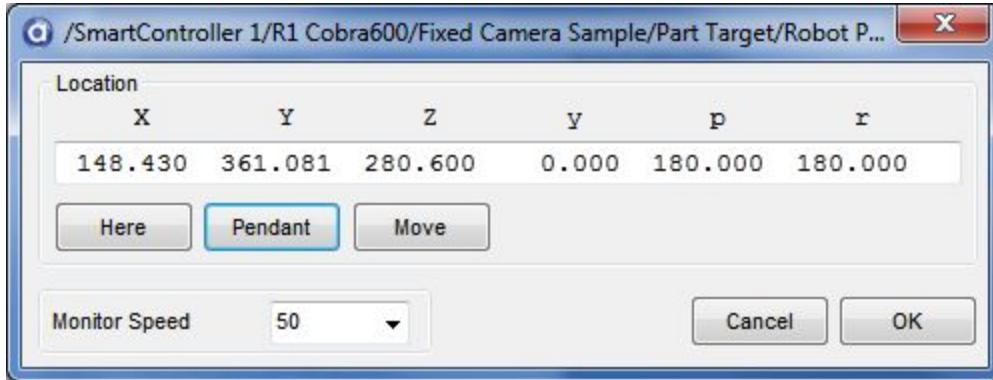
## Configuration Items

---



*Pick Position Editor (Enhanced Position Editor)*

In contrast, the Robot Position editor, shown in the following figure, is a simple position editor, which allows you to enter or teach the location information.



*Robot Position Editor (Simple Position Editor)*

### **Enhanced Editor Parameters**

The upper portion of the enhanced position editor contains the editor parameters grids. These are used to configure various settings on the selected position. The area below that displays online help for the selected parameter.

**NOTE:** Because online help for each parameter is provided directly in the editor, it is not duplicated in this documentation.

The bottom area of the enhanced position editor contains the Location editor and Monitor Speed control.

- Location editor: used to enter or teach the location information.
- Monitor Speed control: sets the speed for any moves initiated by the Move button.

The following are examples of the various enhanced parameter grids:

### **Move Configuration**

These parameters control the configuration of the robot at the selected location. For example, if your workcell contains a SCARA robot (such as the Adept Cobra s600) and you want it to be in a "lefty" configuration, you would set the Righty-Lefty parameter to "Lefty".

The Use Current button allows you to take a snapshot of the robot configuration and apply those settings to the parameters.

### **Approach/Motion/Depart Segment Move Parameters**

These parameters control how the robot moves to and from the selected Part, Vision Refinement Station, or Part Target location. They allow you to "fine tune" the robot

The Absolute option, when enabled, allows you to enter absolute Height values for the Approach and/or Depart motion segments. You can enter positive or negative values, as needed.



**WARNING:** When the Absolute option is selected, you must ensure the approach/depart heights are set correctly. Otherwise, the robot could move to an unintended location or out of range, which may damage the robot or other equipment in the work-cell.

An I/O Timing parameter is included, which controls the timing of the gripper open/close during each part of the motion segment. The I/O Timing (Gripper On) can use either a % value or a distance value, as shown in the following figures. For example, if you set the value to 25 mm, the gripper will activate at 25 mm from the pick position; if you set it at 25%, the gripper will at 25% of the distance from the approach start to the pick position. The distance value is useful when accessing pallets with multiple layers and an absolute approach height has been specified. The time value allows you to set the gripper timing (in milliseconds).

## Configuration Items

---



The Motion Segment tab also contains a Use Custom Program option, which allows you to

specify a custom V+ program that controls the motion segment.

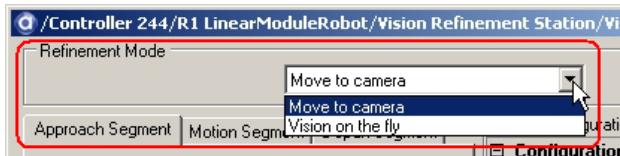
To use this option, select the check box and then click the Browse (Browse) icon to display the wizard. For more details on creating or using a custom V+ program, see Custom V+ Programs on page 937.

## Move Parameters

These parameters control how the robot moves to and from the selected location. They allow you to "fine tune" the robot motion. This is a simplified version of the Approach/Move/Depart Segment Move Parameters, described above.

- To optimize the speed of the robot, apply coarser and faster settings for less-precise motions to and from the location.
- To optimize the precision of the robot, apply finer and slower settings for smoother, more-precise motions to and from the location.

## Vision Refinement Station Parameters

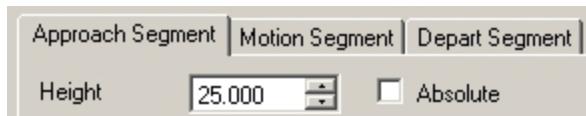


This parameter specifies how the robot moves to and from the Vision Refinement Station.

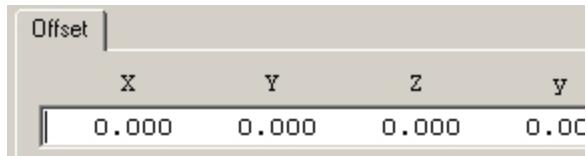
- **Move to camera:** This is a "static" refinement, where the robot pauses at the Vision Refinement Station.
- **Vision on the fly:** This is an "in motion" refinement, where the robot passes through the Vision Refinement Station without any pause in the robot motion.

**NOTE:** The "Vision on the fly" mode will provide faster throughput, but may require more lighting (to achieve a fast shutter speed) than the "Move to camera" mode.

### Move to camera refinement:



The Approach Segment, Motion Segment, and Depart Segment tabs allow you to edit the corresponding parameters for each segment of the robot motion. For details, see Approach/Motion/Depart Segment Move Parameters on page 860.

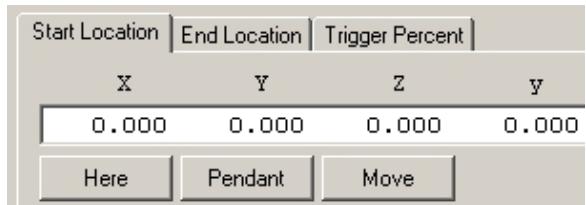


The Offset tab allows you to edit the gripper (tool) offset.

#### **Vision on the fly refinement:**



The Motion Segment tab allow you to edit the parameters for that segment of the robot motion. For details, see Approach/Motion/Depart Segment Move Parameters on page 860.



The Start Location and End Location tabs allow you to edit the start and end points for the robot path through the camera FOV.

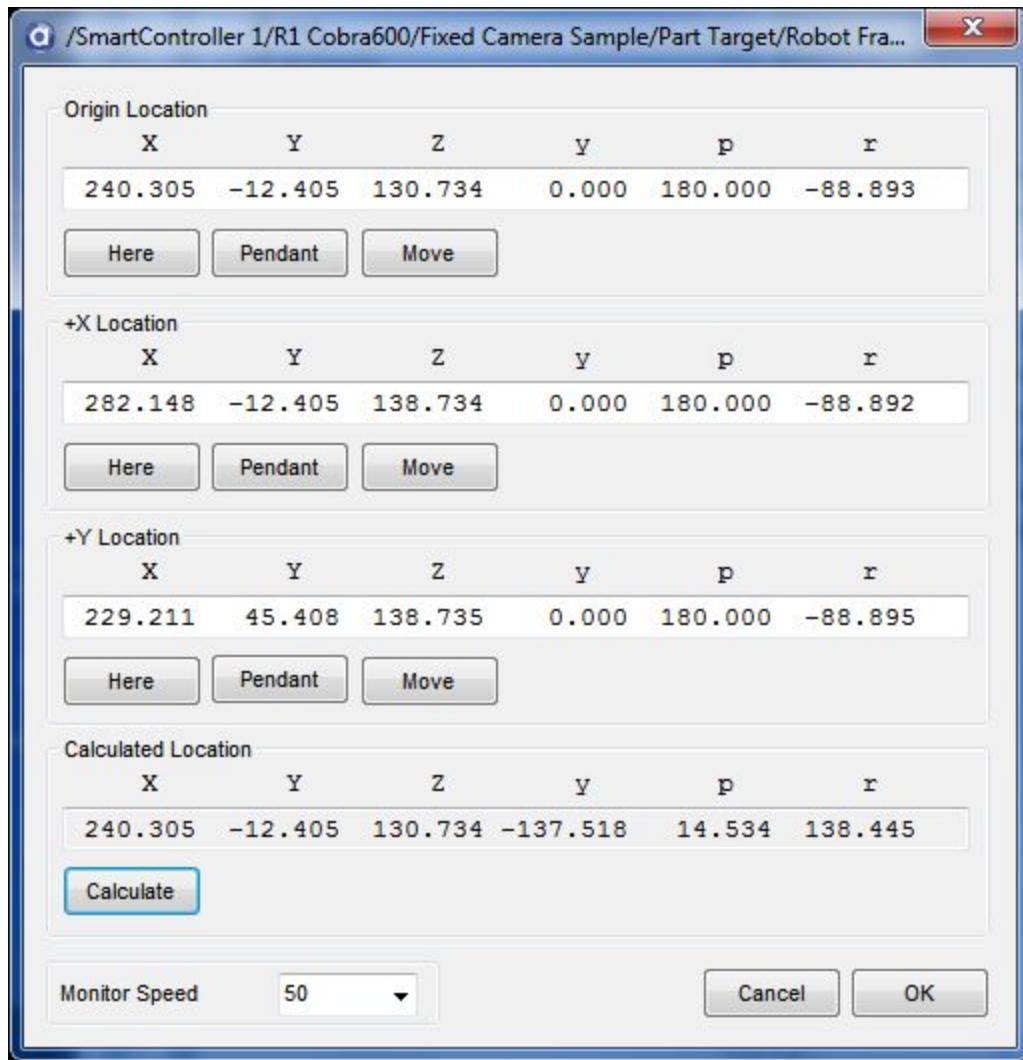
The Trigger Percent tab allows you to edit the robot-motion point (as a percent of the complete motion) where the picture request is triggered.

#### **Robot Frames**

Robot frames (also known as reference frames) are useful because they allow you to teach locations relative to the frame. If the location of the frame changes in the workcell, you simply have to update the frame information to reflect its new location. Then you can use any locations created relative to that frame without further modifications.

A process pallet is typically taught relative to a reference frame. This avoids the problem of teaching many individual pallet positions, having the pallet move for some reason, and then having to reteach all of those positions. Instead, the pallet is taught relative to a frame. If the pallet moves in the workcell, the frame position is re-taught and the part positions relative to that frame remain intact.

The Robot Frame editor is used to teach a reference frame, such as a pallet frame. For example, if you double-click a Pallet object in the Configuration Items group, the Robot Frame editor opens.



### Robot Frame Editor

To teach a robot frame:

1. Teach the Origin Location
  - a. Click **Pendant** to open the Robot Jog Control.
  - b. Use the Robot Jog Control to position the robot tool tip at the origin (in the case of a rectangular pallet, this can be the first pocket position).
  - c. Click **Here** to record the position.
2. Teach the +X Location

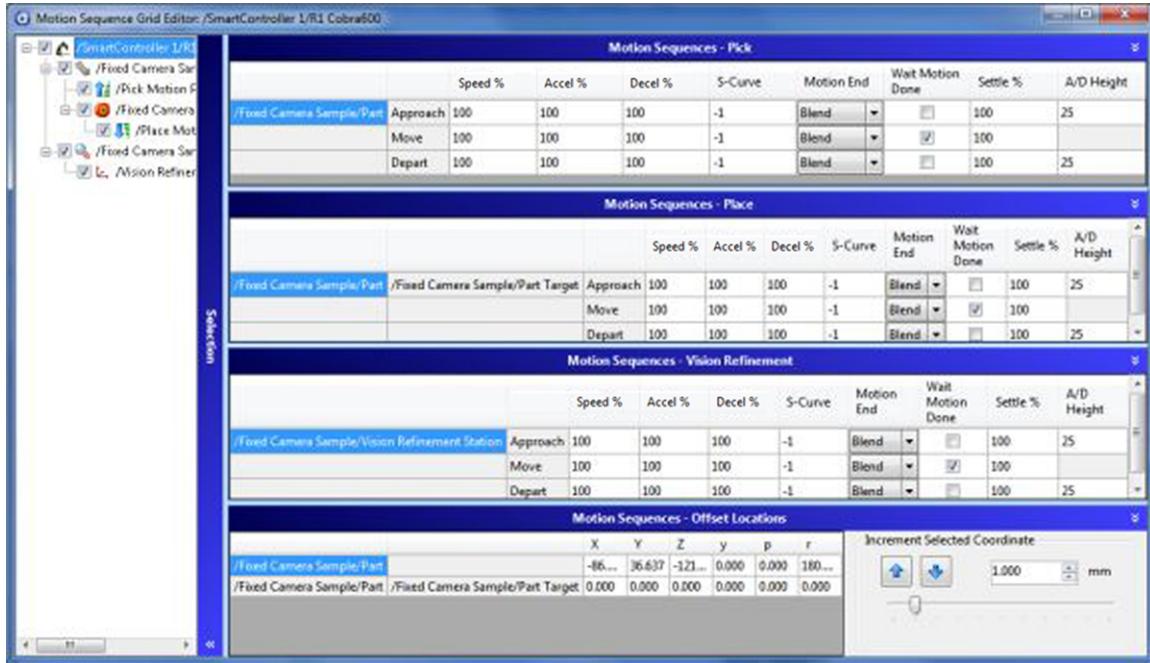
- a. Click **Pendant** to open the Robot Jog Control.
  - b. Use the Robot Jog Control to position the robot tool tip at a point on the +X axis (in the case of a rectangular pallet, this can be any pocket position along the +X axis; however, optimum results will be obtained by using a point as far away from the origin as possible).
  - c. Click **Here** to record the position.
3. Teach the +Y Location
    - a. Click **Pendant** to open the Robot Jog Control.
    - b. Use the Robot Jog Control to position the robot tool tip at a point on the +Y axis (in the case of a rectangular pallet, this can be any pocket position along the +Y axis; however, optimum results will be obtained by using a point as far away from the origin as possible).
    - c. Click **Here** to record the position.
  4. Click **Calculate** to calculate the position of the robot frame relative to the robot.
  5. Click **OK** to close the Robot Frame editor.

### **Motion Sequence Grid Editor**

The Motion Sequence Grid Editor provides a spreadsheet-like (grid) interface that allows you to edit:

- Pick motion parameters
- Place motion parameters
- Vision refinement parameters
- Offset location parameters

To access the grid editor, click **Grid** in the Configuration Items group. The Motion Sequence Grid Editor opens.



### Motion Sequence Grid Editor

The left pane of the grid editor is used to select the items you wish to display for editing. The right pane of the editor contains the three editing groups (Pick, Place, and Offset Location).

### Editing an Item in the Grid Editor

There are two ways to edit the items in the grid:

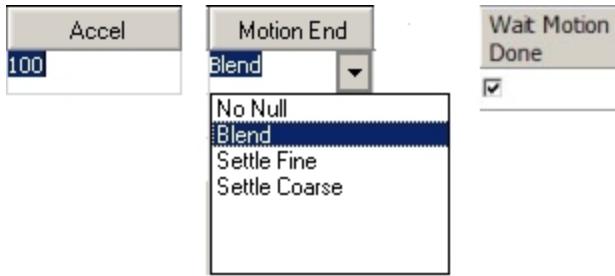
- edit a single item in the grid
- select multiple items and "bulk edit"

To edit a single item in the grid editor

- Click the cell that contains the item you wish to edit.
- If the item is a direct-edit value, you can simply type the desired value into the cell.

If the item is a predefined-list item, click the drop-down arrow to display a list of predefined values, which you use to select the desired value for the cell. Optionally, you can type the desired value, if you know the valid values for that cell.

If the item is a check box item, you can click the check box to select or clear it.

*Direct-edit, Predefined-list, and check box Items*

3. After you have completed editing the values on the grid, you can close the editor by clicking the close icon ( in the upper-right corner of the grid editor window.

To edit multiple items in the grid editor

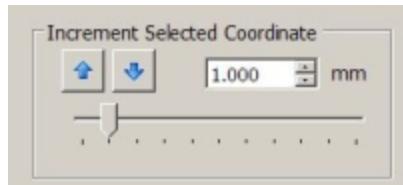
1. Select the cells you wish to edit:
  - Select non-adjacent cells: Press the Ctrl key while clicking the cells
  - Select a series: Click the first cell and then press the Shift key while selecting the last cell; click in the first cell and drag to select nearby cells
  - Select a series: Click in the first cell and drag to select nearby cells

**NOTE:** The selected cells must be the same type.

2. Type the desired value for the selected cells. All cells are updated to show the new value.
3. After you have completed editing the values on the grid, you can close the editor by clicking the close icon ( in the upper-right corner of the grid editor window.

## Offset Controls

The Offset Locations section contains an additional editing control, as shown in the following figure:



### Offset Controls

The control allow you to increment the selected offset coordinates, as follows:

1. Use the mm field or slide control to enter the desired increment value
2. Use the Up or Down arrows ( ) to apply the increment value to the selected cells in the Offset

Locations grid. The Up arrow increases the value by the increment amount; the Down arrow decreases the value by the increment amount.

## Control Sources

This section describes the Control Sources editor in the Process Manager. For details, see Process Manager on page 843.

The Control Sources editor defines the properties for part and target hardware sources, as required by the selected workcell configuration. For example, if you have a part that is being fed on a conveyor belt, you can use the Control Sources button to access the Source Configurations editor, which allows you to configure (select) the controller for that belt and, optionally, a custom V+ belt-control program, as shown in the next section.

### ***Editing a Control Source***

As you add items to the workspace, any item that can have a control source (for example, a conveyor belt) is automatically added to the Source Configurations editor list.

To edit a control source:

1. On the Process Manager menu, click the Control Sources menu item.



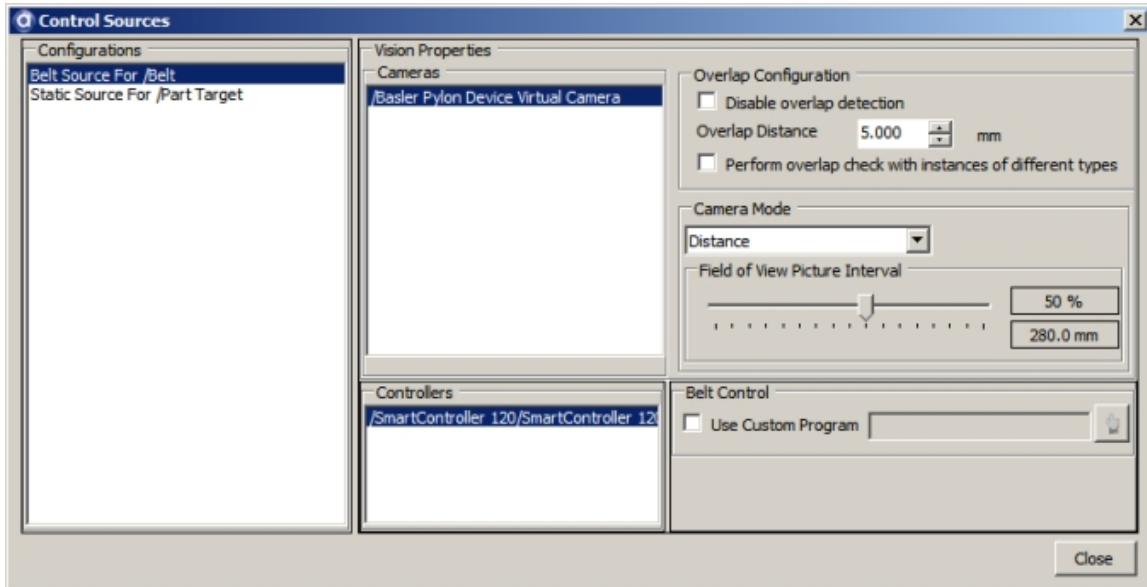
2. The Control Sources editor opens.
3. Select an item for editing. For example, in the following figure, the Belt Source is selected, which displays the corresponding control parameters.
4. Edit the parameters for the selected item.
5. Click **OK** to close the editor.

### Belt Control Sources

**NOTE:** When emulation mode is enabled, this item contains additional features. For more details on Emulation Mode, see Emulation Mode on page 1089.

## Control Sources

---



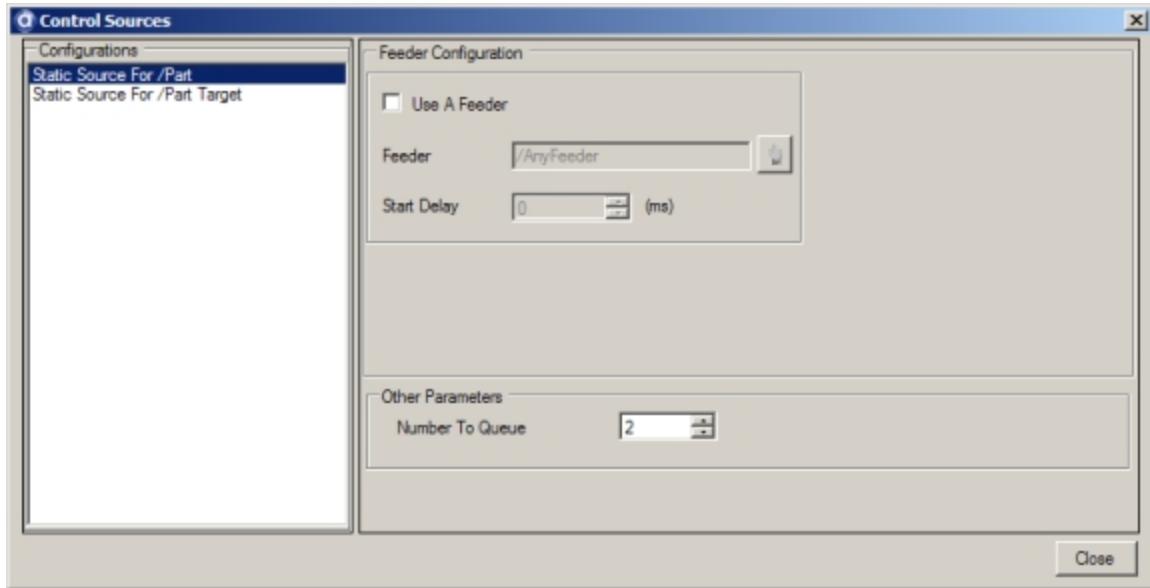
*Control Sources Editor - Belt Control Example*

Item	Description
Configurations	Select the item in the Configurations list that you want to edit. Depending on the type of item selected, the options will change to show the appropriate control sources and options for that item. In this example, a conveyor belt (/Belt) has been selected - the available Controllers and Belt Control program options are displayed for that item.
Cameras	If there are multiple cameras defined in the workcell, these will appear as available cameras on the Cameras list. You can change the Vision Properties for the selected camera. For details, see the description of Vision Properties.
Controllers	If there are multiple controllers defined in the workcell, these will appear as available controllers on the Controllers list. You can change the selected controller here, or you can change it in the <a href="#">Belt editor</a> .
Camera Mode	There are two camera modes available: Distance and Trigger. When Distance is selected, the Field of View Picture interval control is enabled; when Trigger is selected, the Trigger Signal control is enabled.  When emulation mode is enabled, an additional setting (Trigger Period in Emulation Mode) is displayed. This allows you to specify the trigger period in millimeters. For more details on emulation mode, see Emulation Mode on page 1089..
Field of View Picture	This control is used to adjust the picture interval relative to belt travel. The

Item	Description
Interval	setting is displayed in millimeters (mm) and as a % of the field of view.
Trigger Signal	Specifies the signal number to use for the vision trigger. When the specified trigger signal number is activated, a new vision image will be acquired.
Use Custom Program	If you want to use a custom V+ program to control the selected belt, select Use Custom Program in the Belt Control group. For details, see Control Sources on page 869.
Overlap Configuration	When a part is located in an image associated with a conveyor, the position of the object is compared with the position of objects located in previous pictures that were acquired at upstream belt positions.  If Disable Overlap Check is selected, all overlap checking is disabled. When this option is selected, the remaining Overlap Configuration items are not available (grayed-out).  If the newly-located part is within the specified Overlap Distance of a previously-located part (accounting for belt travel), the part is not considered a new part but, rather, a previously-located part that has advanced along the belt.  If "Perform overlap check with instances of different types" is selected, the overlap calculation will check for overlap of any parts, rather than just parts of the same type.

The following are other Control Sources examples:

## Static Control Sources

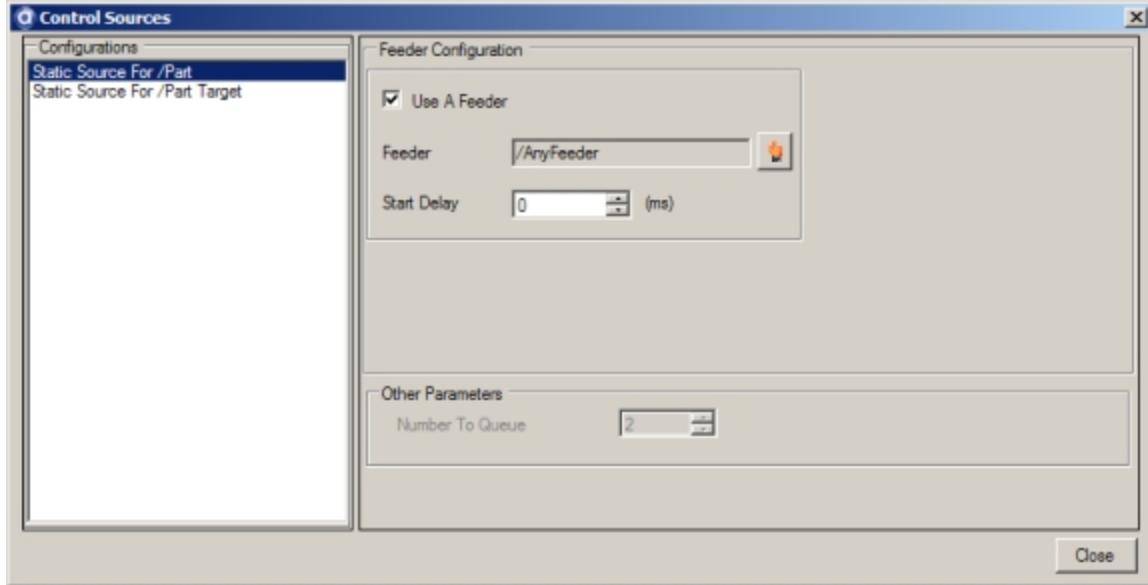


*Control Sources Editor - Static Control Example*

The static control source is for a part or part target that is not being fed on a belt or from a feeder.

Item	Description
Number to Queue	Specifies the number of part instances that can be sent to the controller (the default value is 2). This allows the PC to push the specified number of part instances to the controller to overcome minor disturbances in the communications flow between the controller and PC.

## Feeder Control Sources



*Control Sources Editor - Feeder Control Example*

The feeder control source is for a part or part target that is using a feeder (for example, a feeder could be a bowl feeder or an indexing conveyor belt). Note that the Feeder Configuration "Enable" option must be selected. If you are using an IO Feeder (for example, a bowl feeder), see IO Feeder Object on page 148.

The Feeder Configuration items are described in the following table:

**NOTE:** In ACE version 3.1.x and later, the Feeder Configuration items are located in the IO Feeder Control. For details, see IO Feeder Object on page 148.

Item	Description
Use A Feeder	Enables the selected feeder.
Feeder	Specifies the feeder. Click the Browse (  ) icon to select the feeder.
Start Delay	Specifies the delay (in milliseconds) before the feeder is activated. This delay can be used to ensure the robot has moved out of the pick/image area before the feeder is activated.

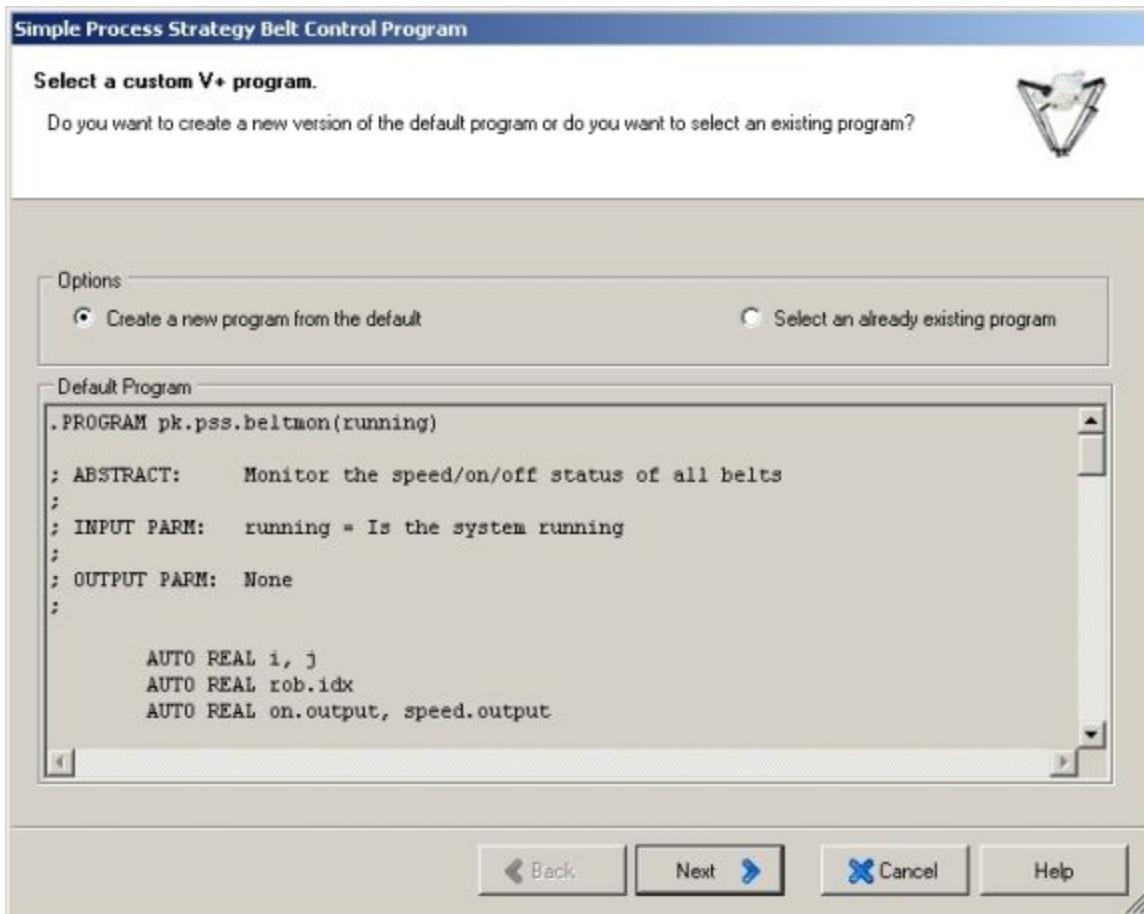
**NOTE:** When the Feeder Configuration option is enabled, the Other Parameters - Number To Queue option is disabled.

## To Create or Select a Custom V+ Program

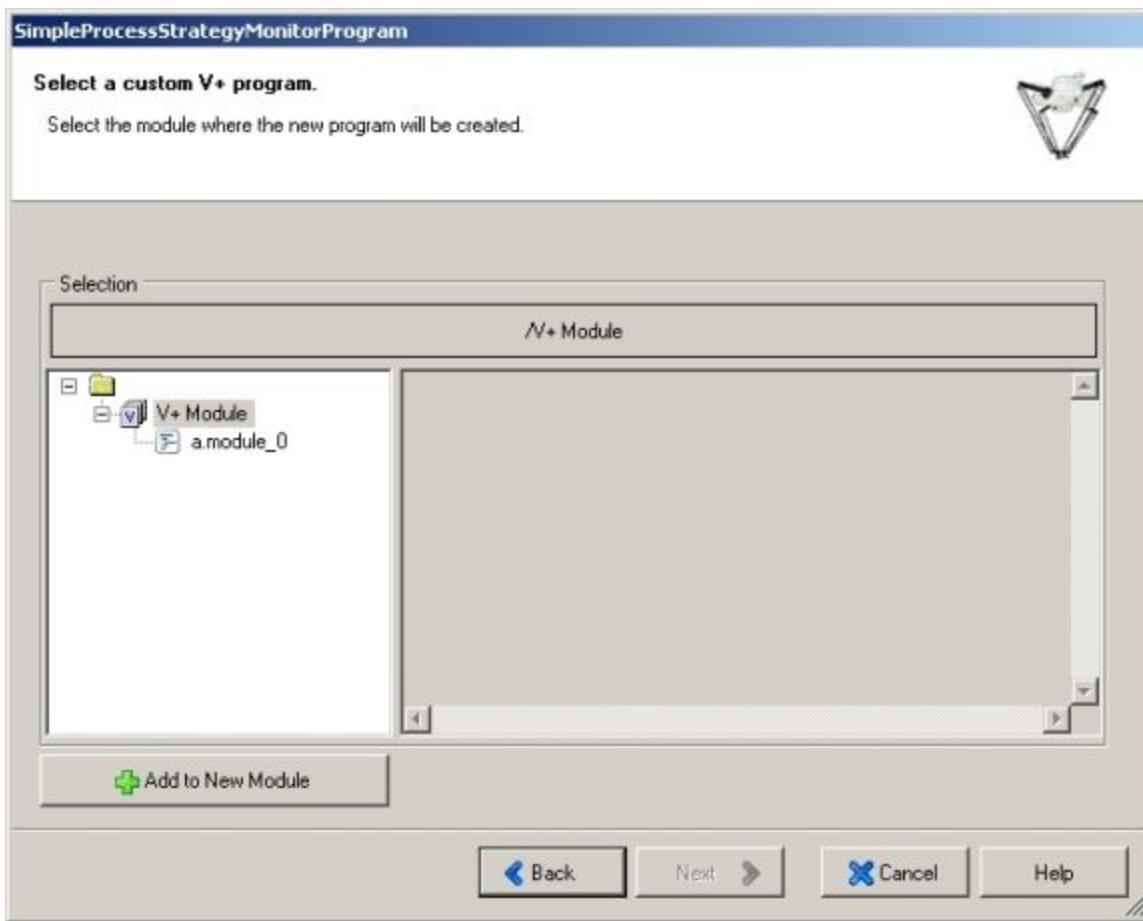
When a "custom" program option is selected, you can enter the name of the custom program, or click the Browse (  ) icon to create a new program or locate an existing program.

### To create a new program:

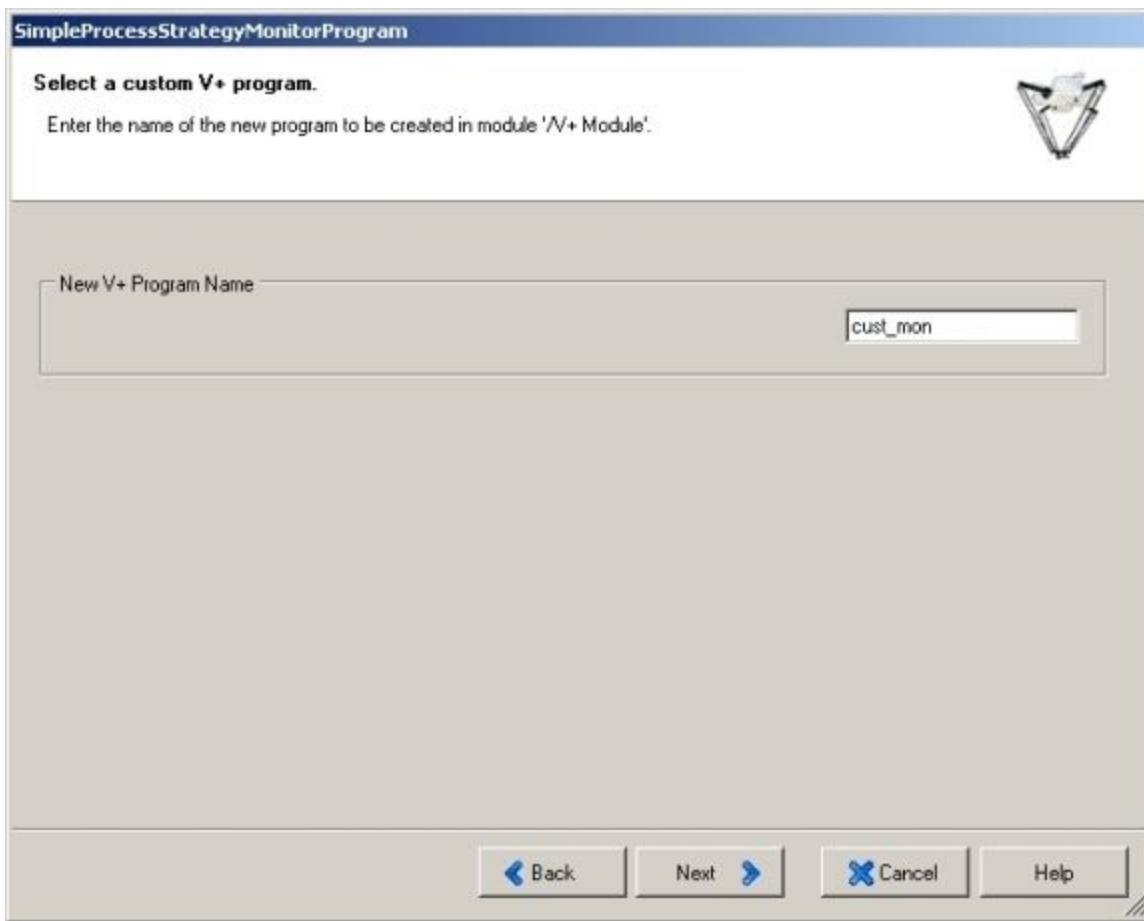
1. Click the Browse icon. A screen similar to the following is opened:



2. Use the radio button options to Create a new program from the default (the default program can be viewed in the Default Program window)
3. Click **Next** to continue. The following screen opens.



4. Select an existing V+ module where the new program will be created, or click **Add to New Module** to create a new V+ module for the program.
5. Click **Next** to continue. The wizard displays a prompt for the new V+ program name. Type the name for the program.

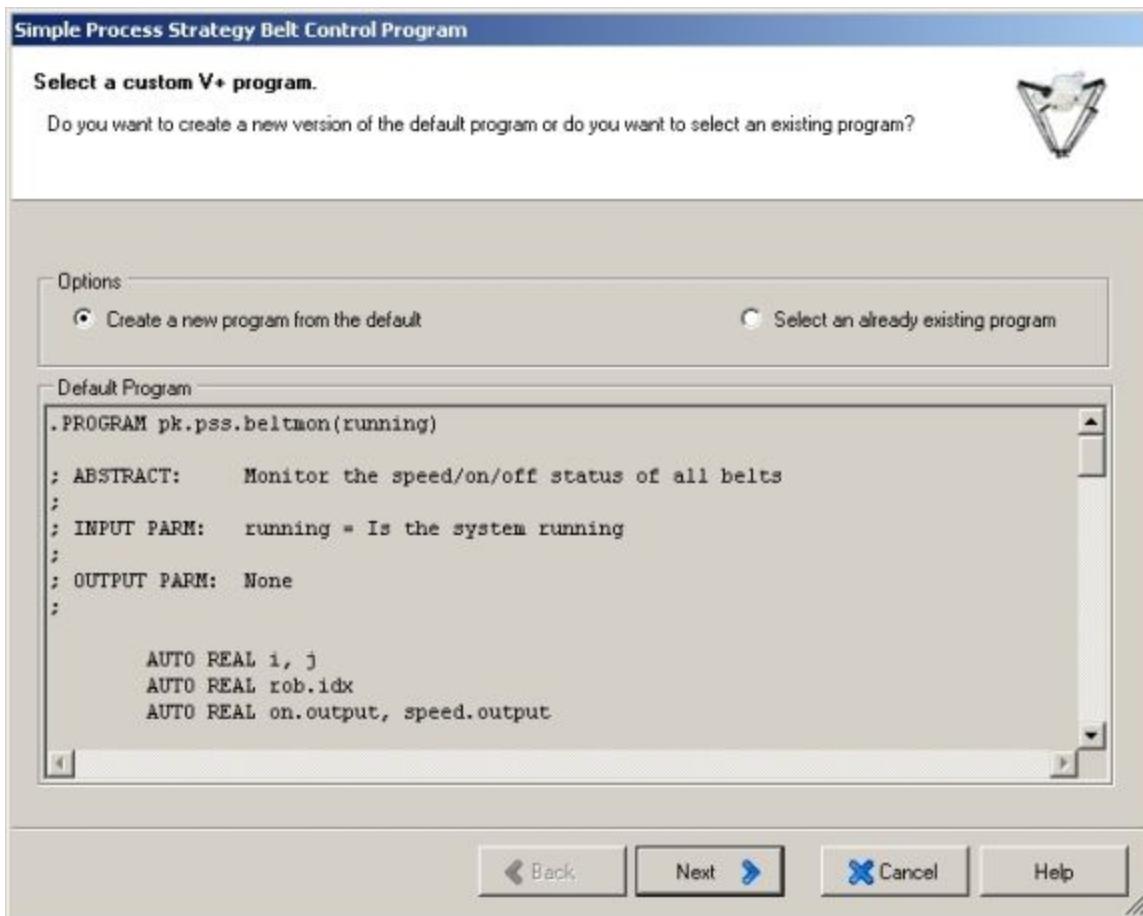


6. Click **Next**. The wizard closes and the custom program name displays in the program name field.

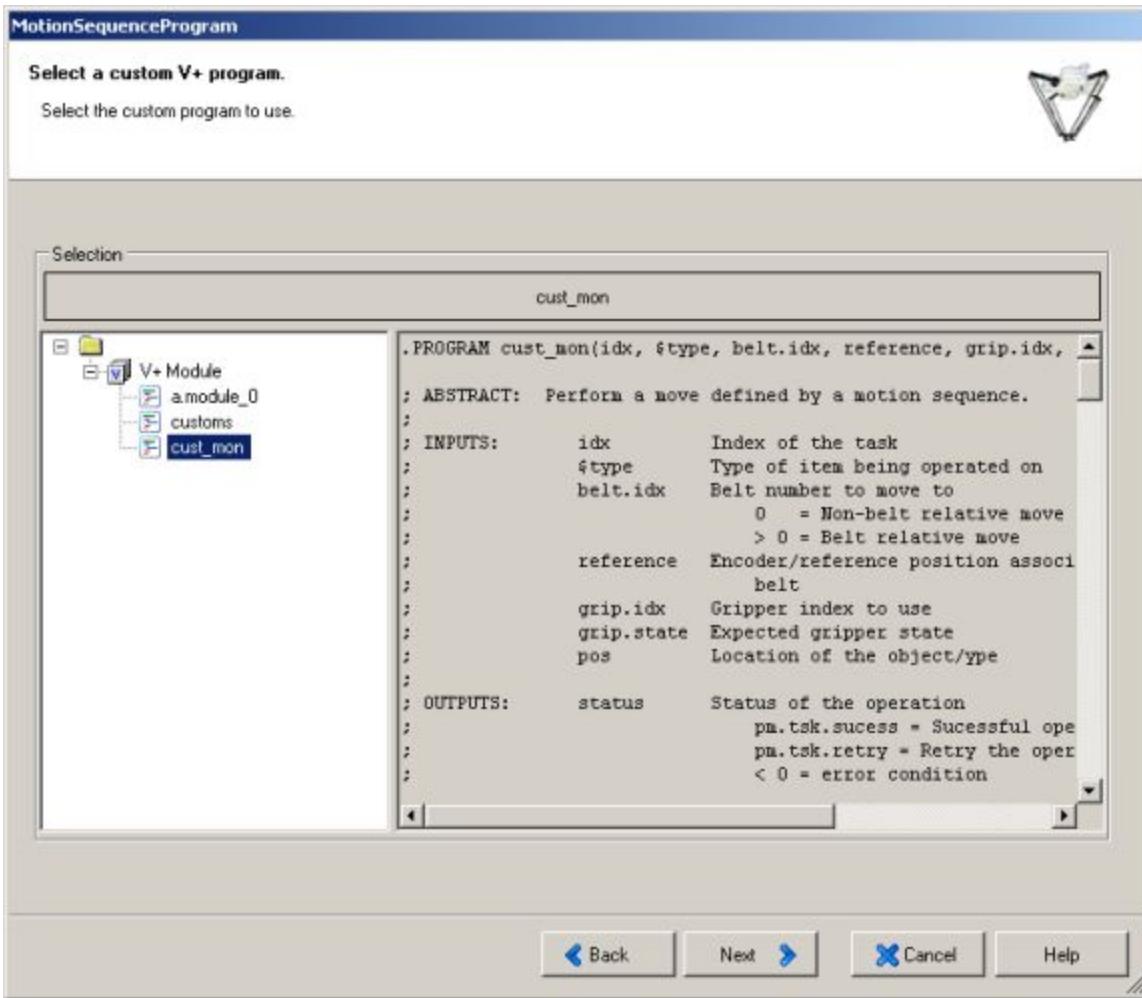


**To select an existing program:**

1. Click the Browse icon. A screen similar to the following is opened:



2. Click the option Select an already existing program .
3. Click **Next** to continue. The following screen opens.



4. Select the program from the module/program list.
5. Click **Next** to continue. The wizard closes and the custom program name displays in the program name field.



## Process Strategies

This section describes the Process Strategy editor in the Process Manager. For details, see Process Manager on page 843.

The Process Manager invokes a Process Strategy to determine how to allocate the Parts and Part Targets identified by the Process Manager. It uses the list of Part Processes to allocate the Parts and Part Targets to

specific robots. The output of this process is passed to the controller queue by the Process Manager. Each Process Strategy operates under certain assumptions based on the process being monitored. Those assumptions determine which algorithms are used to perform the allocation. For more details on Process Strategy, see Line Balancing on page 804.

The Process Strategy Editor provides access to the following parameters editors:

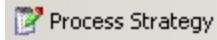
- Controller Parameters on page 880
- Robot Parameters on page 881
- Belt Control Parameters on page 885

The appropriate editor is shown based on the object selected in the left pane of the Process Strategy Editor.

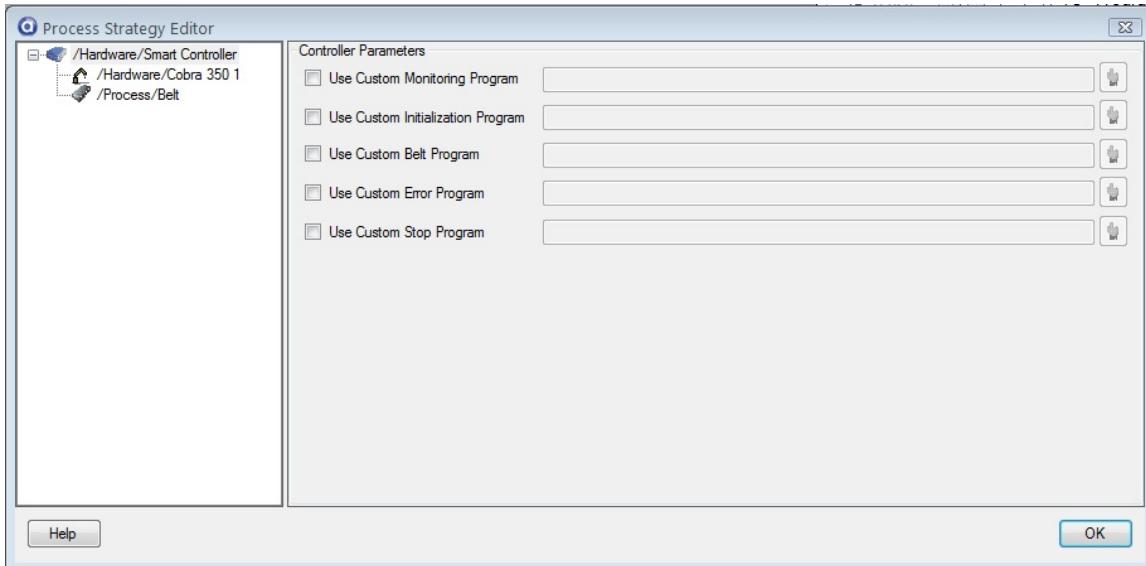
### ***Editing a Process Strategy***

To edit a Process Strategy:

1. On the Process Manager menu, select the Process Strategy menu item.



2. The Process Strategy Editor opens.



*Process Strategy Editor*

3. Select an item for editing. For example, in the previous figure, the controller is selected, which displays the Controller Parameters.

4. Edit the parameters for the selected item.
5. Click **OK** to close the editor.

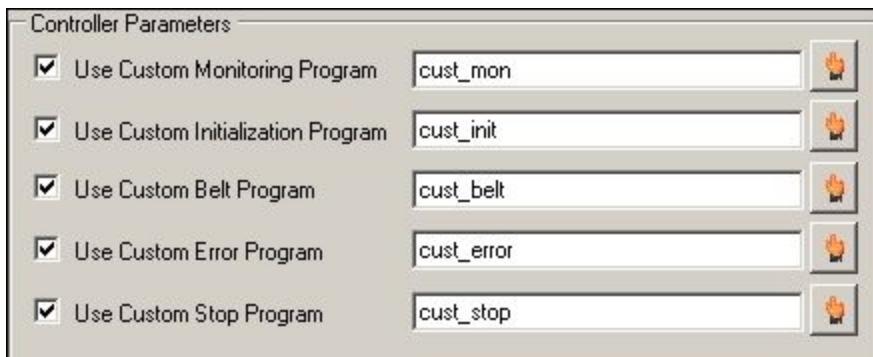
### **Parameters Editors**

The parameters area of the Process Strategy Editor changes, based on the selected item. The available parameter editors are described in this section.

### **Controller Parameters**

The Controller Parameters are displayed when the controller is selected in the Process Strategy Editor.

The Controller Parameters group is used to specify custom V+ programs for the selected controller.



#### *Controller Parameters*

For this screen, the parameters are:

Item	Description
Use Custom Monitoring Program	<p>The default process monitoring:</p> <ul style="list-style-type: none"> <li>• Checks for updates to process strategies</li> <li>• Handles belt monitoring</li> <li>• Monitors parts and part targets</li> <li>• Checks that the ACE server is running</li> </ul> <p>You can copy the default V+ monitoring program for editing, or select an existing program.</p>
Use Custom Initialization Program	<p>The default initialization program initializes the belt control parameters and application before the control programs (robot, belt, process strategy, etc.) are started. You can copy the default V+ initialization program for editing, or select an existing program.</p>

Item	Description
Use Custom Belt Program	The default belt program monitors the speed/on/off status of all belts. You can copy the default V+ belt program for editing, or select an existing program.
Use Custom Error Program	The default error program handles the processing and reporting of robot errors during the execution of a process. You can copy the default V+ error program for editing, or select an existing program.
Use Custom Stop Program	The custom stop program can be used to perform certain operations after the application has stopped. You can copy the default V+ stop program for editing, or select an existing program.

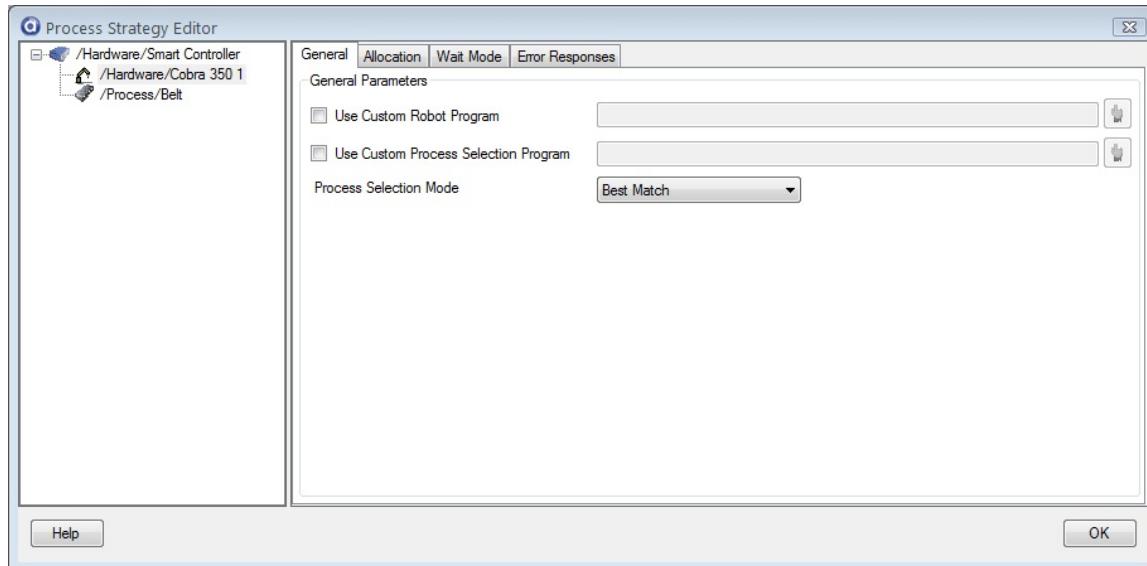
For more details on creating or using a custom V+ program, see [Custom V+ Programs](#) on page 937.

## Robot Parameters

The Robot Parameters are displayed when the robot is selected in the Process Strategy Editor.

There are four groups of robot parameters: the General Parameters group, the Allocation group, the Wait Mode Parameters group, and the Error Response group.

The General Parameters group is used to specify the queue size and pick rate filter for the selected robot; the Wait Mode Parameters group is used to specify the "wait" settings (used when there are no parts or targets to process).



### Robot Parameters

For this screen, the parameters are:

Item	Description								
<b>General Parameters:</b>									
Use Custom Robot Program	<p>Allows you to specify a custom V+ main robot-control program. For example, this would allow you to customize exactly when a given robot executes each process.</p> <p>If this option is selected, a wizard guides you through the program-creation process. For details, see Process Strategies on page 878.</p>								
Use Custom Process Selection Program	<p>Allows you to specify a custom V+ process-selection program. For example, this would allow you to customize the selection and ordering of the processes.</p> <p>If this option is selected, a wizard guides you through the program-creation process. For details, see Process Strategies on page 878.</p>								
Process Selection Mode	<p>Specifies the process-selection mode used by the robot.</p> <table border="1"> <thead> <tr> <th style="background-color: #cccccc;">Mode</th><th style="background-color: #cccccc;">Description</th></tr> </thead> <tbody> <tr> <td>Best Match</td><td>Evaluates all process on the list. It gives priority to the process with belt-relative parts or targets whose product is furthest downstream in the belt window of the associated robot. If no process is belt-relative, it will select the first available process on the list.</td></tr> <tr> <td>As Ordered (no Timeout)</td><td>Selects the processes in the order they are listed. For example, if three processes are listed, it will select process 1, then process 2, and then process 3. The currently-selected process must be completed before the next one can begin. When the last process has completed, the list repeats.</td></tr> <tr> <td>As Ordered with Timeout</td><td>Selects the processes based on their order <i>and</i> user-supplied timeout (the timeout is specified in milliseconds). Normally, it selects the next process in the list, but if the process cannot be selected within the given the timeout, it will move to the next process for possible selection.</td></tr> </tbody> </table>	Mode	Description	Best Match	Evaluates all process on the list. It gives priority to the process with belt-relative parts or targets whose product is furthest downstream in the belt window of the associated robot. If no process is belt-relative, it will select the first available process on the list.	As Ordered (no Timeout)	Selects the processes in the order they are listed. For example, if three processes are listed, it will select process 1, then process 2, and then process 3. The currently-selected process must be completed before the next one can begin. When the last process has completed, the list repeats.	As Ordered with Timeout	Selects the processes based on their order <i>and</i> user-supplied timeout (the timeout is specified in milliseconds). Normally, it selects the next process in the list, but if the process cannot be selected within the given the timeout, it will move to the next process for possible selection.
Mode	Description								
Best Match	Evaluates all process on the list. It gives priority to the process with belt-relative parts or targets whose product is furthest downstream in the belt window of the associated robot. If no process is belt-relative, it will select the first available process on the list.								
As Ordered (no Timeout)	Selects the processes in the order they are listed. For example, if three processes are listed, it will select process 1, then process 2, and then process 3. The currently-selected process must be completed before the next one can begin. When the last process has completed, the list repeats.								
As Ordered with Timeout	Selects the processes based on their order <i>and</i> user-supplied timeout (the timeout is specified in milliseconds). Normally, it selects the next process in the list, but if the process cannot be selected within the given the timeout, it will move to the next process for possible selection.								
<b>Allocation Parameters:</b>									
Part or Target Filtering Mode	<p>Specifies how instances are identified for processing by this robot:</p> <ul style="list-style-type: none"> <li>• <b>No Filtering</b>All instances are sent to the robot</li> <li>• <b>Pick / Skip Instances</b>Processes a certain number of instances then skip a certain number of instances.</li> <li>• <b>Percentage of Instance</b> Allocate a percentage of instances evenly across the range of available instances.</li> </ul>								

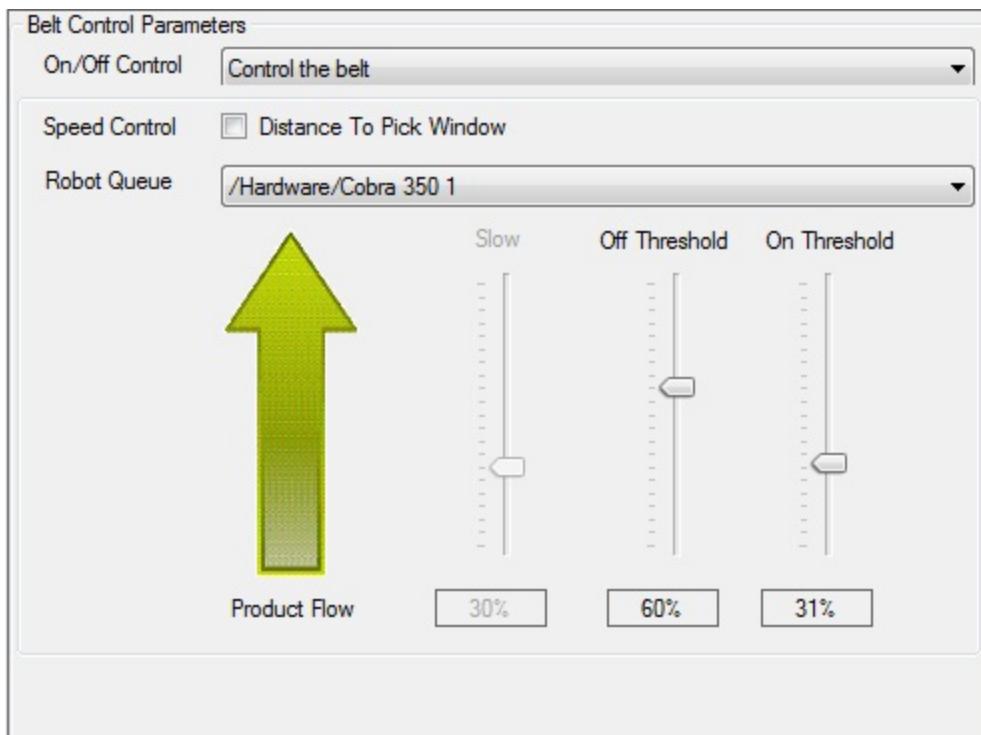
Item	Description
	<p><b>Skip Rate</b> The robot should process instances to maintain an even flow of instances not processed.</p> <p><b>Relative Belt Position</b> Process instances that are separated by a certain distance.</p> <p><b>Pick / Skip Pallets</b> Process a certain number of pallets then skip a certain number of pallets.</p>
Queue Size	<p>Specifies the queue size for the robot. Each robot has a queue size, which represents the maximum number of parts (Part instances) or targets (Part Target instances) that can be sent to the robot. <i>Each part and target is associated with a different queue for a given robot. Therefore, this parameter defines the size of each queue.</i> This parameter defaults to 10.</p> <p>To change the parameter, enter the new value into the Queue Size field, or adjust the value using the up/down arrows.</p>
Allocation Distance	Specifies the distance upstream from the belt window that a part instance must be within before the system is allowed to allocate that part instance to the robot.
Allocation Limit	<p>The distance from the pick line beyond which instances will not be allocated.</p> <ul style="list-style-type: none"> <li>• If set to zero, instances will be allocated if they are at or upstream of the pick line.</li> <li>• If greater than 0, it represents the upstream distance (in mm) from the pick line an instance must be for it to be allocated.</li> </ul> <p>This parameter is useful in the case of fast-moving belts where the robot needs additional "look ahead" to process instances.</p>
Use Custom Allocation	<p>Allows you to specify a Custom Allocation Script. The Custom Allocation Script provides two different entry points that are called: One is for allocating "non-belt" instances (parts and targets); the other is for "belt" instances. The program can manipulate the lists to indicate to the system what should be allocated to a given robot. This is called by the Process Manager when it needs to allocate parts.</p> <p>If this option is selected, click the browse icon ( ) to select the desired Custom Allocation Script.</p>
<b>Wait Mode Parameters:</b>	
Stay at current position	Causes the robot to remain at the current position while waiting for a part or target to process.
Move to idle pos-	Causes the robot to move to the idle position after the specified "After waiting"

Item	Description
Waiting for Part or Target	for" time (in milliseconds) while waiting for a part or target to process. For example, if the "After waiting for" time is 500 ms, when there is a break in the part or part target processing queue, the robot will move to the idle position after 500 ms (1/2 sec).
Use Custom Wait Program	Allows you to specify a custom V+ wait program. The program would be called when the robot does not have a part or target available. The program could check to see if the robot needs to be moved to the idle location or if it should stay at the current location.  <b>NOTE:</b> This program starts with the logic specified by one of the selections above.  If this option is selected, a wizard guides you through the program-creation process. For details, see Process Strategies on page 878.
Use Signal at Cycle Stop	When a cycle stop is issued and this option is enabled, the specified I/O signal will be turned on when the robot has reached the cycle stop state. When the cycle is resumed (cycle stop state is canceled), it will turn the specified signal off and will attempt to enable power, if high power was disabled. For more details on cycle stop, see Process Manager Control on page 920.
<b>Error Responses:</b>	
Output Signal - On Error	Defines a digital signal to assert when an error is detected for the selected robot.
Error Listing	Specifies how specific error conditions are to be handled. By default, all errors will be reported to the user and the software will wait until the user responds to the error. If an error condition is defined, it will override this default error handling. It can be used to define automatically handling of these kinds of errors:  <b>Single Error Code</b> Error code defined by user <b>Range of Error Codes</b> Error code range defined by user <b>Belt Window Access Error</b> Belt window violations <b>Robot Power Errors</b> Problems with power being enabled or enabling power <b>Gripper Errors</b> All gripper actuations <b>All Errors</b> All other errors.  When an error is detected for the robot, it will process the error listing handlers defined by the user and find the first one that can handle the condition. If the error cannot be handled by an item in the list, the error is reported to the user.

## Belt Control Parameters

The Belt Control Parameters are displayed when the belt is selected in the Process Strategy Editor.

The Belt Control Parameters group, shown in the following figure, is used to set the belt control parameters for the selected conveyor belt. These parameters can be set to determine when a conveyor belt is turned on or off. An optional speed control parameter is also provided. The decision point for the belt I/O control is based on the selected robot. If objects on the belt in the selected robot queue reach the specified thresholds, the belt will be turned off or the belt speed will be adjusted.



*Belt Control Parameters*

For this screen, the parameters are:

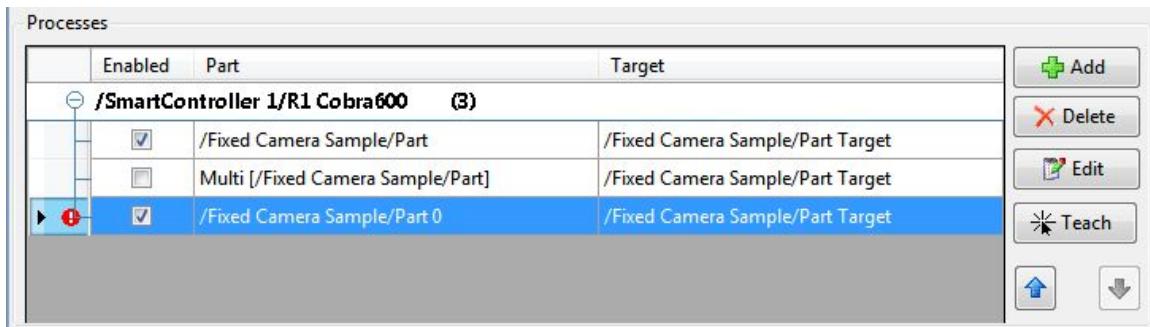
Item	Description
On/Off Control	Specifies the On/Off control of the belt. There are three selections available: <ul style="list-style-type: none"> <li>Do not control the belt: The belt is not controlled by the ACE software. (It can be controlled by another source, such as a PLC, V+, or OPC.)</li> <li>Leave the belt always on: The belt is turned on when the process starts.</li> <li>Control the belt: (Default) The belt is controlled by the ACE software. The belt is turned on/off based on how full the queue is. Also, the belt is automatically</li> </ul>

Item	Description
	turned off when the Process Manager stops the application.
Speed Control	If this is selected, you can use the Slow slide control to adjust the conveyor speed threshold. Otherwise, the conveyor belt operates at a constant speed.
Robot Queue	Specifies a robot for queue monitoring (the queue size for the robot is set in the <a href="#">Robot Parameters</a> group) The selected robot will typically be the last robot in the chain of robots servicing a conveyor. If parts get to the last robot, it needs to slow/stop the conveyor to ensure all parts are processed.
Slow (slide control)	Specifies the point in the belt window for slowing the conveyor if parts reach that point. This is useful for preventing the belt from feeding the robot faster than the robot can pick the parts. For example, 50% means that if a part reaches the mid-point of the belt window, the conveyor will be slowed.
Off Threshold / On Threshold (slide control)	Specifies the part threshold point in the pick window (from the Upstream Limit to the Pick Limit) for stopping the conveyor. For details on the belt limits, see Belt Calibrations on page 848.
	This control is useful for preventing the belt from feeding the robot faster than the robot can pick the parts. For example, if the off threshold is set to 90%, if a part reaches 10% from the end of the belt window, the conveyor will be turned off (assuming the Always On option is not selected). The setting will be higher than that used for the Slow threshold.
	When a belt is turned off by the Off Threshold, the belt will remain off until all instances are removed between the Off Threshold point and the On Threshold point. This can be used to minimize the number of times the belt is started and stopped.
	<p><b>NOTE:</b> The Always On option must be unchecked (not selected) for the Off Threshold setting to function.</p>
	If the Off Threshold point is beyond the pick line, then the robot will not be able to pick this part. For this reason, the stop line is displayed on the 3D Visualization of the belt window as a black line.
Product Flow (arrow)	Shows the product flow (belt window) direction of travel in relation to the Slow and Off Threshold slide controls. It is a reference for the thresholds. The bottom of the arrow represents the start of the belt window; the top of the arrow represents the end of the belt window.

## Processes

This section describes the Processes group in the Process Manager.

The Processes group defines the elements (robot, pick location, place location) for a particular process that will be controlled by the Process Manager.



### Processes Group

Each of the items in the Process group is described in the following table:

Item	Description
	Sets the priority for the selected process. The process at the top of the list receives the highest priority. For details, see Changing the Process Priority on page 893.
Enabled	If checked, the process is enabled for use.
Part	The Part specified for the process. You can double-click this item or click <b>Edit</b> to change it. For details, see Properties Tab: Pick Configuration Group on page 891.
Target	The Part Target specified for the process. You can double-click this item or click <b>Edit</b> to change it. For details, see Properties Tab: Place Configuration Group on page 892.
Add	Click to add a new process to the Processes list. For details, see Creating a Process on page 888.
Delete	Click to remove the selected process from the Processes list.
Edit	Click to edit the selected process. For details, see Creating a Process on page 888.
Teach	Click to teach the selected process. For details, see Teaching a Process on page 894.
	Alert icon, which indicates the process needs to be taught or there is some other problem. Hover the mouse pointer over the icon to view the alert message, which describes the reason(s) for the alert.

Item	Description		
	Enabled	Part	Target
<b>/SmartController 1/R1 Cobra600 (3)</b>			
	<input checked="" type="checkbox"/>	/Fixed Camera Sample/Part	/Fixed Camera Sample
	<input type="checkbox"/>	Multi [/Fixed Camera Sample/Part]	/Fixed Camera Sample
	<input checked="" type="checkbox"/>	/Fixed Camera Sample/Part 0	/Fixed Camera Sample
3 error(s)			
<ul style="list-style-type: none"> <li>- The process has not been taught.</li> <li>- '/Fixed Camera Sample/Part 0' does not identify a Belt.</li> <li>- '/Fixed Camera Sample/Part 0' does not identify a vision tool in the belt properties.</li> </ul>			

In addition to the items above, a shortcut menu is available, which duplicates the functions provided by the Add, Delete, Edit, and Teach buttons. To display the shortcut menu, right-click on a process in the Processes group. The following menu opens:



*Processes Group Shortcut Menu*

For a description of the items on this menu, see the corresponding item in the previous table.

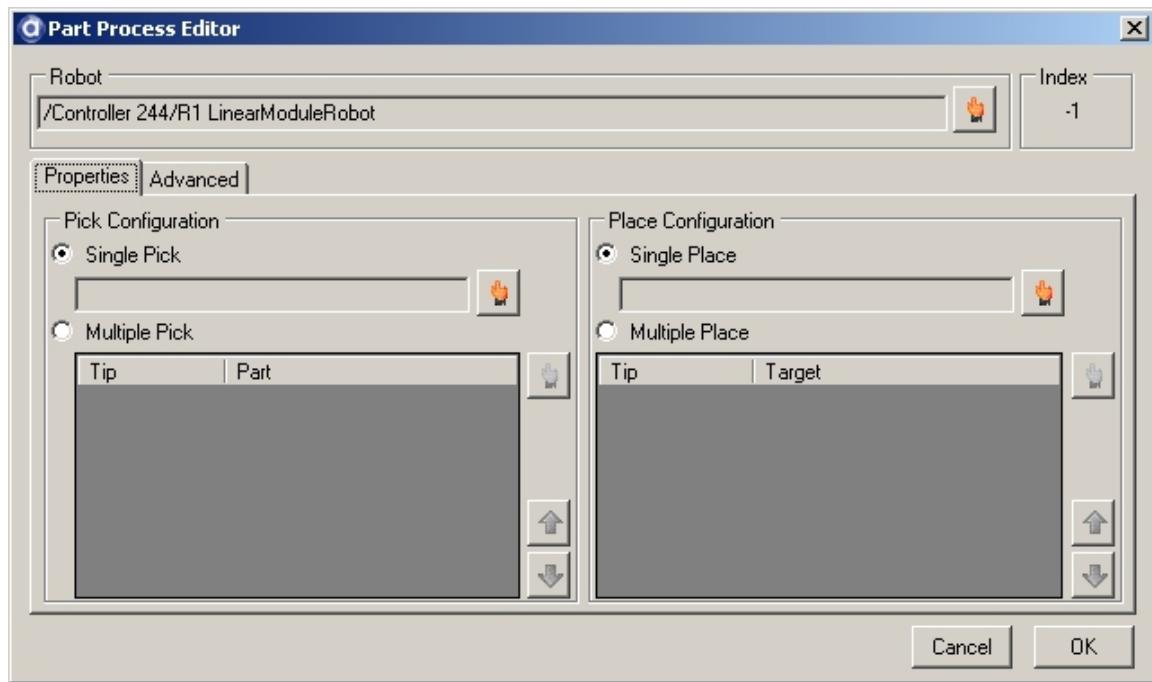
#### ***Creating a Process***

To create a Process, click **Add** in the Processes group. The Part Process Editor opens.

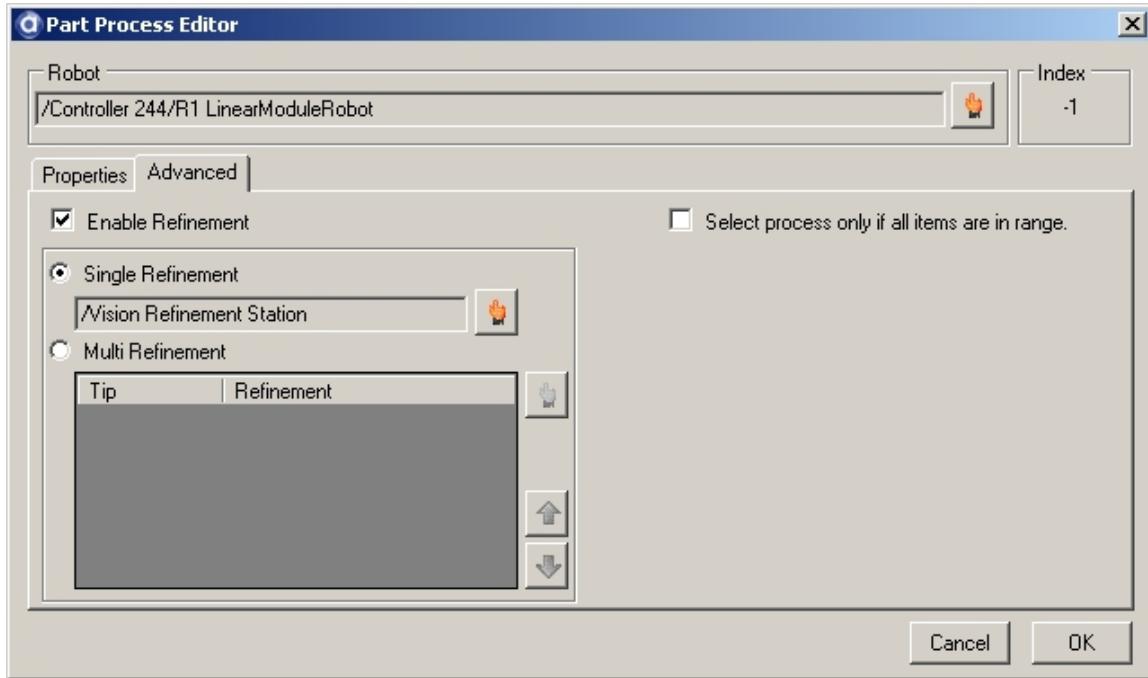
You can also open the Part Process Editor by selecting an existing process and clicking **Edit**.

## Processes

---



*Part Process Editor: Properties Tab*



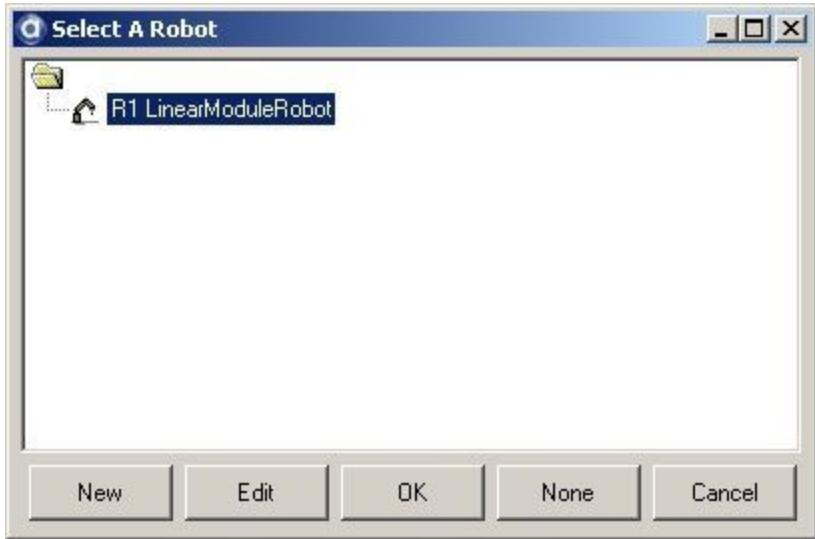
*Part Process Editor: Advanced Tab*

#### ***Editor Parameters***

The Part Process Editor parameters, which specify the items used in the process, are described below.

##### **Robot Group**

The Robot group is used to specify the robot that will be used for the pick-and-place process. Use the browse icon to open the Select a Robot dialog box, which allows you to create or specify the robot that will be used in the pick-and-place process. If there is only one robot in the workcell, that robot is automatically selected for you.



Select a Robot Dialog Box

## Index

This item displays the index number of the process, which can be referenced for custom V+ programs.

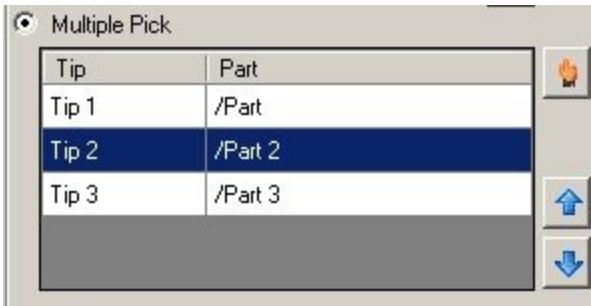
## Properties Tab: Pick Configuration Group

The Pick Configuration group is used to specify the single- or multi-pick Part items for the pick-and-place process, as follows:

- **Single Pick:** Select this item for a single-pick application (only one pick-motion is performed). Use the field below to browse for the part that will be picked.



- **Multi Pick:** Select this item for a multiple-pick application (multiple pick-motions are performed). Use the fields below to specify the tool tip and part for each element of the pick process.

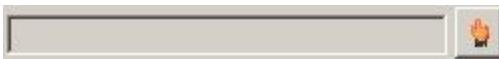


Use the Up/Down arrows to change the order of the tip processing.

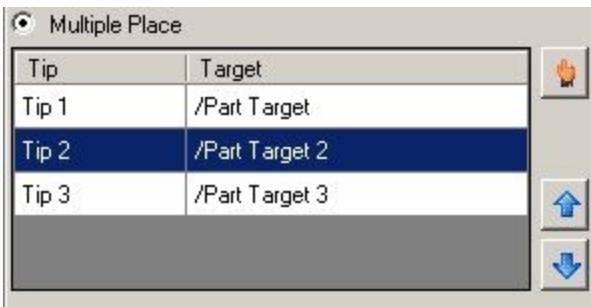
### Properties Tab: Place Configuration Group

The Place Configuration group is used to specify the single- or multi-place Part Target items for the pick-and-place process, as follows:

- **Single Place:** Select this item for a single-place application (only one place-motion is performed). Use the field below to browse for the part target that will be placed.



- **Multi Place:** Select this item for a multiple-place application (multiple place-motions are performed). Use the field below to specify the tool tip and part target for each element of the place process.



Use the Up/Down arrows to change the order of the tip processing.

Single and Multiple can be used together. For example, if you want to pick multiple parts individually and then place them all at the same time, you would use Multiple Pick and Single Place.

### Advanced Tab: Enable Refinement

The Enable Refinement option is used to apply a vision refinement to the picked part. When this option is selected, the part is moved to a predefined vision refinement station for additional processing before being moved to the place (part target) location. For more details on defining the vision refinement station, see Vision Refinement Station on page 941.

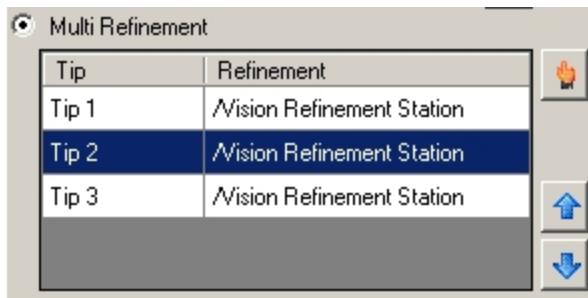
The following items are used to specify single- or multi-part refinement:

- **Single Refinement:** Select this item for a single-vision-refinement application when:
  - Only one part at a time is being moved by the robot, or
  - A multi-tip-gripper is moved to the vision refinement station, but only one picture is taken that includes all parts on the gripper. If you want to process an individual refinement for each part, use the Multi Refinement option.

Use the field below to browse for the vision refinement station.



- **Multi Refinement:** Select this item for a multiple-tip-gripper applications. Use the field below to specify the tool tip and corresponding vision refinement. Each tip will be moved individually to the specified vision refinement station.



Use the Up/Down arrows to change the order of the tip processing.

The *Select process only if items are in range* option tells the system to only select this process if all parts and targets are "in range" of the robot. For example, if parts are upstream of the belt window, you may not want the process to be selected. This allows you to constrain the process selection.

### **Changing the Process Priority**

You can change the priority for a process by using the Up/Down arrows on the Process Group editor, shown in the following figure.

**NOTE:** In addition to the arrows, you can also affect process priority through the Process Selection Mode setting of the Robot Parameters on the Process Strategy dialog. For more details, see Robot Parameters on page 881.



*Up/Down Arrows*

The process priority is used in situations where multiple processes are defined, and a given robot is capable of doing several of the potential processes. The process at the top of the Process Group list receives highest priority.

For example, let's say you have three processes defined in the Process Manager, as follows:

1. Pick from input (Part), place to output (Part Target)
2. Pick from Part Buffer, place to output (Part Target)
3. Pick from input (Part), place to output (Part Buffer)

In this case, the Process Manager would always execute process #1, if there are parts at the input, and move them to the output. If no parts are present at the input, it will then check for process #2 and #3.

However, if you change the order (and, therefore, the priority), you will get a different behavior, as follows:

1. Pick from Part Buffer, place to output (Part Target)
2. Pick from input (Part), place to output (Part Target)
3. Pick from input (Part), place to Part Buffer

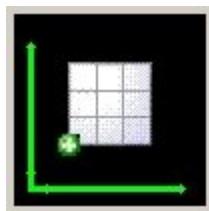
In this case, the Process Manager will *always* remove parts from the Part Buffer, until no parts remain in the Part Buffer, before processing the input (Part).

### **Teaching a Process**

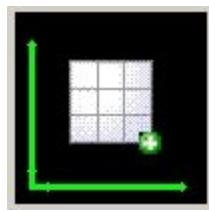
The last step in creating the application is to teach the process. This is done using the Teaching Process wizard. For details on the buttons and indicators on the wizard pages, see Wizards on page 181.

**NOTE:** The following process illustrates the steps for a basic pick-and-place application. The steps in your wizard will vary, depending on the application and types of Parts and Part Targets you have defined for your application.

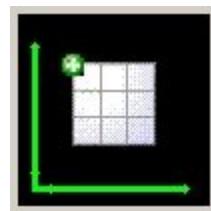
If a Process Pallet is used for a Part or Part Target, you will see additional screens in the wizard for teaching the pallet frame (the orientation of the pallet in the workspace) and the first position in the pallet. Each of these steps contain an image of the pallet item being taught, which provides a visual reference.



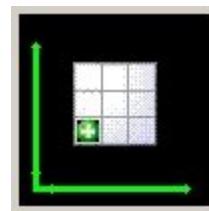
Frame Origin



Frame X-axis



Frame Y-axis



First Pallet Position

For more details on a Process Pallet, see Process Pallet on page 927.

To access the Teaching Process wizard:

1. Click **Teach** in the Processes group of the Process Manager editor. The Teaching Process wizard initializes and opens the first page of the wizard.

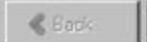
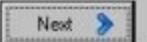
**Teaching Process**

**Teaching a Process.**

Instructions

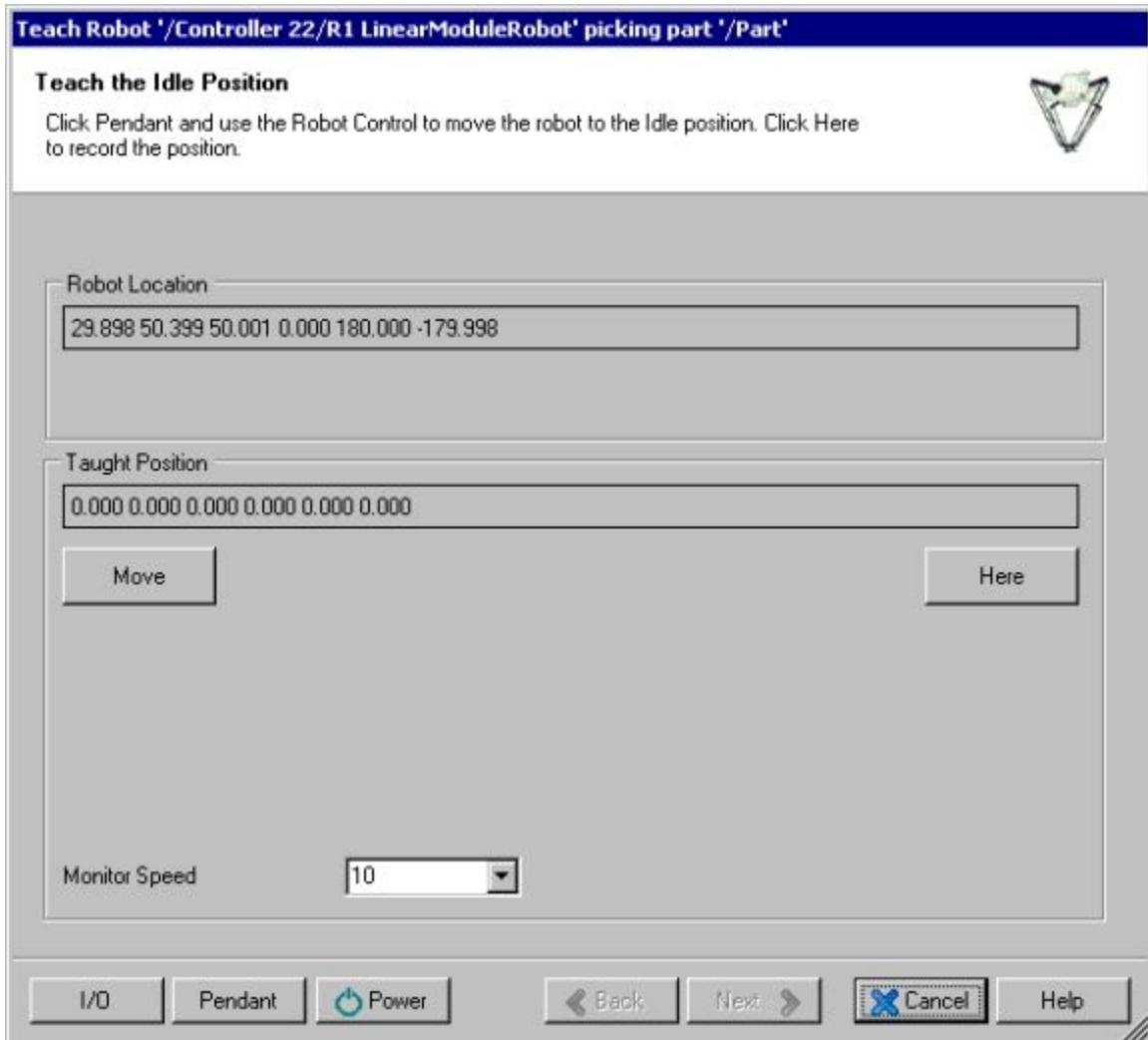


You are teaching the selected process. This procedure will first take you through the steps of picking all the parts associated with the process. Once the robot has all the parts, you will be taken through the steps of teaching the part(s) at the target(s) configured in the process.

 Back  Next  Cancel  Help

*Teaching a Process (Instructions)*

2. Click **Next** to proceed. The Teach the Idle Position screen opens.

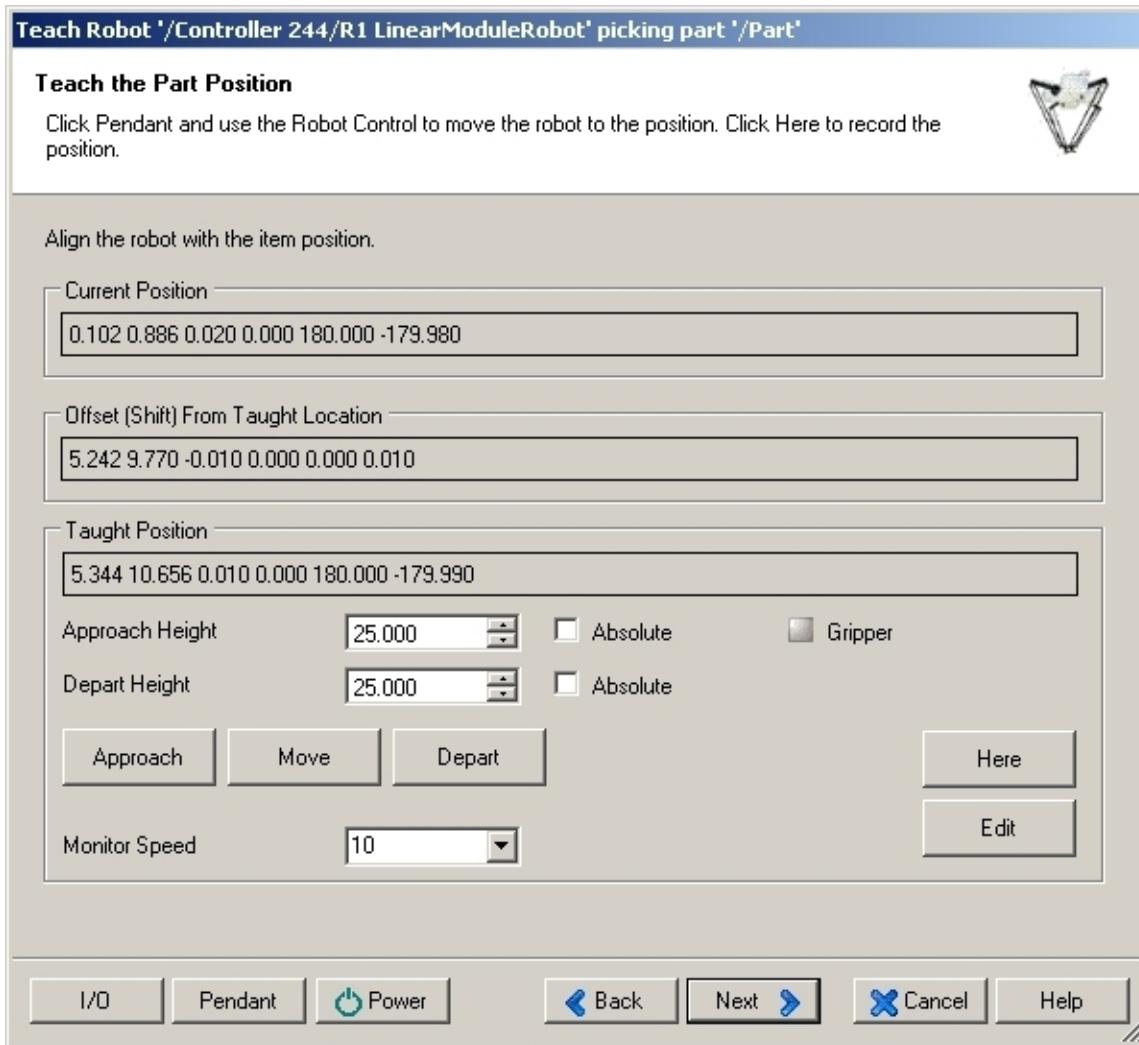


*Teaching a Process (Teach the Idle Position)*

- a. Click **Pendant** to display the Robot Jog Control.
- b. Use the Robot Jog Control to move the robot to the Idle position.
- c. Click **Here** to record the position.
- d. (Optional) You can use the Robot Jog Control to move the robot away from the recorded (taught) position, and then click **Move** to move the robot back to the taught position.

**NOTE:** If the Robot Jog Control is open when Move is clicked, the Robot Jog Control temporarily switches to Comp (computer control) mode while the move is executed. When the move has completed, the Robot Jog Control switches back to its previous mode.

- Click **Next** to proceed. The Teach the Part Position screen opens.



#### *Teaching a Process (Teach the Part Position)*

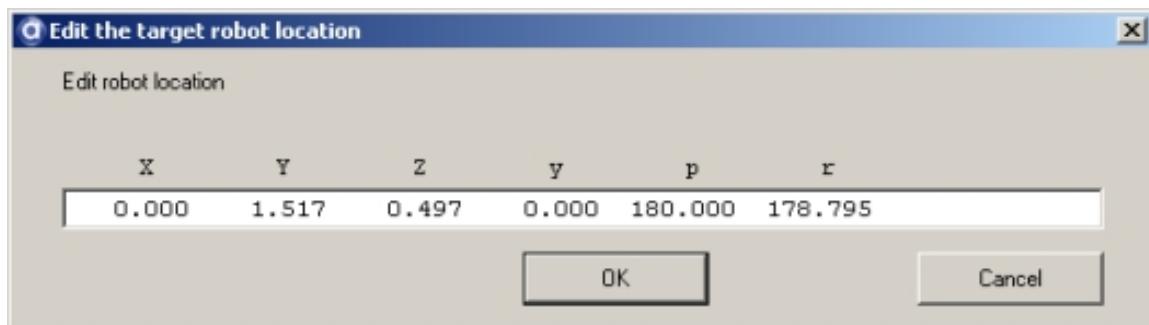
- Click **Pendant** to display the Robot Jog Control.
- Use the Robot Jog Control to move the robot to the Part position.
- (Optional) You can adjust the Approach Height and Depart Height, using positive or negative values, as needed. You can also select the Absolute check box, if you want the values to be

absolute values instead of relative to the taught position.



**WARNING:** When the Absolute option is selected, you must ensure the approach/depart heights are set correctly. Otherwise, the robot could move to an unintended location or out of range, which may damage the robot or other equipment in the workcell.

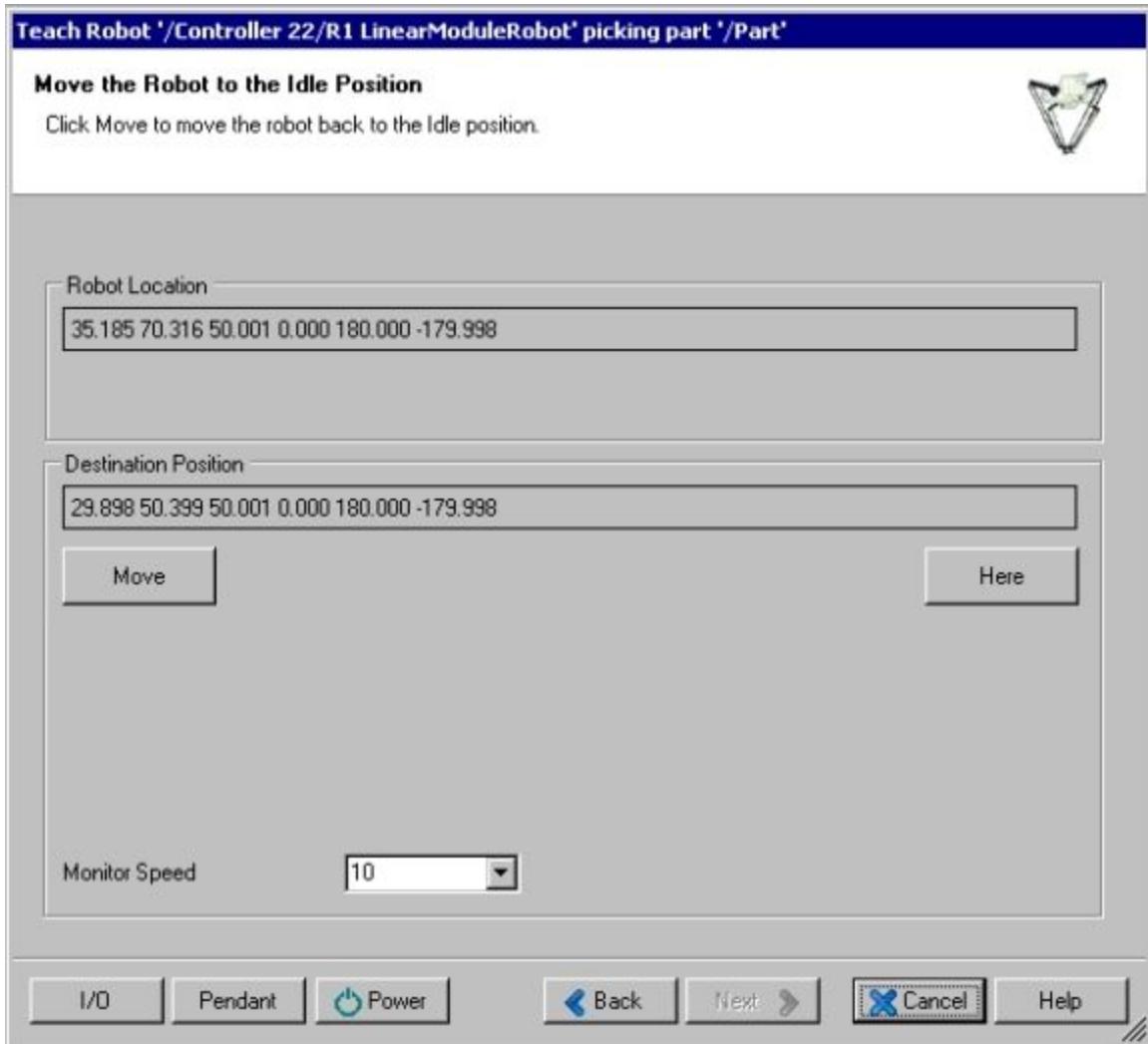
- d. Click **Here** to record the position.
- e. (Optional) Click **Edit** to adjust the recorded (taught) robot location. The following window opens:



Edit the location values and then click **OK** to close the window.

- f. (Optional) You can use the Robot Jog Control to move the robot away from the recorded (taught) position, and then click **Move** to move the robot back to the taught position. When there is a difference (offset or shift) from the current position to the taught position, that difference is shown in the Offset (Shift) From Taught Location field.

4. Click **Next** to proceed. The Move the Robot to the Idle Position screen opens.

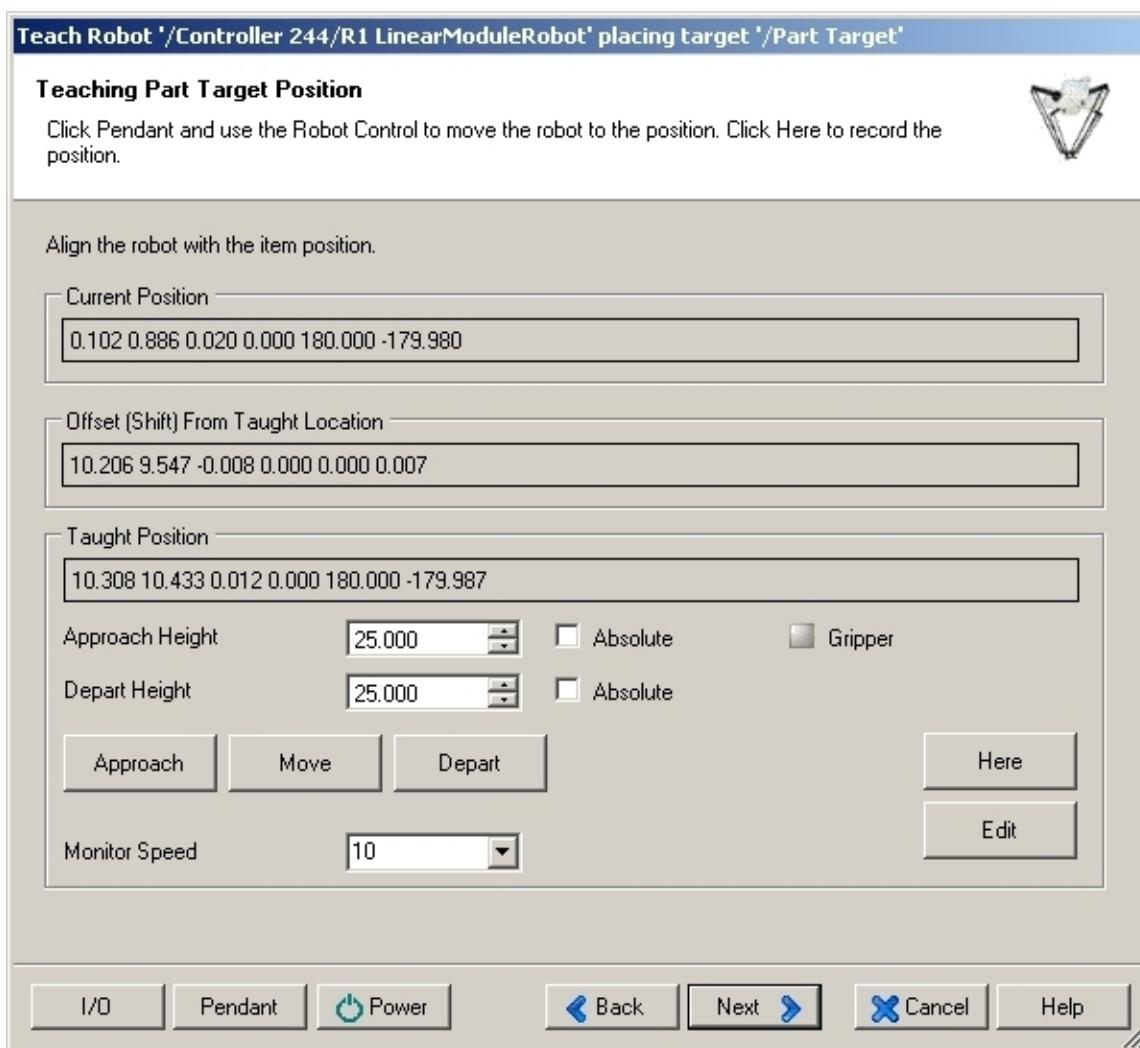


*Teaching a Process (Move the Robot to the Idle Position)*

- Click **Move** to move the robot to the Idle position.
- (Optional) You can use the Robot Jog Control to move the robot to a new Idle position and then click **Here** to record that position.
- (Optional) You can use the Robot Jog Control to move the robot away from the recorded (taught) position, and then click Move to move the robot back to the taught position.

If your process is using a vision refinement, there are several additional steps here that are not part of the "normal" teach process. For details on vision refinement, see Vision Refinement Station on page 941.

- For a "Move to camera" refinement, you will teach the camera location and then move back to the idle position. For details, see Move to Camera Additional Steps on page 903.
  - For a "Vision on the fly" refinement, you will teach the starting and ending points of the path through the camera FOV, and then move back to the idle position. For details, see Vision on the Fly Additional Steps on page 905.
5. Click **Next** to proceed. The next screen of the wizard opens, which you will use to teach the Part Target position.



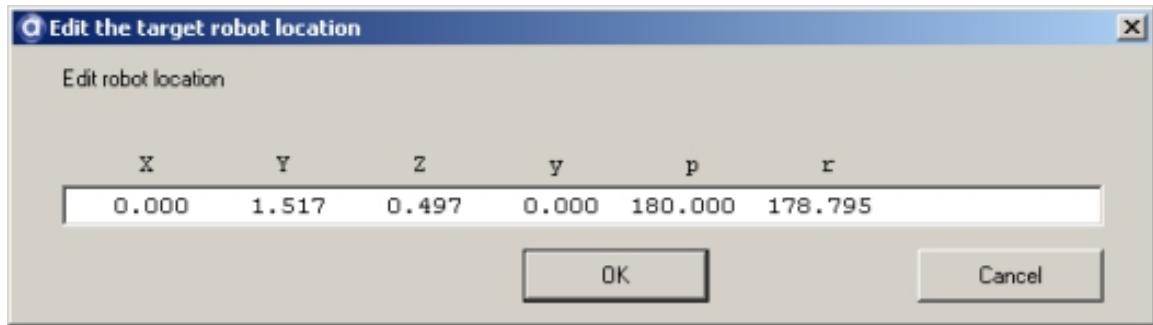
*Teaching a Process (Teaching Part Target Position)*

- a. Click **Pendant** to display the Robot Jog Control.
- b. Use the Robot Jog Control to move the robot to the Part Target position.
- c. (Optional) You can adjust the Approach Height and Depart Height, using positive or negative values, as needed. You can also select the Absolute check box, if you want the values to be absolute (instead of relative) values.



**WARNING:** When the Absolute option is selected, you must ensure the approach/depart heights are set correctly. Otherwise, the robot could move to an unintended location or out of range, which may damage the robot or other equipment in the workcell.

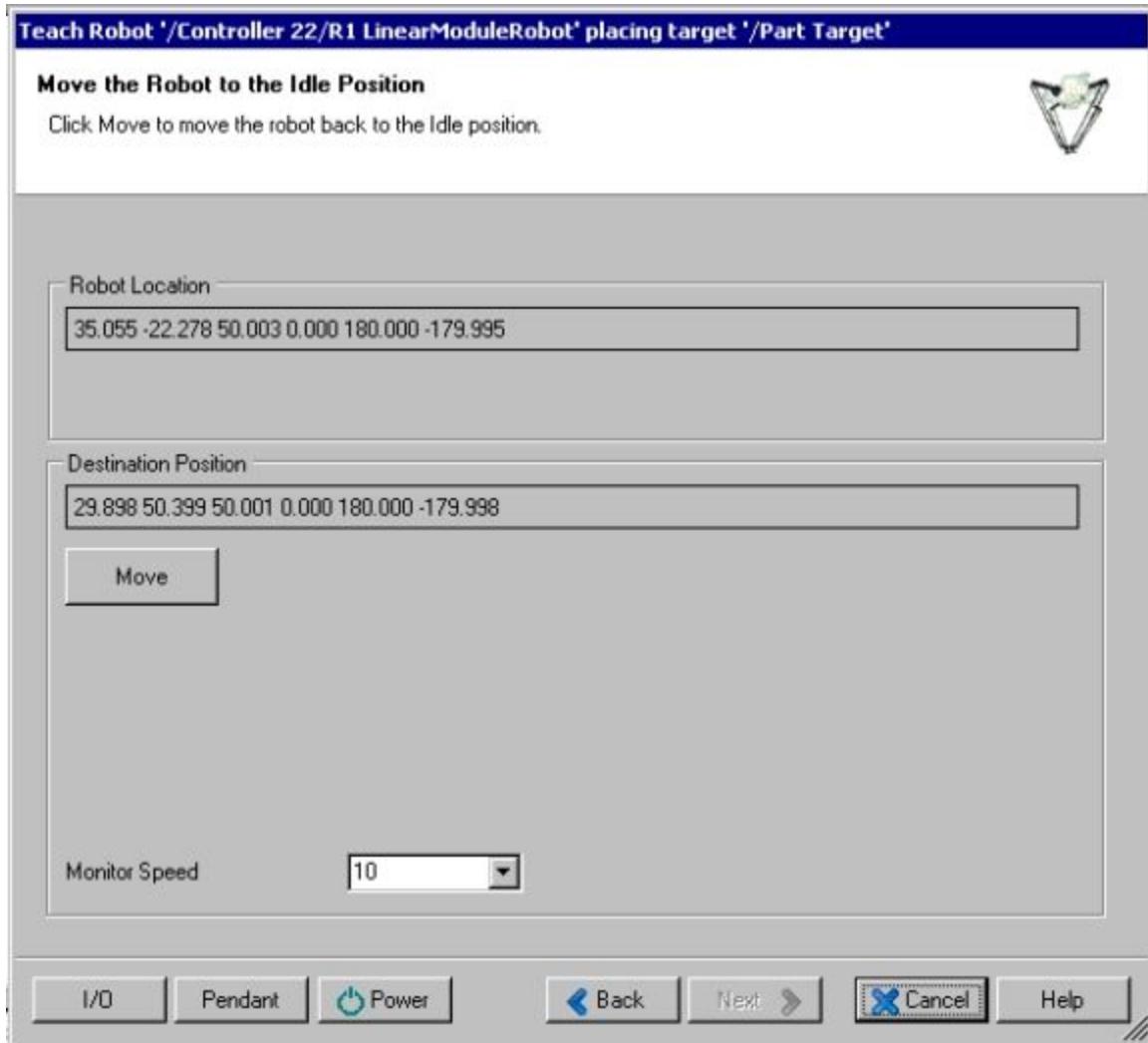
- d. Click **Here** to record the position.
- e. (Optional) Click **Edit** to adjust the recorded (taught) robot location. The following window opens:



Edit the location values and then click **OK** to close the window.

- f. (Optional) You can use the Robot Jog Control to move the robot away from the recorded (taught) position, and then click **Move** to move the robot back to the taught position. When there is a difference (offset or shift) from the current position to the taught position, that difference is shown in the Offset (Shift) From Taught Location field.

6. Click **Next** to proceed. The Move the Robot to the Idle Position screen opens.

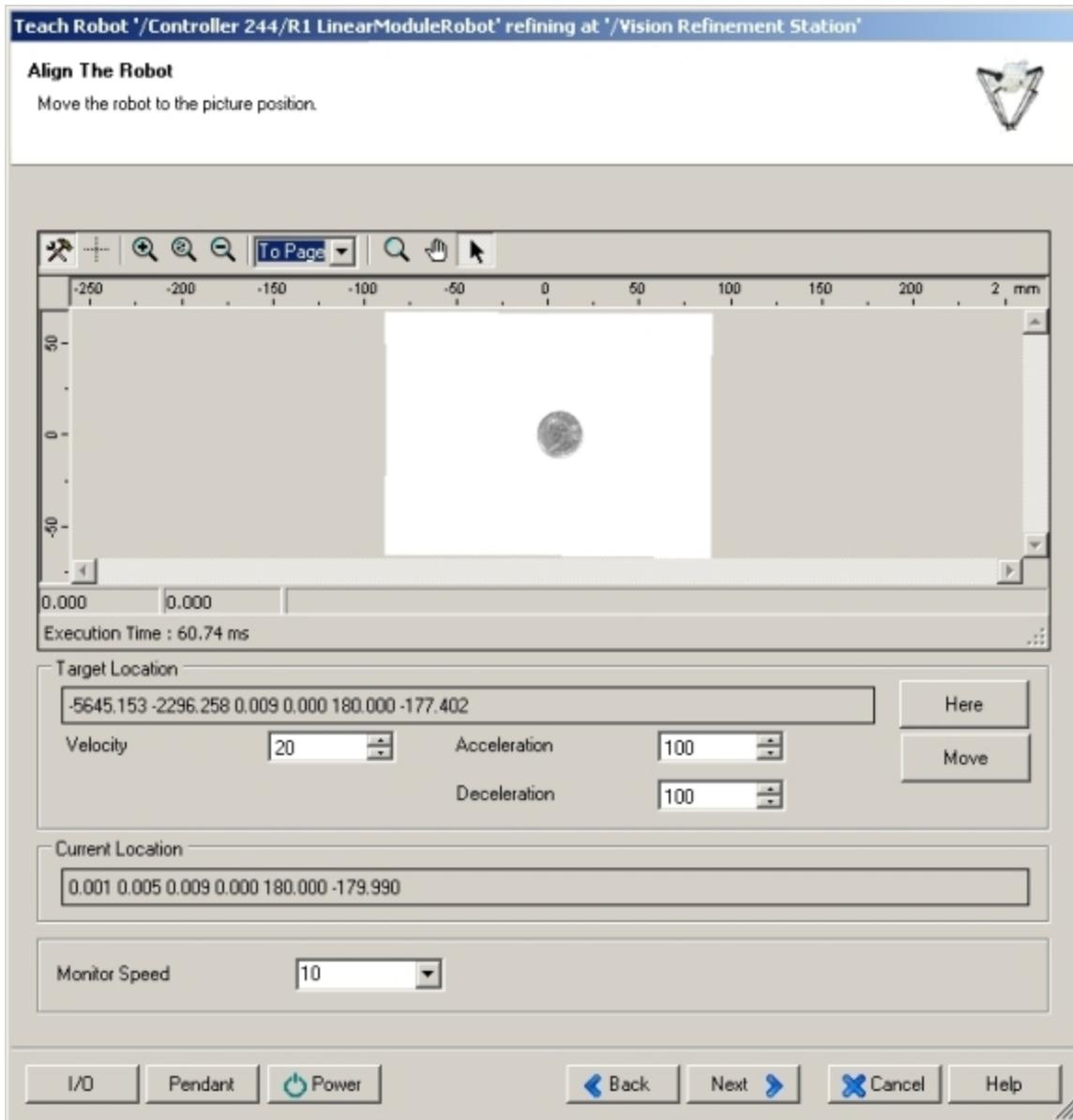


*Teaching a Process (Move the Robot to the Idle Position)*

7. Click **Move** to return the robot back to the idle (home) position. After the robot has completed its movement, the wizard closes.

## Move to Camera Additional Steps

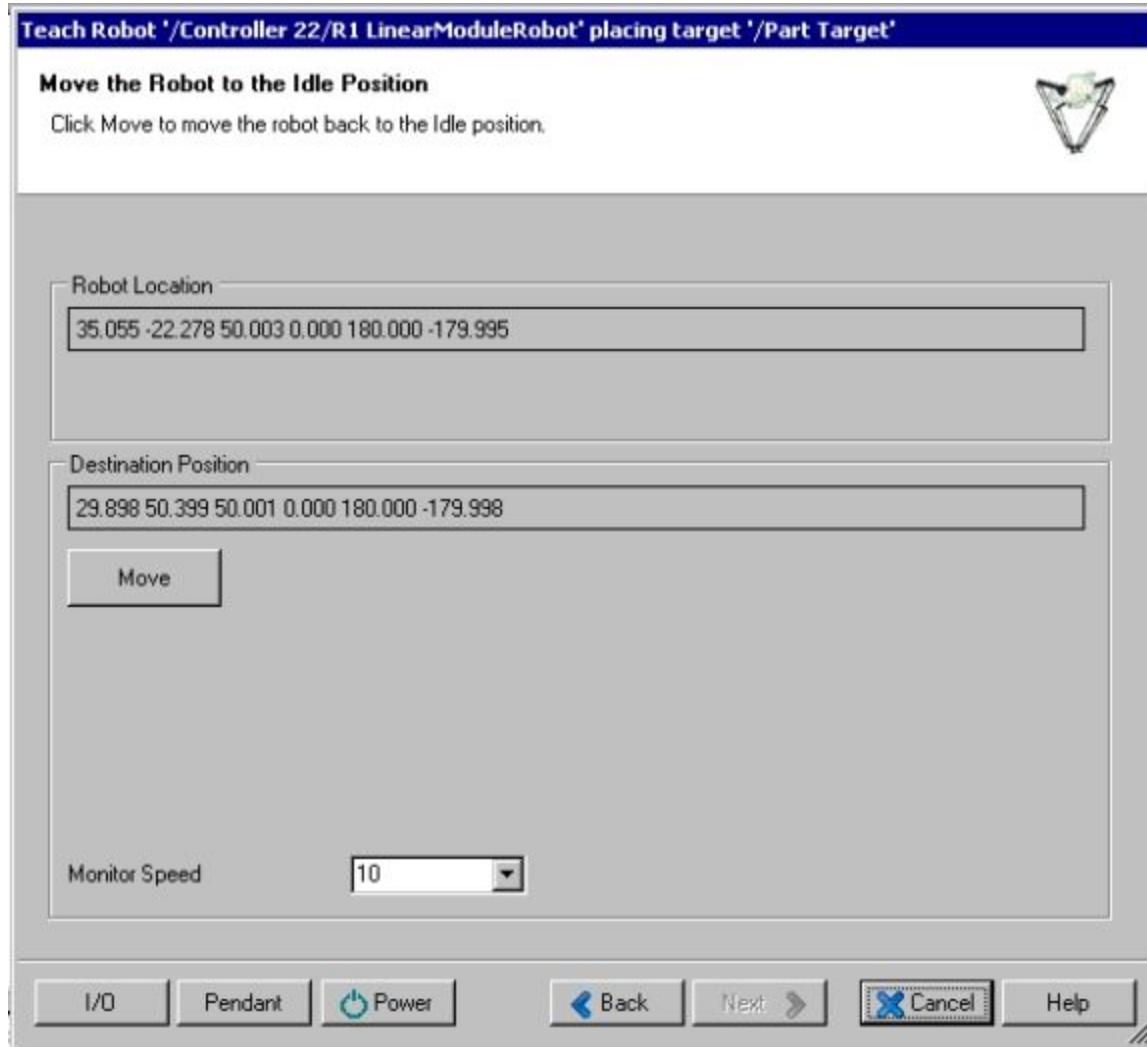
1. Click **Next** to proceed. The Align the Robot screen opens.



*Teaching a Move to Camera Process (Move the Robot to the Camera Position)*

- a. Click **Pendant** to display the Robot Jog Control.
- b. Use the Robot Jog Control to move the robot to the Part Target position.

- c. (Optional) You can adjust the Velocity, Acceleration, and Deceleration values, as needed. These parameters control the robot motion to the camera location.
  - d. Click **Here** to record the position.
  - e. (Optional) You can use the Robot Jog Control to move the robot away from the recorded (taught) position, and then click **Move** to move the robot back to the taught position.
2. Click **Next** to proceed. The Move the Robot to the Idle Position screen opens.

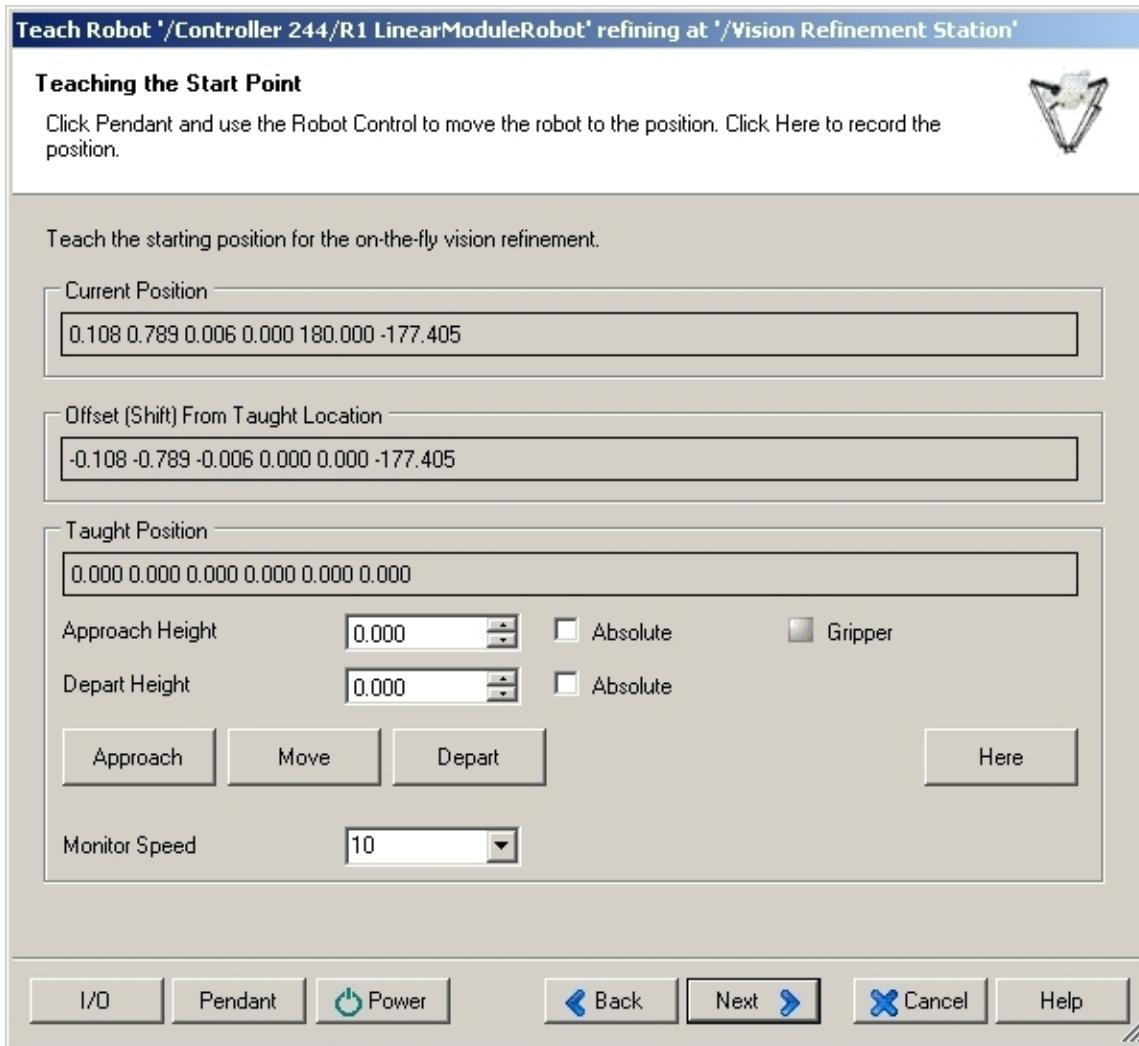


*Teaching a Move to Camera Process (Move the Robot to the Idle Position)*

3. Complete the remaining steps of the wizard, as described in Teaching a Process on page 894.

## Vision on the Fly Additional Steps

1. Click **Next** to proceed. The Move to Start Position screen opens.



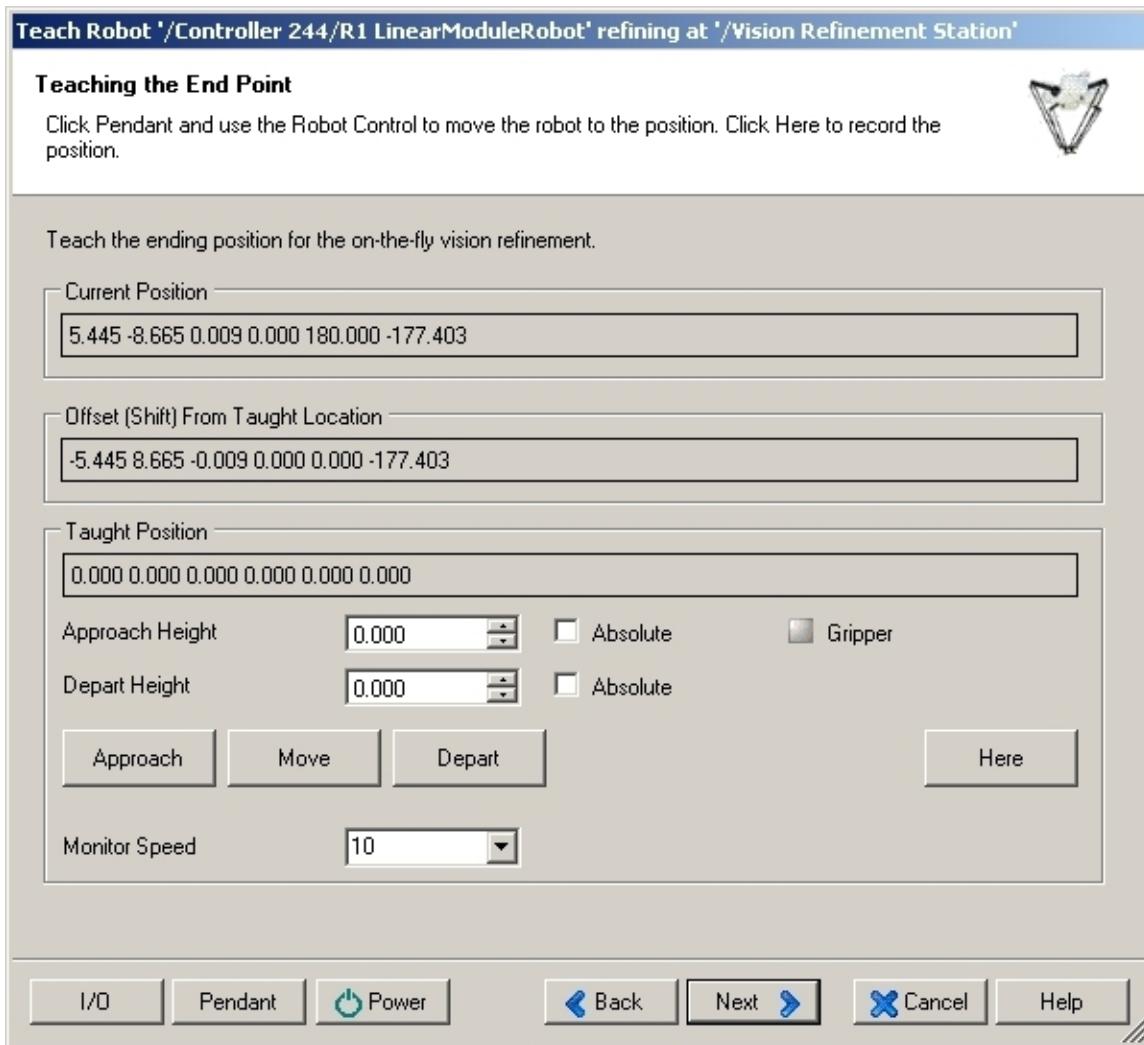
*Teaching a Vision on the Fly Process (Move the Robot to the Start Position)*

- a. Click **Pendant** to display the Robot Jog Control.
- b. Use the Robot Jog Control to move the robot to the Part Target position.
- c. (Optional) You can adjust the Approach Height and Depart Height values, as needed. You can also select the Absolute check box, if you want the values to be absolute (instead of relative) values.

**WARNING:** When the Absolute option is selected, you must ensure the approach/depart heights are set correctly. Otherwise, the robot could move to an unintended location or out of range, which may damage the robot or other equipment in the workcell.

- d. Click **Here** to record the position.
- e. (Optional) You can use the Robot Jog Control to move the robot away from the recorded (taught) position, and then click **Move** to move the robot back to the taught position. When there is a difference (offset or shift) from the current position to the taught position, that difference is shown in the Offset (Shift) From Taught Location field.

2. Click **Next** to proceed. The Move to End Position screen opens.



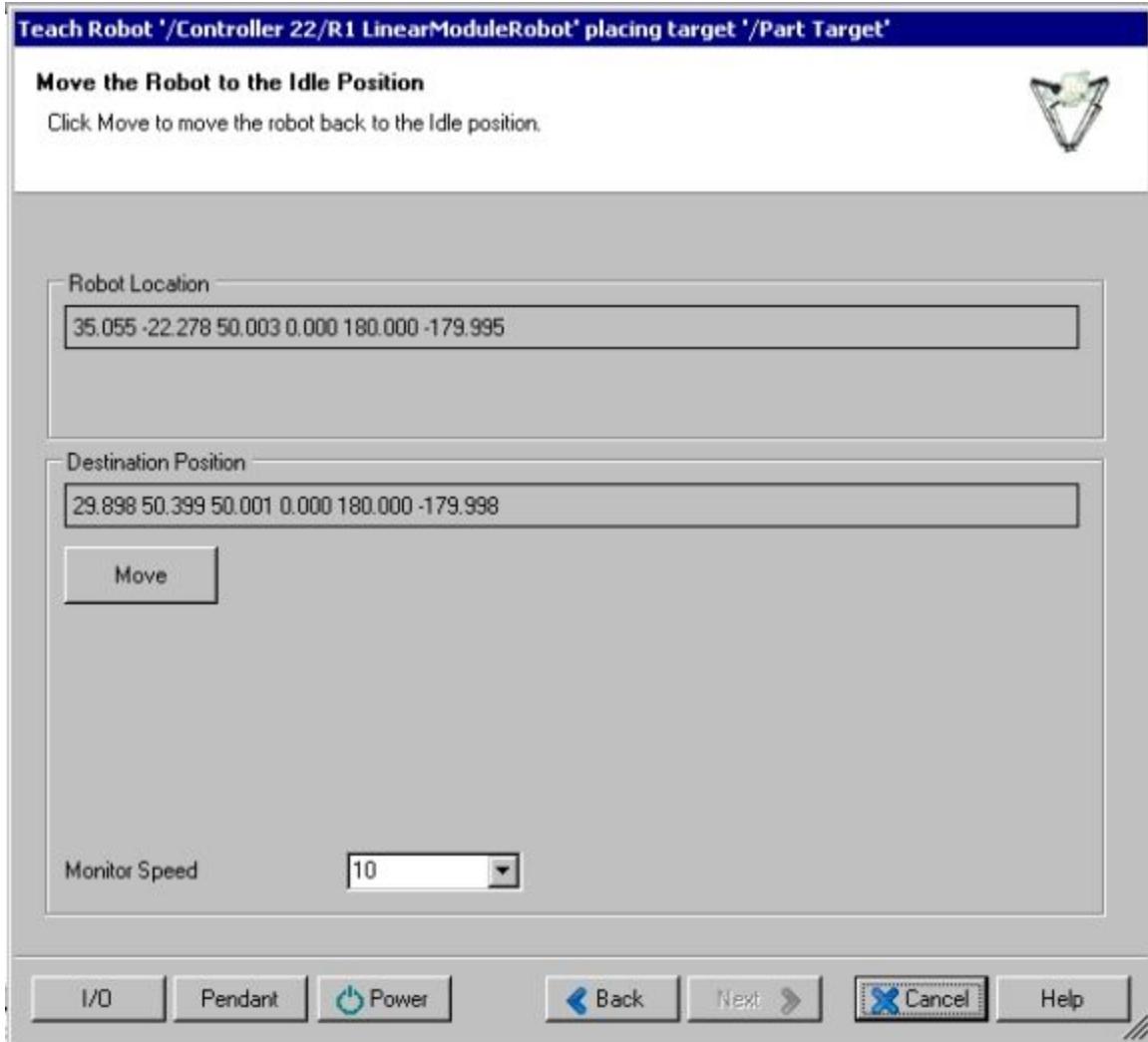
#### *Teaching a Vision on the Fly Process (Move the Robot to the End Position)*

- Click **Pendant** to display the Robot Jog Control.
- Use the Robot Jog Control to move the robot to the Part Target position.
- (Optional) You can adjust the Approach Height and Depart Height values, as needed. You can also select the Absolute check box, if you want the values to be absolute (instead of relative) values.

**WARNING:** When the Absolute option is selected, you must ensure the approach/depart heights are set correctly. Otherwise, the robot could move to an unintended location or out of range, which may damage the robot or other equipment in the workcell.

- d. Click **Here** to record the position.
- e. (Optional) You can use the Robot Jog Control to move the robot away from the recorded (taught) position, and then click **Move** to move the robot back to the taught position. When there is a difference (offset or shift) from the current position to the taught position, that difference is shown in the Offset (Shift) From Taught Location field.

3. Click **Next** to proceed. The Move the Robot to the Idle Position screen opens.



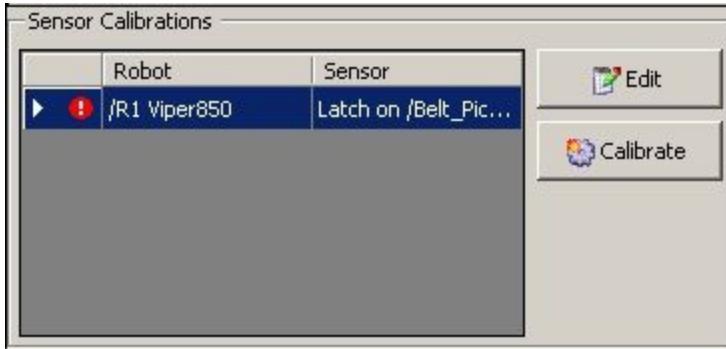
*Teaching a Vision on the Fly Process (Move the Robot to the Idle Position)*

4. Complete the remaining steps of the wizard, as described in Teaching a Process on page 894.

## Sensor Calibrations

This section describes the Sensor Calibrations group in the Process Manager. For details, see Process Manager on page 843.

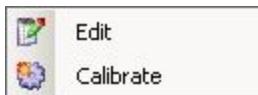
The Sensor Calibrations group defines the robot-to-sensor calibrations for the selected workcell process. For more details on defining a workcell process, see [Processes](#).

*Sensor Calibrations Group*

Each of the items in the Sensor Calibrations group is described in the following table:

Item	Description
Robot	The robot specified for the belt calibration. Double-click this item or click <b>Edit</b> to display the Belt Calibration Editor. For details, see Editing the Calibration Parameters on page 911.
Sensor	The sensor specified for the process. Double-click this item or click <b>Edit</b> to display the Belt Calibration Editor. For details, see Editing the Calibration Parameters on page 911.  A "sensor" can be any of the following: <ul style="list-style-type: none"> <li>• belt camera</li> <li>• fixed camera (downward-looking camera)</li> <li>• latch sensor</li> <li>• spacing reference</li> <li>• refinement camera (upward-looking camera)</li> </ul>
Edit	Click to edit the selected belt calibration. For details, see Editing the Calibration Parameters on page 911.
Calibrate	Click to teach the selected process. For details, see Using the Calibration Wizard on page 914.

In addition to the items above, a shortcut menu is available, which duplicates the functions provided by the Edit and Calibrate buttons. To display the shortcut menu, right-click on a process in the Processes group. The following menu opens:

*Sensor Calibrations Group Shortcut Menu*

For a description of the items on this menu, see the corresponding item in the previous table.

### ***Creating a Sensor Calibration***

When a sensor calibration is required, the Process Manager displays the Sensor object name with an alert icon () in the Sensor Calibrations Group.

There are four types of sensor calibrations (the sensor calibration required depends on the type of part sensor selected for the Part or Part Target). For details, see Part on page 829 and Part Target on page 841. The sensor calibration types are:

- Belt camera (see Belt Camera Calibration on page 914 for details)
- Fixed camera [downward-looking camera] (see Fixed Camera Calibration on page 917 for details)
- Latch sensor (see Belt Latch Calibration on page 915 for details)
- Spacing reference (see Spacing Reference Calibration on page 918 for details)
- Refinement camera [upward-looking camera] (see Refinement Camera Calibration on page 918 for details)

For details on the vision windows and image-editing controls in the wizards, see the *ACE Sight User's Guide*.

The sensor is calibrated using the Sensor Calibration wizard, which is accessed from the Calibrate button in the Sensor Calibrations group. After the sensor is calibrated using the wizard, the stored calibration values can be manually edited.

### ***Editing the Calibration Parameters***

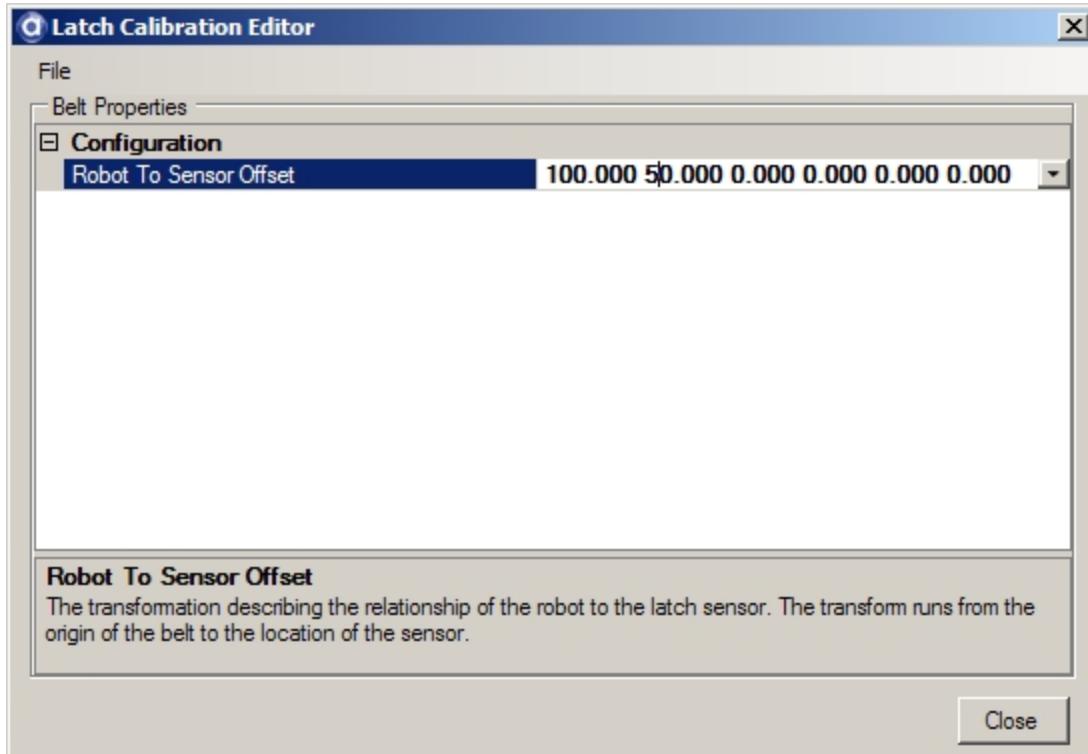
**NOTE:** The sensor must be calibrated, or loaded from a previously-saved calibration data file, before the values can be manually edited. For details, see Using the Calibration Wizard in the next section.

After the sensor is calibrated through the Sensor Calibration wizard, you can manually edit the stored sensor-calibration parameters, such as the robot-to-sensor offset. These parameters are edited using the sensor Calibration Editor. To access the sensor Calibration Editor, click **Edit** in the Sensor Calibrations group. The Sensor Calibration Editor opens.

The Sensor Calibration Editor contains the sensor properties configuration parameters. These are used to configure various settings of the selected sensor. The bottom area of the Sensor Calibration Editor displays online help for the selected parameter.

**NOTE:** Because online help for each parameter is provided directly in the editor, it is not duplicated in this documentation.

For example, the following figure shows the Latch Calibration Editor, which has one property for controlling the calibration offset.

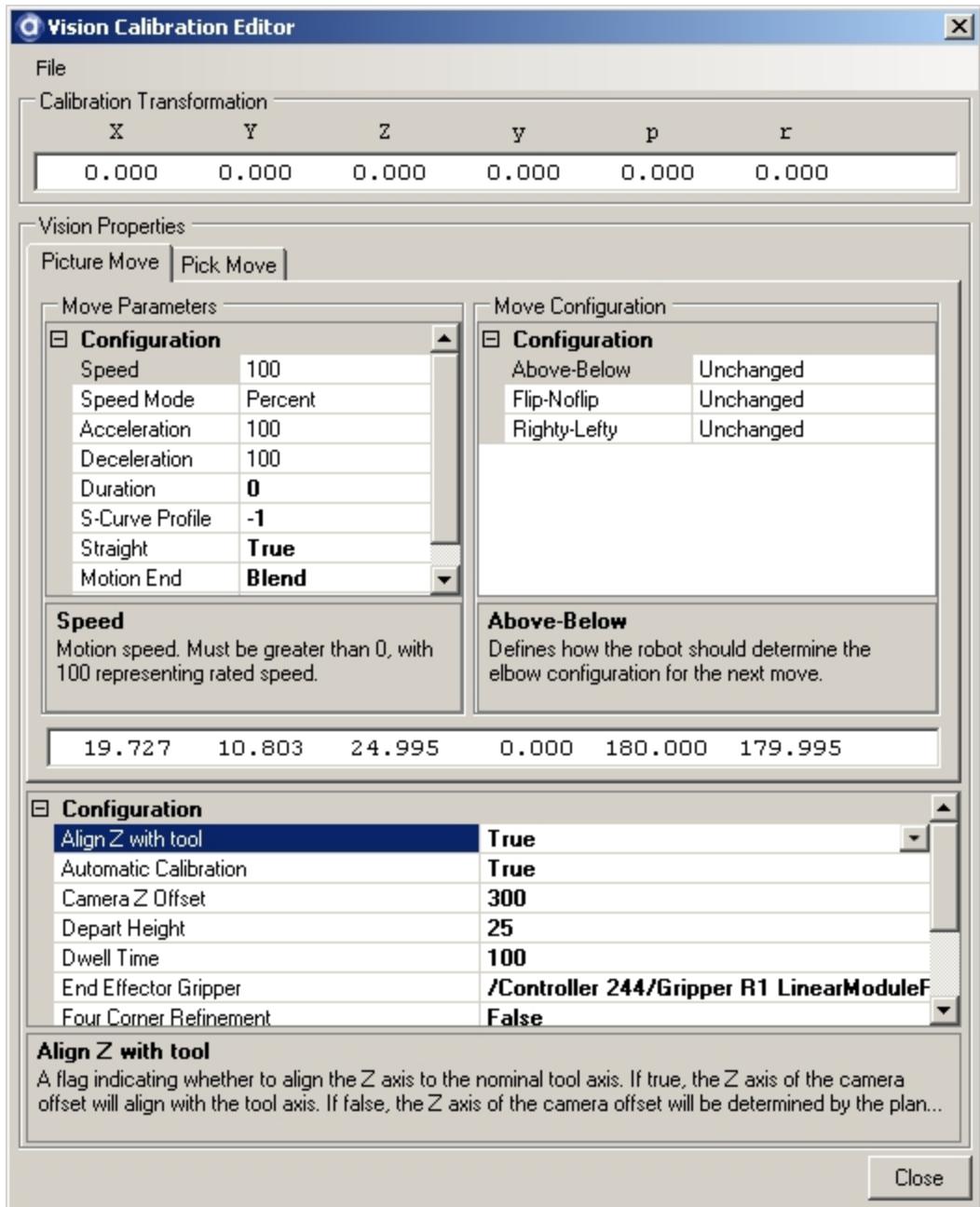


*Sensor Calibration - Latch Calibration Editor*

In this example, the following figure shows the Vision Calibration Editor, which contains a calibration offset, along with additional parameters for controlling the robot motion during the picture-taking and part-pick operations.

## Sensor Calibrations

---



Sensor Calibration - Vision Calibration Editor

## Saving and Loading a Calibration

After a calibration has been completed, the data can be saved by selecting **File > Save To** on the calibration editor menu. You can load a previously-saved calibration file by selecting **File > Load From** on the calibration editor menu.

## Automated versus Manual Calibrations

The calibrations can be performed using either the calibration wizard (preferred method) or the manual calibration procedure. In the calibration wizard (automated process), you teach the initial locations and then the wizard automatically performs the robot movements to acquire enough data points to calibrate the system. In the manual procedure, you have to move the robot through each step of the process until enough data points have been acquired. The manual method is provided for cases where obstructions in the workcell do not allow for automated movement of the robot during the calibration process.

**NOTE:**

1. Adept recommends that you use the calibration wizard (automated process), in order to obtain the optimum performance from your system.
2. The manual calibration procedure is available for the Fixed Camera and Refinement Camera calibrations.

## Using the Calibration Wizard

**NOTE:** This feature operates differently in emulation mode. For details on emulation mode wizards, see Emulation Mode Wizards on page 185.

The Sensor Calibration wizard provides an interactive series of steps that guide you through the calibration process. Each step of the wizard contains graphical aids and text that describe the particular step being performed. As each step of the wizard is completed, the information from that step is collected and used to populate the fields in the Sensor Calibration Editor, which is described in the previous section.

For more details on the buttons and indicators on the wizard pages, see Wizards on page 181.

### Belt Camera Calibration

**NOTE:** This feature operates differently in emulation mode. For details on emulation mode wizards, see Emulation Mode Wizards on page 185.

This section describes the Belt Camera Calibration wizard in the Process Manager, which is accessed from the Sensor Calibrations group. The Sensor Calibrations group defines the robot-to-sensor calibrations for the selected workcell process. For more details on the Sensor Calibrations group, see Sensor Calibrations on page 909.

---



### *Sensor Calibrations Group*

The calibration wizard provides an interactive series of steps that guide you through the calibration process. Each step of the wizard contains graphical aids and text that describe the particular step being performed. As each step of the wizard is completed, the information from that step is collected and used to populate the fields in the Sensor Calibration Editor. For details, see [Sensor Calibrations on page 909](#).

For details on the buttons and indicators on the wizard pages, see [Wizards on page 181](#).

To access the Belt Camera Calibration wizard:

1. Click **Calibrate** in the Sensor Calibrations group. The wizard initializes and opens the first page.  
*Belt Camera Calibration Wizard (Initialize and Select End Effector)*
2. Read and follow the instructions shown on the wizard page.
3. When you have completed the procedure for the current wizard page, click **Next** to continue. The next page of the wizard opens.
4. Repeat this process until you get to the end of the wizard procedure. When you complete the last page, the sensor calibration is finished and the wizard closes.

### ***Belt Latch Calibration***

**NOTE:** This feature operates differently in emulation mode. For details on emulation mode wizards, see [Emulation Mode Wizards on page 185](#).

This section describes the Belt Latch Calibration wizard in the Process Manager, which is accessed from the Sensor Calibrations group. The Sensor Calibrations group defines the robot-to-sensor calibrations for the selected workcell process. For more details on the Sensor Calibrations group, see [Sensor Calibrations on page 909](#).

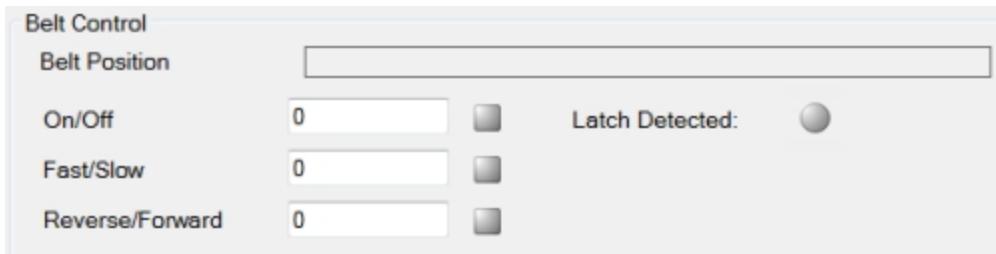


### *Sensor Calibrations Group*

The calibration wizard provides an interactive series of steps that guide you through the calibration process. Each step of the wizard contains graphical aids and text that describe the particular step being performed. As each step of the wizard is completed, the information from that step is collected and used to populate the fields in the Sensor Calibration Editor. For details, see [Sensor Calibrations on page 909](#).

For details on the buttons and indicators on the wizard pages, see [Wizards on page 181](#).

The Belt Latch Calibration wizard includes controls for moving the belt, along with visual indicators for belt on/off, fast/slow, reverse/forward, and latch detected, which show the current state of the conveyor and latch signal.



### *Belt Controls and Visual Indicators*

To access the Belt Latch Calibration wizard:

1. Click **Calibrate** in the Sensor Calibrations group. The wizard initializes and opens the first page.
2. Read and follow the instructions shown on the wizard page.
3. When you have completed the procedure for the current wizard page, click **Next** to continue. The next page of the wizard opens.
4. Repeat this process until you get to the end of the wizard procedure. When you complete the last page, the sensor calibration is finished and the wizard closes.

### **Fixed Camera Calibration**

**NOTE:** This feature operates differently in emulation mode. For details on emulation mode wizards, see Emulation Mode Wizards on page 185.

This section describes the Fixed Camera Calibration wizard in the Process Manager, which is accessed from the Sensor Calibrations group. The Sensor Calibrations group defines the robot-to-sensor calibrations for the selected workcell process. For more details on the Sensor Calibrations group, see Sensor Calibrations on page 909.



*Sensor Calibrations Group*

The calibration wizard provides an interactive series of steps that guide you through the calibration process. Each step of the wizard contains graphical aids and text that describe the particular step being performed. As each step of the wizard is completed, the information from that step is collected and used to populate the fields in the Sensor Calibration Editor. For details, see Sensor Calibrations on page 909.

For details on the buttons and indicators on the wizard pages, see Wizards on page 181.

**NOTE:** If your workcell contains obstacles that may interfere with automated movement of the robot during calibration, you must use the manual calibration procedure. For details, see Sensor Calibrations on page 909.

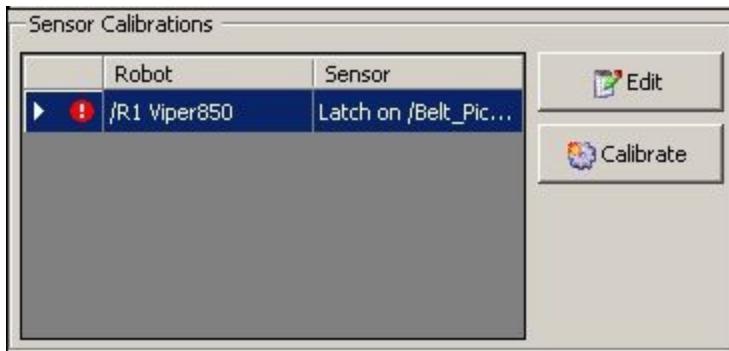
To access the Fixed Camera Calibration wizard:

1. Click **Calibrate** in the Sensor Calibrations group. The wizard initializes and opens the first page.
2. Read and follow the instructions shown on the wizard page.
3. When you have completed the procedure for the current wizard page, click **Next** to continue. The next page of the wizard opens.
4. Repeat this process until you get to the end of the wizard procedure. When you complete the last page, the sensor calibration is finished and the wizard closes.

### **Spacing Reference Calibration**

**NOTE:** This feature operates differently in emulation mode. For details on emulation mode wizards, see Emulation Mode Wizards on page 185.

This section describes the Spacing Reference Calibration wizard in the Process Manager, which is accessed from the Sensor Calibrations group. The Sensor Calibrations group defines the robot-to-sensor calibrations for the selected workcell process. For more details on the Sensor Calibrations group, see Sensor Calibrations on page 909.



*Sensor Calibrations Group*

The calibration wizard provides an interactive series of steps that guide you through the calibration process. Each step of the wizard contains graphical aids and text that describe the particular step being performed. As each step of the wizard is completed, the information from that step is collected and used to populate the fields in the Sensor Calibration Editor. For details, see Sensor Calibrations on page 909.

For details on the buttons and indicators on the wizard pages, see Wizards on page 181.

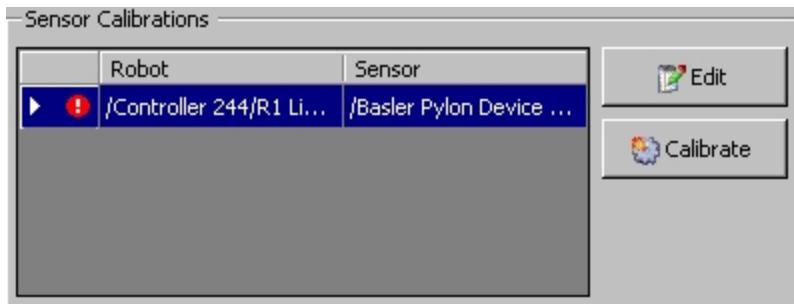
To access the Spacing Reference Calibration wizard:

1. Click **Calibrate** in the Sensor Calibrations group. The wizard initializes and opens the first page.
2. Read and follow the instructions shown on the wizard page.
3. When you have completed the procedure for the current wizard page, click **Next** to continue. The next page of the wizard opens.
4. Repeat this process until you get to the end of the wizard procedure. When you complete the last page, the sensor calibration is finished and the wizard closes.

### **Refinement Camera Calibration**

**NOTE:** This feature operates differently in emulation mode. For details on emulation mode wizards, see Emulation Mode Wizards on page 185.

This section describes the Refinement Camera Calibration wizard in the Process Manager, which is accessed from the Sensor Calibrations group. The Sensor Calibrations group defines the robot-to-sensor calibrations for the selected workcell process. For more details on the Sensor Calibrations group, see Sensor Calibrations on page 909.



### *Sensor Calibrations Group*

The calibration wizard provides an interactive series of steps that guide you through the calibration process. Each step of the wizard contains graphical aids and text that describe the particular step being performed. As each step of the wizard is completed, the information from that step is collected and used to populate the fields in the Sensor Calibration Editor. For details, see Sensor Calibrations on page 909.

For details on the buttons and indicators on the wizard pages, see Wizards on page 181.

**NOTE:** If your workcell contains obstacles that may interfere with automated movement of the robot during calibration, you must use the manual calibration procedure. For details, see Sensor Calibrations on page 909.

To access the Refinement Camera Calibration wizard:

1. Click **Calibrate** in the Sensor Calibrations group. The wizard initializes and opens the first page.
2. Read and follow the instructions shown on the wizard page.
3. When you have completed the procedure for the current wizard page, click **Next** to continue. The next page of the wizard opens.
4. Repeat this process until you get to the end of the wizard procedure. When you complete the last page, the sensor calibration is finished and the wizard closes.

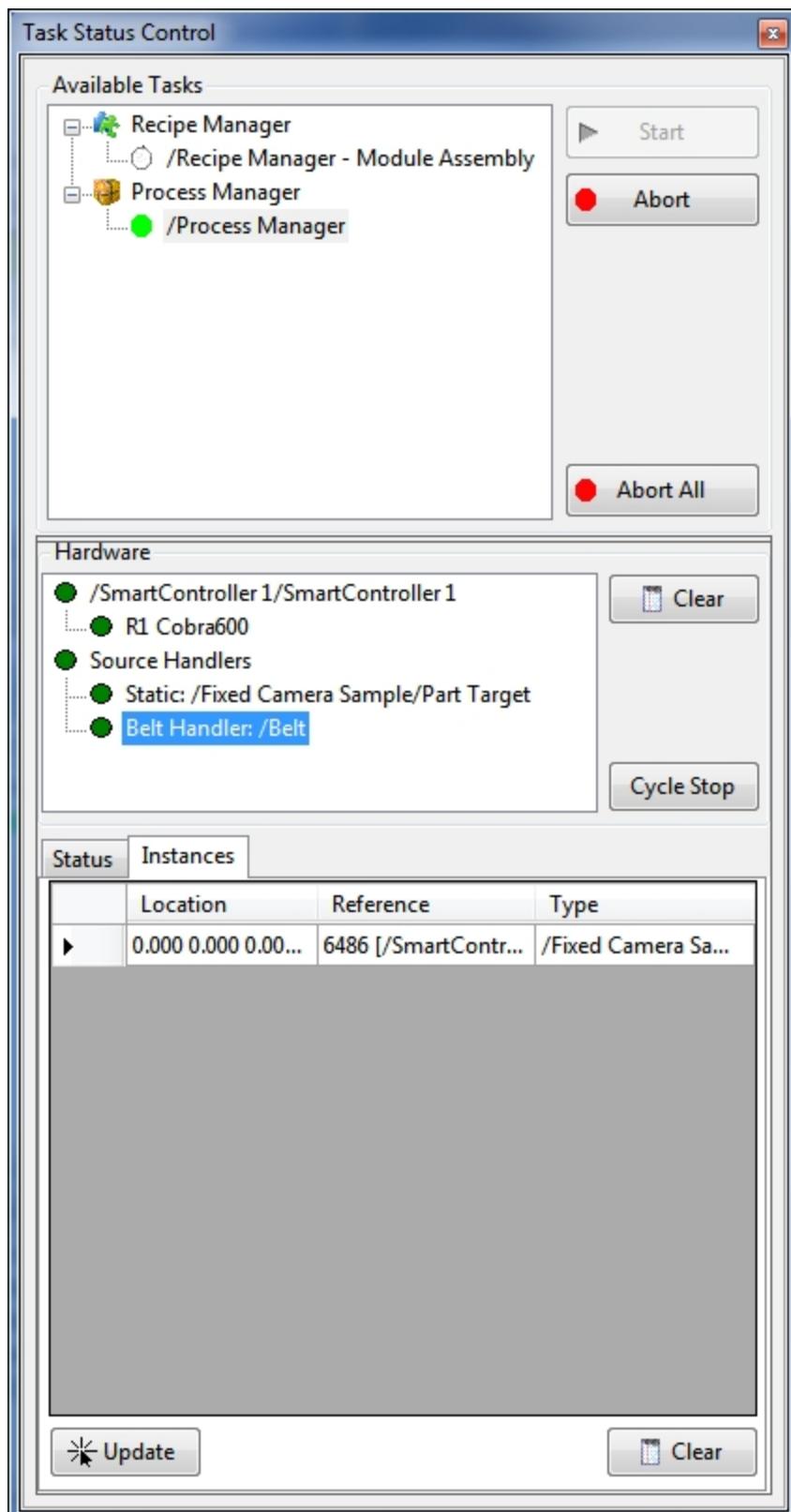
## Process Manager Control

**NOTE:** The Process Manager Control requires the USB hardware key (dongle), and Adept controller licenses, to enable complete functionality. See Licensing Requirements on page 28 for details.

The Process Manager Control is used to start and stop a process-managed application, such as an ACE Pack-Xpert packaging application. For details on creating a process-managed application, see Process Control on page 799.

### Using the Process Manager Control

To open the Task Status Control, select **View > Task Status Control** from the ACE menu. The Task Status Control opens. Use the pull-down selector to select an existing Process Manager task. The Process Manager control items are added to the Task Status Control, as shown in the following figure.



*Process Manager Control embedded in the Task Manager*

## **Editor Parameters**

The Process Manager Control is used to select the Process Manager, start and stop the selected application, and to view status and instances on the application while it is operating.

**NOTE:** Process Manager statistics are viewed and plotted using the System Monitor tool. For details, see System Monitor on page 1117.

### **Available Tasks Group**

The Available Tasks group is used to select the task (program) for execution. The available tasks are arranged in a "tree view", which is grouped by task type (C# Program, Process Manager, etc.).

When you double-click a task to select it, the selected object is also highlighted in the Workspace Explorer folder pane.

Item	Description
Abort	Enabled during execution, pauses or exceptions, this button causes the execution to stop for the selected task.
Abort All	Enabled during execution, pauses or exceptions, this button causes the execution to stop for all tasks.
Start	Starts (executes) the selected task.

Programs that are waiting for user interaction are marked as follows:

Icon	Description
	The program is running
	The program is paused
	The program is stopped (aborted)

### **Hardware Group**

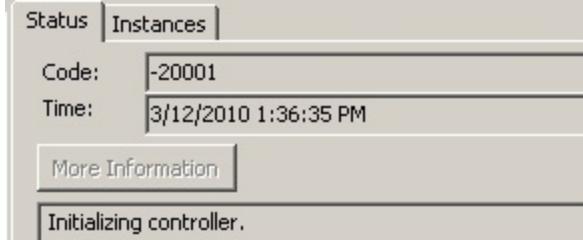
This area displays the hardware items in the selected Process Manager, and their status, as follows:

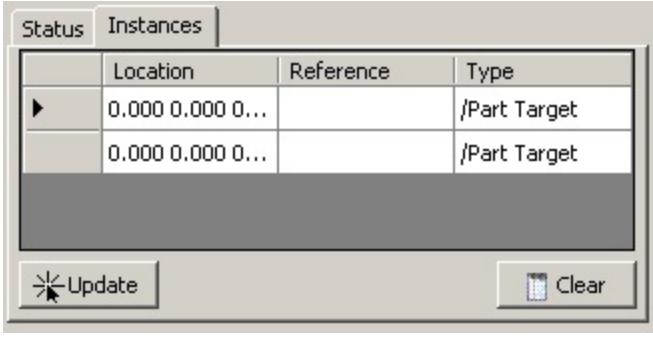
Item	Description
Idle state	Indicates the application is idle (not operating).
Operating state	Indicates the application is operating.
Warning state	Indicates a warning condition. The specific warning message is displayed on the Status tab. See the <a href="#">Status tab description</a> (in the next section) for details.
Error state	Indicates an error condition. The specific error message is displayed on the Status tab. See the <a href="#">Status tab description</a> (in the next section) for details.
Clear	Clears all Part and Part Target instances from the system.
Cycle Stop	<p>Sends a signal to each robot to stop after the current process cycle has completed. Each robot stops after it reaches the end of its current process, and then the following message is displayed:</p> <p>The screenshot shows a portion of the Process Manager control interface. At the top, there are two tabs: 'Status' and 'Instances'. Below the tabs, there is a message area containing the following information:</p> <ul style="list-style-type: none"> <li>Code: 2002</li> <li>Time: 11/8/2010 11:35:14 AM</li> <li>A button labeled 'More Information'</li> <li>A message box containing the text: "Robot is waiting because of a cycle stop request."</li> </ul> <p>You can resume the robot(s) and process operations by clicking the Cycle Stop button.</p> <p>The Cycle Stop feature gives the capability to implement a variety of situations. For example, you could click Cycle Stop and leave the system running. When the system is in this state, all tracking is enabled. Therefore, you could either click Abort (which stops everything), or you could simply resume the current cycle by clicking Cycle Stop again.</p>

### Application Information

There are three application information pages: Status and Instances, which provide feedback on the operation of the selected application. To view an information page, click the corresponding tab.

**NOTE:** When the robot is waiting for parts or part targets, a yellow warning condition is displayed on the Process Manager control. It shows what part or part target the robot is waiting for. If the robot is waiting because a process has not been selected, it shows a similar warning indicating that a process has not been selected. For details on selecting and running a process, see the next section Controlling an Application on page 925.

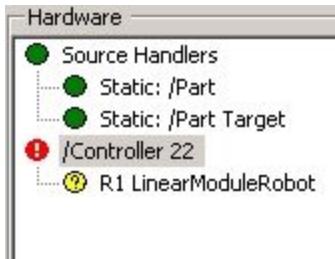
Item	Description										
Status	<p>This tab displays information on the status of the "control components" driving the process. This is similar to the <a href="#">Task Status Control</a> in the ACE software. It shows the hardware in the system and the status of the selected item. For more details on status codes and messages, see Status/Error Codes on page 932.</p>  <p><i>Status Tab with Initialization Message</i></p>  <p><i>Status Tab with Latch Signal Error Message</i></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #cccccc;">Item</th><th style="background-color: #cccccc;">Description</th></tr> </thead> <tbody> <tr> <td>Code</td><td>Displays the error number for the message.</td></tr> <tr> <td>Time</td><td>Displays the controller time when the error occurred.</td></tr> <tr> <td>More Information</td><td>Displays the details of an ACE exception by showing the contents of the V+ program stack, when available (the exception source must be a V+ error).</td></tr> <tr> <td>(Message area)</td><td>Displays status and error-message text.</td></tr> </tbody> </table>	Item	Description	Code	Displays the error number for the message.	Time	Displays the controller time when the error occurred.	More Information	Displays the details of an ACE exception by showing the contents of the V+ program stack, when available (the exception source must be a V+ error).	(Message area)	Displays status and error-message text.
Item	Description										
Code	Displays the error number for the message.										
Time	Displays the controller time when the error occurred.										
More Information	Displays the details of an ACE exception by showing the contents of the V+ program stack, when available (the exception source must be a V+ error).										
(Message area)	Displays status and error-message text.										

Item	Description	
	Item	Description
Instances	Retry	These buttons, and their functions, are enabled and controlled by the V+ error message.
	Skip	
	Abort	
Instances	<p>This tab displays information on the parts and part targets that are associated with each control source.</p>  <p><i>Instances Tab with Part Instances</i></p> <p>When the tab is initially selected, the list is refreshed. After that, you must use the Update button to view the latest results. This is done to conserve system resources.</p> <p>The information can be removed using the Clear button. This clears the information for this tab only. To remove all information, use the Clear button at the top of the Process Manager Control.</p>	

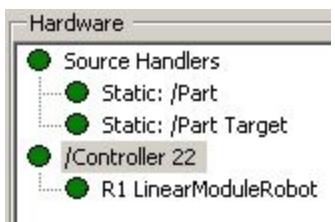
## Controlling an Application

The Process Manager toolbar or Task Status Control allows you to control (start and stop) the operation of the selected application, as follows:

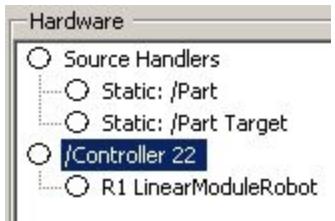
1. Use the drop-down list box to choose the Process Manager (application) you wish to use.
2. Click **Start**. The Start button dims, the application initializes and the Process Manager establishes communication with the hardware components. During initialization, the Hardware group items will look similar to this:



Note that some items on the Hardware list are in Error and Warning states until the Process Manager establishes communications with and initializes those items. After initialization has completed, the Hardware group items will look similar to this:



3. The selected Process Manager application runs continuously. To stop the application, click **Stop**. The Stop button dims and the Hardware group displays the idle state.



## Process Pallet

This section describes the Process Pallet option on the Process menu.

The Process Pallet object defines the layout of a pallet, which can be used to pick parts from or place parts to.

**NOTE:** When used with a camera or belt, the camera or belt will be configured to locate the origin of the pallet, not the parts in the pallet.

The Process Pallet editor provides an interface for setting various pallet-related parameters, such as the pallet configuration, location, and properties. The pallet is created as a separate object that can be linked with a frame. The Process Pallet object defines the dimensional information only. When linked to a frame it will position the pallet in Cartesian space.

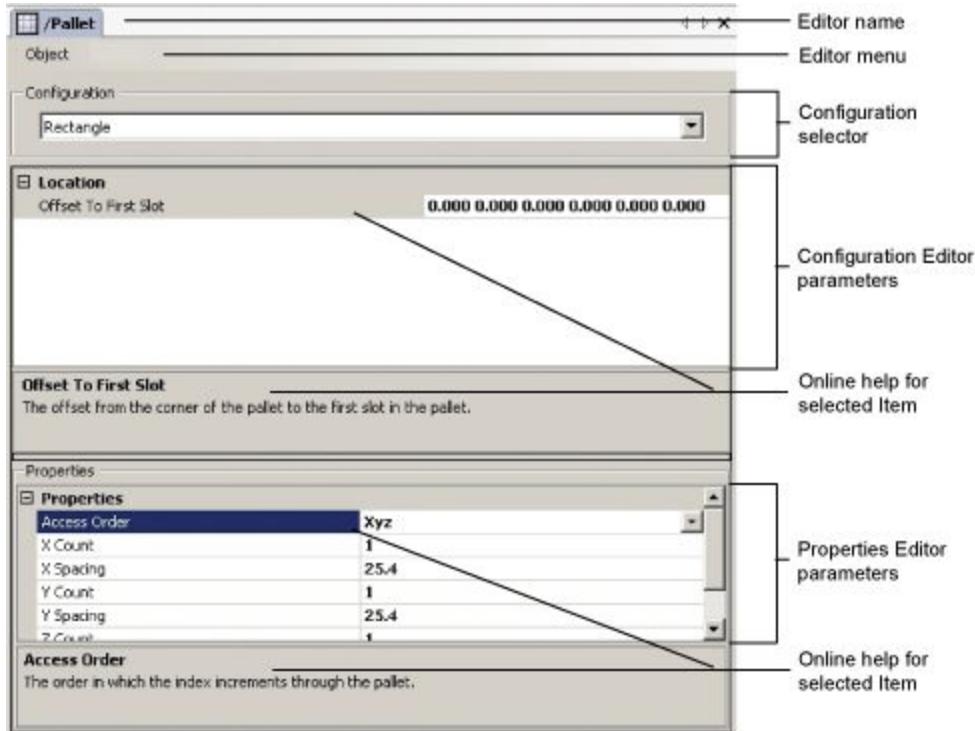
The Process Pallet editor allows you to specify a custom pallet configuration. This allows you to define individual X, Y, and Degree positions of each slot. Further, you can define circular pallets or pallets with offset rows. The software calculates the individual positions based on the input data and defines the positions in the Process Pallet object.

To create a Pallet object, right-click in the folder area of the Workspace Explorer and select **New > Process > Process Pallet** from the menu. The Packaging Pallet object is added to the Workspace Explorer and the Pallet object editor opens.

You can also open the Process Pallet editor by double-clicking the Pallet object in the folder area of the Workspace Explorer.

## Menu Items

---



Pallet Editor

## Menu Items

This section describes the selections available from the Pallet editor menu.

### Object Menu



#### Help

Displays the online help for the object editor.

#### Refresh Editor

Refreshes the contents of the object editor window.

#### Close

Closes the object editor.

## Configuration, Editor Parameters, and Online Help

The middle portion of the Process Pallet editor contains the Configuration drop-down list box, Configuration editor items, Properties editor and online help for the editor parameters.

### Configuration Drop-down List Box

The Configuration drop-down list box is used to specify the type of pallet being used. The choices are:

- **Rectangle** - parts are arranged on the pallet in rows and columns.
- **Custom** - parts are arranged in rectangular or radial pattern. For a rectangular pallet, you specify the offsets, spacing, and part counts for X, Y, and Z. For radial pallets, you specify the start angle, angle spacing, part count, and radius.

#### **Configuration Editor**

The Configuration editor is used to specify configuration, location and 3D visualization information for the pallet. Each item is described in the online help for that item.

#### **Properties Editor**

The Properties editor is used to specify the access order, part count and part spacing for X, Y, and Z for the standard (rectangular) pallet. Each item is described in the online help for that item.

If a custom pallet is selected in the Configuration drop-down list box, this area changes to a table that contains information collected from the Add Pattern dialog. For details, see [Creating a Custom Pallet](#).

#### **Online Help**

The bottom parts of the Pallet editor displays online help for the selected parameters.

**NOTE:** Because online help for each parameter is provided directly in the editor, it is not duplicated in this documentation.

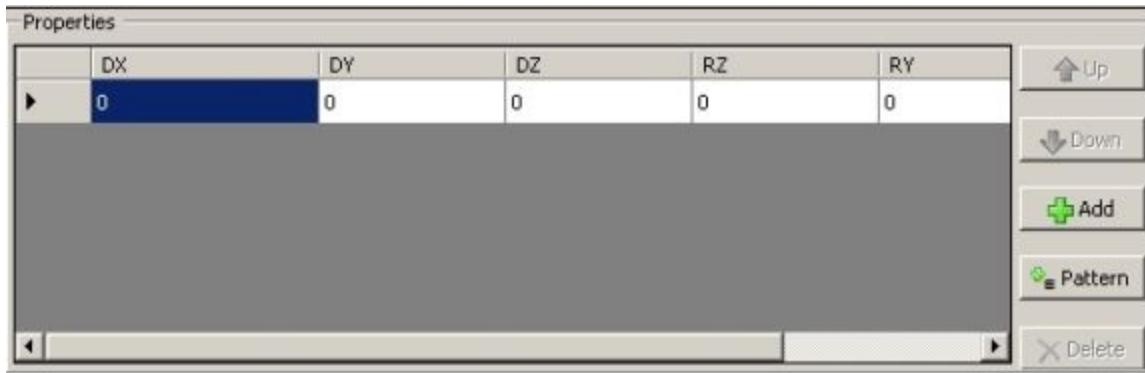
### **Creating a Custom Pallet**

To create a custom pallet with either a rectangular or radial configuration:

1. Click the Configuration drop-down list box to specify a custom pallet.



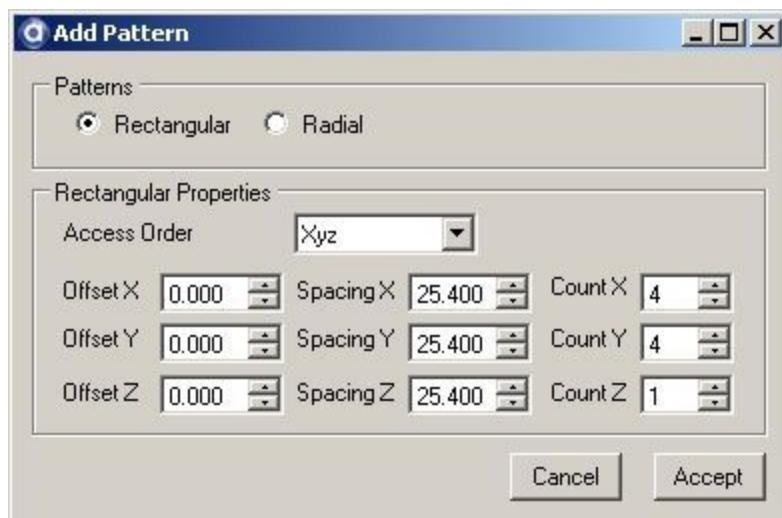
The Properties area changes to a grid, which is used to enter the pallet part-location values.



2. Use the Properties grid to enter the part-location values. Each row of the grid represents one part position on the pallet. For example, if your pallet is 4 x 4 x 1, you will have 16 parts on the pallet and 16 rows defined in the Properties grid.

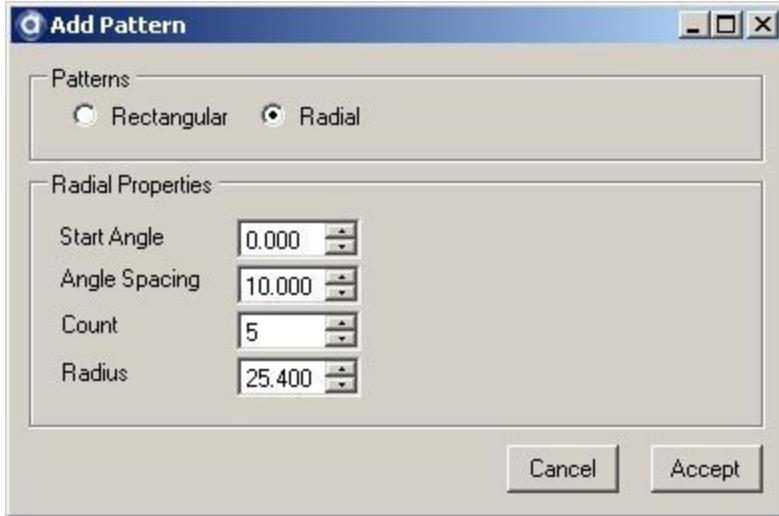
Optionally, you can click **Pattern** to define the pallet in the Add Pattern dialog, as described in step 4, below.

3. Use the Add button to add another row to the Properties grid.
  4. Optionally, click **Pattern** to display the Add Pattern dialog, which is used to define a Rectangular or Radial pallet configuration.
- If Rectangular is selected, the Add Pattern dialog displays the Rectangular Properties configuration items, as shown in the following figure:



- If Radial is selected, the Add Pattern dialog displays the Radial Properties configuration items,

as shown in the following figure:



- After the pallet is defined, click Accept to close the Add Pattern dialog. The part information is automatically added to the Properties grid. The following figure shows the results of the Rectangular pallet configuration, which was defined in the Add Pattern dialog above.

Properties						
	DX	DY	DZ	RZ	RY	RZ
▶	0	0	0	0	0	0
	25.4	0	0	0	0	0
	50.8	0	0	0	0	0
	76.1999...	0	0	0	0	0
	0	25.4	0	0	0	0
	25.4	25.4	0	0	0	0
	50.8	25.4	0	0	0	0

## Pallet Visualization

You can select a shape to represent the pallet in the 3D Visualization window. The shape is specified on the Part or Part Target object editor. The shape can be selected from: Box, Cylinder, or STL (STL mesh file). For more details on the Part or Part Target object editor, see Part on page 829 or Part Target on page 841. For more details on the 3D Visualization, see 3D Visualization on page 176.

## Status/Error Codes

This section describes the Process Manager status/error codes and their descriptions.

Configuration errors are displayed as prompts in the Process Manager (note that the error code is not displayed); runtime status/error codes and their descriptions are displayed on the Status tab of the Process Manager Control. For details on the Process Manager, see Process Manager on page 843. For details on the Process Manager Control, see Process Manager Control on page 920.

### Category Key

Each code/message shown in the next section is marked with the corresponding category. Note that some codes apply to more than one category. The categories are as follows:

C	Controller	F	Feeder
R	Robot	P	Process
G	Gripper	PT	Part / Part Target
CV	Camera / Vision	Pa	Pallet
BE	Belt / Encoder	Cs	Custom
L	Latch	O	Other

### Code and Description List

The following table provides a listing of the Process Manager status/error codes and corresponding descriptions. The codes are listed in ascending order.

#### Description Placeholders

Some description in the table have "placeholders", such as '{0}', in the text. For example, in the following status code/description:

-20112	'{0}' has not been calibrated to the spacing reference for belt '{1}'.
--------	--

In the example above, the '{0}' refers to a name of the robot, and '{1}' refers to the name of the belt.

Code	Description	C	R	G	CV	BE	L	F	P	PT	Pa	Cs	O
-20151	The Feeder is enabled but not selected.							x					
-20150	The Feeder device is not properly configured							x					
-20149	An incorrect sequence of latches was detected. Controller '{0}' reported a latch out of sequence.	x					x						

Code and Description List

---

<b>Code</b>	<b>Description</b>	<b>C</b>	<b>R</b>	<b>G</b>	<b>CV</b>	<b>BE</b>	<b>L</b>	<b>F</b>	<b>P</b>	<b>PT</b>	<b>Pa</b>	<b>Cs</b>	<b>O</b>
-20148	Custom process strategy selection program is not defined.							x			x		
-20147	The camera calibration references a latch that does not exist in '{1}'			x		x							
-20146	The latch calibration cannot have multiple latches					x							
-20145	The Custom Process Strategy Allocation Algorithm Has Failed To Execute.							x			x		
-20144	Custom Robot Process Strategy Algorithm Not Defined.		x					x			x		
-20143	The controller associated with belt '{1}' does not appear in the process list.	x				x		x					
-20142	Refinement station not defined in the process.			x				x					
-20141	The process has not been taught.							x					
-20140	There are no latches referenced by '{0}' for the belt '{1}'.					x	x						
-20139	Custom robot wait program is not defined.	x									x		
-20138	Custom robot program is not defined.	x									x		
-20137	Custom stop program is not defined.										x		
-20136	Custom error program is not defined.										x		
-20135	The expected latch signal associated with camera '{0}' was triggered multiple times on controller '{1}'.	x			x		x						
-20134	The expected latch signal associated with camera '{0}' was not detected on controller '{1}'.	x			x		x						
-20133	A configuration item name is too long: {2}.											x	
-20132	The calibration key is too long to write to the controller.	x											
-20131	The robot end-effector key name associated with '{1}' is too long to write to the controller.	x	x										
-20130	'{0}' is too long to write to the controller.	x											
-20129	'{0}' is defined as a belt spacing configuration.	x	x			x							

---

Code and Description List

---

<b>Code</b>	<b>Description</b>	<b>C</b>	<b>R</b>	<b>G</b>	<b>CV</b>	<b>BE</b>	<b>L</b>	<b>F</b>	<b>P</b>	<b>PT</b>	<b>Pa</b>	<b>Cs</b>	<b>O</b>
	It cannot be accessed by robots on different controllers.												
-20128	The controller associated with the feeder is not defined for source '{0}'.	x						x					
-20127	Custom strategy monitoring program is not defined.										x		
-20126	Custom initialization program is not defined.										x		
-20125	Custom belt program is not defined.					x					x		
-20124	Robot queue for belt monitoring has not been defined.		x			x							
-20123	The pallet associated with '{0}' does not have any available slots.										x		
-20122	Robot '{1}' end-effector does not have any tips.		x										
-20121	Robot '{1}' does not have a selected end-effector.		x										
-20120	Unable to execute the vision tool '{0}'			x									
-20119	The belt encoder in '{1}' associated with '{0}' cannot have multiple encoders associated with the spacing part.					x			x				
-20118	'{0}' must have one latch signal enabled on a controller associated with the camera.	x			x		x						
-20117	The encoder referenced by '{0}' is not configured to latch to the latching controller in the belt '{1}'.	x				x	x						
-20116	The encoder referenced by '{0}' is not wired into the robot controller '{2}' in the belt '{1}'.	x	x			x							
-20115	The encoder referenced by '{0}' is not wired into the robot controller in the belt '{1}'.	x	x			x							
-20114	'{0}' does not identify a Belt encoder.					x							
-20113	Robot '{1}' is not associated with a controller.	x	x										
-20112	'{0}' has not been calibrated to the spacing reference for belt '{1}'.					x							

---

Code and Description List

---

<b>Code</b>	<b>Description</b>	<b>C</b>	<b>R</b>	<b>G</b>	<b>CV</b>	<b>BE</b>	<b>L</b>	<b>F</b>	<b>P</b>	<b>PT</b>	<b>Pa</b>	<b>Cs</b>	<b>O</b>
-20111	'{0}' has not been calibrated to the latch for belt '{1}'.					x	x						
-20110	'{0}' does not identify a vision tool in the belt properties.				x	x							
-20109	'{0}' has not been calibrated to '{1}'.	x				x							
-20108	'{0}' has not been calibrated to the belt camera '{1}'.				x	x							
-20107	'{0}' has not been calibrated to '{1}'.		x		x								
-20106	'{0}' does not define the pallet in the pallet properties.										x		
-20105	'{0}' uses vision tool that does not reference a virtual camera.				x								
-20104	'{0}' does not define the vision tool being used.					x							
-20103	'{0}' does not identify a Belt.						x						
-20102	Robot is not defined in process.		x						x				
-20101	Part target not defined in the process.								x	x			
-20100	Part not defined in the process.								x	x			
-20027	Unable to retract the gripper tip			x									
-20026	Unable to extend the gripper tip			x									
-20025	The refinement operation did not detect the expected latch signal.				x		x						
-20024	Failure in the execution of the refinement vision tool			x									
-20023	There are too many results from the refinement operation				x								
-20022	There are no results from the refinement operation				x								
-20021	The refinement operation has timed-out				x								
-20020	The specified robot main control program does not exist.	x											

Code and Description List

---

<b>Code</b>	<b>Description</b>	<b>C</b>	<b>R</b>	<b>G</b>	<b>CV</b>	<b>BE</b>	<b>L</b>	<b>F</b>	<b>P</b>	<b>PT</b>	<b>Pa</b>	<b>Cs</b>	<b>O</b>
-20019	Starting the V+ code on the controller.	x											
-20018	Preparing the robots for control.		x										
-20017	Defining the belt parameters on the controller.	x				x							
-20016	Initializing code on the controller.	x											
-20015	Writing data objects to controller.	x											
-20014	The startup program does not exist.											x	
-20013	The specified robot place program does not exist.		x										
-20012	The specified robot pick program does not exist.		x										
-20010	The specified gripper index does not exist.			x									
-20009	The specified process index is invalid.										x		
-20008	The custom strategy program does not exist.											x	
-20007	The custom belt program does not exist.					x						x	
-20005	Vision tool execution failure.				x								
-20004	Unable to open the gripper tip			x									
-20003	Unable to close the gripper tip			x									
-20002	Unable to allocate a part to the robot.	x								x			
-20001	Initializing controller.	x											
-20000	Unknown Error.											x	
2000	No processes has been selected for the robot.		x							x			
2001	The robot is waiting for instances of type '{0}' to become available.		x										
2002	Robot is waiting because of a cycle stop request.		x										
2003	One or all of the encoders associated with the belt do not appear to be moving.					x							

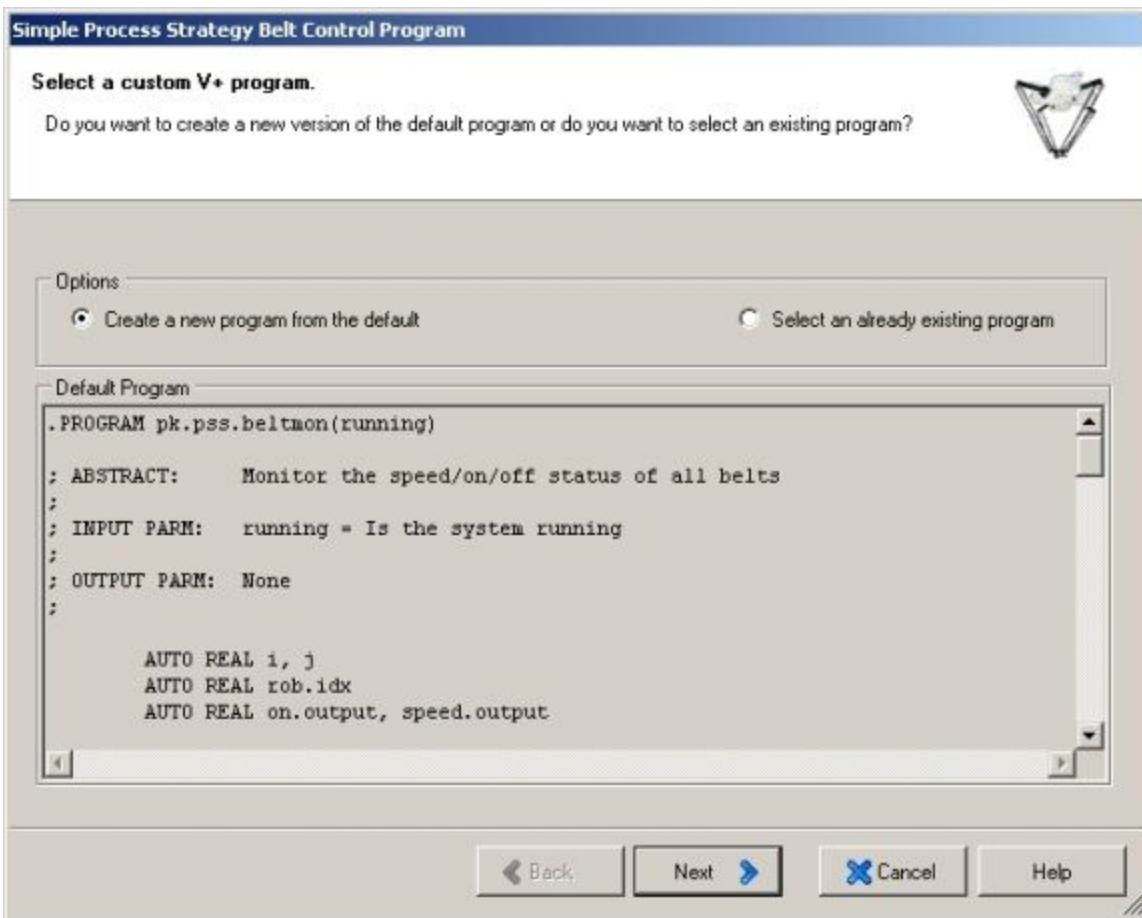
## Custom V+ Programs

This section describes the option for creating or selecting a custom V+ program. A custom V+ program can be used with different components of the Process Manager, such as a Process Strategy, a Control Source, or a Configuration Item. This provides greater control for system developers and a means for using an existing library of custom V+ programs.

### To create or select a custom V+ program

When a "custom" program option is selected, you can enter the name of the custom program, or click the Browse (  ) icon to create a new program or locate an existing program, as follows:

1. Click the Browse icon, a screen similar to the following is opened:



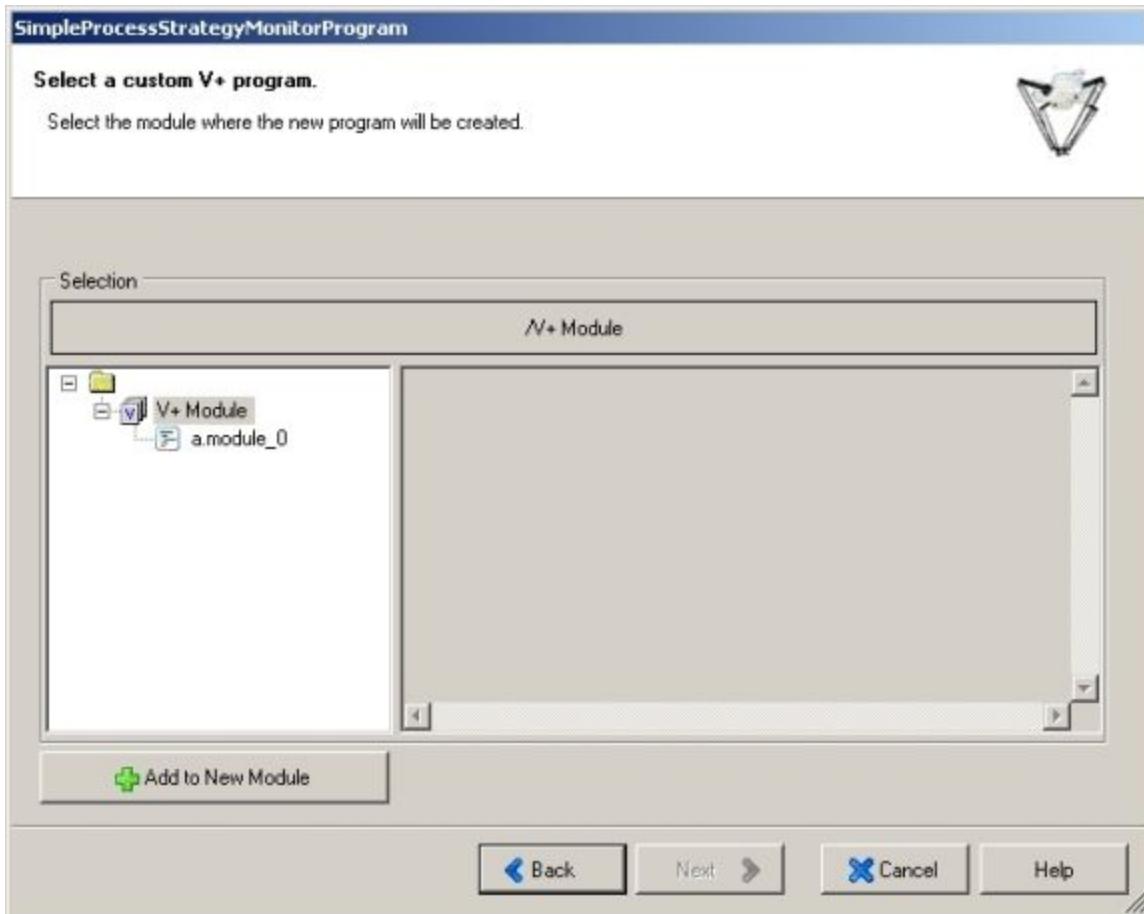
2. Use the radio button options to:

To create or select a custom V+ program

---

- Create a new program from the default program (the default program can be viewed in the Default Program window), or
  - Select an existing program.
3. Click **Next** to continue.

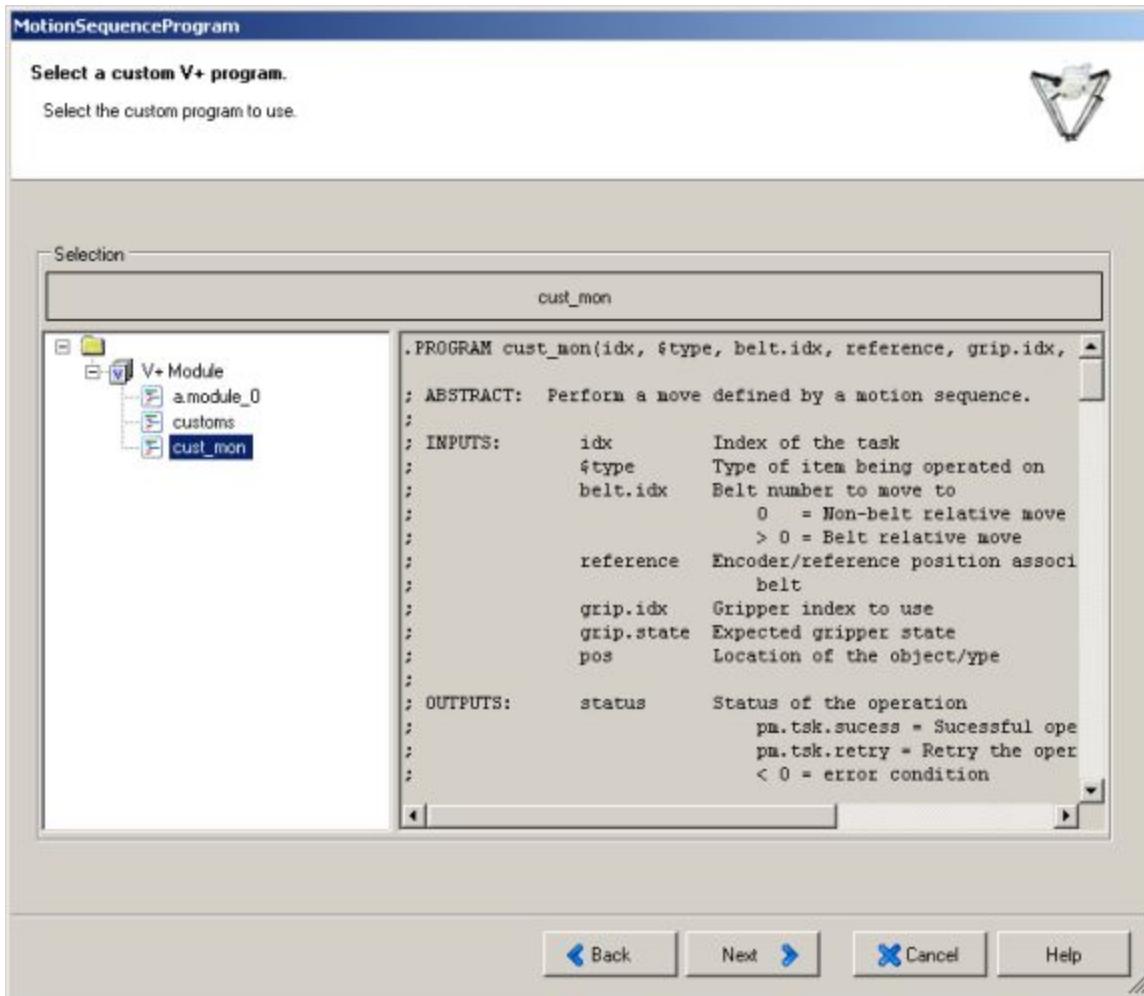
If you chose to create a new program, the following screen opens.



If you chose to select an existing program, the following screen opens.

To create or select a custom V+ program

---



4. If you chose to create a program, select an existing V+ module where the new program will be created, or click **Add to New Module** to create a new V+ module for the program.

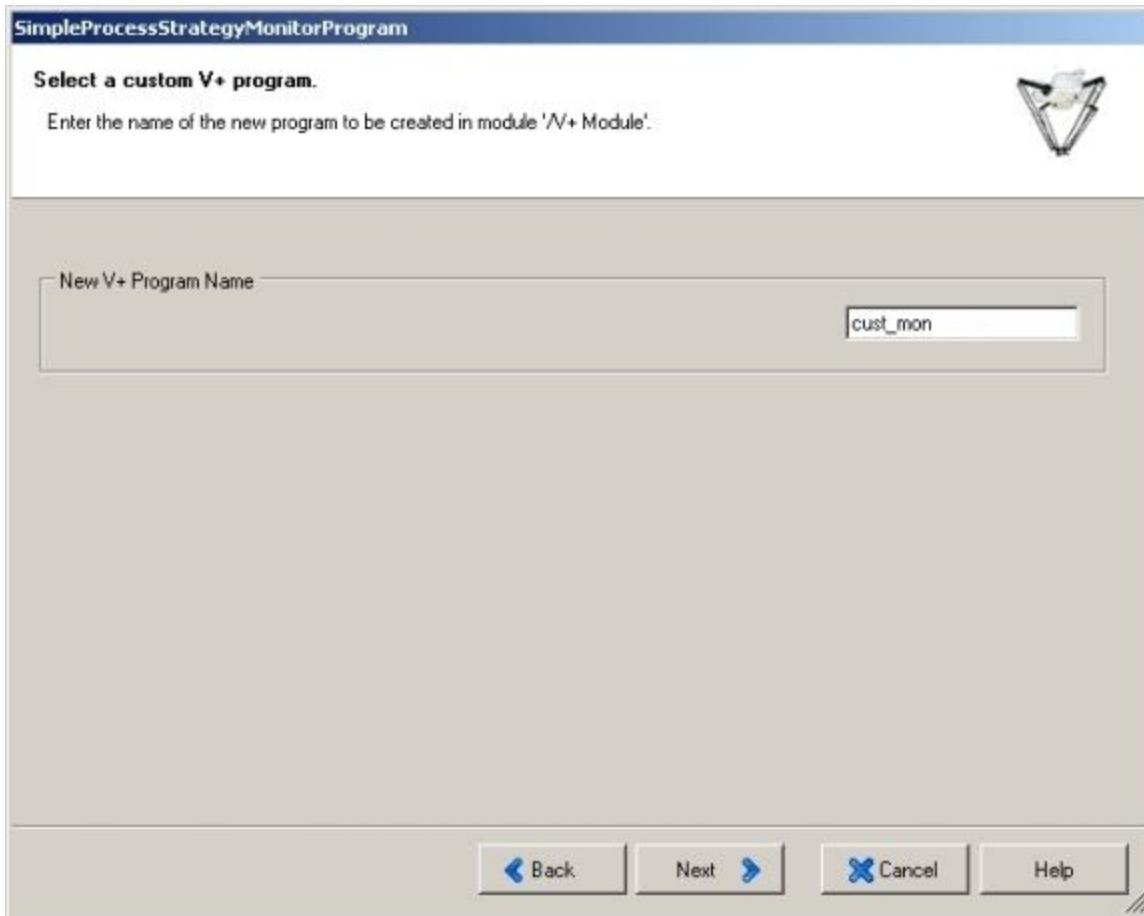
If you chose to select an existing program, select the program from the module/program list.

5. Click **Next** to continue.

If you chose to create a program, the wizard displays a prompt for the new V+ program name. Type the name for the program.

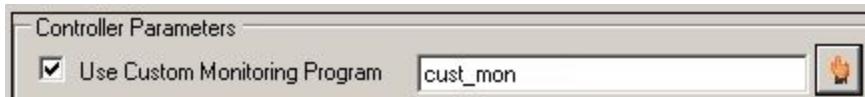
To create or select a custom V+ program

---



If you chose to select an existing program, the wizard skips the screen above and goes to the next step.

6. Click **Next**. The wizard closes and the custom program name displays in the program name field.



## Vision Refinement Station

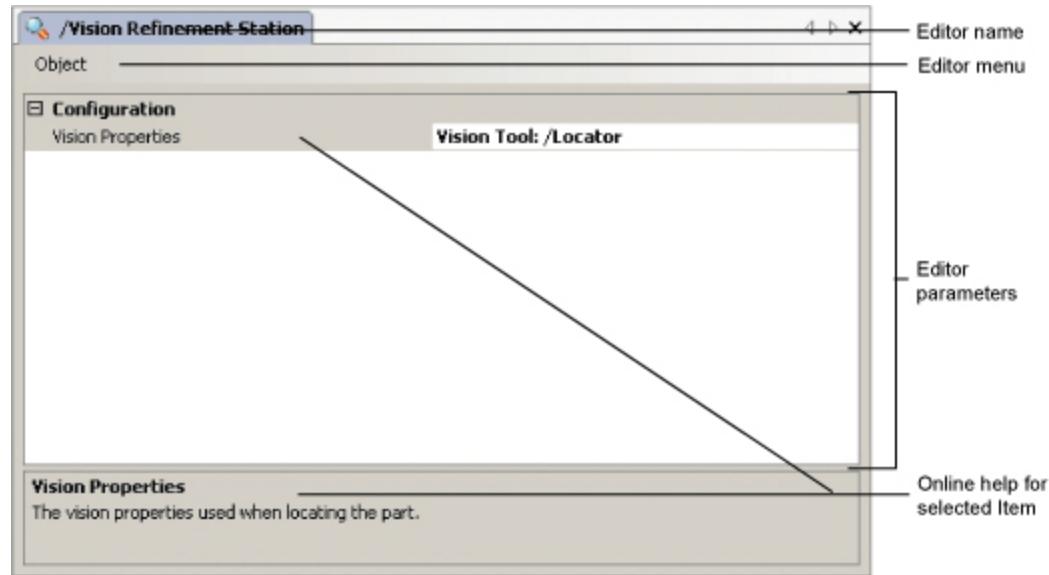
This section describes the Vision Refinement Station option on the Process menu.

**NOTE:** The following information assumes you have already installed a physical camera, created a virtual camera, calibrated the camera, and created a vision tool and model. For details, see the *ACE Sight User's Guide*.

The Vision Refinement Station object defines a location with an upward-mounted camera that is used to improve the part-to-gripper orientation for more accurate placement of the part. For more details on vision refinement, see Selecting a Refinement Mode on page 943.

To create a Vision Refinement Station object, right-click in the folder area of the Workspace Explorer and select **New > Process > Vision Refinement Station** from the menu. The Vision Refinement Station object is added to the Workspace Explorer and the Vision Refinement Station object editor opens.

You can also open the Vision Refinement Station editor by double-clicking the Vision Refinement Station object in the folder area of the Workspace Explorer.

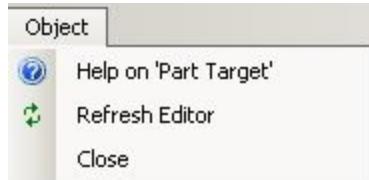


*Vision Refinement Station Editor*

After you create the object, you can set the motion parameters (how the robot moves to and from that location) using the location editors. For details, see Location Editor Types on page 858.

### Menu Items

This section describes the selections available from the Vision Refinement Station editor menu.

**Object Menu****Help**

Displays the online help for the object editor.

**Refresh Editor**

Refreshes the contents of the object editor window.

**Close**

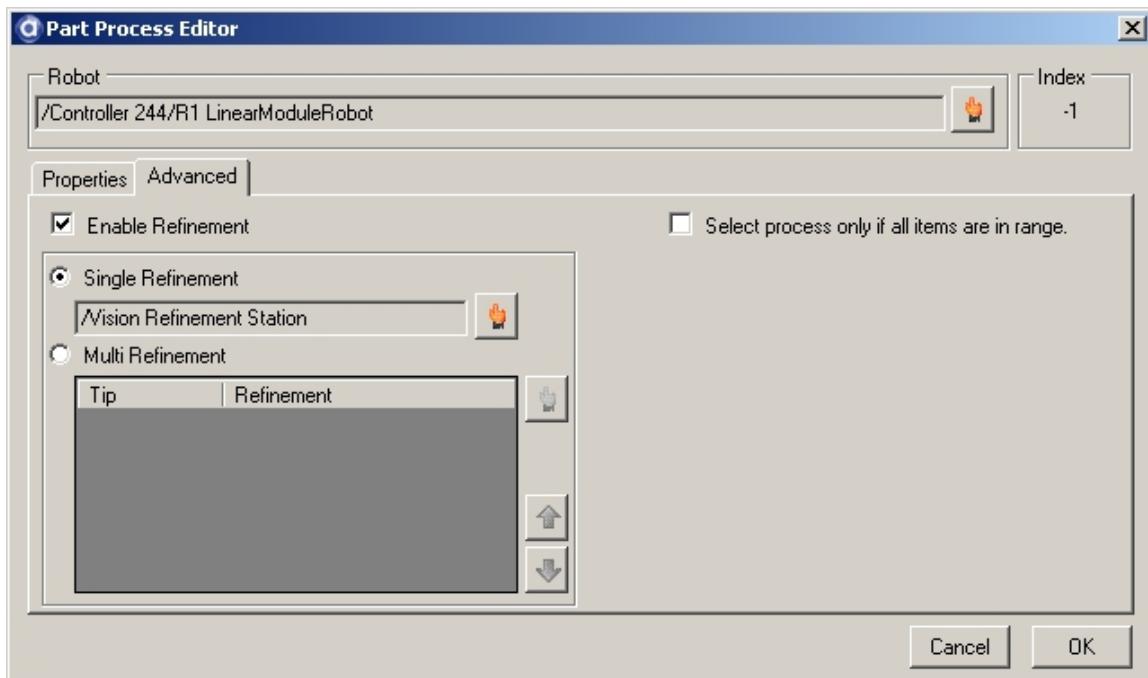
Closes the object editor.

**Configuration, Editor Parameters, and Online Help**

The middle portion of the Vision Refinement Station editor contains the Configuration drop-down list box, Configuration editor items and online help for the editor parameters. There is a single item, called Vision Properties, which is used to specify the vision tool that will be used to locate the part in the gripper. For more details on adding vision tools to the workspace, see the *ACE Sight User's Guide*.

**Adding the Vision Refinement Station to a Process**

After you create the Vision Refinement Station, it must be added to a pick-place process. This is done using the Advanced tab of the Part Process Editor, shown in the following figure. For more details, see Creating a Process on page 888.



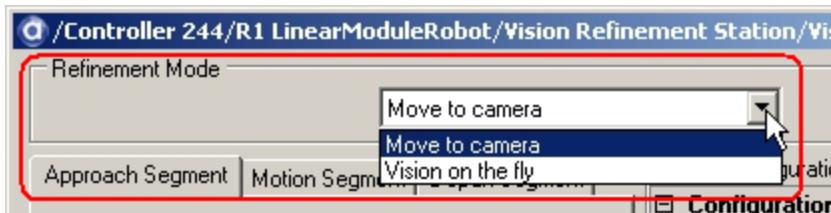
*Part Process Editor: Advanced Tab*

## Editing the Motion Parameters

After you add the Vision Refinement Station to the pick-place process, you can optionally edit the motion parameters for the station. This is done using the Location Editors that are accessed from the Configuration Items group. For details, see Location Editor Types on page 858.

### Selecting a Refinement Mode

The Vision Refinement Station has a Refinement Mode option that specifies the type of vision refinement that is applied to the part on the gripper. The Refinement Mode option is added to the Location Editor when editing the Configuration items for a Vision Refinement Station, as shown in the following figure.



*Refinement Mode Option for a Vision Refinement Station*

The refinement can be done in one of two ways:

- **Move to camera:** This is a "static" refinement, where the robot pauses at the Vision Refinement Station.
- **Vision on the fly:** This is an "in motion" refinement, where the robot passes through the Vision Refinement Station without any pause in the robot motion.

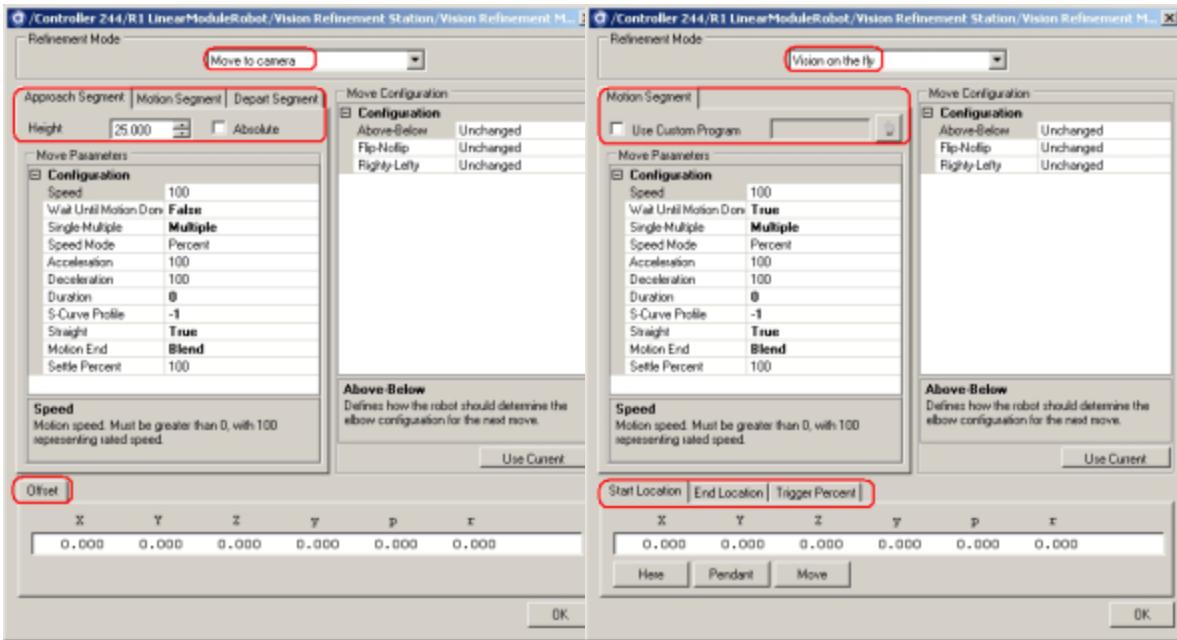
**NOTE:** The "Vision on the fly" mode will provide faster throughput, but may require more lighting (to achieve a fast shutter speed) than the "Move to camera" mode.

To select the refinement mode for the Vision Refinement Station:

1. Create the Vision Refinement Station, as described at the beginning of this topic.
2. Add the Vision Refinement Station to a pick-place process.
3. Select the Vision Refinement Station in the Configurations Items group of the Process Manager, and click **Edit**. The Location Editor opens. For details, see Location Editor Types on page 858. Because this is a Vision Refinement Station, the Refinement Mode options are included at the top of the Location Editor.
4. Select the desired refinement mode from the Refinement Mode options. The Configuration Items parameters change, based on the selected refinement mode. For details, see Configuration Items on page 856.

## Editing the Motion Parameters

---



*Refinement Mode: Move to camera*

*Refinement Mode: Vision on the fly*

5. Click **OK** on the Location Editor to save the changes.
6. Perform a sensor calibration, which calibrates the robot to the upward-mounted camera. For details, see Refinement Camera Calibration on page 918.
7. In the Processes group of the Process Manager, locate the process that is using the vision refinement and teach that process. For details, see Teaching a Process on page 894.
8. Optionally, you can return to the Location Editor, and adjust the other parameters for the Vision Refinement Station. For details, see Configuration Items on page 856.

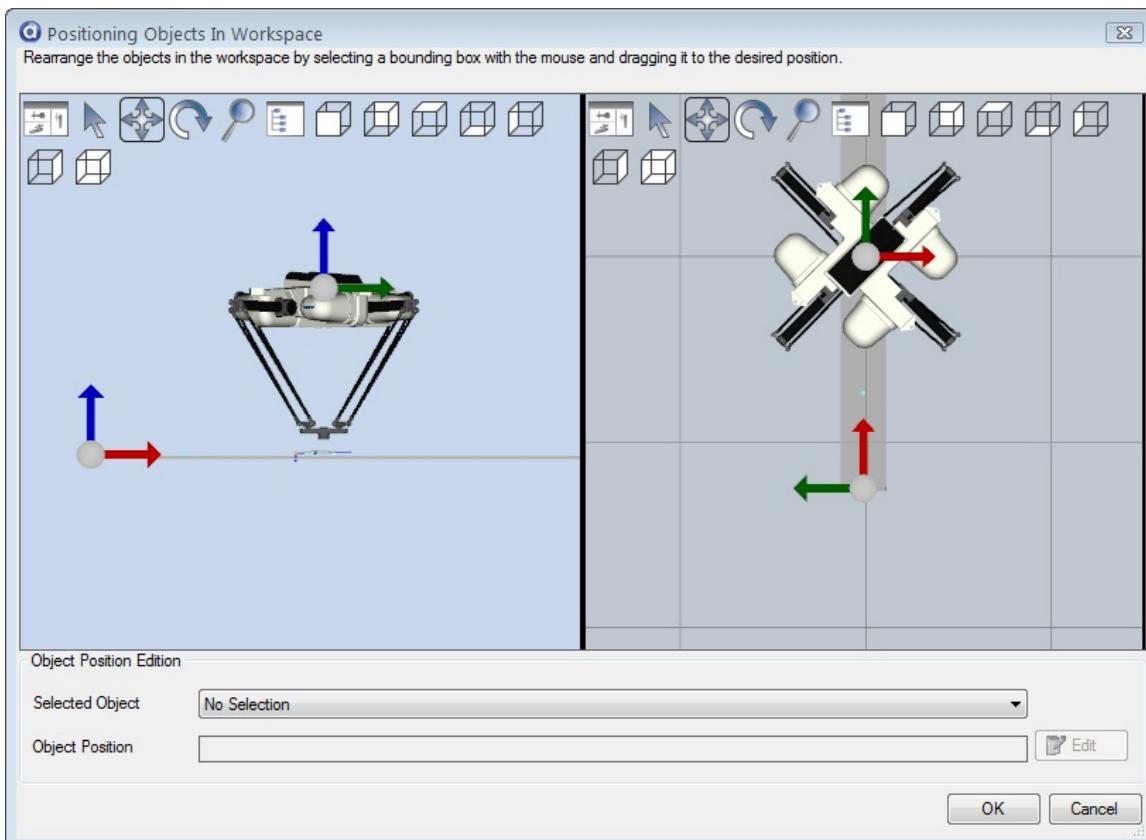
## Workspace Positioning

This section describes the Workspace Positioning tool.

The Workspace Positioning tool relates the positions of all robots and conveyor belts used in an application to a common workspace coordinate system. When multiple robots are used to access the same conveyor, the procedure sets the order (position) of those robots to that conveyor.

To use the Workspace Positioning tool:

- From the ACE menu bar, select **Tools > Workspace Positioning**. The Workspace Positioning tool opens, as shown in the following figure.



*Workspace Positioning Tool*

- Drag the selected object to the desired location in the workspace. Optionally, you can click **Edit** and type the coordinates for the object in the Edit Position dialog box.

**NOTE:** Each floor tile is sized to 1 square meter, which aids in positioning items in the workspace.

### *Object Coordinates*

3. Repeat the previous steps for the remaining workcell objects.
4. After you have completed positioning the workcell objects, click **OK** to save the changes and close the tool. Or, click **Cancel** to close the tool without saving the changes.

## Application Samples

The topics in this chapter describe the Application Samples feature of the ACE software.

Overview .....	948
Selecting the Application Sample .....	949
Cobra i-Series Pick and Place Samples .....	953
Static Pick and Place Sample .....	961
Belt Camera Part Pick Sample .....	966
Fixed Camera Part Pick Sample .....	974

## Overview

**NOTE:** This feature operates differently in emulation mode. For details on emulation mode, see Emulation Mode on page 1089.

The ACE software provides several application samples that can be used to learn about the basic features and functions of the software.

### ACE Sight: V+

The following ACE Sight vision application samples are currently available:

- Smart Controller Pick and Place Sequence

For more details on the ACE Sight vision application samples, see the *ACE Sight User's Guide*.

### MicroV+ Cobra i-Series/e-Vario

The following MicroV+ (Cobra i600/i800 robot) application samples are currently available:

- Cobra i-Series Pick and Place, which includes:
  - Pick and Place
  - Pick, Place to Pallet
  - Pick from Pallet, Place
  - Pick from Pallet, Place to Pallet
- MicroV+ Arm- or Fixed-Mount Camera (for details, see the *ACE Sight User's Guide*)

### Process Manager

The following Process Manager application samples are currently available:

- Static Pick and Place
- Belt Camera Part Picking
- Fixed Camera Part Picking

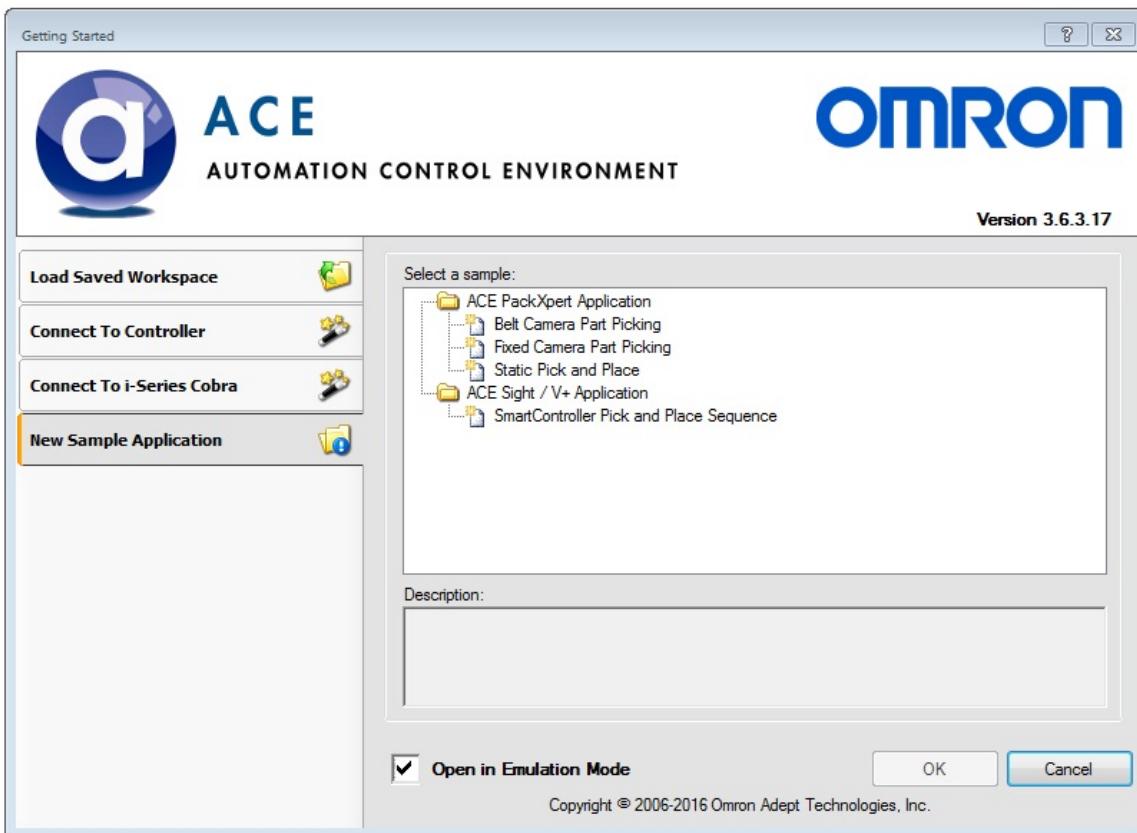
The setup and programming of the application samples is performed through an interview wizard. To use an application sample, you prepare the workcell and ACE workspace, select the application sample you wish to create, and click Select to launch the corresponding interview wizard. The interview wizard will guide you through the programming process.

---

## Selecting the Application Sample

To select the application sample:

1. Open the application sample selector by selecting **File > Load > New Sample Application**. The Select an Application Sample dialog opens, as shown in the following figure.



*Application Sample Selector*

2. Select an application sample from the Samples list.
3. Click **Select**. The interview wizard opens for the selected application sample. Follow the instructions shown in the interview wizard to complete your application.

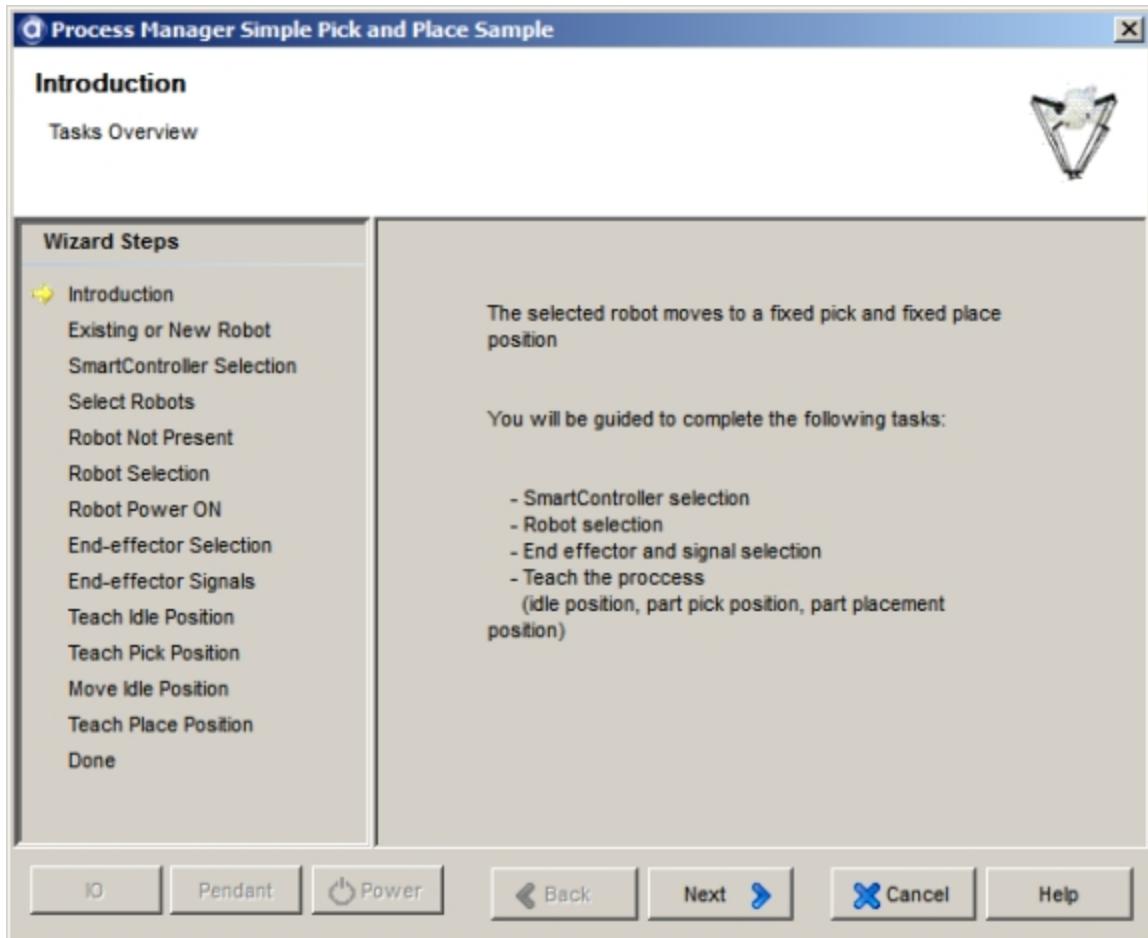
## Introduction and Tasks Overview

After the interview wizard opens, it displays the Introduction and Tasks Overview screen. The screen is divided into two parts: the left-hand area shows the complete list of steps needed to complete the interview wizard; the right-hand part shows the information and controls for the current step in the interview wizard. The current step (shown in the screen title) is also highlighted in blue text in the left-hand area, as shown in the following figure.

---

## To Program the Application:

---



### *Introduction and Tasks Overview*

**NOTE:** To emphasize the current task (right-hand area), the Wizard Steps pane (left-hand area) is omitted from the screen shots in the documentation.

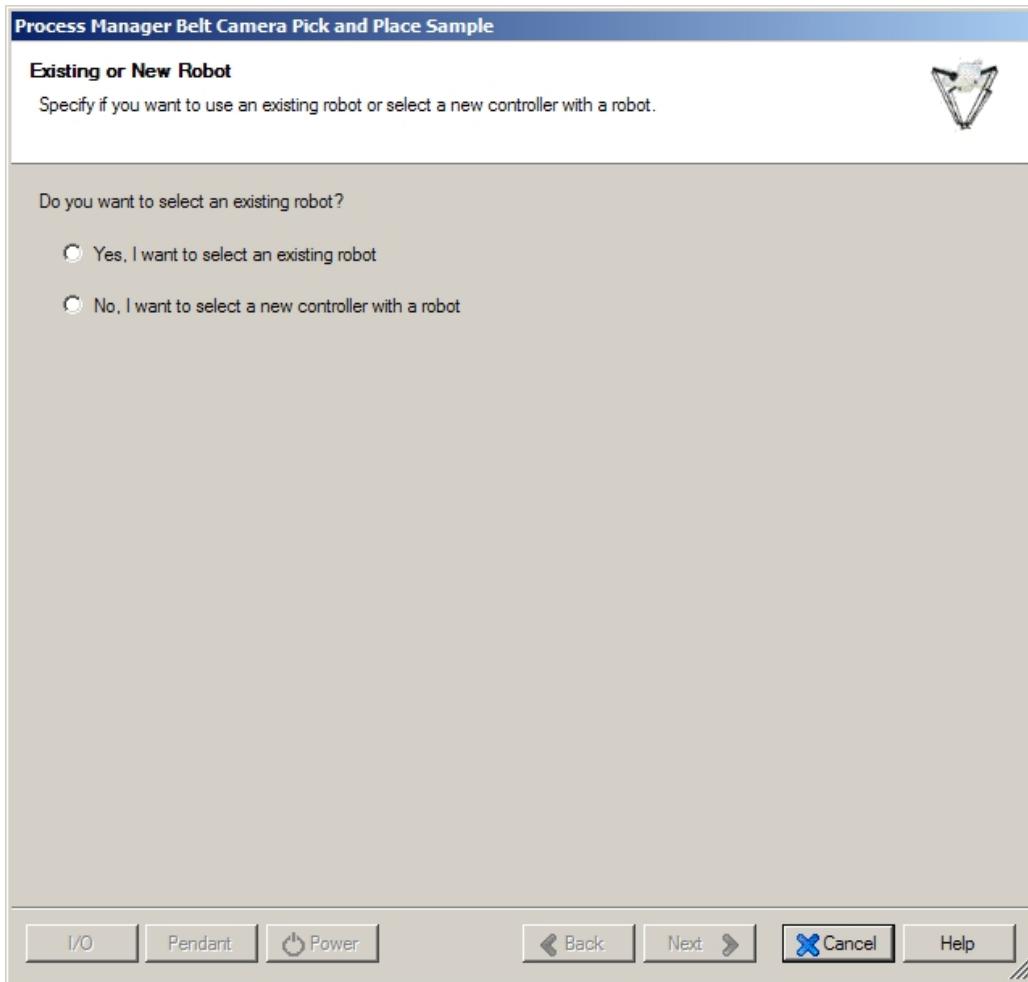
## To Program the Application:

After you select the application sample, the first page of the interview wizard opens. An introduction screen displays that describes the selected application sample and the steps that will be used to create it.

1. After you have finished reading this introductory information, click Next to proceed. The next screen of the interview wizard opens.
2. Specify if you want to use an existing robot, or select a new controller and robot. If there is no controller and robot in the workspace, this step will be skipped.

## To Program the Application:

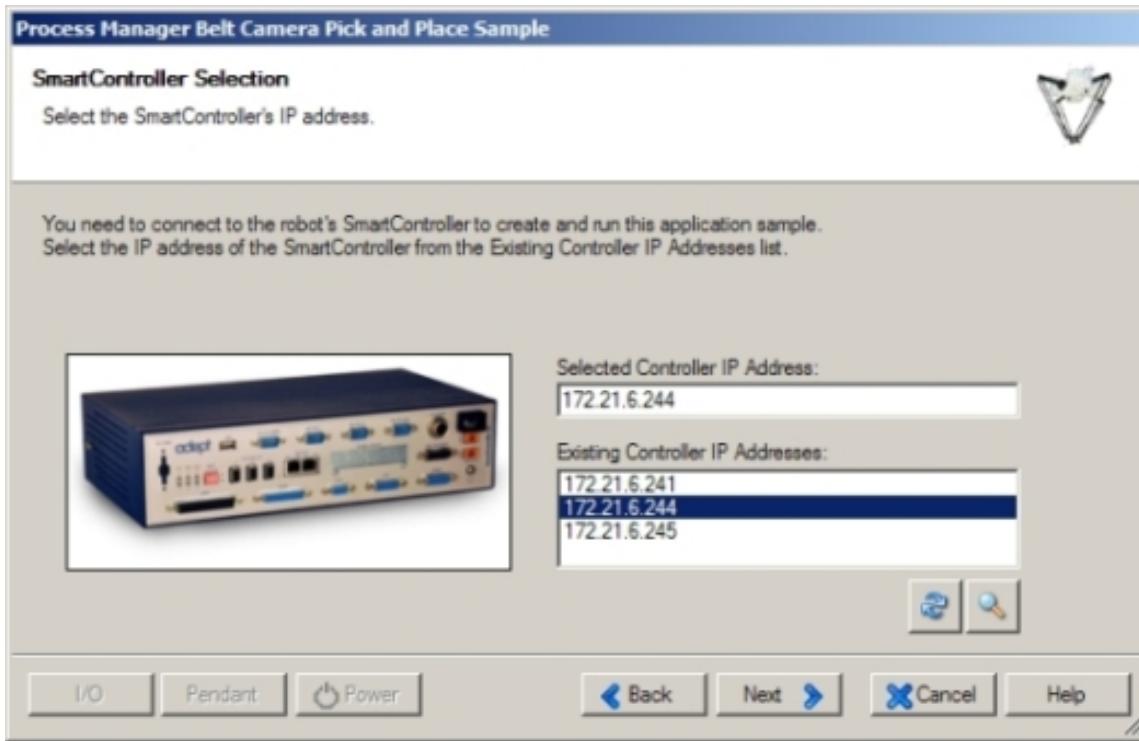
---



3. If you decide to use a new controller and robot, or there is no controller and robot in the workspace, the following screen opens. It allows you to select a SmartController. After you make a selection, the selected SmartController and attached robot(s) are installed in the workspace.

To Program the Application:

---



For the remaining steps, select the desired application sample from the following list:

- Cobra i-Series Pick and Place Samples on page 953
- Static Pick and Place Sample on page 961
- Belt Camera Part Pick Sample on page 966
- Fixed Camera Part Pick Sample on page 974

For more details on using the ACE Sight vision application samples, see the *ACE Sight User's Guide*.

## **Cobra i-Series Pick and Place Samples**

After you select the application sample, you are ready for the setup and programming of the application. An interview wizard guides you through the process, as described below.

The following instructions are provided as an example of a MicroV+ static pick and place application. The steps for your application sample may vary, depending on the application sample selected in the previous section. However, this example will provide an overview of the basic steps for programming an application sample.

### **What happens next?**

You will use the interview wizard to:

- Select the robot
- Teach the process. You will teach the Safe, Pick, and Place positions. As part of the Pick (or Place) teach process, you will select a mode for each (pick from a static position, or pick from a pallet).

The interview wizard will pause momentarily to create the objects and write the program code, then you will:

- Enable the signal numbers defined in the generated code.
- Run the process using the Task Manager panel of the Controller Development Tools window.

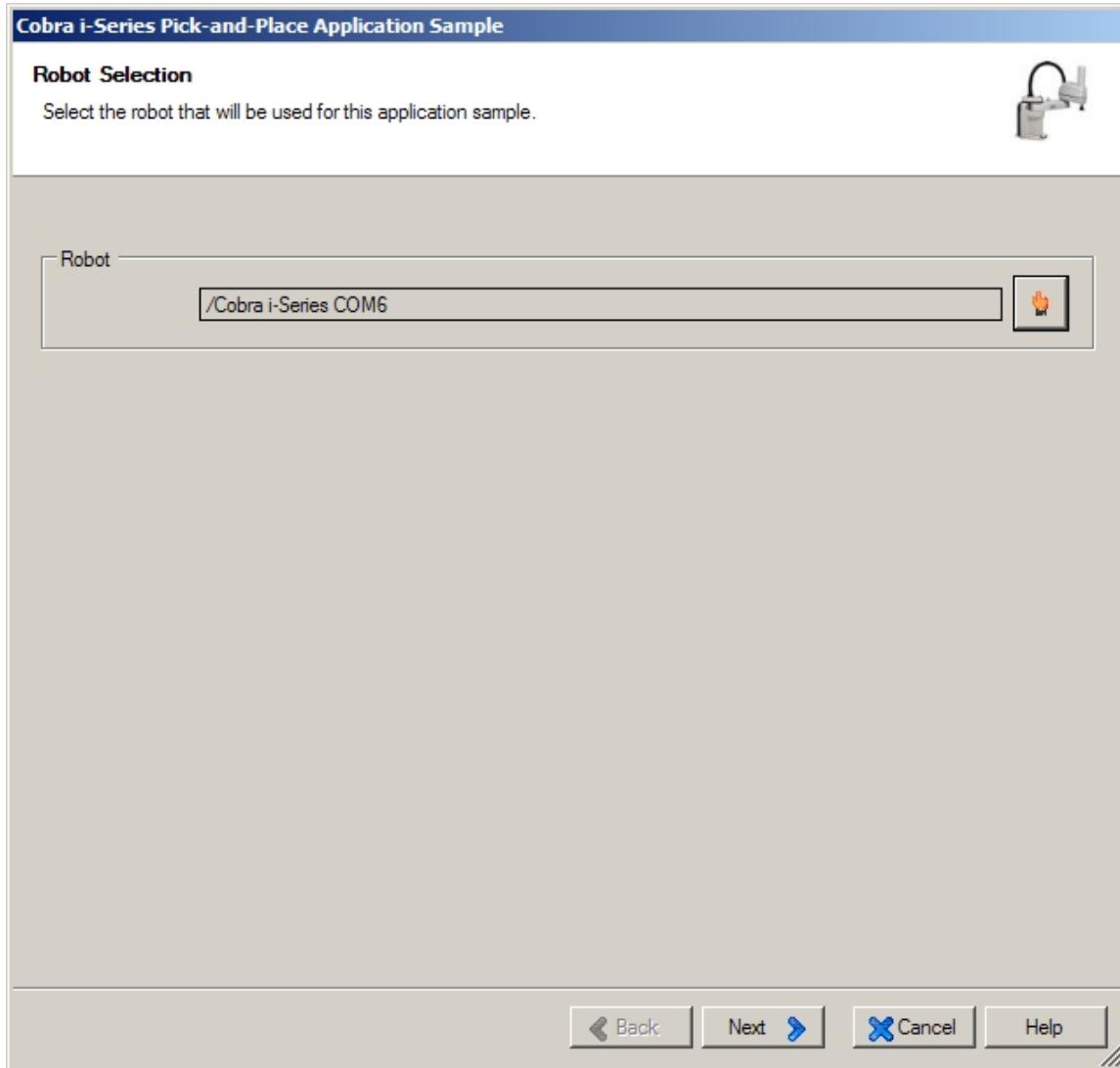
### **To program the application:**

After you select the application sample, the first page of the interview wizard opens.

If you already have a Cobra i-Series robot in your workspace, you will see the following screen.

To program the application:

---

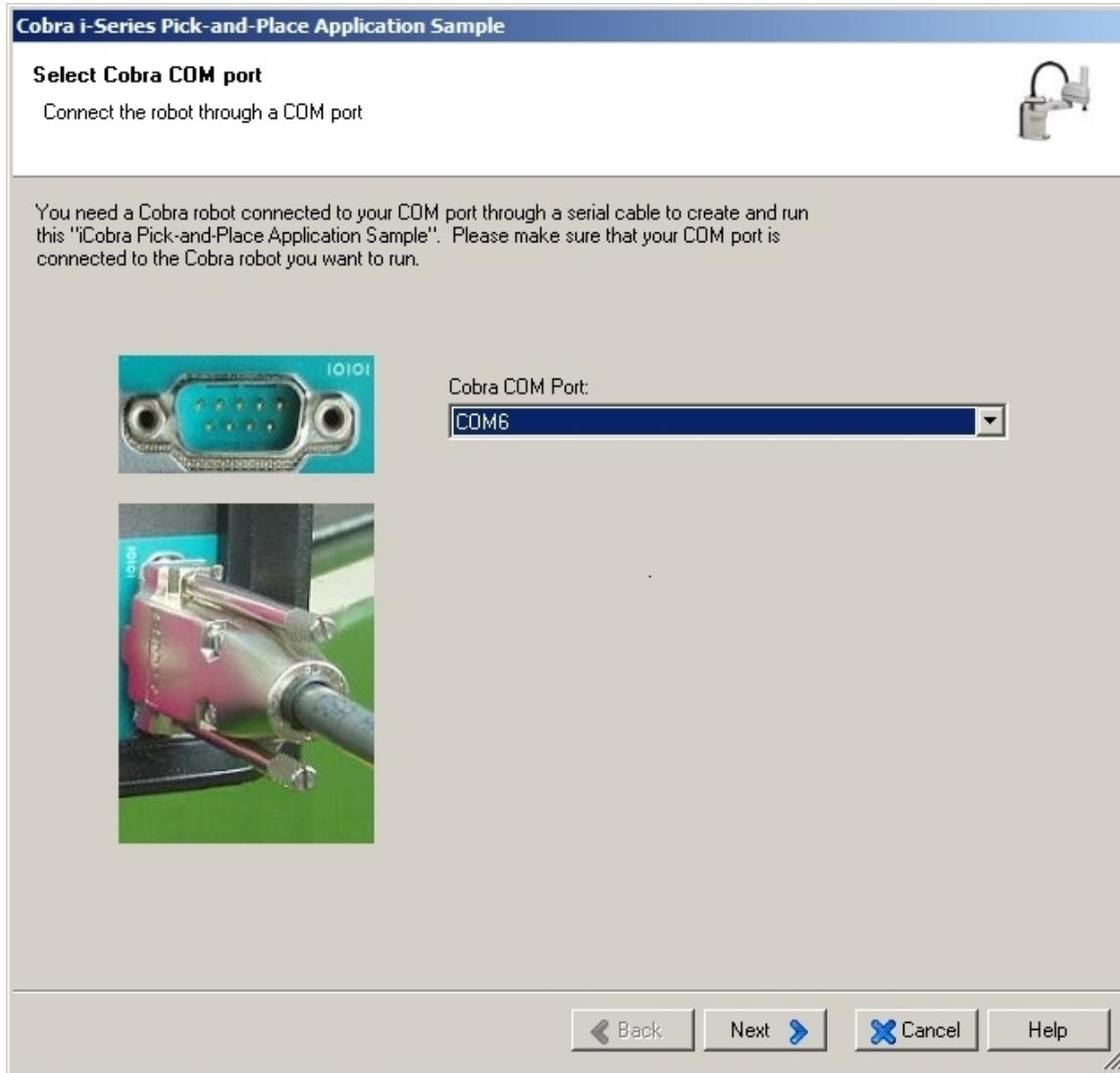


*Selecting an Existing Cobra i-Series Robot*

If you do not have a Cobra i-Series robot in your workspace, you will see the following screen. After selecting the COM port, the connected robot will be added to your workspace.

To program the application:

---



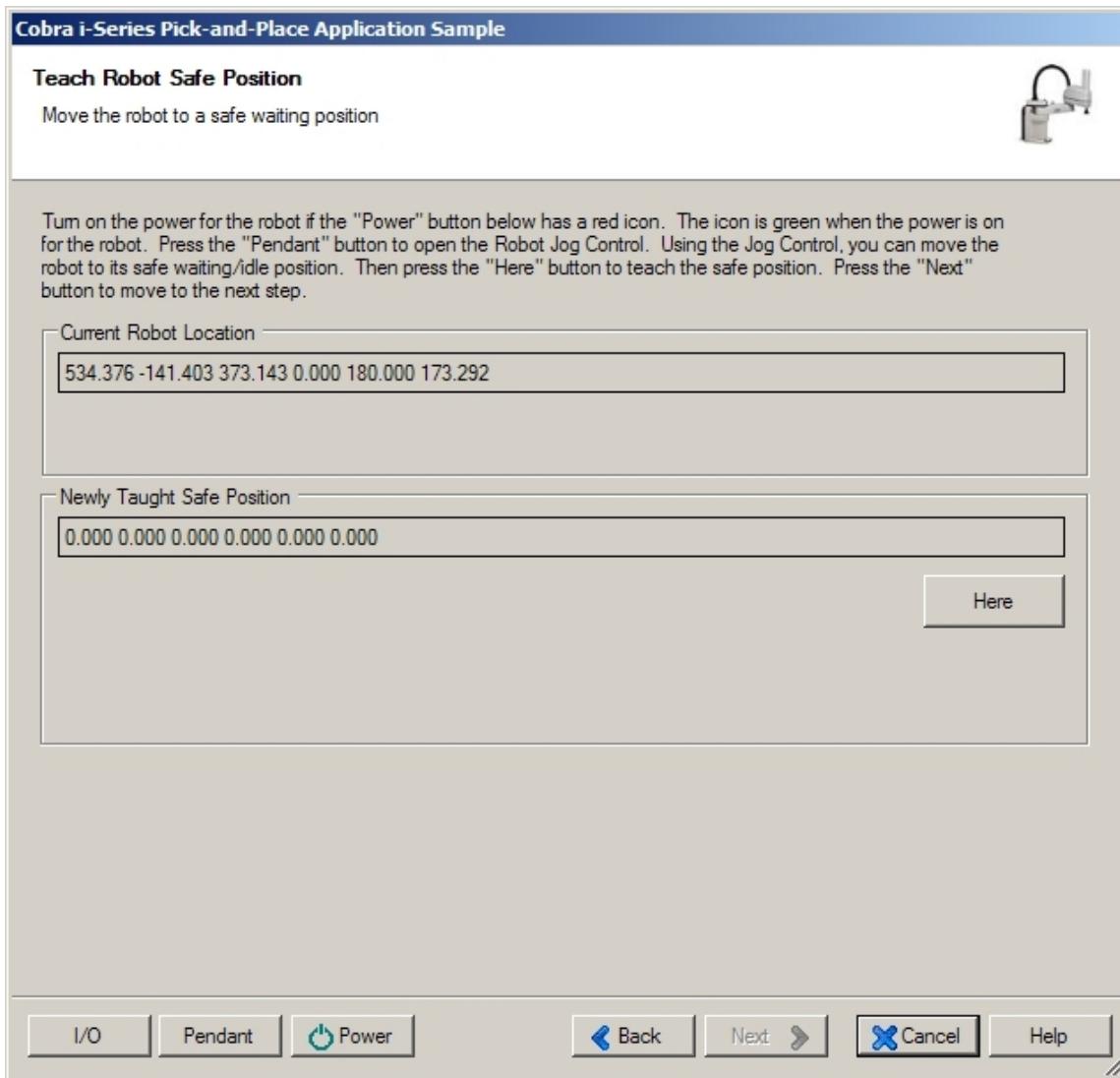
#### *Adding a Cobra i-Series Robot to the Workspace*

1. Select the robot.
  - a. Select the COM port that is connected to your robot.
  - b. Click **Next** to proceed. The interview wizard records the robot information and opens the next page.

To program the application:

---

2. The Teach robot safe position page opens.



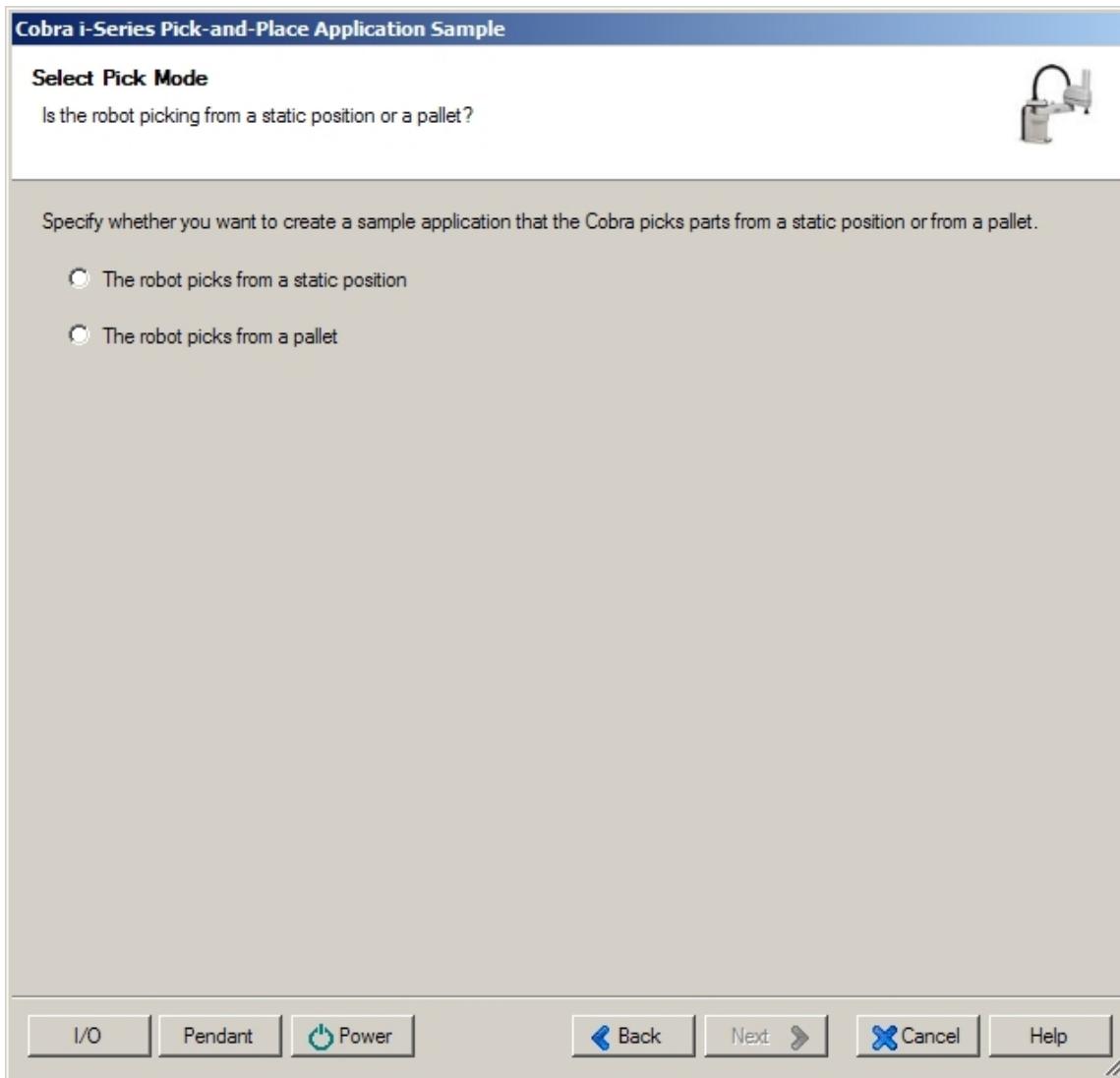
*Completing the Sample*

3. Read the instructions, and then use the Pendant and Here buttons to move the robot and record the location.
4. Click **Next** to proceed. The interview wizard opens the next page.
5. Teach the robot safe position. Click **Next** to proceed.

To program the application:

---

6. Use the radio buttons to select the pick process that represents your application.



*Selecting a Static Position or Pallet*

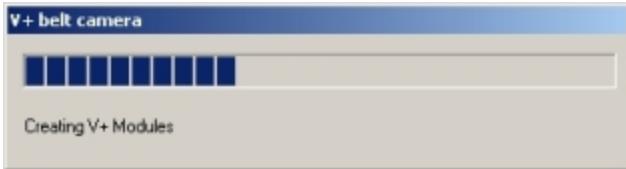
For more details on teaching the process, see [Teaching a Process on page 894](#).

7. Click **Next** to begin teaching the process. The Teach Robot Pick Position page opens.
8. Teach the static pick location (or pallet origin, +X, and +Y).
9. Use the radio buttons to select the place process that represents your application.
10. Teach the static place location (or pallet origin, +X, and +Y).

To program the application:

---

11. Activate the Cycle Start Digital Input signal number 2010, as instructed. Use the **Open Digital IO Window** button to verify that the signal is ON.
12. The interview wizard completes the interview process and generates the application code. A progress indicator displays the status.

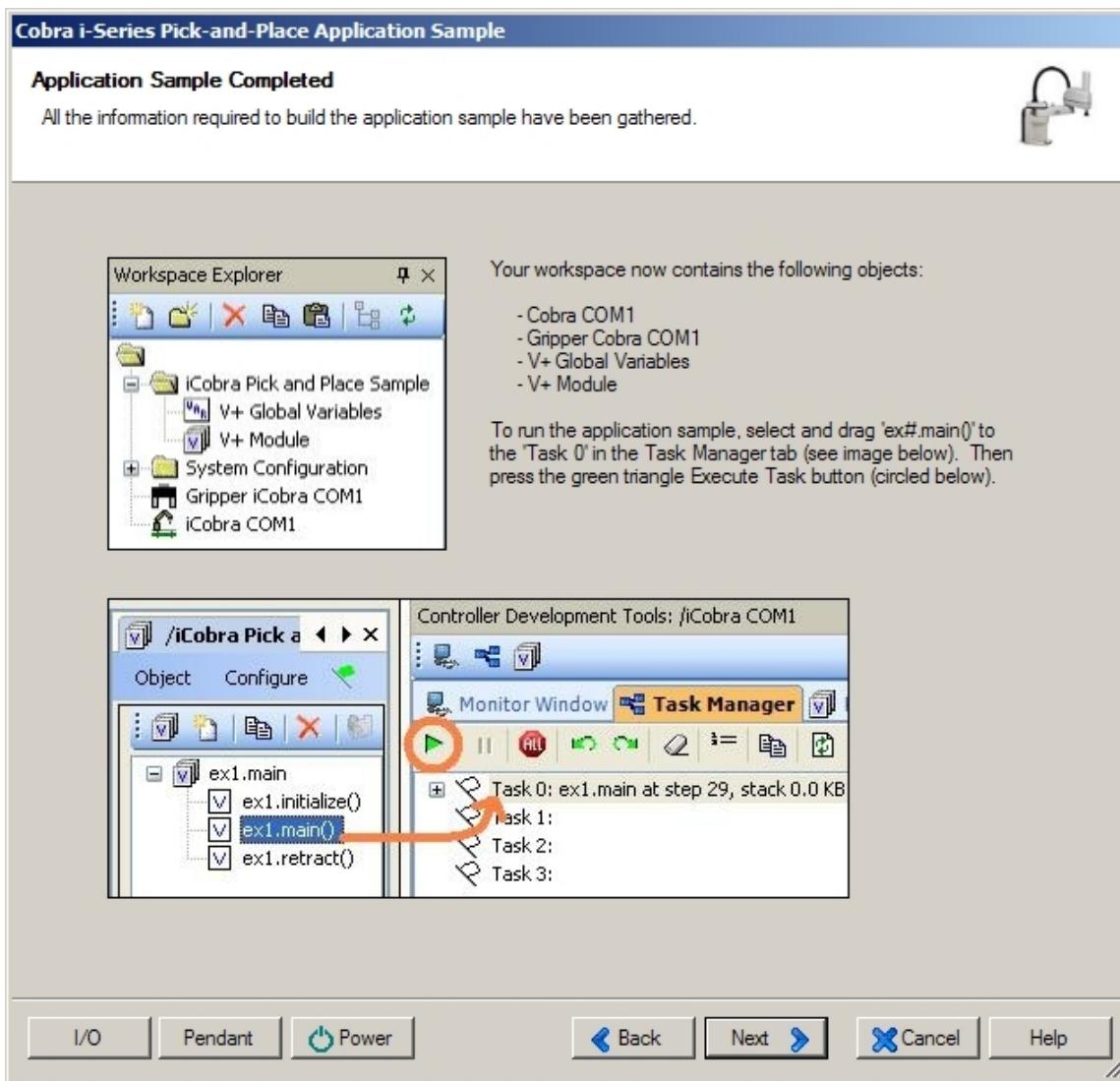


*Progress Indicator*

13. The Application Sample Completed page confirms that the process has completed.

To program the application:

---



*MicroV+ Application Example Completed Page*

14. Open the V+ Task Manager and then drag the "main( )" program onto Task 0 of the Task Manager panel, as shown in the previous figure. For details, see V+ Task Manager on page 772.
15. On the ACE toolbar, use the Monitor Speed control to set the monitor speed to 25%. For details, see Toolbar on page 157.
16. In the Task Manager panel of the Controller Development Tools window:

---

#### Auto starting the application:

---

- a. Select Task 0.
- b. Click the Start icon ( ) to start that program. The robot will move to the Safe location and then begin executing the pick and place cycle.
- c. To stop the program, click the Pause icon ( ). The robot motion stops.

#### **Auto starting the application:**

After you complete the application sample, you have the option of "auto starting" it. The autostart option allow the application to load and prepare for operation when the robot is powered-up.

To enable the autostart option:

1. In ACE, open the Monitor window and type the following command at the dot prompt:

ENABLE AUTOSTART

2. Save the ACE workspace.
3. Turn off high power to the robot.
4. Turn on high power to the robot.
5. After the robot boots up, the white button on the Front Panel blinks. Press the button to complete the power-up process.
6. Enable signal 2010 to activate the application. This can be done by starting the ACE software, and using the Digital I/O panel to activate soft signal 2010. For more details on the Digital I/O panel, see Digital I/O Window on page 166.

## Static Pick and Place Sample

After you select the application sample, you are ready for the setup and programming of the application. An interview wizard guides you through the process, as described below.

The following instructions are provided as an example of a Process Manager static pick and place application. The steps for your application sample may vary, depending on the application sample selected in the previous section. However, this example will provide an overview of the basic steps for programming an application sample.

### What happens next?

You will use the interview wizard to:

- Select the robot
- Select the gripper

The interview wizard will pause momentarily to create the objects and write the program code, then you will:

- Teach the process in the Process Manager. You will teach the Idle, Part (pick) and Part Target (place) positions.
- Run the process using the Process Manager Control.

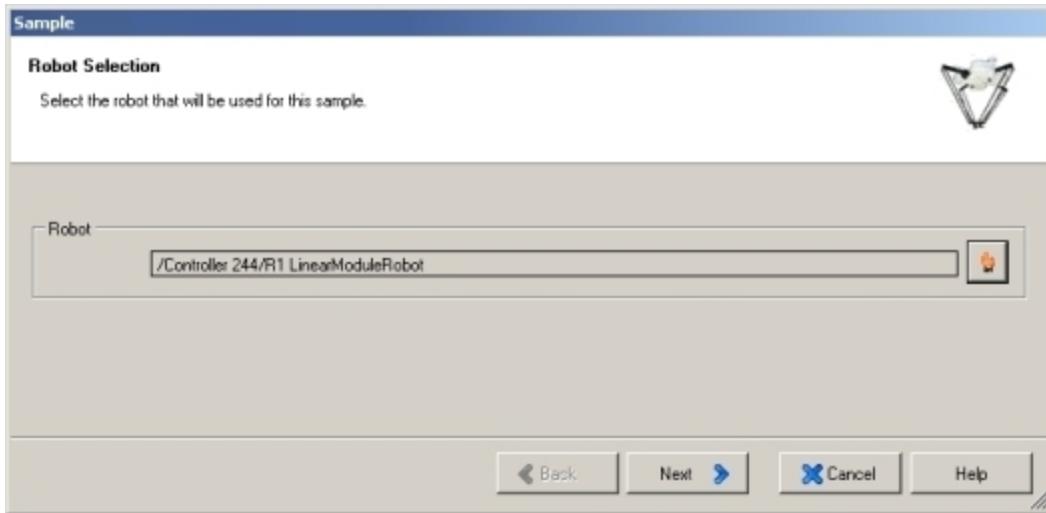
### To program the application:

After you complete the introduction pages of the interview wizard, continue with the steps described below.

To program the application:

---

1. Select the robot and gripper.

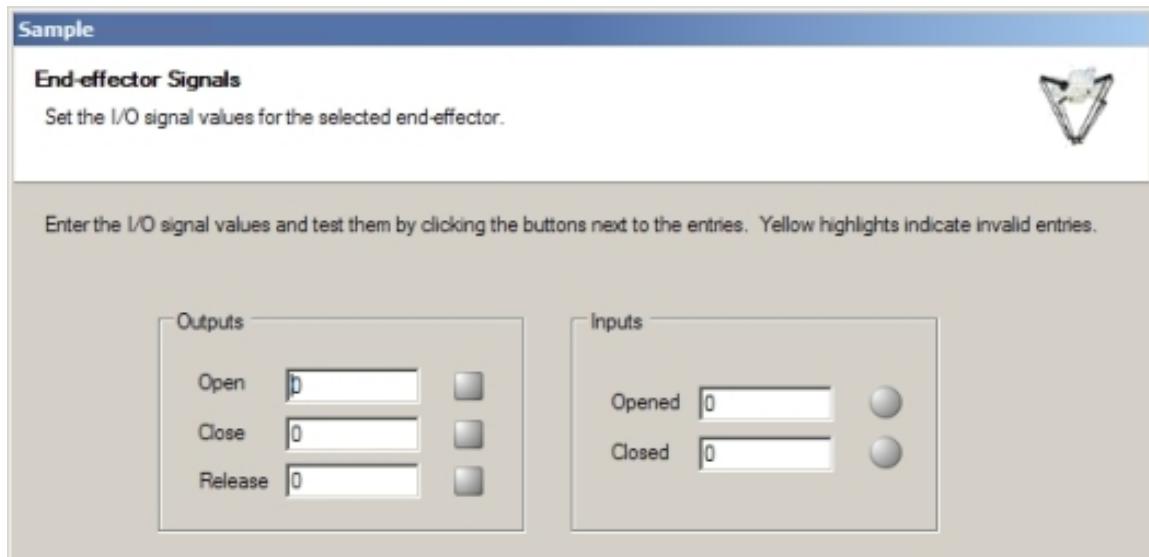


*Selecting the Robot*

- a. Click the Browse ( ) icon to view the list of available robots and then select a robot. Click **Next** to proceed. The interview wizard records the selection and opens the next page.
- b. Click the Browse ( ) icon to view the list of available grippers and then select an end effector (gripper). Click **Next** to proceed. The interview wizard records the selection and opens the next page.
- c. Specify the I/O signals for the selected end effector (gripper).

To program the application:

---

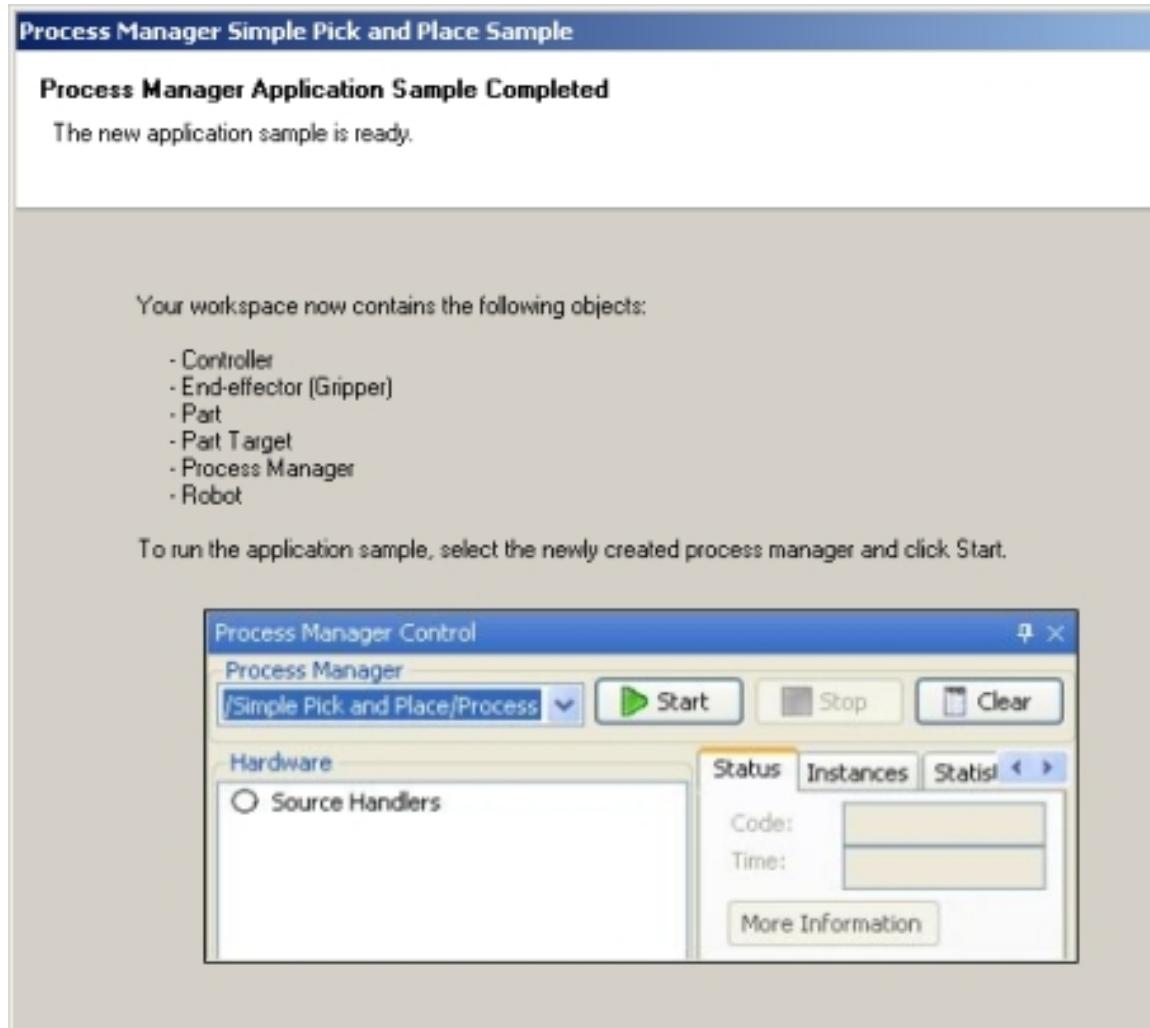


*Specifying the I/O Signals*

- d. Click **Next** to proceed. The interview wizard records the signal information and opens the next page.
2. Now the interview wizard takes you through the final steps for teaching the Process (teaching the idle position, part position and part target position). For details on teaching the Process, see Processes on page 886.
3. After the Process teach steps have been completed, the Sample Completed page opens. Read the instructions and click **Finish** to proceed.

To program the application:

---

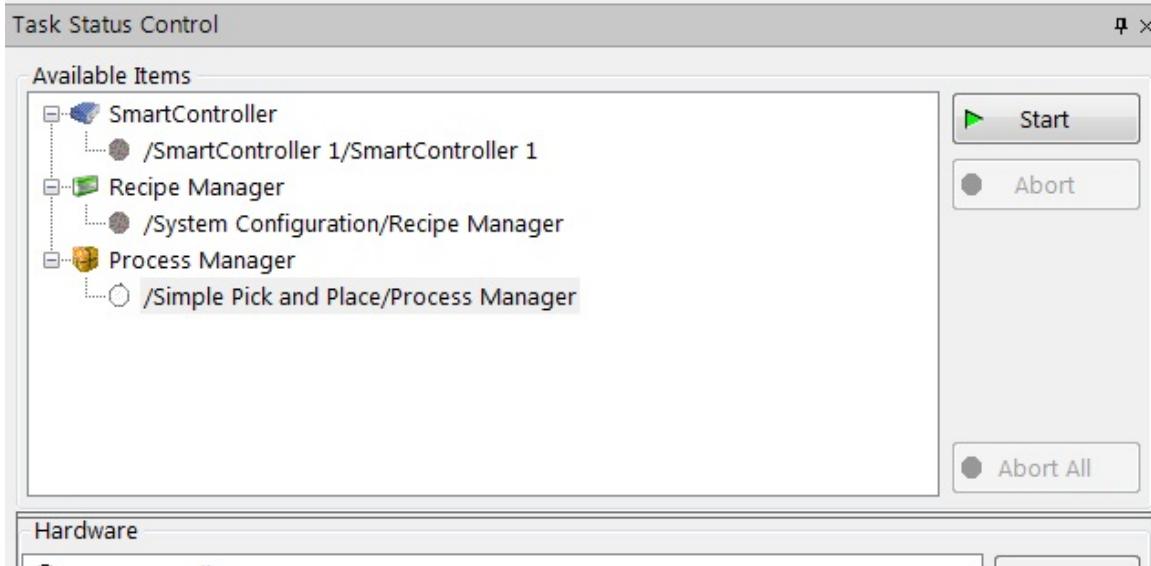


*Completing the Sample*

To program the application:

---

4. Run the application using the ACE Task Manager Control.



*Running the Application*

- a. Use the Process Manager drop-down list box to select the process.
- b. Click **Start** to start the selected process.
- c. Click **Stop** to stop the process.

For more details on the Process Manager Control, see [Process Manager Control](#)

## **Belt Camera Part Pick Sample**

After you select the application sample, you are ready for the setup and programming of the application. An interview wizard guides you through the process, as described below.

The following instructions are provided as an example of a Process Manager belt camera pick and place application (in other words, a vision-guided pick from a conveyor belt and place to a static location). The steps for your application sample may vary, depending on the application sample selected in the previous section. However, this example will provide an overview of the basic steps for programming an application sample.

### **What happens next?**

You will use the interview wizard to:

- Select the robot
- Select the gripper
- Select the belt (conveyor) encoder
- Select the virtual camera
- Teach the vision model

The interview wizard will pause momentarily to create the objects and write the program code, then you will:

- Calibrate the robot to the belt (Belt calibration).
- Calibrate the robot to the belt camera (Sensor calibration).
- Teach the process in the Process Manager. You will teach the Idle, Part (pick) and Part Target (place) positions.
- Run the process using the Process Manager Control.

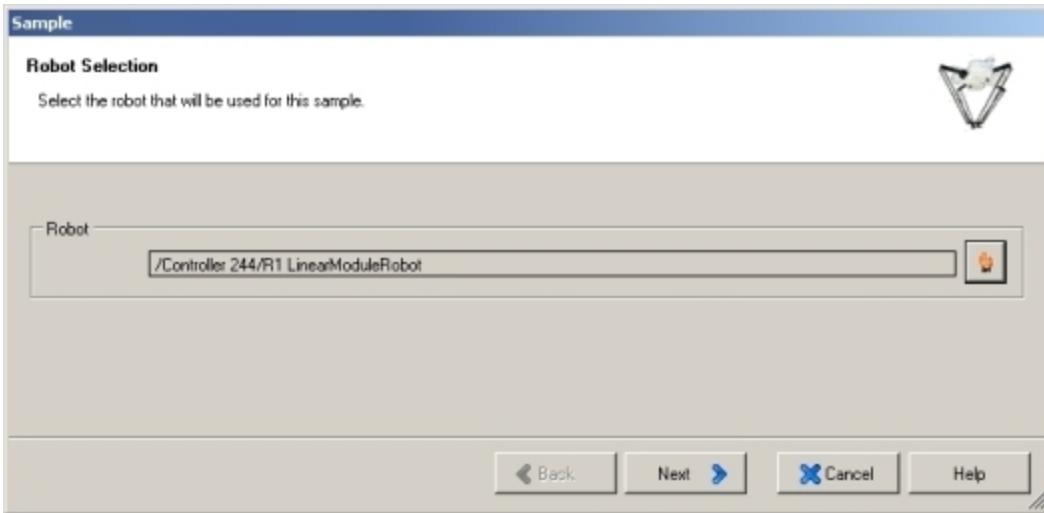
### **To program the application:**

After you complete the introduction pages of the interview wizard, continue with the steps described below.

To program the application:

---

1. Select the robot and gripper.

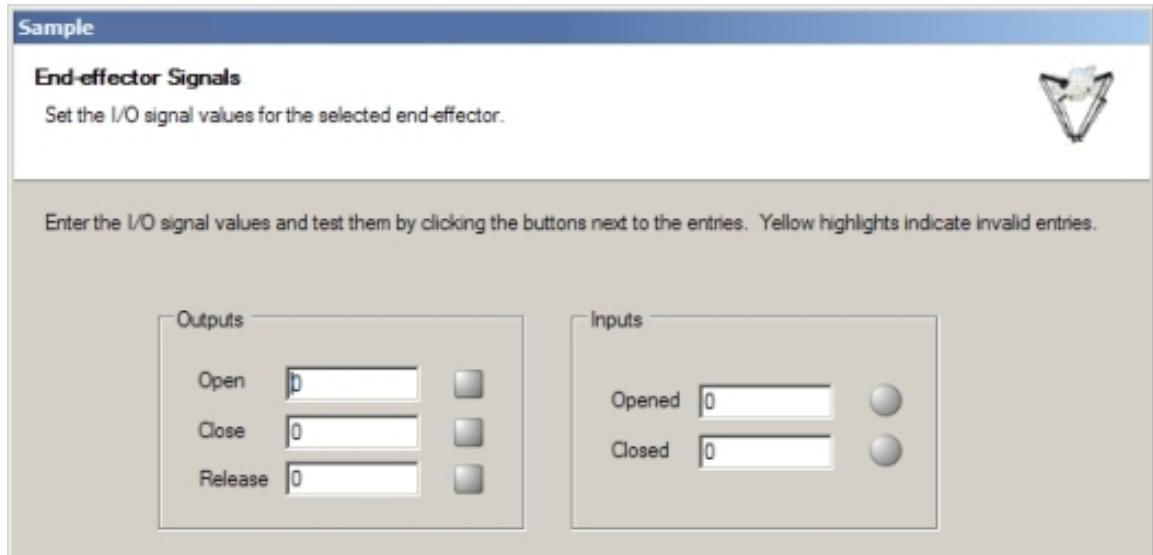


*Selecting the Robot*

- a. Click the Browse () icon to view the list of available robots and then select a robot. Click **Next** to proceed. The interview wizard records the selection and opens the next page.
- b. Click the Browse () icon to view the list of available grippers and then select an end effector (gripper). Click **Next** to proceed. The interview wizard records the selection and opens the next page.
- c. Specify the I/O signals for the selected end effector (gripper).

To program the application:

---

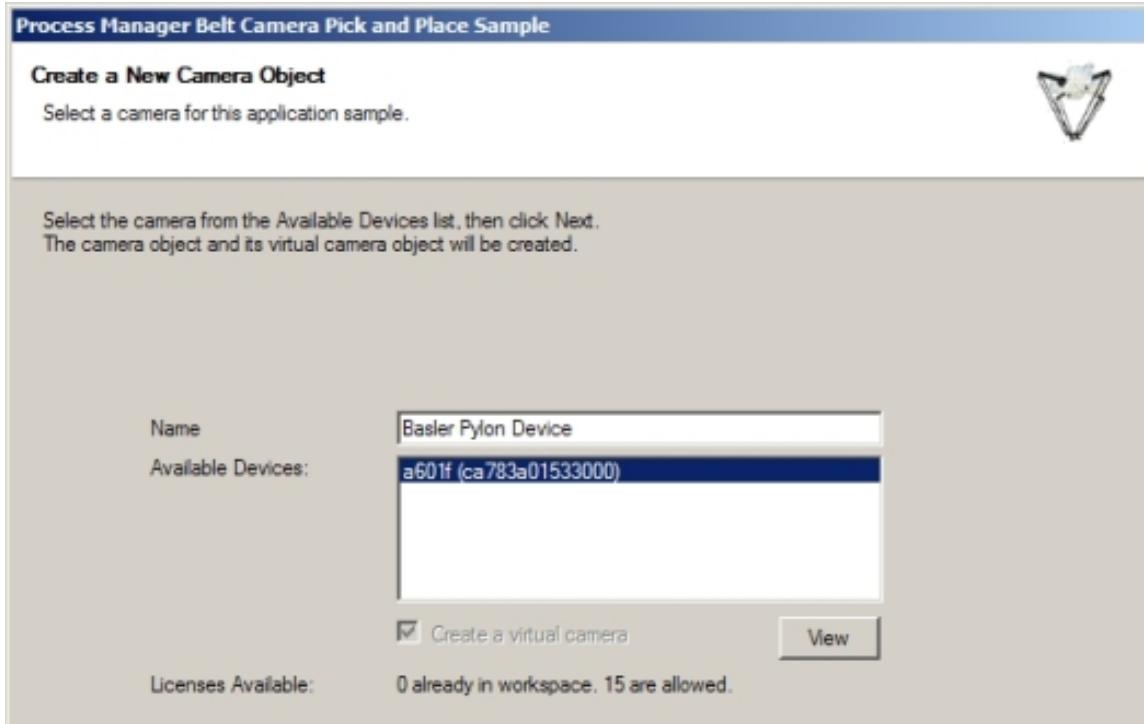


*Specifying the I/O Signals*

- d. Click **Next** to proceed. The interview wizard records the signal information and opens the next page.
2. Specify if you want to use an existing camera or select a create and select a new one. If there is no camera in the workspace, this step will be skipped.
3. If you decide to create a new camera, or there is no camera in the workspace, you will create and select a camera. After you make a selection, the selected camera is installed in the workspace.

To program the application:

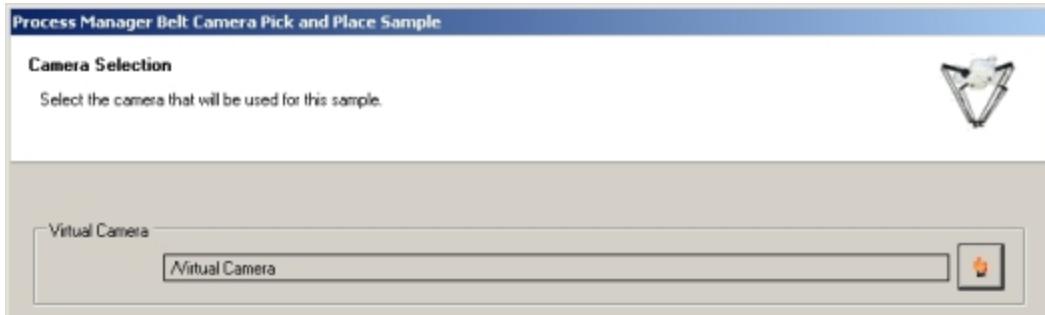
---



#### *Creating a New Camera Object*

Otherwise, you will proceed to the next step.

4. Select the camera.

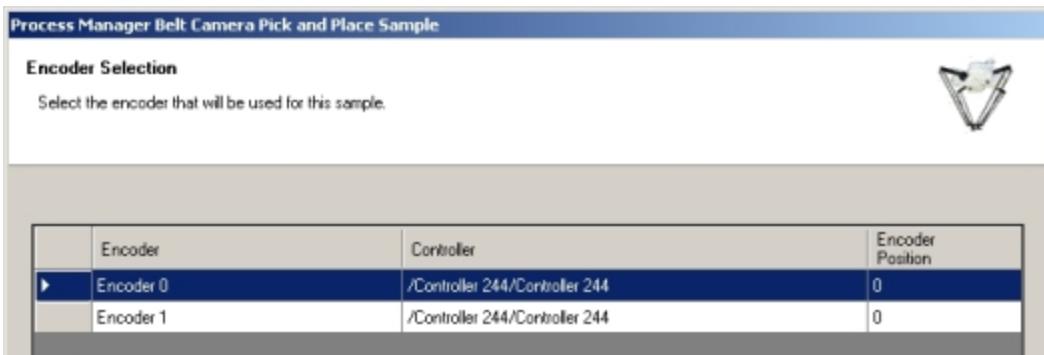


#### *Selecting the Camera*

- a. Click the Browse ( ) icon to view the list of available cameras and then select your camera.
- b. Click **Next** to proceed. The interview wizard records the camera information and opens the

next page.

- c. Adjust the camera properties. For details, see the *ACE Sight User's Guide*. Click **Next** to proceed. The interview wizard records the camera adjustments and opens the next page.
  - d. Perform a Grid Calibration of the camera. Click **Launch Grid Calibration** to begin the procedure. For details on performing a grid calibration, see the *ACE Sight User's Guide*. Click **Next** to proceed. The interview wizard records the camera calibration and opens the next page.
5. Select the belt (conveyor) encoder.



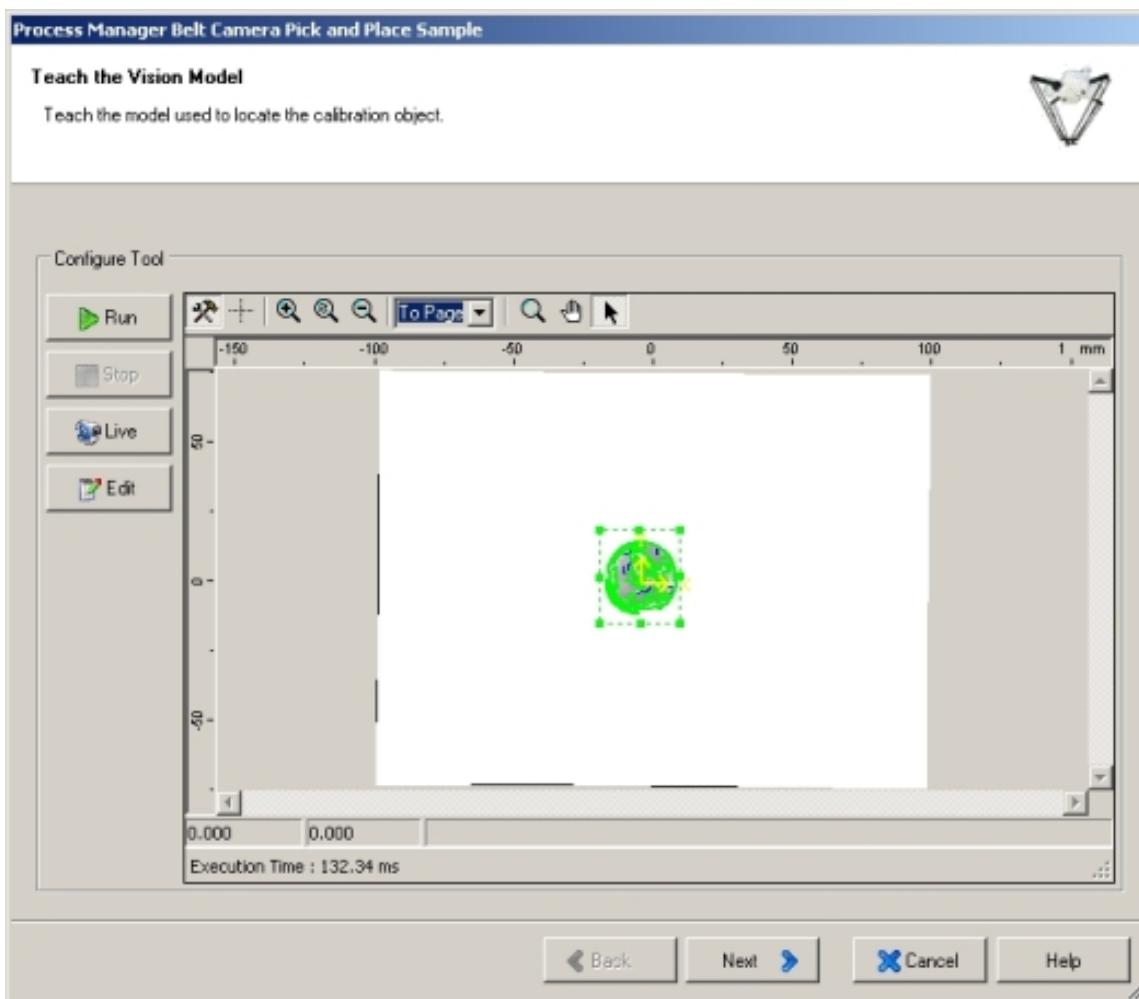
*Selecting the Encoder*

- a. Click the encoder you wish to use in your sample application. You can quickly locate the encoder by activating the conveyor and watching the Encoder Position column—the count for the corresponding encoder should change when the conveyor is running.
  - b. Click **Next** to proceed. The interview wizard records the encoder information and opens the next page.
6. Teach the picture-taking position. This is the location to robot will move to when a picture is taken. The robot must be out of the camera field-of-view (FOV), so the image is not obscured by the robot.
- a. Move the robot to a location that is out of the camera field-of-view (FOV).
  - b. Click **Here** to record the position.

To program the application:

---

7. Teach the vision model.



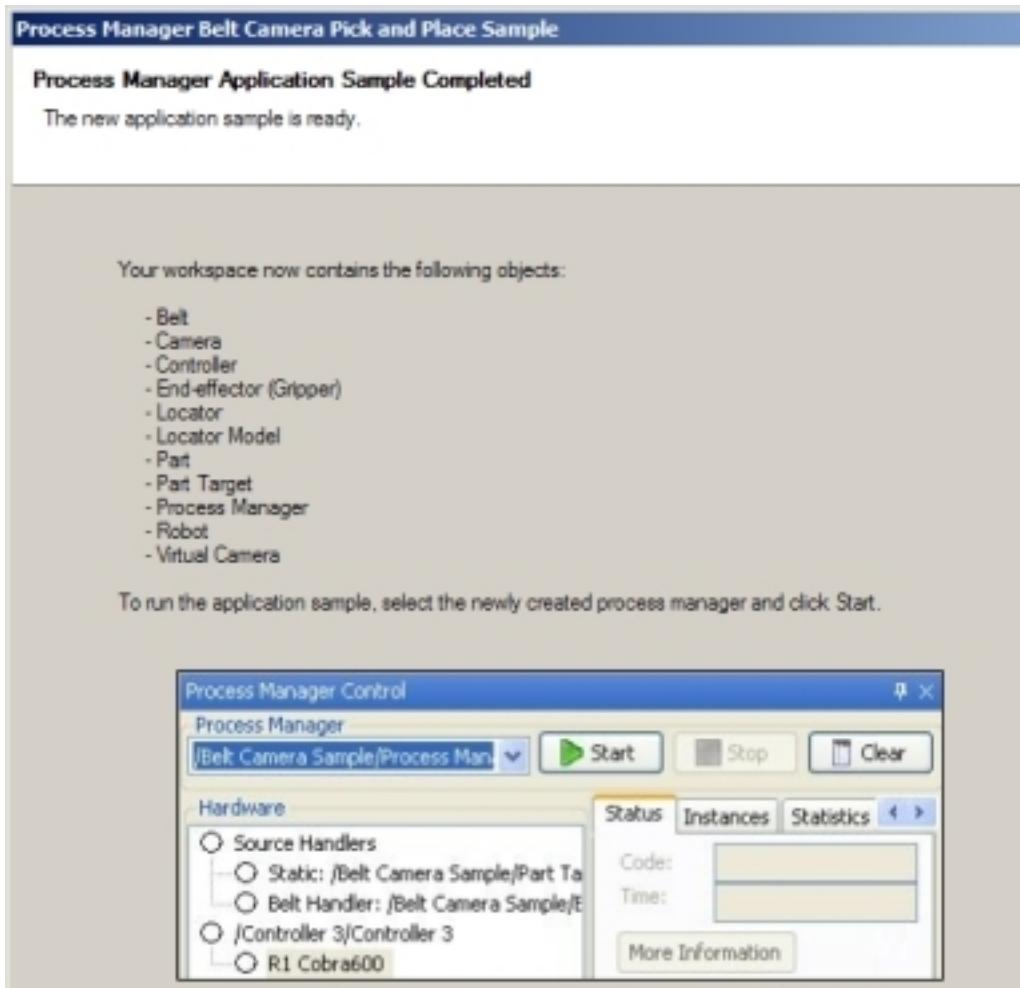
*Teaching the Vision Model*

- a. Use the handles to set the bounding box around the outside of the object.
  - b. Position the orientation indicator on the object.
  - c. Click **Run** to check the operation of the model. The sample part will be matched to the model by the vision system.
  - d. Click **Next** to proceed. The interview wizard records the model information and opens the next page.
8. At this point, the interview wizard steps you through two calibration procedures:

To program the application:

---

- The first procedure creates the robot-to-belt relationship. For details on this calibration procedure, see Belt Calibrations on page 848.
  - The second procedure creates the robot to belt-camera relationship. For details on this calibration procedure, see Belt Camera Calibration on page 914.
9. Now the interview wizard takes you through the final steps for teaching the Process (teaching the idle position, part position and part target position). For details on teaching the Process, see Processes on page 886.
10. After the Process teach steps have been completed, the Sample Completed page opens. Read the instructions and click **Finish** to proceed.



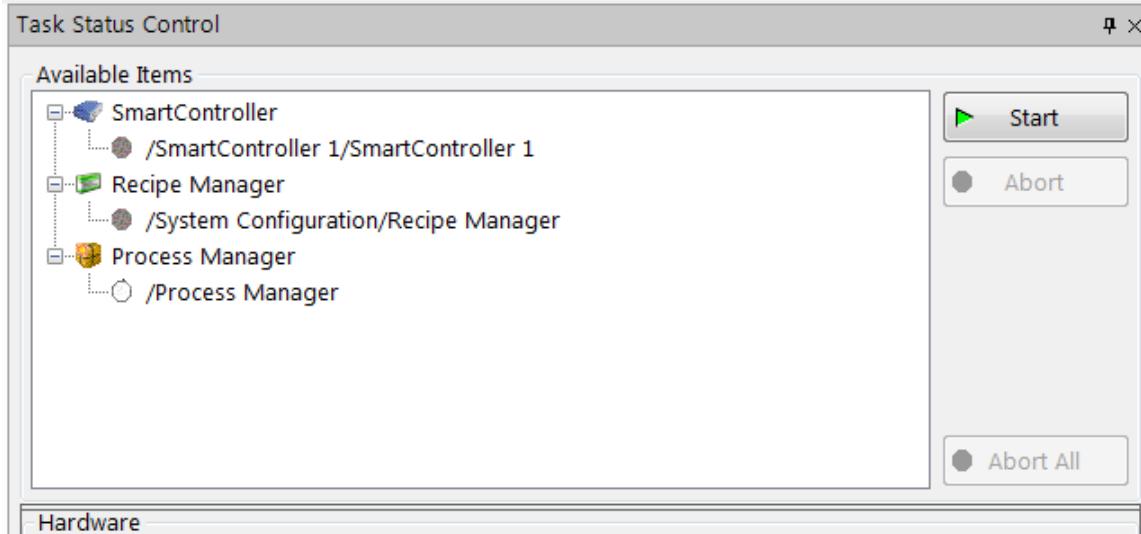
*Completing the Sample*

---

To program the application:

---

11. Run the application using the ACE Task Manager Control.



*Running the Application*

- a. Use the Process Manager drop-down list box to select the process.
- b. Click **Start** to start the selected process.
- c. Click **Stop** to stop the process.

For more details on the Process Manager Control, see [Process Manager Control](#)

## Fixed Camera Part Pick Sample

After you select the application sample, you are ready for the setup and programming of the application. An interview wizard guides you through the process, as described below.

The following instructions are provided as an example of a Process Manager fixed camera pick and place application (in other words, a vision-guided pick from a location under a fixed-mount camera and place to a static location). The steps for your application sample may vary, depending on the application sample selected in the previous section. However, this example will provide an overview of the basic steps for programming an application sample.

### What happens next?

You will use the interview wizard to:

- Select the robot
- Select the gripper
- Select the virtual camera
- Teach the vision model

The interview wizard will pause momentarily to create the objects and write the program code, then you will:

- Calibrate the robot to the fixed camera (Sensor calibration).
- Teach the process in the Process Manager. You will teach the Idle, Part (pick) and Part Target (place) positions.
- Run the process using the Process Manager Control.

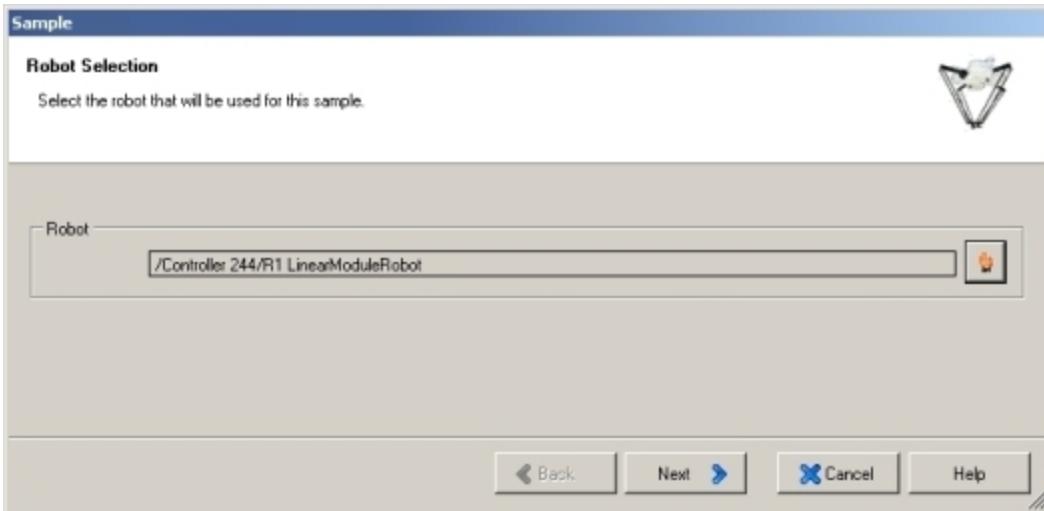
### To program the application:

After you complete the introduction pages of the interview wizard, continue with the steps described below.

To program the application:

---

1. Select the robot and gripper.

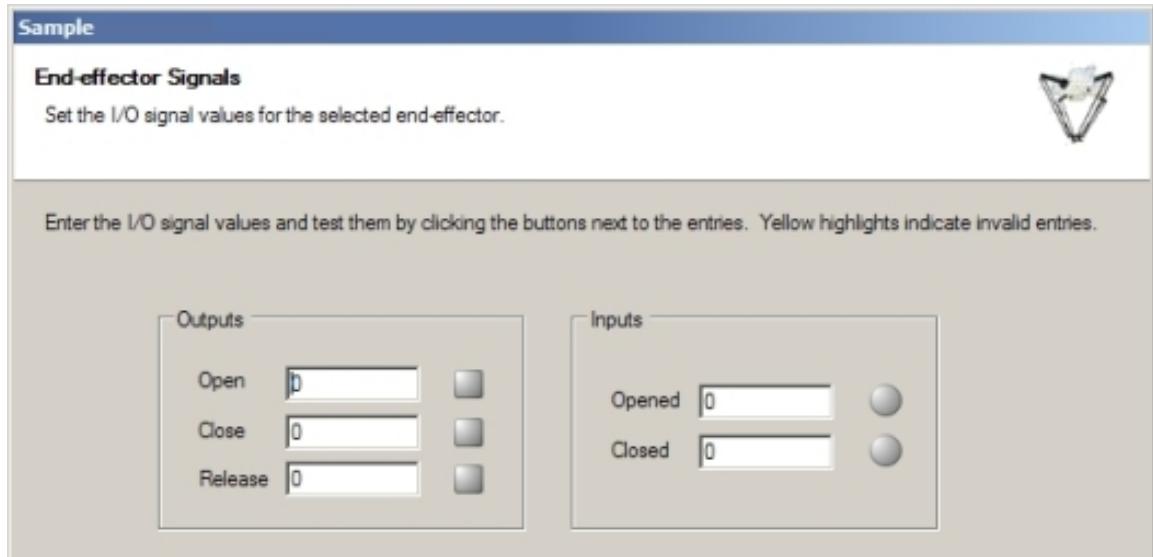


*Selecting the Robot*

- a. Click the Browse ( ) icon to view the list of available robots and then select a robot. Click **Next** to proceed. The interview wizard records the selection and opens the next page.
- b. Click the Browse ( ) icon to view the list of available grippers and then select an end effector (gripper). Click **Next** to proceed. The interview wizard records the selection and opens the next page.
- c. Specify the I/O signals for the selected end effector (gripper).

To program the application:

---

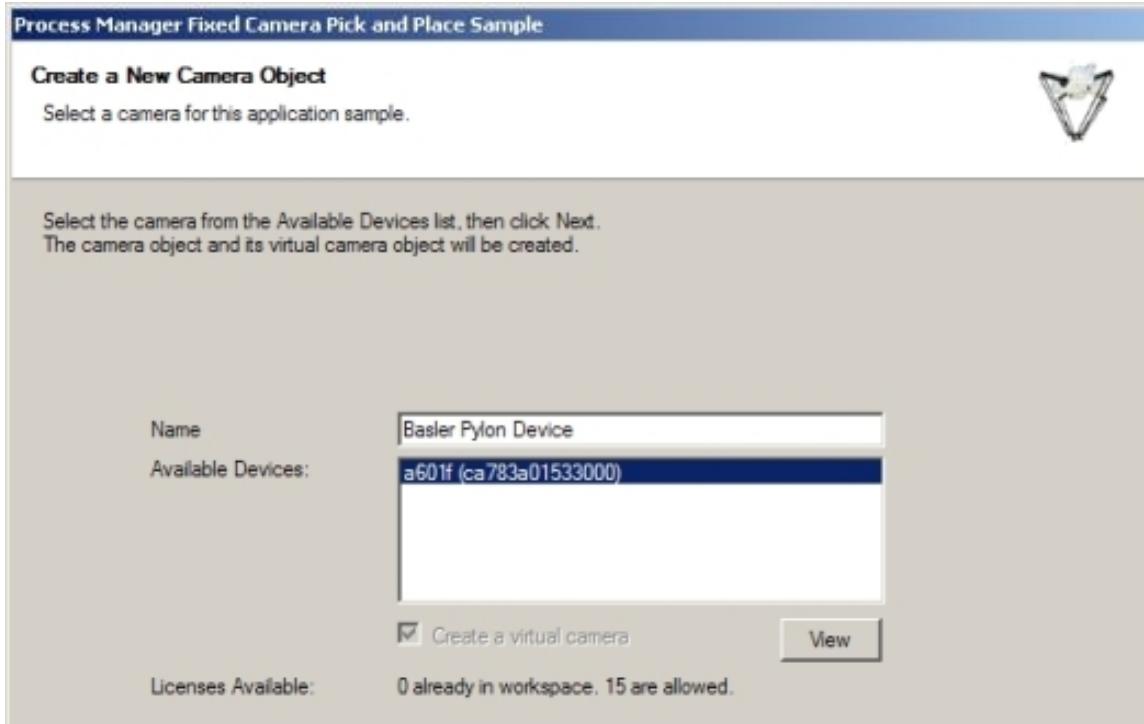


*Specifying the I/O Signals*

- d. Click **Next** to proceed. The interview wizard records the signal information and opens the next page.
2. Specify if you want to use an existing camera or select a create and select a new one. If there is no camera in the workspace, this step will be skipped.
3. If you decide to create a new camera, or there is no camera in the workspace, you will create and select a camera. After you make a selection, the selected camera is installed in the workspace.

To program the application:

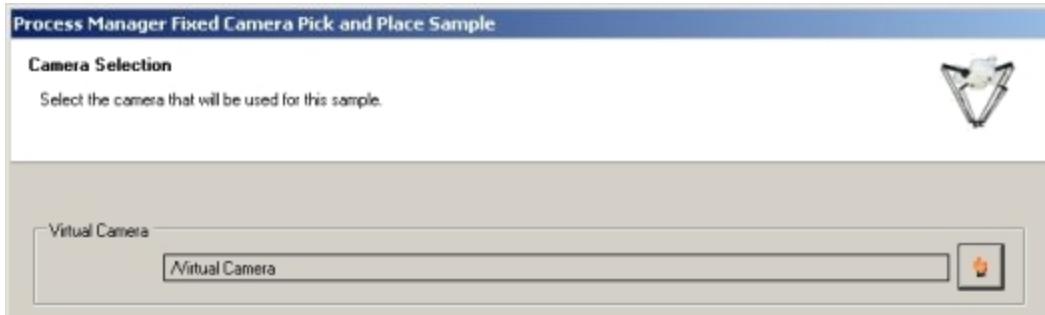
---



#### *Creating a New Camera Object*

Otherwise, you will proceed to the next step.

4. Select the camera.



#### *Selecting the Camera*

- a. Click the Browse (📁) icon to view the list of available cameras and then select your camera.
- b. Click **Next** to proceed. The interview wizard records the camera information and opens the

To program the application:

---

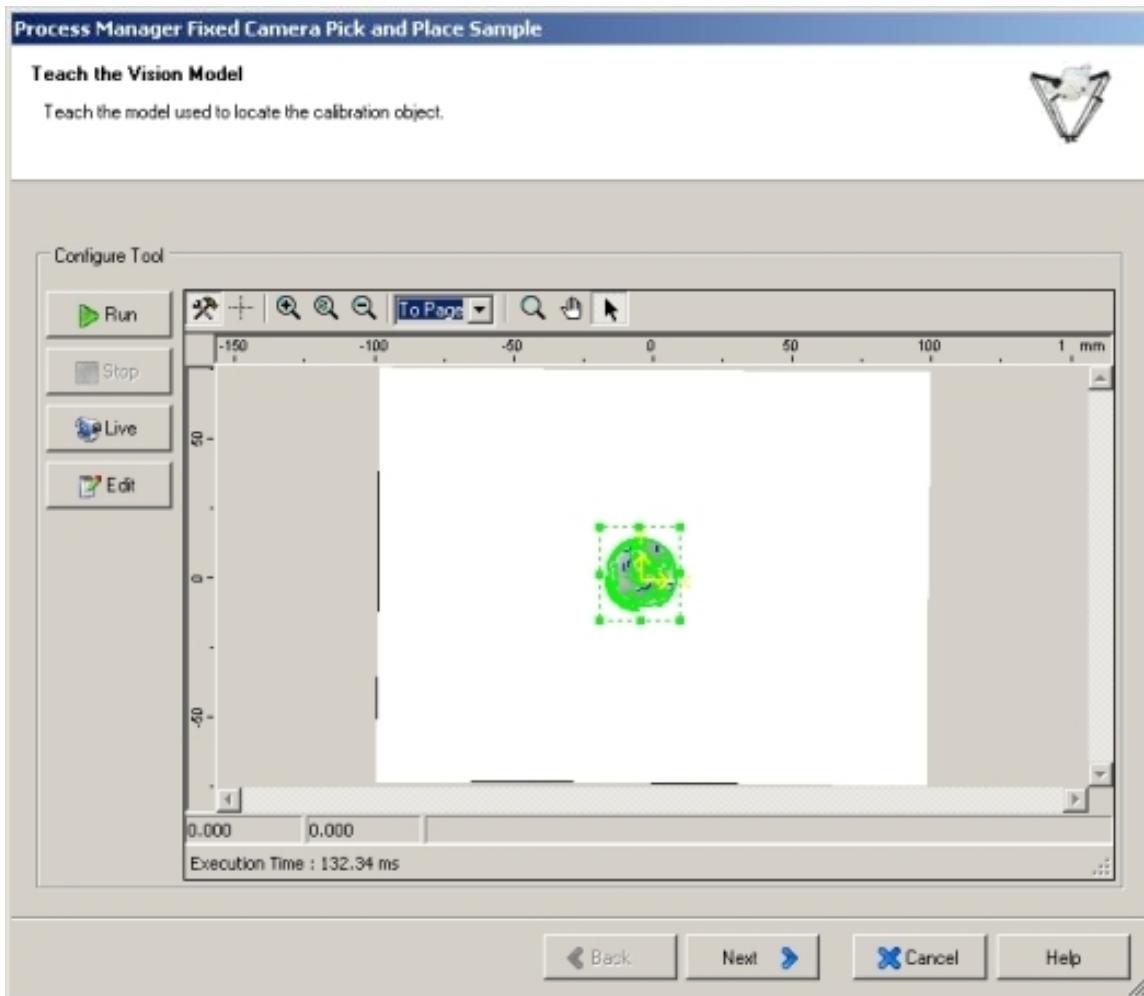
next page.

- c. Adjust the camera properties. For details, see the *ACE Sight User's Guide*. Click **Next** to proceed. The interview wizard records the camera adjustments and opens the next page.
  - d. Perform a Grid Calibration of the camera. Click **Launch Grid Calibration** to begin the procedure. For details on performing a grid calibration, see the *ACE Sight User's Guide*. Click **Next** to proceed. The interview wizard records the camera calibration and opens the next page.
5. Teach the picture-taking position. This is the location to robot will move to when a picture is taken. The robot must be out of the camera field-of-view (FOV), so the image is not obscured by the robot.
    - a. Move the robot to a location that is out of the camera field-of-view (FOV).
    - b. Click **Here** to record the position.

To program the application:

---

6. Teach the vision model.



*Teaching the Vision Model*

- a. Use the handles to set the bounding box around the outside of the object.
  - b. Position the orientation indicator on the object.
  - c. Click **Run** to check the operation of the model.
  - d. Click **Next** to proceed. The interview wizard records the model information and opens the next page.
7. Teach the Region of Interest (ROI) for locating the part.

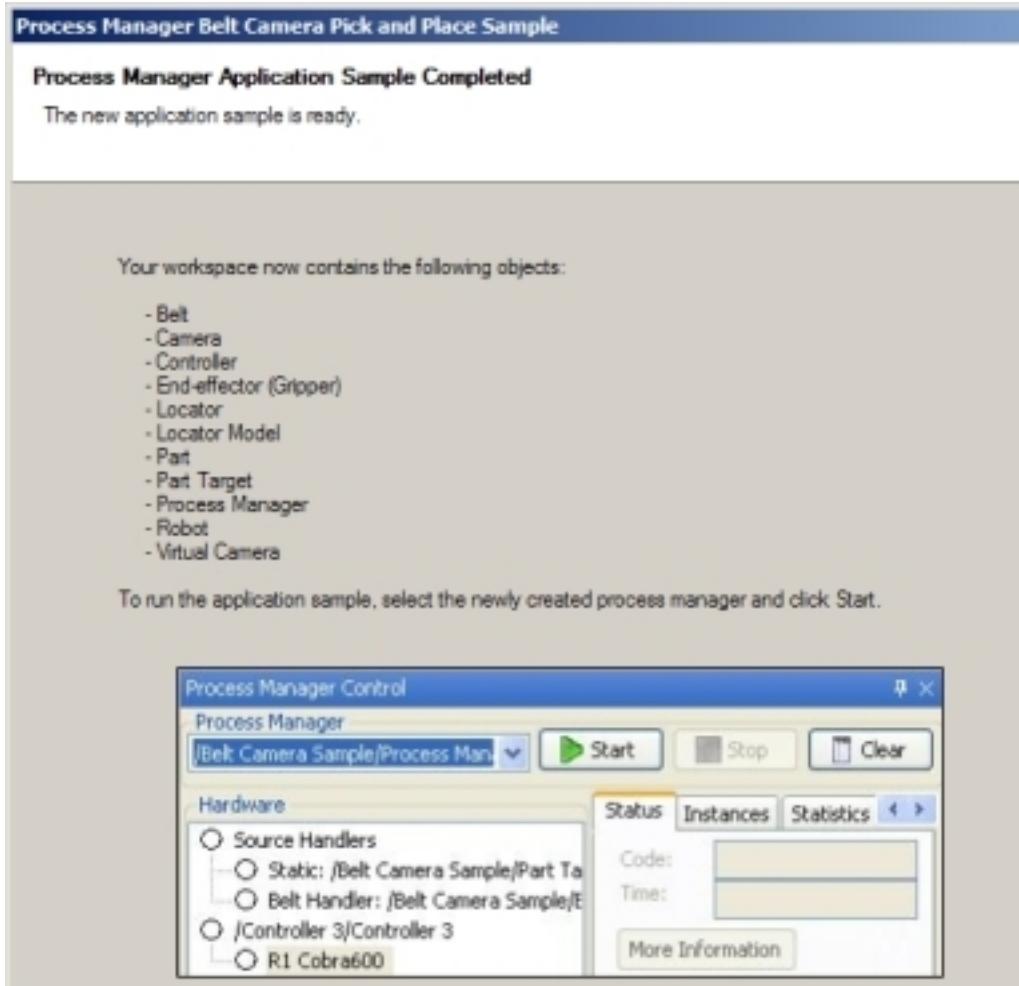
To program the application:

---

- a. Use the handles to set the bounding box for the area where the part will be located.
  - b. Click **Run** to check the operation of the model.
  - c. Click **Next** to proceed. The interview wizard records the ROI information and opens the next page.
8. Teach the idle, part (pick) and part target (place) positions, as instructed by the interview wizard.
    - a. Click the Pendant button and use the controls to move the robot to the desired position.
    - b. Click **Next** to proceed. The interview wizard records the position information and opens the next page.
  9. At this point, the interview wizard steps you through the automated sensor calibration procedures, which calibrate the robot to the fixed-camera, along with the final steps for teaching the process.
  10. After the Process teach steps have been completed, the Sample Completed page opens. Read the instructions and click **Next** to proceed.

To program the application:

---

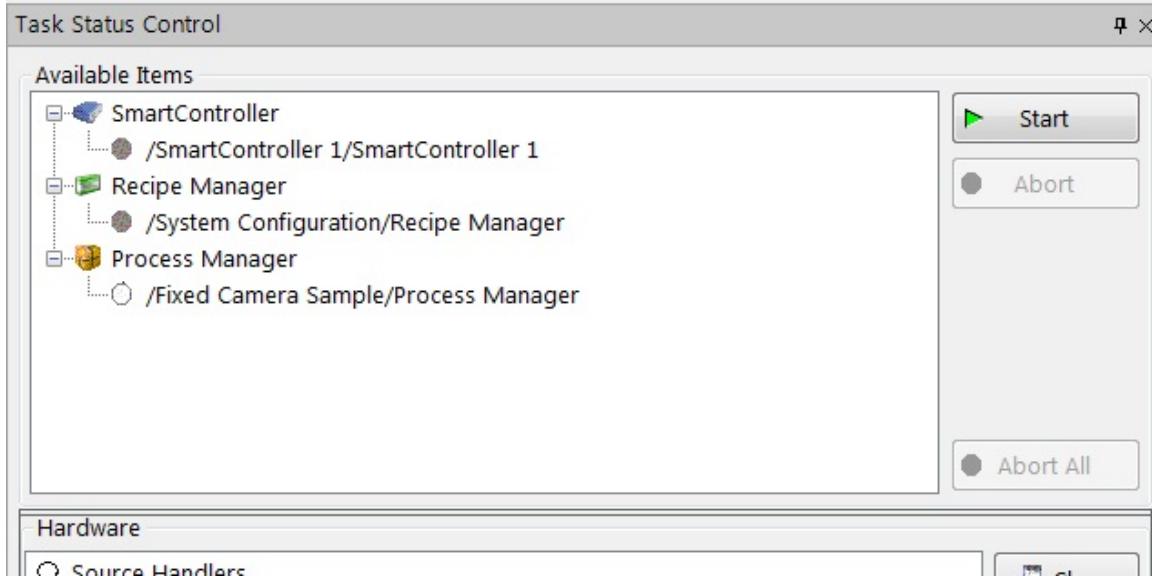


*Completing the Sample*

To program the application:

---

11. Run the application using the Process Manager Control.



*Running the Application*

- a. Use the Process Manager drop-down list box to select the process.
- b. Click **Start** to start the selected process.
- c. Click **Stop** to stop the process.

For more details on the Process Manager Control, see [Process Manager Control](#)

## Recipe Manager

The topics in this chapter describe the Recipe Manager features in the ACE software.

Overview .....	984
Using the Recipe Manager .....	985
Editing the Recipe .....	991

## Overview

The goal of the Recipe Manager is to provide a place for saving and restoring production data and other specific information that describes the product-manufacturing process. The ACE object that stores this information is called a "Recipe Manager". The Recipe Manager manages two aspects of recipe management:

- Recipe Definition
- Recipe Management

There are two primary steps to building a recipe:

1. Identifying objects in the workspace that need to be included in the recipe definition.
2. Create, Edit, and Delete recipes.

The remainder of this chapter details the configuration and use of the Recipe Manager.

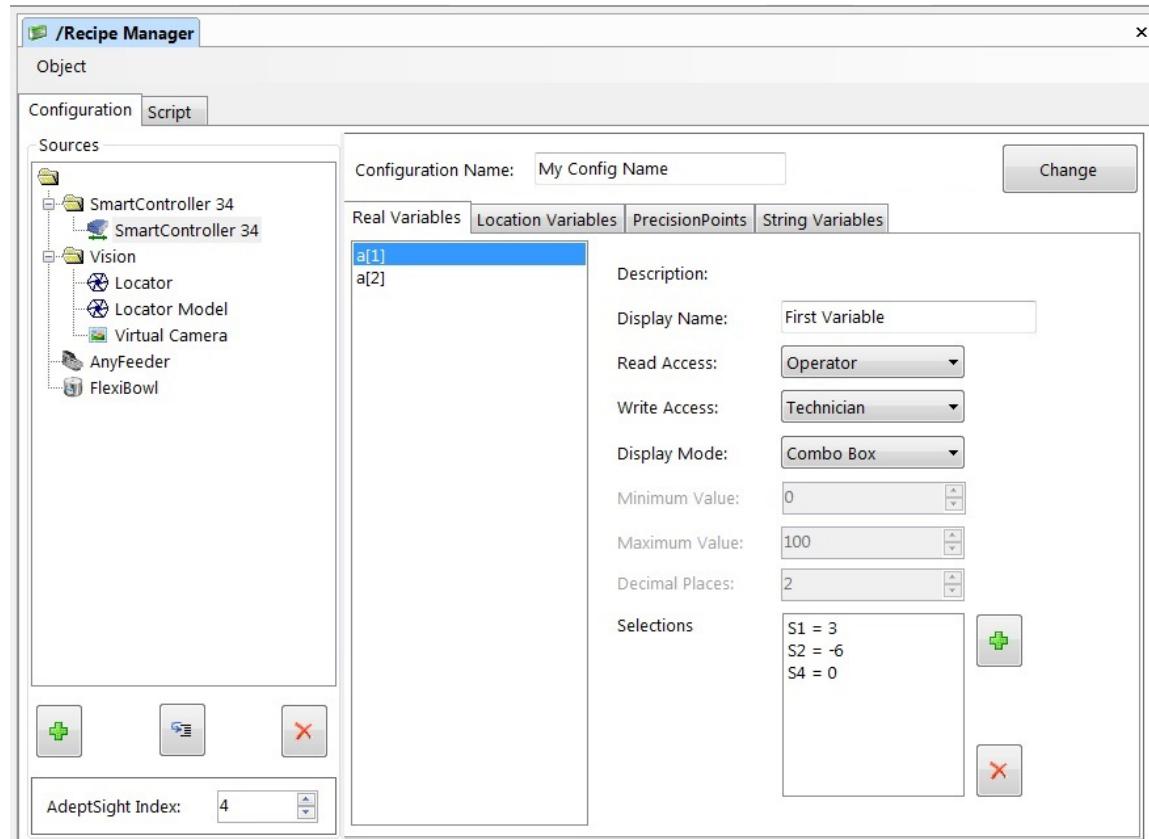
## Using the Recipe Manager

The Recipe Manager provides a tool for creating manufacturing recipes. For a brief overview of the Recipe Manager, see Overview on page 984.

To open the Recipe Manager and create a new recipe, right click in the folder area of the Workspace Explorer and select:

### New > Configuration > Recipe Manager

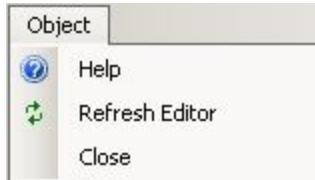
The Recipe Manager opens, which provides the work area and tools for defining the recipe configuration.



*Recipe Manager*

## Understanding Editor Menus

This section describes the selections available on the Recipe Manager menu bar:

**Object Menu****Help**

Displays the online help for the editor.

**Refresh Editor**

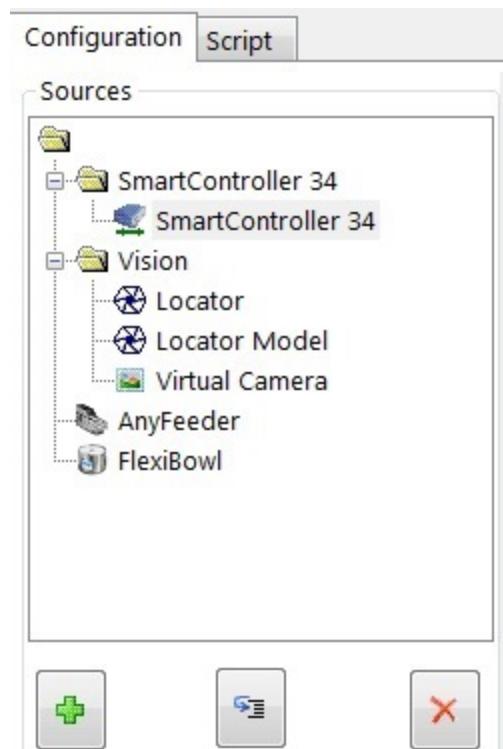
Refreshes the contents of the editor window.

**Close**

Closes the editor.

**Configuration Sources**

The Configuration section is used to identify which objects in the workspace need to be included in the recipe configuration. You can click on the Add and Remove buttons to change which workspace objects are included.



*Recipe Configuration Group*

The **Add Button** and **Remove Button** can be used to change what objects are associated with the recipe configuration. The **Sort Order Button** can be used to change the order the recipe sources are displayed to the user in the Recipe display.

The following objects are capable of being added to the recipe configuration:

Item	Description
Adept Controller	User can identify V+ global variables which are included in a recipe. This includes identifying how the variable is displayed to the user and what kind of access an operator would have with each item. The user can set the value of the variables through the recipe editor.
Blob Analyzer Gripper Clearance Image Histogram Locator	Recipe contains a copy of each vision tool and all vision tool properties are included in each recipe. The user can edit the advanced properties of each tool through the recipe editor.
Locator Model	Recipe contains a copy of the locator model which is included in each recipe. The user can train the model through the recipe editor.
Virtual Camera	Recipe contains a copy of the virtual camera which is included in each recipe. The user can modify the acquisition settings through the recipe editor.
Any Feeder Flexibowl Feeder	Recipe contains a copy of the feeder which is included in each recipe. The user can modify the typical feeding parameters through the recipe editor.

## ACE Sight Index

The ACE Sight index defines the index used when accessing the recipe manager using the V+ ACE Sight protocol. More details are included in the ACE Reference Guide.



*ACE Sight Index Group*

## Configuration Editor

When a configuration source is selected in the source list, an editor will be displayed to identify the recipe configuration for that source. The editor will be different depending on the type of object that is selected.

### Adept Controller

The AdeptController configuration allows the user to identify a set of V+ variables which will be included in the recipe definition. For each recipe the user creates, each V+ variable will have a separate value stored within the recipe.

Configuration Name: My Config Name Sort Order: 0 Change

Real Variables Location Variables PrecisionPoints String Variables

a[1]  
a[2]

Description: First Variable

Display Name: First Variable

Read Access: Operator

Write Access: Technician

Display Mode: Combo Box

Minimum Value: 0

Maximum Value: 100

Decimal Places: 2

Selections

S1 = 3
S2 = -6
S4 = 0

+    -

*AdeptController Configuration*

**Change Button** Displays a list of V+ variables the user can select the included variables.

**Configuration Name** The name displayed to the user in the recipe editor.

**Real / Location / Precision Point / String Variables** Displays all the variables included in the recipe configuration organized by type. Each type of variable contains different properties that affect how the variable is presented to the user in the recipe editor. For example, you can define a display name and access levels.

### Vision Tools

There are several different kinds of vision tools that can be added to a recipe configuration : Blob Analyzer, Gripper Clearance, Image Histogram, Locator, Locator Model, and Virtual Camera. For each recipe the user creates, a copy of the vision tool will be saved with each recipe.

## Script

---

When a recipe is selected, the selected recipe is linked with vision tools in workspace they correspond to. When a vision tool included in the recipe configuration is modified in the workspace, the selected recipe copy of the vision tool is updated. Likewise, when the vision tool is modified in the recipe editor, the workspace vision tool is also updated.

Because of this linking between the recipe and workspace, the user can use the vision tool in the workspace to configure a vision tool and it will be saved with the active recipe.

Depending on the type of vision tool, the recipe editor will be different. Typically, the recipe editor is only a small subset of vision tool properties:

Configuration Name:

Teach Access:

*Typical Vision Tool Configuration*

### **AnyFeeder / Flexibowl Feeder**

The feeder configurations store the feeder parameters with each recipe. When a recipe is selected, the selected recipe is linked with the feeder in the workspace they correspond to. When a feeder included in the recipe configuration is modified in the workspace, the selected recipe copy of the feeder is updated. Likewise, when the feeder is modified in the recipe editor, the workspace feeder is also updated.

Configuration Name:

Teach Access:

*Feeder Configuration*

## Script

When certain events occur in the lifetime of a recipe, the recipe manager will invoke a C# method defined in this section. This allows the person configuring the recipe manager to change the default behavior.

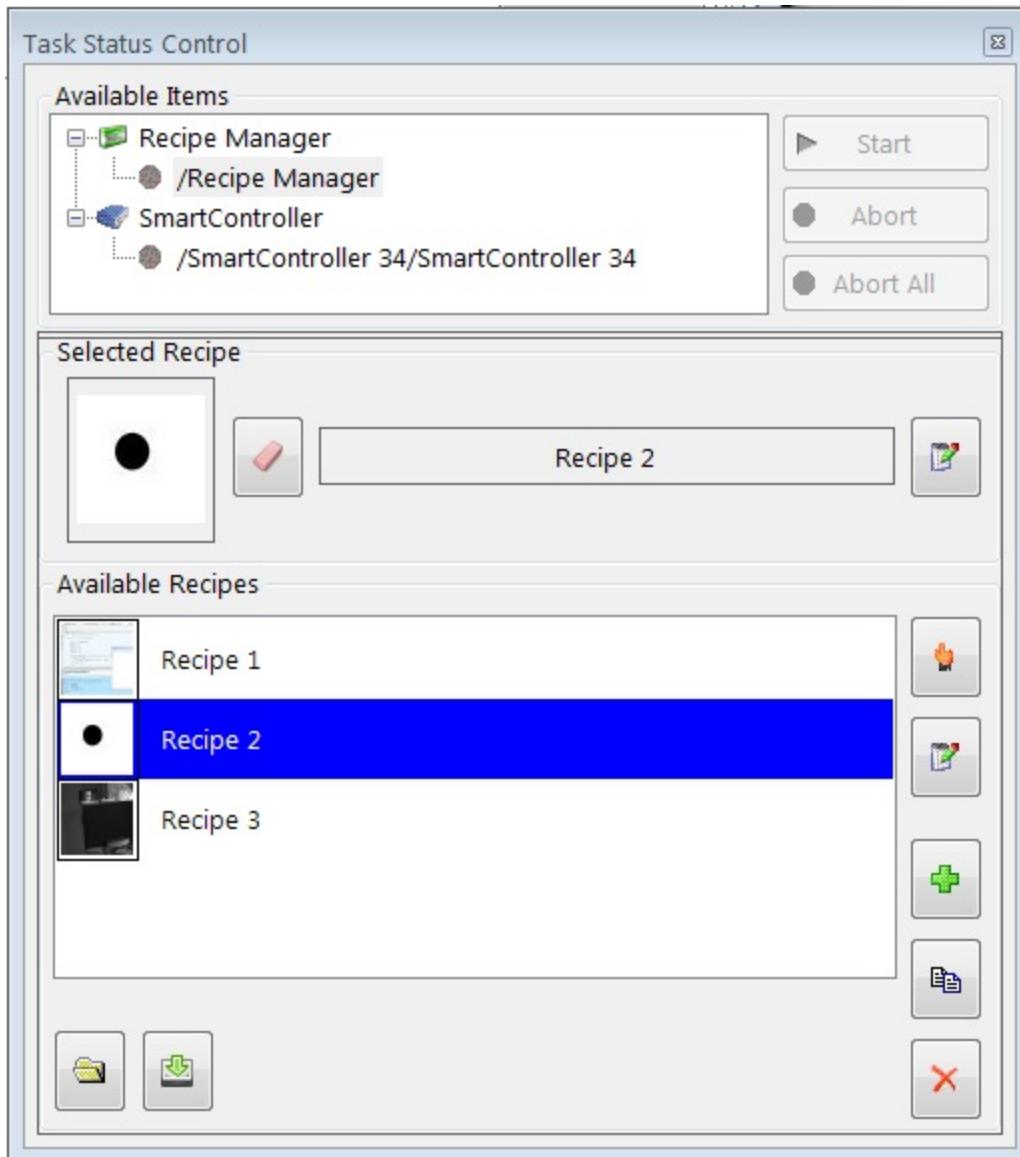
Script Method	Description
string CanEdit(Recipe recipe)	Called to check if a recipe can be edited. If this method returns an empty string, the recipe can be edited. If it returns a non-empty string, editing will be prevented an

Script Method	Description
	the string will be displayed to the user.
void BeforeEdit(Recipe recipe)	If a recipe can be edited, this method is called before the editor is displayed.
void AfterEdit(Recipe recipe)	This method is called after the recipe editor is closed by the user.
string CanSelect(Recipe recipe)	Called to check if a recipe can be selected. If this method returns an empty string, the recipe can be selected. If it returns a non-empty string, selection will be prevented and the string will be displayed to the user.
void BeforeSelection(Recipe recipe)	If a recipe can be selected, this method is called before the editor is selected.
void AfterSelection(Recipe recipe)	This method is called after the recipe is selected.

## Editing the Recipe

### ACE Task Status Control: Recipe Editing

The recipe manager displays the available recipes in the ACE Task Status Control:



*Recipe Selection in ACE Task Status Control*

#### **Selected Recipe**

The selected recipe section identifies which recipe is the active recipe.

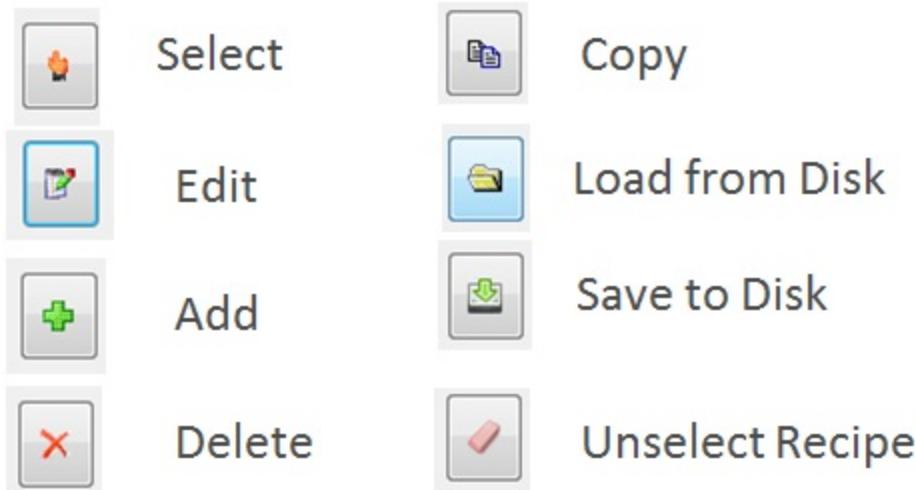
---

### **Available Recipes**

A listing of all recipes that are available for selection.

### **Buttons**

The various buttons on the user interface are:



*Recipe Editor Buttons*

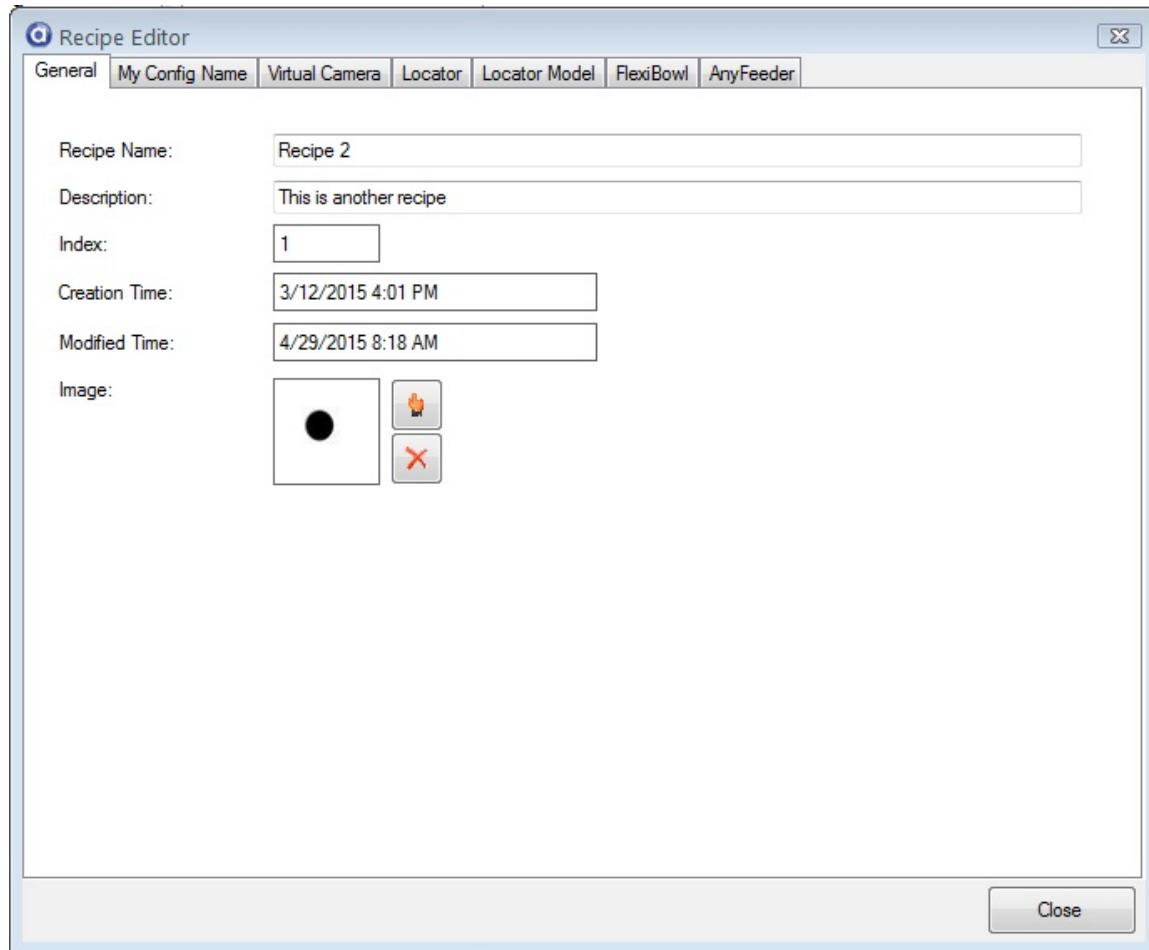
Button	Description
Select	Makes the highlighted recipe in the recipe list the active recipe
Edit	Edit the recipe
Add	Add a new recipe
Delete	Delete the highlighted recipe in the recipe list
Copy	Copy the highlighted recipe in the recipe list
Load from Disk	Load a recipe from a file into the recipe list
Save to Disk	Save the highlighted recipe in the recipe list to a file
Unselect Recipe	Deselect the currently selected recipe

## Recipe Editor

---

### Recipe Editor

When a recipe is edited, a recipe editor is displayed to the user:



### Recipe Editor

#### General Tab

All recipes contain a tab defining the general properties of the recipe.

Property	Description
Recipe Name	The name of the recipe
Description	User definable description associated with the recipe

---

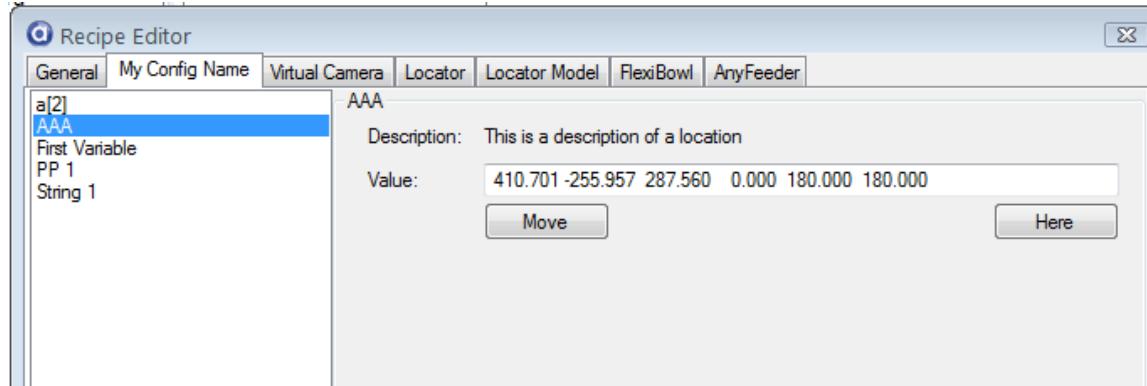
Property	Description
Index	The index of the recipe used when accessing the recipe through ACE Sight or via the C# API.
Creation Time	The time the recipe was created
Modified Time	The last time the recipe was modified
Image	User definable picture associated with the recipe.

### Recipe Source Tabs

All other tabs in the recipe editor correspond with sources added to the recipe configuration. The exact order of the tabs and the content will depend on the recipe selection.

### Adept Controller Source

Each selected variable is displayed in a list. The display is changed as each variable is selected based on the settings in the recipe configuration.



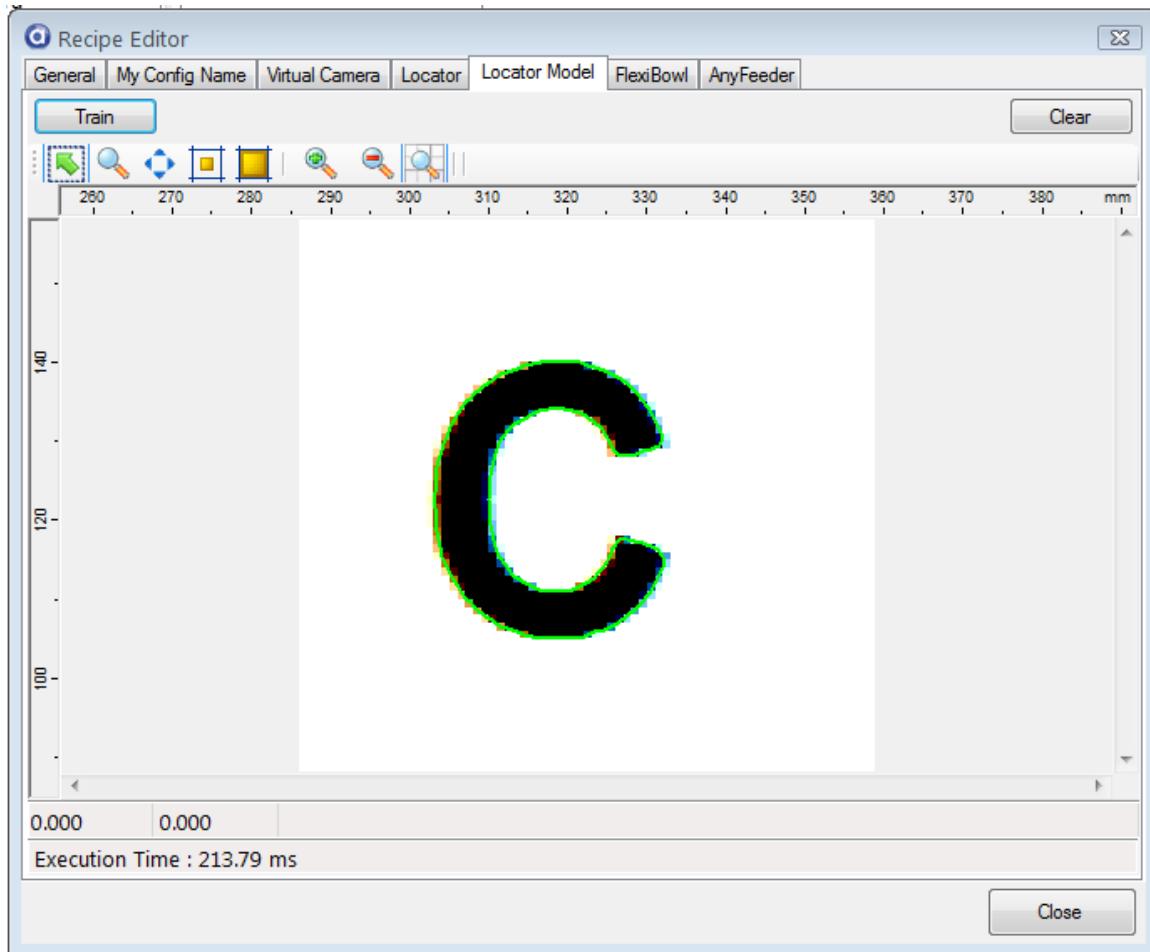
*Adept Controller Recipe Editor*

### Locator Model Source

The user is displayed the currently trained locator model and can edit or retrain the locator model.

## Recipe Editor

---



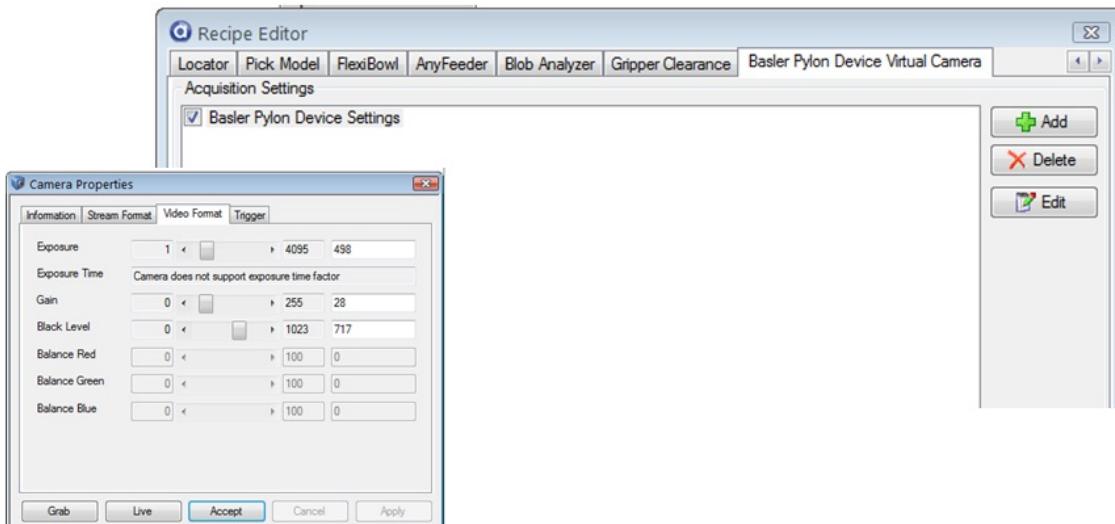
*Locator Model Editor*

### Virtual Camera Source

The acquisition properties are displayed in a list. The user can modify, add, or remove acquisition settings as needed.

## Recipe Editor

---

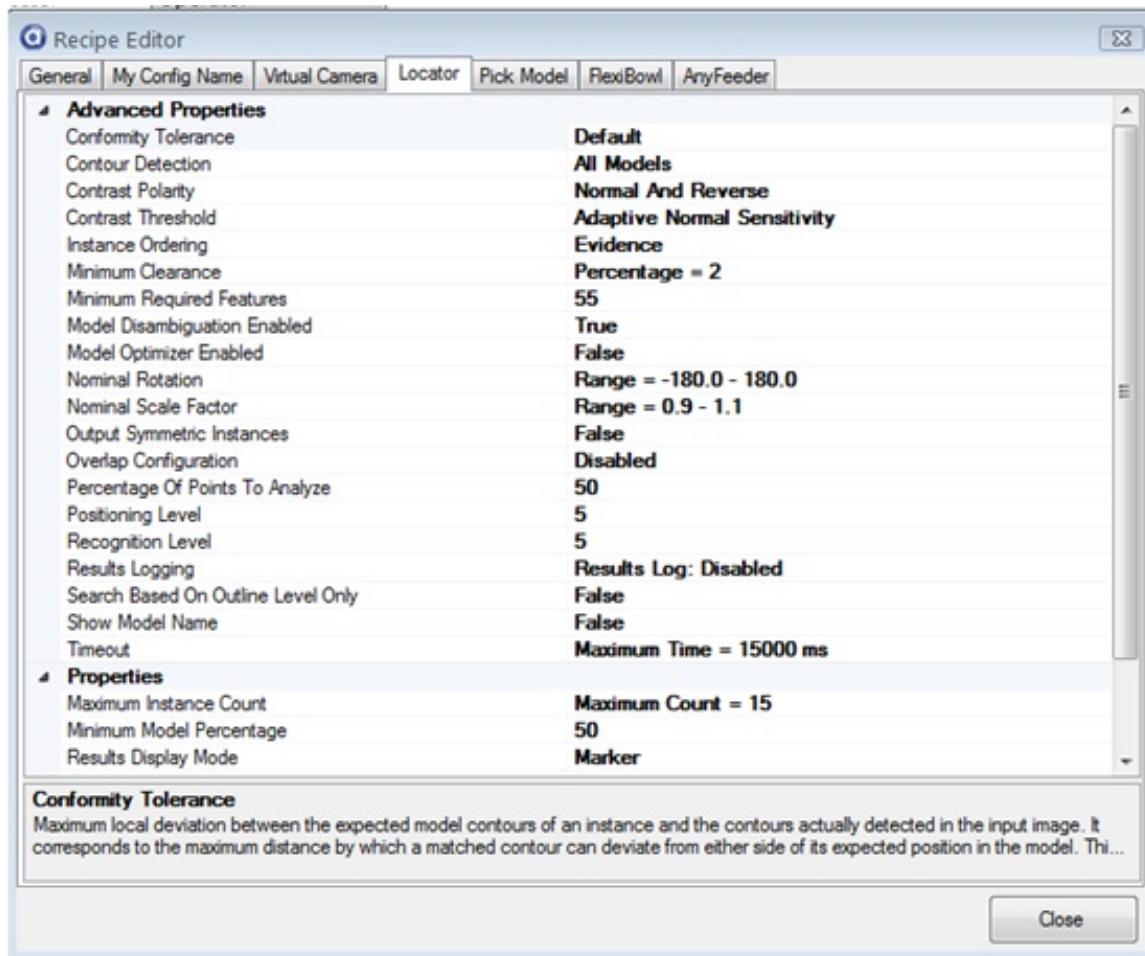


*Virtual Camera Editor*

## Other Vision Tool Sources

The user is displayed the Properties and Advanced Properties of the vision tool.

## Recipe Editor



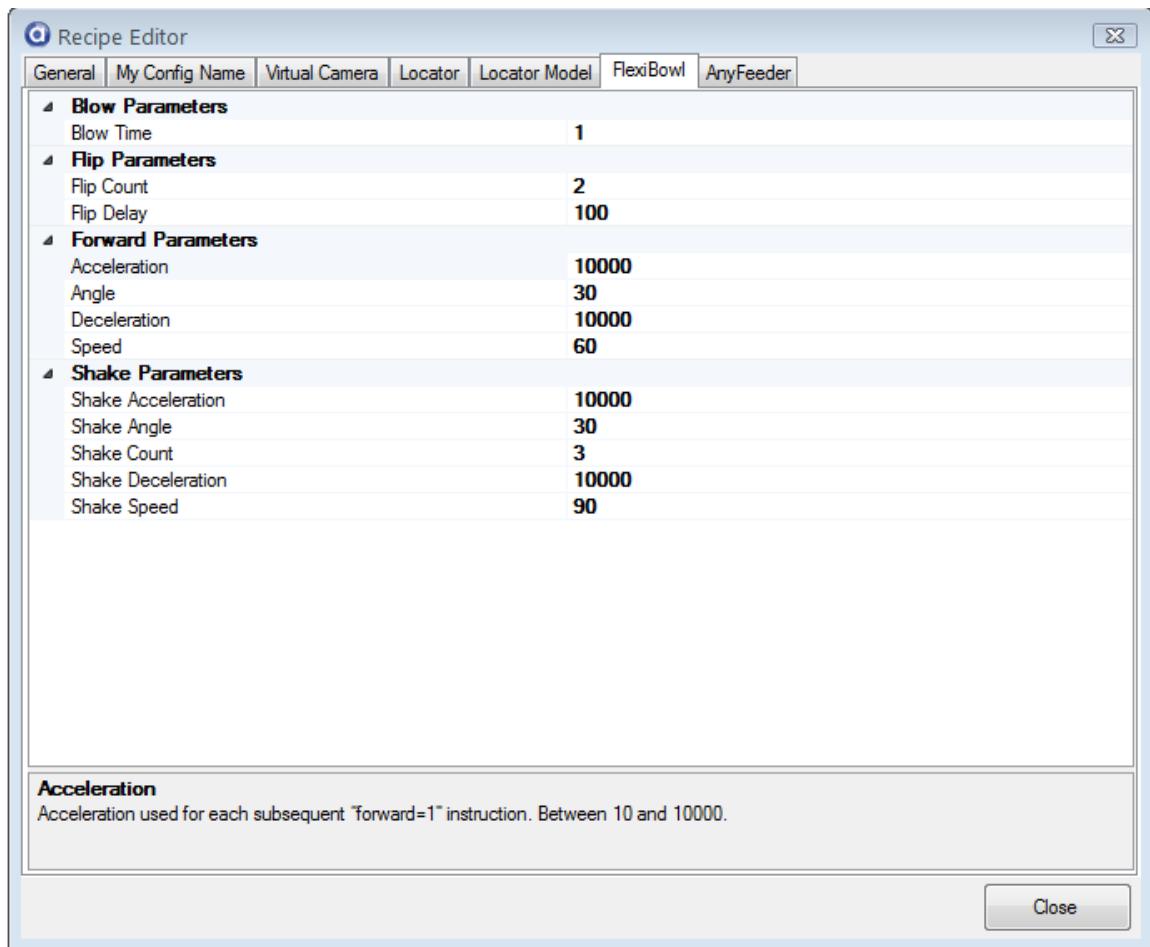
## Locator Model Editor

### Feeder Source

The user is displayed the general feed properties of the feeder.

## Recipe Editor

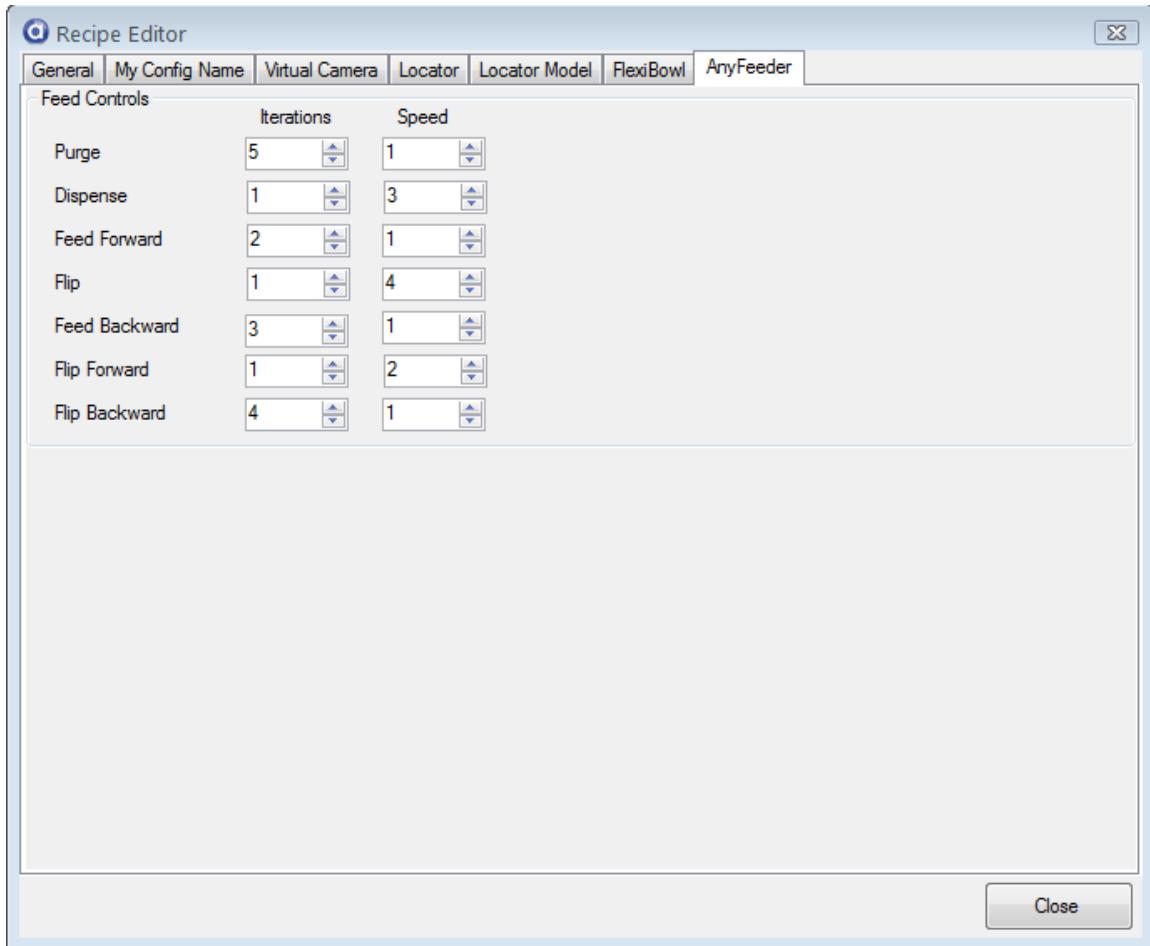
---



Flexibowl Feeder Editor

## Recipe Selection

---



*AnyFeeder Editor*

### Recipe Selection

When a recipe is selected, the parameters saved in the recipe are applied to the workspace. All V+ variables will be set to the corresponding values. All vision tool and feeder properties will be copied into the appropriate sources in the workspace .

## User Interface Designer

The topics in this chapter describe the User Interface Designer features of the ACE software.

Overview .....	1001
Understanding the Interface .....	1003
Adding and Editing UI Tools .....	1016
Using the Appearance, Behavior and Layout Controls .....	1025
Using the Condition Editor .....	1030
Using Property Links .....	1039
Using the User Access Controls .....	1043
Creating a Combo Box or List Box .....	1045
Creating a Graph .....	1052
Creating an Image List .....	1058
Creating a Split Form .....	1066
Creating a Tabbed Form .....	1070
Creating an Example User Interface .....	1076
Deploying the Custom User Interface .....	1082
Controlling the User Interface through Scripts .....	1087

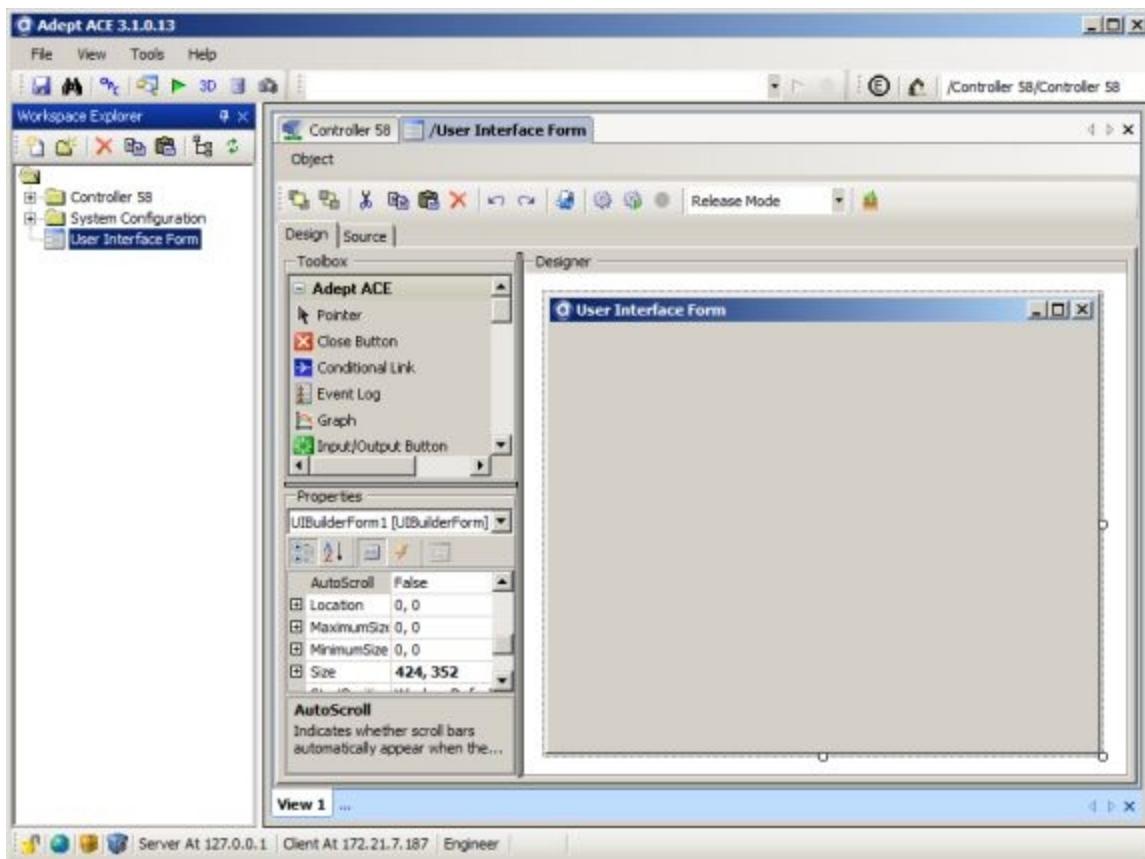
## Overview

The ACE software contains a User Interface (UI) Form designer, which is used to create a custom UI for your application. The UI Form object contains a resizable form, along with a library of tools that you can drag onto the form, to create a custom UI. The UI Form designer also includes controls for customizing the look and feel of your UI.

To create a new UI Form object, right-click in the folder area of the Workspace Explorer, and select:

### New > Program > User Interface Form

To open the UI Form designer, double-click the User Interface Form object in the folder area of the Work-space Explorer.



### User Interface Form Object

Like other ACE folders and objects, the User Interface Form can be saved to a file and loaded into other ACE projects. This allows you to create a "design template" and then reuse it as a starting point for your UI design in other project. Note that any linked tools in your template UI will need to be relinked when it is imported into a new ACE project. For details on saving and loading workspace objects, see Shortcut Menus on page 76.

---

## Overview

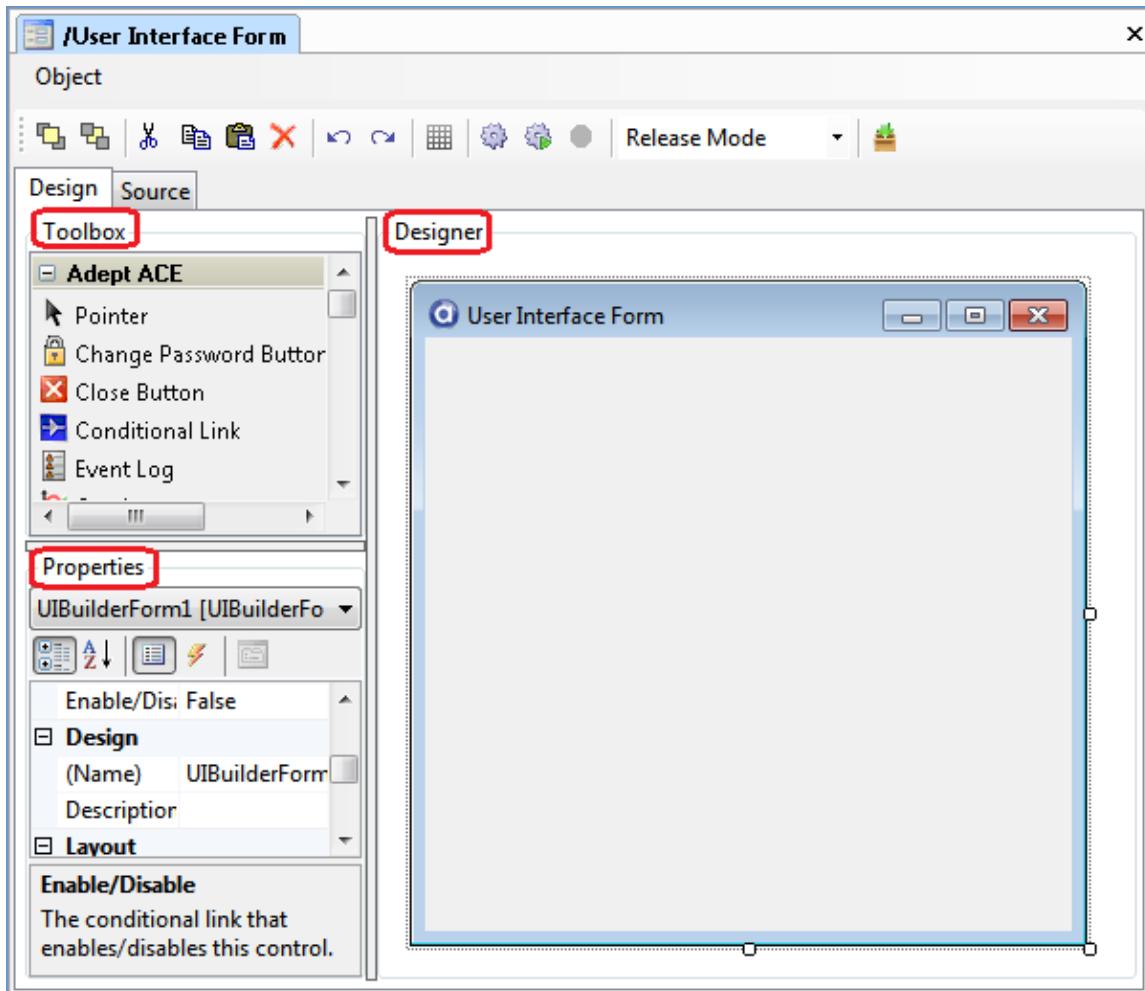
---

The following sections describe the tools and functions that are available in the UI Form designer. For more details on the tools that are available, see [Understanding the Interface](#) on page 1003. For an example of creating a simple UI, see [Creating an Example User Interface](#) on page 1076.

## Understanding the Interface

The User Interface Form object has three areas (see the following figure):

- The upper-left area is the Toolbox—it provides a library of tools that you place on the UI Form to create the custom interface.
- The lower-left area is the Properties editor—it provides categories of editable properties that control the look/feel and actions of the items on the UI Form.
- The right-hand area is the Designer—it provides the design area for building the custom user interface.



*UI Form Object - Toolbox, Properties, and Designer*

## Tab Items

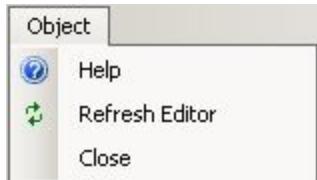
This section describes the tab items available from the UI Form object.

Tab	Description
Design	Displays the UI design tools and properties grids, which are used to select, place, and edit the items in your custom user interface.
Source	<p>Displays the event handler C# code for each item on the UI form. The Source tab lets you add custom code to the UI form. For details on locating an event handler for a specific tool on the form, see <a href="#">Viewing the Events for a Tool on page 1013</a>. For more details on the C# Program Editor, see <a href="#">C# Language Programming on page 750</a>.</p> <p><b>NOTES:</b></p> <ol style="list-style-type: none"> <li>1. You can go directly to an item's default event handler code by double-clicking the item in the Design area. When the code opens, the cursor will be located on the default event handler for that item.</li> <li>2. If you close the Source window, it also closes the UI form. You can reopen the UI form from the ACE Workspace Explorer pane. For details on the Workspace Explorer, see <a href="#">Workspace Explorer on page 75</a>.</li> </ol>

## Menu Items

This section describes the selections available from the UI Form object menu.

### Object Menu



#### Help

Displays the online help for the User Interface Form object.

#### Refresh Editor

Refreshes the contents of the object.

#### Close

Closes the object.

## Toolbar Icons

This section describes the toolbar icons:

Icon	Description
	Moves the selected item to the front (top) of a stack of items.

## Toolbar Icons

---

<b>Icon</b>	<b>Description</b>
	Moves the selected item to the back (bottom) of a stack of items.
	Deletes the selection and moves it to the Windows clipboard.
	Copies the selection to the Windows clipboard.
	Pastes the contents of the Windows clipboard onto the UI Form Designer.
	Deletes the selection but does not store a copy in the Windows clipboard.
	Undoes the last action performed.
	Undoes the last undo action.
	Displays a grid in the Designer area, which helps with placement and alignment of items on the form. For more details, see Alignment Guides and Grid Pattern on page 1018.
	Clears the messages in the Trace Messages tab.
	Compiles the program.
	Builds and runs the form. This starts a compiler, which builds a form from the items you've placed in the UI Designer. A progress bar displays while the form is being compiled. After the form has opened, you can test the buttons, controls, etc., on the form to verify that it is working correctly.  If the form is already running, and you click this icon again, it will close the current form, recompile, and run a new instance. A progress bar shows the status of this process.
	(Active when program is running) Stops the program.
<b>Release Mode</b> ▾	Selects the editor mode: Release Mode is the normal execution mode; Debug Mode is used to debug the program. When Debug Mode is selected, the Step Over and Step Into icons are shown on the toolbar.  <b>NOTE:</b> If the UI designer is closed while in Debug mode, when reopened during the same ACE session, it will reopen in Debug mode. If the ACE session is restarted, when the UI designer is opened, it will open in Release mode.
	(Debug Mode only) Used to step over code that is executed outside the current method.

Icon	Description
	(Debug Mode only) Executes the next line of the program (single step through the program).
	Deploys the custom user interface. The User Interface Deployment dialog box opens, which allows you to create a shortcut for deploying the custom user interface. For details, see Deploying the Custom User Interface on page 1082.

## Toolbox Items

The Toolbox contains a list of tools that you drag into the UI Form to create the custom interface. The following table describes each of the Toolbox items.

**NOTE:** Some items allow you to control user-access by setting the user-access level for the item. For more details on controlling user access, see User Manager on page 1013.

Item	Description
<b>ACE</b>	
<Pointer>	This is a special item in the Toolbox. It is used for tool selection in the design area. After one or more tools have been added to the design area, the Pointer tool is used to select a tool for editing.
Change Password Button	Provides a button that opens the Change Password form, which is used to change the password of the current user. For more details on the Change Password form, see Creating or Changing a Password on page 129.
Close Button	Provides a button that is used to close the form. This provides an alternate way for a user to close the form in addition to using the standard Windows "close" icon (☒).
Conditional Link	Inserts a Conditional Link item, which is used to specify the criteria for a condition (for example, if a variable equals a specified value). When that condition is met, any Tool that is linked to that condition will be enabled/disabled, shown/hidden, etc., based on how the tool is linked to the condition. For more details, see Using the Condition Editor on page 1030.
<b>NOTE:</b> When this tool is added, it appears below the UI form design area.	
Event Log	Inserts an Event Log tool. For more information on the Event Log, see Event Log on page 1108.
Graph	Inserts a Graph tool. For more information on the Graph tool, see Creating a Graph on page 1052.

Item	Description
	<p><b>NOTE:</b> This tool requires a special license. For details, see Licensing Requirements on page 28.</p>
Input/Output Button	<p>Provides a specialized button used to toggle a digital output or display the state of a digital input. Its operation is similar to the Toggle Button (see below); however, it is used exclusively for digital input/output signals, and it uses a simplified set of configuration controls.</p>
Robot Jog Control	<p>Inserts the Robot Jog Control. For more details on the Robot Jog Control, see Robot Jog Control on page 163.</p>
Sign In Button	<p>Provides a specialized button that displays the user Sign In dialog. For details on the Sign In dialog, see Signing In on page 128.</p> <p><b>NOTE:</b> Certain functionality is limited, based on the sign-in level. For details, see Appendix 2: User Access Item List on page 1168.</p>
Sign Out Button	<p>Provides a specialized button that is used to sign out of the current access level. For details on signing out, see Signing Out on page 128.</p>
Transform Editor	<p>Provides controls used for entering transform coordinates (X, Y, Z, yaw, pitch, roll). For example, if this was linked to the Gripper object offset, it would allow the operator to use the UI to modify the tool offset. It could also be used to modify a frame or location origin. For details on tool offsets, see Using a Tool Offset on page 261. For details on locations, see The Coordinate System on page 58.</p>
User Interface Form Display Button	<p>Provides a button that is used to launch another User Interface Form. For example, you could have a "master" or "parent" form that contains a button to launch a sub-form or "child" form. You can use the Connection properties group to set the mode of the "child" form. For information on the Connection properties, see Properties on page 1011.</p> <p><b>NOTE:</b> The access level of the child (sub-form) is inherited from the parent form. For details on access levels, see User Manager on page 1013.</p>
Vision Display	<p>Displays a vision image from the specified vision camera source. The Image Source property must be set to the camera whose image should be displayed in this area.</p>
<b>Advanced Tools</b>	
<p><b>NOTE:</b> These tools require a special license. For more details on licenses, see Licensing Requirements on page 28.</p>	

Item	Description
<Pointer>	This is a special item in the Toolbox. It is used for tool selection in the design area. After one or more tools have been added to the design area, the Pointer tool is used to select a tool for editing.
Button	Similar to the Windows Forms Button tool. When pressed, it sets a value.
Digital Indicator	Displays a digital (Boolean) value on a tool that looks similar to an indicator light.
Gauge	Displays a numeric value on an analog-style gauge.
Input/Output Button	Displays the state of an input signal, or allows you to toggle the state of an output signal.
Numeric Indicator	Displays a numeric value on a digital-style gauge. This tool includes a DecimalPlaces property, which is used to set the number of decimal places displayed on the gauge.  <b>NOTE:</b> When there are not enough digits to display the entire value, an error icon (!) will be shown next to the tool and all digits will be turned off.
Slider	Sets a numeric value, which is based on the position of the slide in relation to the data minimum and maximum settings.
Toggle Button	A dual-state button. Pressing it changes its state, which allows you to toggle the state of an item. For example, if you connect this to the controller high power data item, you can toggle the power on/off.
<b>Windows Forms</b>	
<Pointer>	This is a special item in the Toolbox. It is used for tool selection in the design area. After one or more tools have been added to the design area, the Pointer tool is used to select a tool for editing.
Button	Creates a button that executes a specified action. For example, this tool is typically used to start the execution of a C# script.
Check Box	Creates a check box, which is used to toggle or display the state of a numeric value. It can be linked to a numeric ACE item to directly control its state or display its state.
Combo Box	Creates a text box with a closed list box attached. The list box opens when you click the arrow next to the text box. Use the Data > Items (Collection) property to display the String Collection Editor, which is used to add items to the list. The list is available when you compile and run the interface.  Optionally, you can choose to have the list sorted when the interface is compiled. If you select this option, the list will also be sorted in the String Col-

Item	Description
	lection Editor.
Group Box	<p>Used to group (frame) related tools. It is similar to the Panel tool, but it provides a visible frame along with a name area at the top of the frame. For example, if you have controller and robot tools in your UI, you could put the controller tools into a "Controller" group box, and the robot tools into a "Robot" group box.</p> <p><b>NOTE:</b> The default access level for this tool is Operator.</p>
Image List	<p>Provides a list (collection) of images that will be displayed by a Picture Box.</p> <p><b>NOTE:</b> When this tool is added, it appears below the UI form design area.</p>
Label	Displays static text on the form. For example, you could place a label at the top of the form as a title, or you could place a label above/below a progress bar to describe what the progress bar is showing.
List Box	Used to display a box containing a list of items the user can select.
List View	Used to display a list of items with item text. Optionally, an icon can be included to help identify the item type.
Numeric Up Down	Provides a field and up/down controls for numerical input. This is a combination of a field into which the user may type the desired value, and up-/down arrows that modify the value by a specified increment. The acceptable range of values and increment/decrement, as well as the number of decimal places displayed, are all configurable. It can be linked to any exposed numeric ACE item to directly control its value.
Panel	<p>Used as a container for related tools. In most applications, the frame is not visible to the user. It is similar to the Group Box, but it does not provide a name at the top of the frame.</p> <p><b>NOTE:</b> The default access level for this tool is Operator.</p>
Picture Box	Used to display a static or animated graphic in the UI. Usability and understanding can be greatly enhanced through the proper use of graphics (a picture is worth a thousand words). The Picture Box can also be linked to an Image List to display different images based on the value of the connected data item. For more details, see Creating an Image List on page 1058.
Progress Bar	Used to display a numerical value. As the name implies, a common usage is to display the percent done of a task while the task is running. The minimum (start of progress) and maximum (completion of progress) values are configurable. It can be linked to any exposed numeric ACE item for direct display

Item	Description
	of its value.
Radio Button	Creates a radio button, which is used to toggle or display the state of a numeric value. It can be linked to a numeric ACE item to directly control its state or display its state.
Split Container	Provides a boundary line that separates a panel into two resizable regions. It is typically used with the Group Box to make resizable containers for the UI tools. For more details, see Creating a Split Form on page 1066.
	<b>NOTE:</b> The default access level for this tool is Operator.
Status Bar	Creates a status bar, which is used to display the current status of the linked item. For example, you could display the controller power status (Power ON or Power OFF).
Tab Control	Used to visually organize the tools in the operator interface. For example, related tools can be displayed on the different tabs in the control. For more details, see Creating a Tabbed Form on page 1070.
Text Box	Used to display text to the operator or get text input from the operator. It can be linked to any exposed string ACE item to directly control its value.
Timer	Creates a timer control, which is used to run a C# script at a specified interval.
	<b>NOTE:</b> This tool requires a special license. For details, see Licensing Requirements on page 28.
	<b>NOTE:</b> When this tool is added, it appears below the UI form design area.
Toggle Button	Creates a toggle, a true/false-type control, which can be in the form of a digital input, digital output, or custom image. A toggle can be an indicator as well as a control. For example, if you create a toggle to control soft signal 2009, it will also indicate the state of that signal if it is changed from outside of your UI (such as from a program or process).
Track Bar	Used for numerical input from the operator. It can be linked to any exposed numeric ACE item to directly control its value.

#### A Note on Toggle Buttons

The User Interface Form object provides two types of toggle buttons: the Input/Output Button and the Toggle Button.

- The Input/Output Button is a "simplified" toggle button, which provides links and settings that are commonly used to create toggles and indicators for digital input and output signals.
- The Toggle Button is a full-feature toggle button, which provides links and advanced settings for creating toggles and indicators for digital signals, power and calibration, controller connection, variables, and other items in the ACE workspace.

## Properties

The Properties editor provides access to linking and formatting (appearance) parameters for the form tools. The following table describes the main purpose of each property group. Only those property groups that apply to the selected form tool will be displayed. When multiple form tools are selected, the common properties are shown, which allows you to edit those properties for multiple tools at once, rather than individually.

**NOTE:** Some items allow you to control user-access by setting the user-access level for the item. For more details on controlling user access, see User Manager on page 1013.

### Properties Toolbar

This section describes the Properties toolbar icons:

Icon	Description
	Arranges the available properties by category (such as Appearance). The categories are sorted in alphabetical order.
	Arranges the available properties in an alphabetical list.
	Displays the properties for the selected tool.
	Displays the events for the selected tool. For details, see Viewing the Events for a Tool on page 1013.
	Not used at this time.

### Properties Items

**NOTE:** The list of available parameters will vary, based on the tool/item that is currently selected.

This section describes the Properties items:

Item	Description
[Pull-down list]	Provides quick access for selecting a form item for editing. All items on the

Item	Description
	<p>form will appear on the list. Simply click the item to select it and edit its properties.</p> <p>This list is also useful for locating form items that you've hidden under another item or hidden with the Layout &gt; Parent property.</p>
Appearance	This group controls the appearance (background color, foreground color, font, border, etc.) of the selected tool. For details, see Using the Appearance, Behavior and Layout Controls on page 1025.
Behavior	This group controls the behavior (enabled, range, etc.) of the selected tool. For example, the range (minimum and maximum values) for a Track Bar tool.
Conditional Links	This group controls the display and/or enabling of the selected tool based on a specified condition. The condition is defined using the Conditional Link tool, described in the previous table. For details, see Using the Condition Editor on page 1030.
Connection	This group connects the tool to another UI form. This allows a tool on one form to open another UI form (typically referred to as a "child" form). It contains a Modal property that, when set to "true", flags the form as a "modal" (while the child form is open, it prevents access to the parent form).
Data	Specifies minimum and maximum values, decimal places and increment (the size of the step each time the control is activated).
Design	<p>Specifies a name and description for the selected form item.</p> <p>The name is used to reference the item in custom scripts. A default name is provided, which can be changed.</p> <p>The description is used to provide a meaningful description of the form item for your reference.</p> <p><b>NOTE:</b> The name must be unique for each UI form item. However, names can be reused on other UI forms in the project.</p>
Images	<p>This group controls the display of images that will be placed onto the selected tool. For example, for a toggle button, you could display a different image for each state of the button (pressed/active, pressed/inactive, not pressed/active, and not pressed/inactive).</p> <p><b>NOTE:</b> To activate the images for the selected tool, you must select the Custom option of the DisplayMode property in the Images group.</p>
Layout	This group controls the position and size of the tool. For details, see Using the

Item	Description
	Appearance, Behavior and Layout Controls on page 1025.
Property Links	This group exposes items in the ACE workspace to the selected tool. Although the other properties groups are important, it is the Property Links group that allows you to use your interface elements to view or write (control) items in the ACE application. For example, you can connect a Button tool to the Power and Calibration data item to toggle power on/off and initiate robot calibration. For details, see Using Property Links on page 1039.
User Manager	This group sets the access level and access visibility for the selected tool. For more details, see Using the User Access Controls on page 1043.
Window Style	This group controls the visibility (Z-order) of the selected base form in relation to other forms. When true, the form will appear on top of (in front of) all other forms on the desktop that do not have this property set to "true".

### ***Setting the UI Form Title***

You can set the title of the base form (the window title of the form when it is deployed) by selecting the base form in the Designer area and then typing the desired name in the Appearance > Text parameter of the Properties group. For more details on controlling the appearance of your UI form, see Using the Appearance, Behavior and Layout Controls on page 1025. For more information on deploying the UI form, see Deploying the Custom User Interface on page 1082.

**NOTE:** The ACE server version is displayed in the title bar of the form at run time. It uses the following pattern:

ACE [version] - [Text]

This can be customized by adding an event handler to the Load event and manually setting the Text property of the form. For details on controlling the UI through scripting, see Controlling the User Interface through Scripts on page 1087.

### ***Viewing the Events for a Tool***

The UI Form designer allows you to view (and modify) an event tied to a particular tool. For example, if you have placed a Button tool on your form, you can select the button, click the Events ( ) icon on the Properties toolbar, select an event for that tool, and then view (and modify) the event handler for the selected event.

As a shortcut, if you double-click the tool in the UI form designer, it will display the default event handler for that tool. When the code opens, the cursor will be located on the default event handler for that item.

For example, to view a specific event for a Button tool:

## Properties

---

1. Drag a button onto your form.
2. In the Properties selector, locate (select) the Button tool you just added to the form.



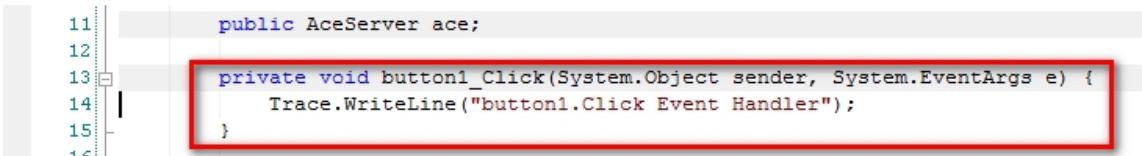
*Properties Item Selector*

3. Click the Event ( ) icon. The Event selector opens.



*Event Selector*

4. Select an event to view for the tool. In the case of the Button tool, the available event is "button1\_Click", which is selected from Action > Click > button1\_Click. When you select the event, the code view for the UI form opens, and the cursor is placed at that event handler in the code.



*Cursor at Event Handler Code*

Note the following about the event viewer:

- You cannot modify the event handler line (it is marked by a gray background, as shown in the previous figure, which means it is protected and cannot be edited or removed).
- If you do NOT add any code to the event handler, when you select "none" from the Event selector, the event handler code is removed, which makes the code "cleaner".
- If you add code to the event handler, the event handler code will NOT be removed when you select "none" from the Event selector.

***A Note on Parent/Child User Interface Forms***

When you compile and run parent/child User Interface forms, you should be aware of the following special behavior/restrictions:

- When you sign in (or sign out) on a child UI form, the parent UI form does not sign in (or sign out).
- When you sign in (or sign out) on a parent UI form, if a child UI form is already open, the child UI form does not sign in (or sign out).
- When the parent UI form is open (compiled and running), if you make changes to one of its child UI forms, you need to compile/run the changed UI form and then save the ACE workspace, so the changes are stored (and visible) when you deploy the multi-form User Interface.

## **Adding and Editing UI Tools**

As discussed in the previous section, the UI Form Designer allows you to create and deploy a custom user interface for your ACE application. This section describes how to add tools to your UI form and then edit their properties.

### **Adding Tools to the Form**

There are three ways to add a tool to your form:

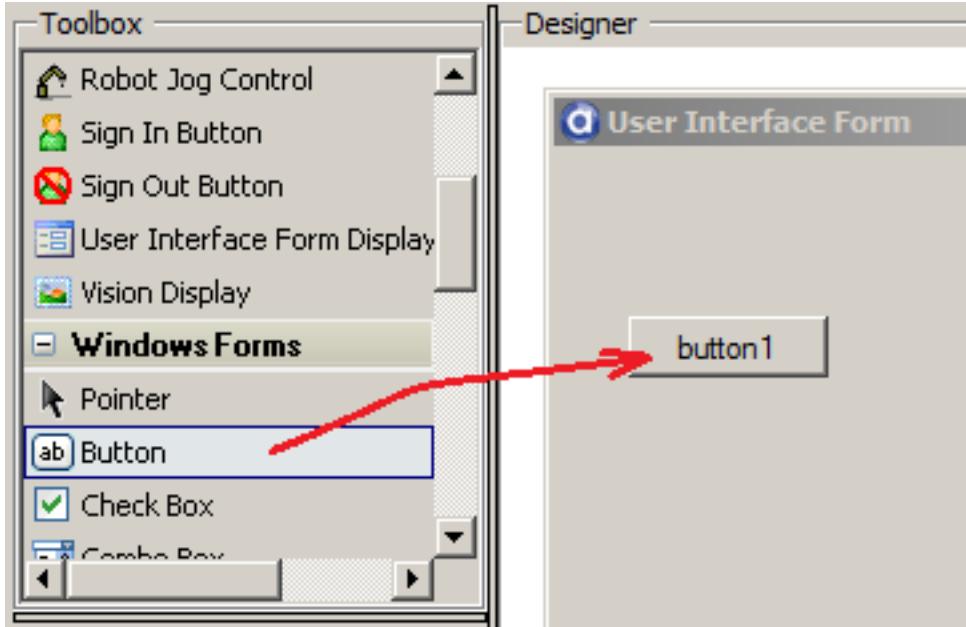
- Double-click
- Drag and drop
- Select and place

To add a tool by double-clicking:

1. Locate the desired tool in the Toolbox
2. Double-click the tool. The selected tool is placed on the upper-left corner of the UI form.
3. Move the tool to the desired position on the UI form.

To drag and drop a tool onto your UI form:

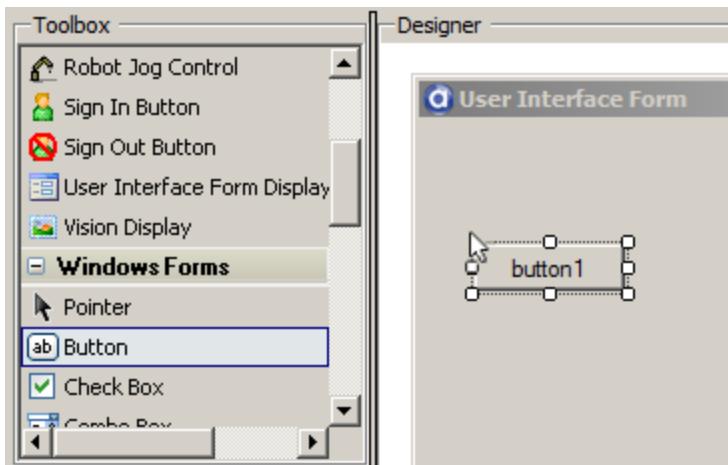
1. Locate the desired tool in the Toolbox
2. Select the desired tool and hold the left mouse button down. The selected tool is highlighted.
3. Without releasing the left mouse button, drag the selected tool to the desired position on your form.
4. Release the left mouse button. The selected tool is positioned on the UI form.



*Drag and Drop Operation*

To select and place a tool onto your UI form:

1. Locate the desired tool in the Toolbox
2. Select the desired tool.
3. Click on the desired position on your form. The selected tool is positioned on the UI form.



*Select and Place Operation*

**NOTE:** If the position or size of the tools on the form exceeds the form boundaries, vertical and/or horizontal scroll bars are automatically added to the UI form.

### **Alignment Guides and Grid Pattern**

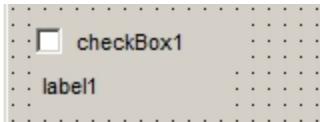
The Designer area contains two useful tools for aligning the items on the UI form: the alignment guides and the grid pattern. The grid pattern is enabled/disabled using the grid icon (grid icon) on the UI Builder toolbar.

- When the grid is disabled, the alignment guides automatically show as you drag a form item into proximity of the X and/or Y axis of another form item.



*Alignment Guides*

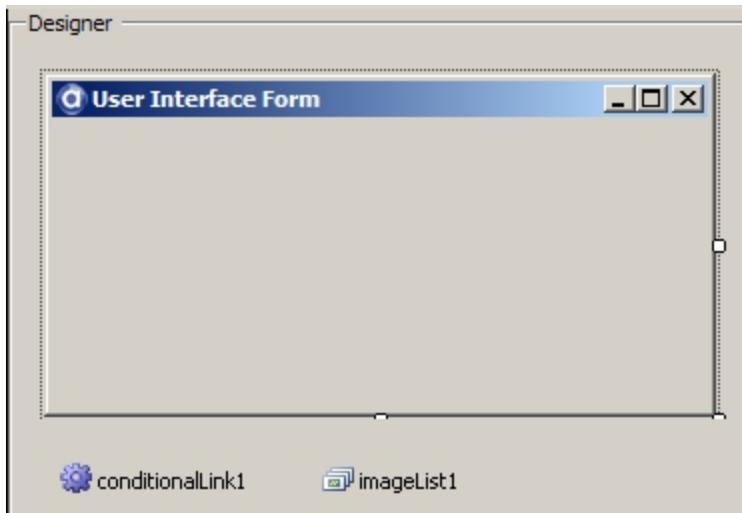
- When the grid is enabled, an X - Y pattern of dots appear on the form, which are used to align the form items. The alignment guides are disabled when the grid is enabled.



*Grid Pattern*

### **A Note About Conditional Link and Image List Tools**

The Conditional Link and Image List are special tools that are associated with the UI form but not located directly on the form. When a Conditional Link or Image List tool is added to the UI form, it is automatically relocated to the area below the form, as shown in the following figure.



*Conditional Link and Image List Tools Below the UI Form*

For more details on the Conditional Link tool, see Using the Condition Editor on page 1030. For more details on the Image List tool, see Creating an Image List on page 1058.

## **Editing a Tool on the Form**

After you place a tool onto your form, you will need to edit various properties of the tool. For example, if you have just placed a button on the form, you will want to edit the name, size, color, position, function, and other properties.

The ACE UI Form Designer contains a Properties area that is used to edit the various properties of the selected tool. Only those properties that apply to the selected tool are shown. The properties are organized into groups, such as Font, Layout, Property Links, etc., to make it easier to locate the property that you wish to edit.

To edit a tool on the form:

1. Locate the desired tool on the form.
2. Select the tool. A dashed border with "handles" appears on the selected tool. The "handles" serve two purposes:
  - They show that the tool is selected for editing
  - They provide a quick way of resizing the tool. Simply drag a handle to resize the tool.
3. You can reposition the selected tool by:
  - Dragging it to the desired position, or
  - Use the Location properties to specify the desired position.

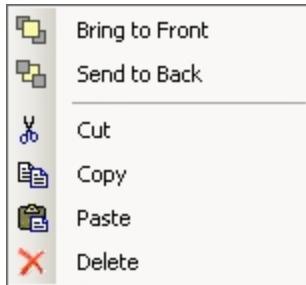
<input type="checkbox"/> Location	<b>78, 138</b>
X	<b>78</b>
Y	<b>138</b>

*Location Properties*

4. Use the Properties area of the form designer to edit the desired properties for the selected tool. For example, to edit the label on a button, you would:
  - a. Select the button, as described above.
  - b. In the Properties area of the UI Form Designer, locate the Text item in the Appearance group.
  - c. Edit the Text item so that it contains the text you want to show on the button label.

## **Using the Editing Commands and Z-Order**

The UI Form Designer provides a shortcut menu that is used for editing operations, such as cutting, copying, pasting, etc. The shortcut menu opens when you right-click on an item in the UI form.



*UI Form Designer Shortcut Menu*

The shortcut menu items are described in the following sections.

In addition to the shortcut menu, the same editing functions are also provided on the UI Form Designer toolbar. For more details, see User Interface Designer on page 1000.

### **Selecting an Item for Editing**

There are several methods for selecting an item (or group of items) for editing, as follows:

To select a single item:

- Click the item to select it, or
- Drag around the item

To select multiple items:

- Drag around the items, or
- Hold the Ctrl (Control) key while clicking each item, then release the Ctrl key

**NOTE:** When multiple items are selected, only the common properties for those items are shown in the Properties area of the UI Form Designer.

### ***Using the Editing Functions***

The UI Form Designer uses the standard Windows functions for selecting, copying, pasting, cutting, and deleting items on the form. The following table provides a list and description of the available editing functions.

<b>Icon</b>	<b>Shortcut</b>	<b>Description</b>
	Ctrl+X	Deletes the selection and moves it to the Windows clipboard.
	Ctrl+C	Copies the selection to the Windows clipboard.
	Ctrl+V	Pastes the contents of the Windows clipboard onto the UI Form Designer.
	Delete	Deletes the selection but does not store a copy in the Windows clipboard.

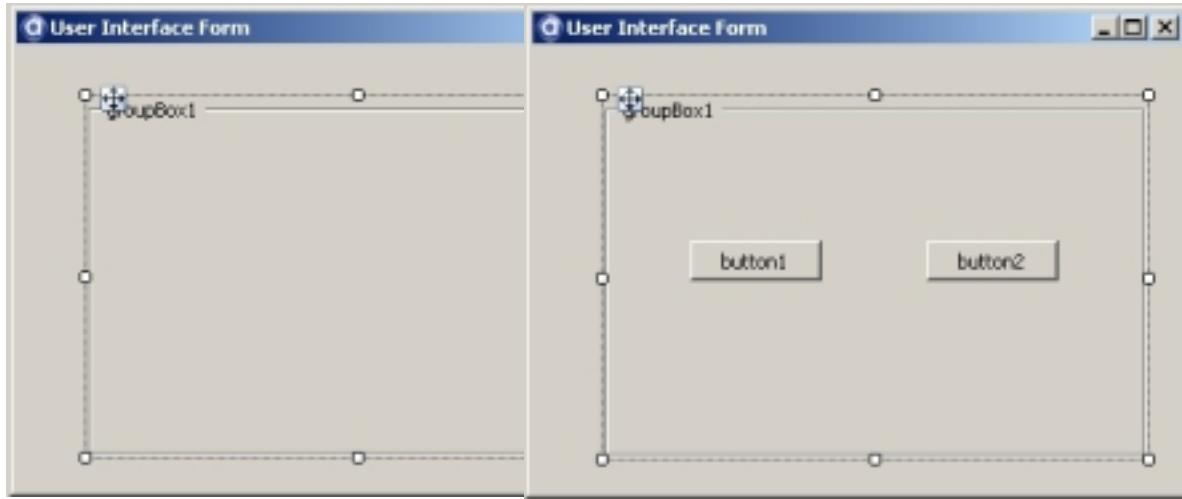
### ***Changing the Z-Order***

In addition to resizing a tool, you can also change its Z-order (the front-to-back position of the tool when it is in a stack of tools). The UI Builder provides two special tools for this operation:

<b>Icon</b>	<b>Shortcut</b>	<b>Description</b>
	N.A.	Brings the selected item to the front (top) of the stack.
	N.A.	Sends the selected item to the back (bottom) of the stack.

For example, suppose you have created a form with some buttons and other related items. Now you want to group all of those items with a Group Box tool, to make it visually friendly for the user. When you add the Group Box tool to your form, it appears on top of the other tools that are already on the form, which hides them. To fix this, you need to change the Z-order of the tools (move the Group Box tool to the back of the stack), as follows:

1. Select the Group Box item.
2. Right-click to open the shortcut menu.
3. Select Send to Back from the shortcut menu. The Group Box is moved to the back (bottom) of the stack, which allows the buttons to come into view.

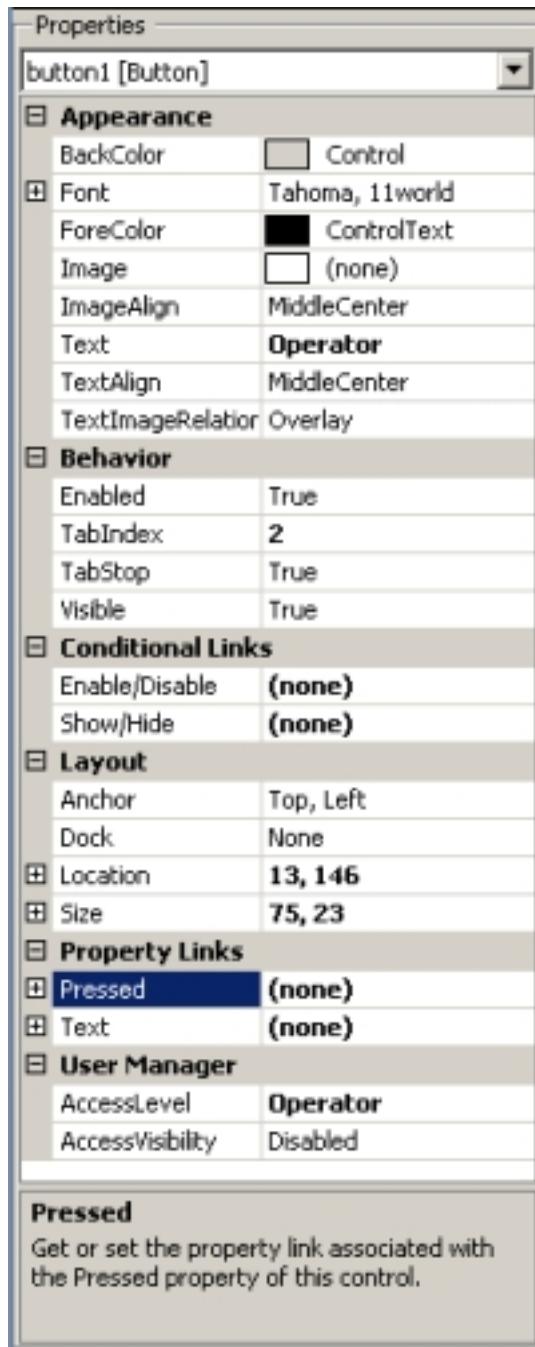


*Group Box in Front with Buttons Hidden*

*Group Box in Back with Buttons Exposed*

### **Editing the Item Properties**

The Properties editor is used to edit the appearance, visibility, and various functions of the selected UI Form item. The properties are available based on the particular item selected for editing. For example, a button will have different set of properties than a label. The following figure shows the Properties editor set for a button.



*Example Properties Editor Set for a Button*

The complete list of Properties groups, and their descriptions, is provided in Properties on page 1011.

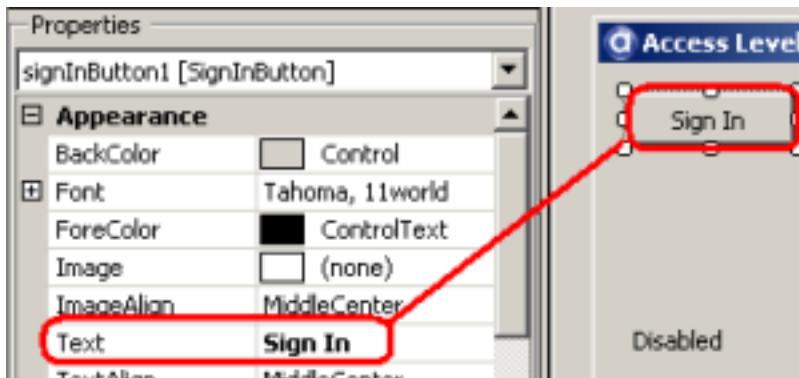
**NOTE:** The list of available parameters will vary, based on the tool/item that is currently selected.

**NOTE:** Online help for the parameters is provided directly in the editor. Therefore, it is not duplicated in this documentation.

To edit the properties for a UI Form item:

1. Select the item you wish to edit.
2. Go to the Properties area of the UI Form Designer and locate the group that contains the properties you want to change.
3. As needed, click the "+" sign next to a group heading to expand that group and show its properties.
4. Click the field next to the property item you want to change. In most cases, the editor opens a pick list or other predefined set of values for you to select from.
5. Select the value from the provided list to complete the edit. As you make a selection, the change is immediately applied to the selected UI Form item.

There are a few properties, for example, Appearance > Text, which require you to type the information into the field. In the case of Appearance > Text, you would type the text, which becomes the label for the item (for example, the label on a button, as shown in the following figure).



*Appearance > Text Property for Button Label*

## Using the Appearance, Behavior and Layout Controls

Each tool in the UI Form Designer contains a set of Appearance and Layout properties. These are used to control the font, color, size, position, etc., of the selected tool. For example, you can use these to set the colors and size of a button, and the font for the button label. Many tools also contain a set of Behavior properties. These are used to control tab index, tab stop, enabled, visibility, etc., of the selected tool.

**NOTE:** The list of available parameters will vary, based on the tool/item that is currently selected.

### The Appearance Parameters

The Appearance parameters control the font, colors, image and text for the selected tool. The following figure shows an example of some of the Appearance parameters.

Appearance	
BackColor	 Control
Font	Tahoma, 11world
Name	ab
Size	11
Unit	World
Bold	False
GdiCharSet	1
GdiVerticalFont	False
Italic	False
Strikeout	False
Underline	False
ForeColor	 ControlText
Image	 (none)
ImageAlign	MiddleCenter
Text	<b>signInButton1</b>
TextAlign	MiddleCenter
TextImageRelation	Overlay

#### *Appearance Parameters Example*

These parameters are used to control the appearance of the selected tool.

**NOTE:** Online help for the parameters is provided directly in the editor. Therefore, it is not duplicated in this documentation.

### The Behavior Parameters

The Behavior parameters control tab, visibility, and of the selected tool. The following figure shows an example of some of the Behavior parameters.

Behavior	
Alignment	Top
Appearance	<b>FlatButtons</b>
Enabled	True
ItemSize	<b>25, 18</b>
SizeMode	<b>Fixed</b>
TabIndex	<b>20</b>
TabPages	(Collection)
TabStop	True
Visible	True

*Behavior Parameters Example*

These parameters are used to control the behavior of the selected tool. For examples of using the Behavior parameters, see Applying Behavior Parameters to the Tab Control on page 1074.

**NOTE:** Online help for the parameters is provided directly in the editor. Therefore, it is not duplicated in this documentation.

## The Layout Parameters

The Layout parameters control the location and size of the selected tool. The following figure shows an example of some of the Layout parameters.

Layout	
Anchor	Top, Left
Dock	None
Location	<b>13, 95</b>
X	<b>13</b>
Y	<b>95</b>
Size	<b>75, 23</b>
Width	<b>75</b>
Height	<b>23</b>

*Layout Parameters Example*

These parameters are used to control the layout of the selected tool.

**NOTE:** Online help for the parameters is provided directly in the editor. Therefore, it is not duplicated in this documentation.

## Applying the Appearance and Layout Parameters

The following are several examples of how the appearance and layout parameters can be applied.

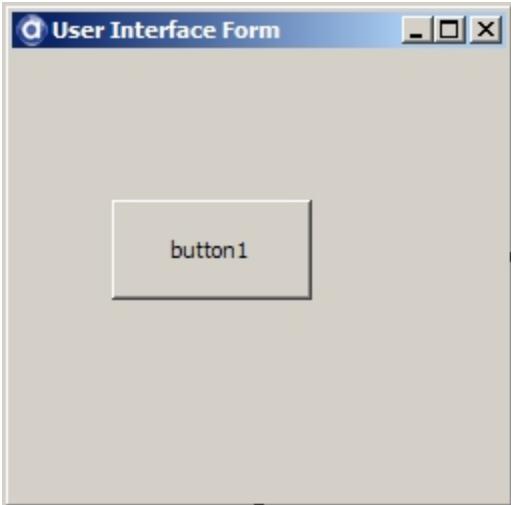
### ***Applying Layout Parameters to a Single Item***

This example shows how to apply Layout (size and location) parameters to precisely position and size a button. It is also possible to simply drag the button to the desired location on the form, and then drag the handles on the button to size it, as desired. However, if you want to position or size the button (or other tool) to specific values, it is recommended that you enter those values, as described in the following steps.

1. In the UI Form Designer, drag a Button tool onto your UI Form.
2. Select the button.
3. In the Properties area, scroll down to the Layout group.
4. Set the Layout parameters to the values shown in the following table.

Parameter	Value
Anchor	Top, Left
Location	50, 75
X	50
Y	75
Size	100, 50
Width	100
Height	50

5. The button appearance is set to the new values, as shown in the following figure.



*Button with Layout Parameters Applied*

#### **Applying the Appearance Parameters to a Single Item**

This example shows how to apply the Appearance parameters to the button that was positioned and sized in the previous example.

1. In the UI Form Designer, select the button that was used in the previous example.
2. In the Properties area, scroll down to the Appearance group.
3. Set the Appearance parameters to the values shown in the following table.

Parameter	Value
BackColor	Web > LightBlue
Font	
Name	Arial
Size	10
Unit	Point
Bold	True
GdiCharSet	0
GdiVerticalFont	False

Parameter	Value
Italic	True
Strikeout	False
Underline	False
ForeColor	Web > MidnightBlue
Image	(none)
ImageAlign	MiddleCenter
Text	My Button
TextAlign	MiddleCenter
TextImageRelation	Overlay

4. The Font category displays "Arial, 10pt, style=Bold, Italic" for the value.
5. The button appearance is set to the new values, as shown in the following figure.



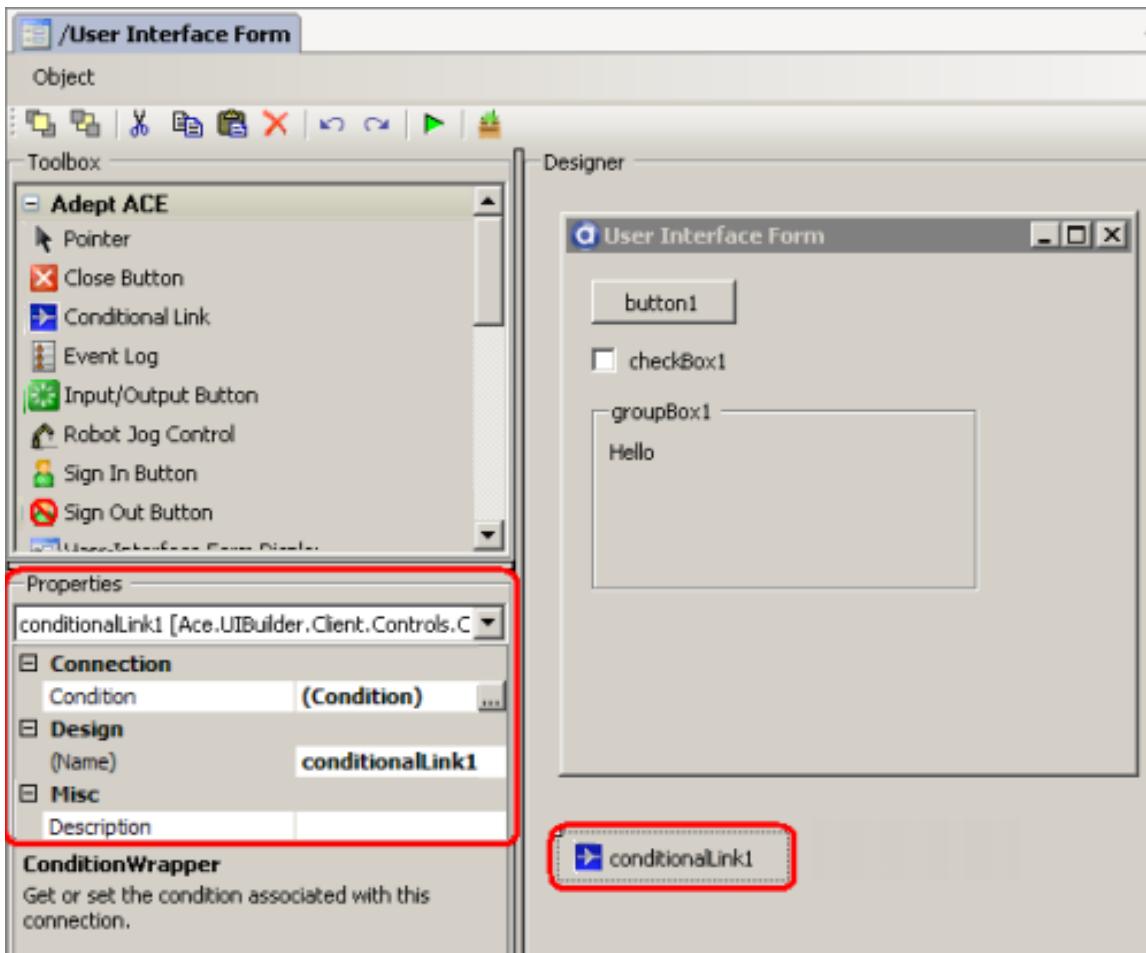
*Button with Appearance Parameters Applied*

## Using the Condition Editor

The Condition Editor is used to specify the criteria for a condition (for example, if a variable equals a specified value). When that condition is met, any tool that is linked to that condition will be activated/deactivated, enabled/disabled or shown/hidden, based on how the tool is linked to the condition. Note that the Condition Editor allows you to create and "relate" (with AND and OR) multiple conditions.

To open the Condition Editor:

1. Select the Conditional Link item in your UI Form designer. The Properties for the selected Conditional Link are shown at the bottom left column.



*UI Form Designer Showing Conditional Link Item Properties*

2. In the Connection group, locate the Condition parameter and click the browse (...) button. The Condition Editor opens. For details , see Condition Editor on page 1031.

## Conditional Properties

In the UI Form Designer, you use the properties for displaying or editing the condition item, and for assigning a descriptive name to the condition item.

Condition	(Condition)
(Name)	conditionalLink1
Description	Static numeric var. condition

*Conditional Item Parameters*

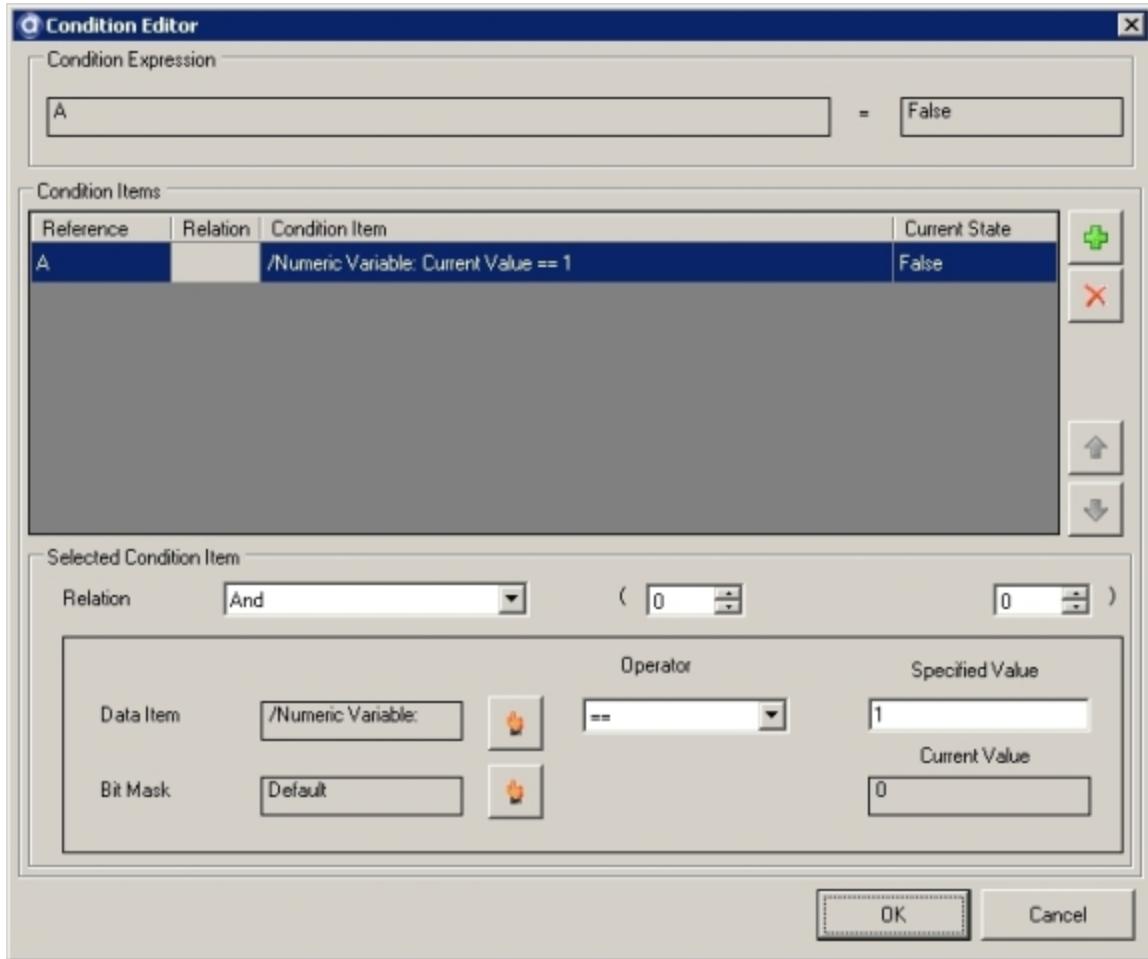
**NOTE:** Online help for the parameters is provided directly in the editor. Therefore, it is not duplicated in this documentation.

## Condition Editor

The Condition Editor is used to create one or more conditions. After the conditions are created, you can assign them to buttons, check boxes and other UI Builder tools, in order to control their appearance, actions, etc.

## Condition Editor

---

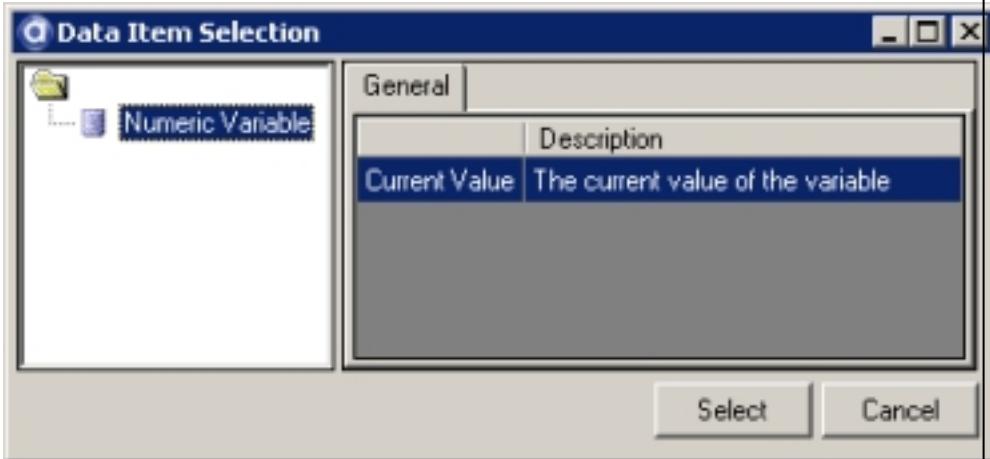


*Condition Editor with One Condition Defined*

<b>Item</b>	<b>Description</b>
Condition Expression	Shows the current state of the selected item in the Condition Items list.
Condition Items	This group displays information on each defined condition. To edit the condition, you must select it and then use the Selected Condition Item group controls.
Reference	(Automatically assigned) A designation for the condition.
Relation	Shows the relationship between multiple conditions.
Condition	Shows the description (data item, operator, and value) of the condition.

## Condition Editor

---

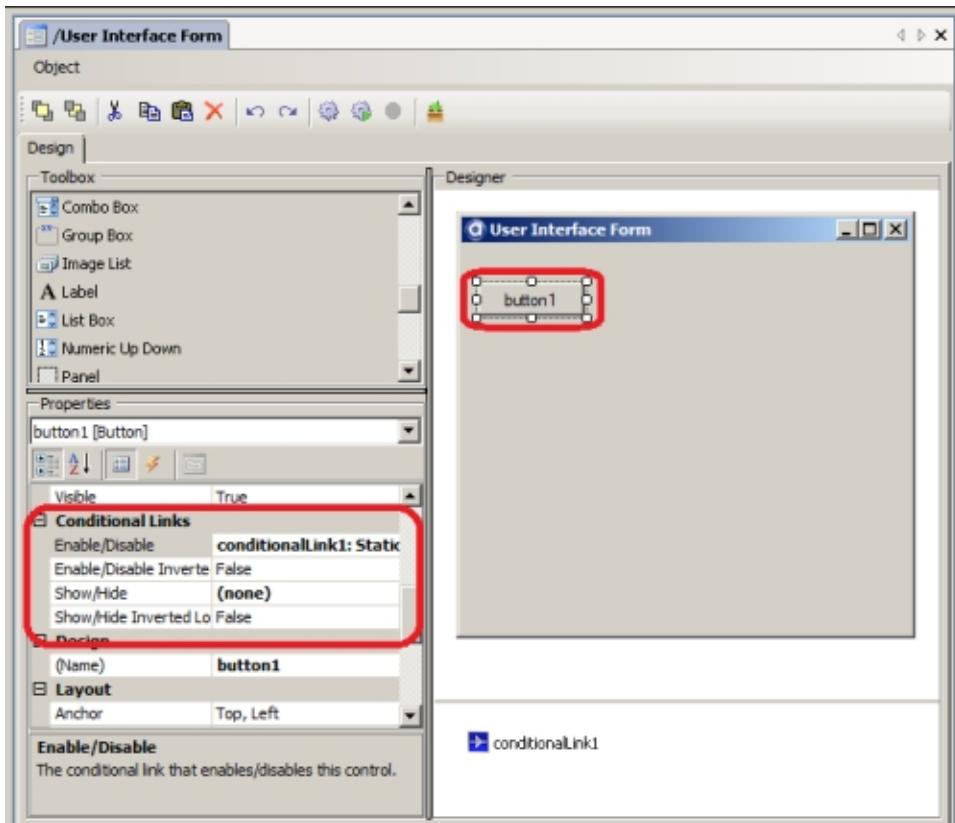
<b>Item</b>	<b>Description</b>
Item	
Current State	Shows the current state of the condition (true or false).
	Adds a new Condition Item to the list.
	Deletes the selected Condition Item from the list.
	Moves the selected Condition Item up or down in the list.
Selected Condition Item	This group of controls allows you to edit the condition selected in the Condition Items list.
Relation	When there are multiple conditions, this control allows you to specify the relation between the selected condition and the preceding condition.
( ) Bracket controls	When multiple conditions are present, these are used to place brackets in the condition statement, where needed. For example: (A Or B) And C
Data item	Shows the data item (for example, a numeric variable) that was selected for the condition. Click the browse (  ) icon to open the Data Item Selection dialog, which allows you to view and select from the available data items.   <i>Data Item Selection Dialog</i>
Operator	Specifies the operator for the condition statement:

Item	Description
	<code>==</code> equal to <code>!=</code> not equal to <code>&lt;</code> less than <code>&gt;</code> greater than <code>&lt;=</code> less than or equal to <code>&gt;=</code> greater than or equal to
Specified Value	Specifies the value for the condition statement.
Current Value	Shows the current value of the data item.
Bit Mask	<p>Specifies the bit mask that will be applied to the data. Click the browse () icon to open the Edit Bit Mask dialog, which is used to select one or more bits for the bit mask. You can use the Toggle button to select/deselect all bits.</p>  <p>The dialog shows four rows of eight checkboxes each, labeled Bits 1-8, Bits 9-16, Bits 17-24, and Bits 25-32. Below the grid are two buttons: 'Toggle' and 'Accept'.</p> <p><i>Edit Bit Mask Dialog</i></p> <p>The Bit Mask is primarily used for I/O (digital and soft) signals. It is used to select the bits of the data you want to filter. For example, selecting bits 1, 3, and 5 will filter bits 1, 3 and 5 (1-based). Bit Mask is based on 32 bit values.</p>

## Applying a Condition to a Tool

After you create a condition, you need to do something with it, such as use it to control the look or action of a specified tool on the UI Builder form. The section describes how to do that.

Each tool in the UI Builder toolbox has a Properties category called Conditional Links. The parameters in this category are used to tie (link) a conditional statement to the selected tool, and to specify a specified change in the tool state (enable/disable, show/hide) when the condition is met.



*Conditional Links Group for the Selected Button*

**NOTE:** Online help for the parameters is provided directly in the editor. Therefore, it is not duplicated in this documentation.

In the previous figure, the selected button (button1) is enabled/disabled by the condition specified in the Enable/Disable field (My Conditional). In this case, when the assigned condition is true, the button will be enabled . Note that the Show/Hide parameter does not have a condition assigned to it, so it is unaffected by the state of any condition defined in the UI Builder form.

To create the example condition:

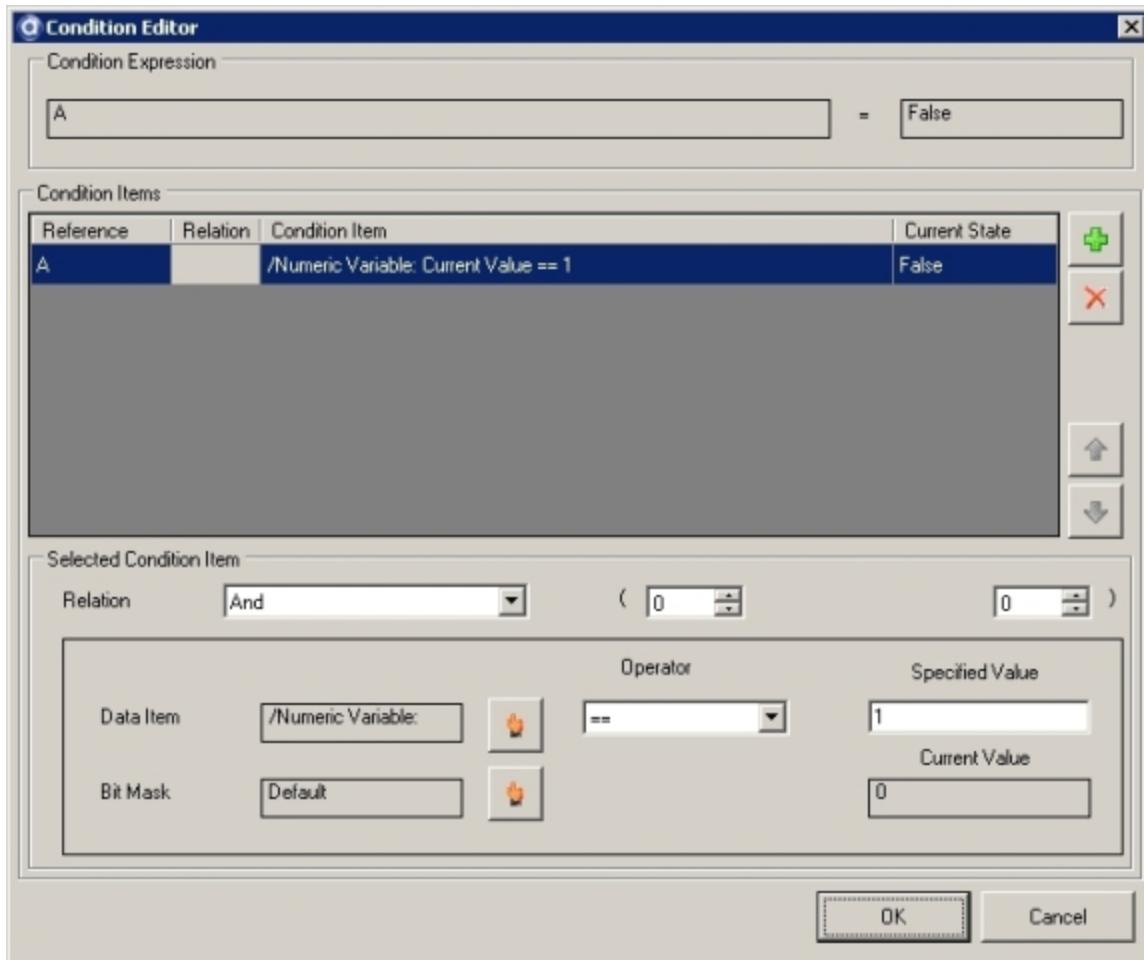
1. In the ACE Workspace Explorer, create a Static Numeric Variable item. You will alter the value of this variable to enable/disable the linked UI Form item. For details on creating a Static Numeric Variable, see Variable Editor on page 131.

2. In the UI Form Designer object, drag a Conditional Link item onto your UI Form. If this is the first Conditional Link item on your form, it appears as "conditionalLink1" at the bottom of the form.
3. Select the "conditionalLink1" item, at the bottom of the form.
4. Edit the Description parameter.

Condition	(Condition)
(Name)	conditionalLink1
Description	Static numeric var. condition

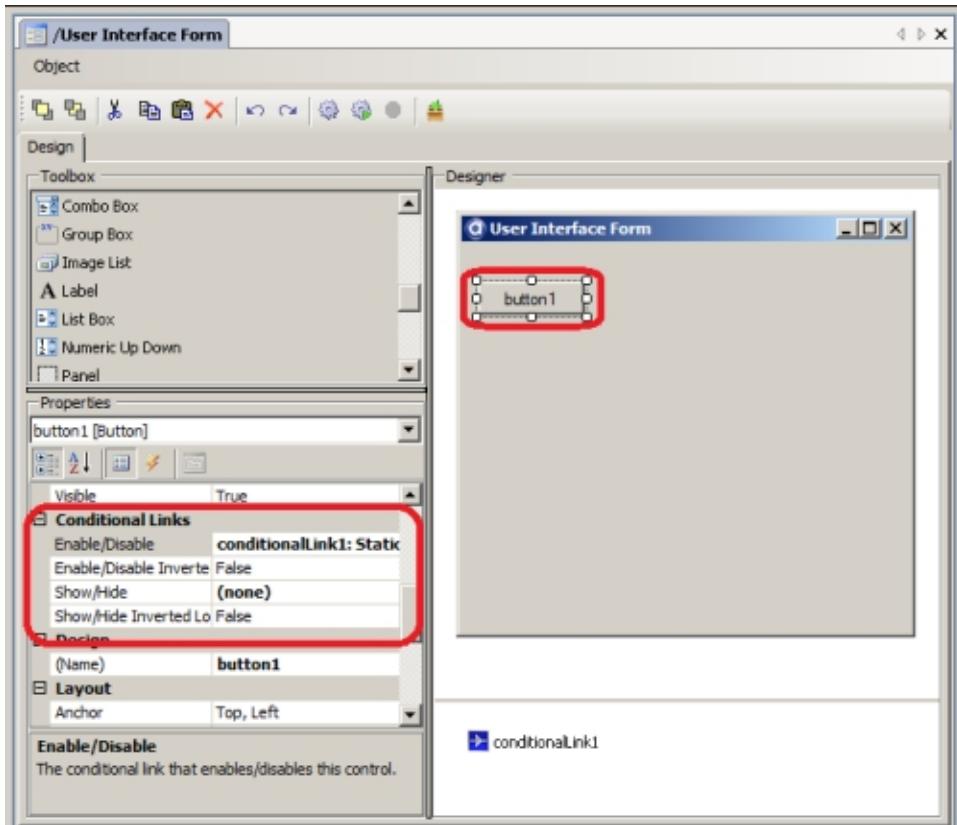
*Conditional Item Parameters*

5. Click the Browse ( ...) icon at the end of the Condition field to open the Condition Editor.
6. Click the "+" sign to add a new condition.
7. Use the controls in the Selected Condition Item group to edit the condition, as shown in the following figure.



*Condition Added to the Condition Editor*

8. Click OK to save and close the Condition Editor.
9. On the UI Form, select the button (or other item) that you want to control with a condition.
10. In the Conditions group, edit the parameters as shown in the following figure.



*Condition Links Parameters for Selected Button*

11. Click the Compile and Run ( ) icon to execute the form and test your interface. The item will be enabled and disabled, based on the state of the specified condition. You can change the condition state by altering the value of the Static Numeric Variable item in your ACE workspace.

## Using Property Links

Each connection between a data item and a UI Form Designer object (any tool, property, or other item that requires access to an ACE server) contains a set of Property Links parameters, as shown in the following figure.

Property Links	
ItemValue	/SmartController/SmartController CX: Power
ValueWhenChecked	-1
ValueWhenUnchecked	0

*Property Links Parameters*

These parameters are used to expose a data item to the UI Form Designer object, and control how the data is handled (filtered and formatted). The categories of Property Links parameters are shown in the following table.

Category	Description
ItemChecked	Links an object that can be checked (for example, a check box or radio button) to a data item.
ItemPressed	Links an object that can be pressed (for example, a button or toggle button) to a data item.
ItemText	Links an object containing text (for example, a button, label, or group box) to a data item.
ItemValue	Links an object that displays a value (for example, a progress bar or track bar) to a data item.

The number and types of parameters shown depend on the object that is currently selected.

**NOTE:** Help for the remaining parameters is provided directly in the editor. Therefore, it is not duplicated in this documentation.

### Linking the Object to a Data Item

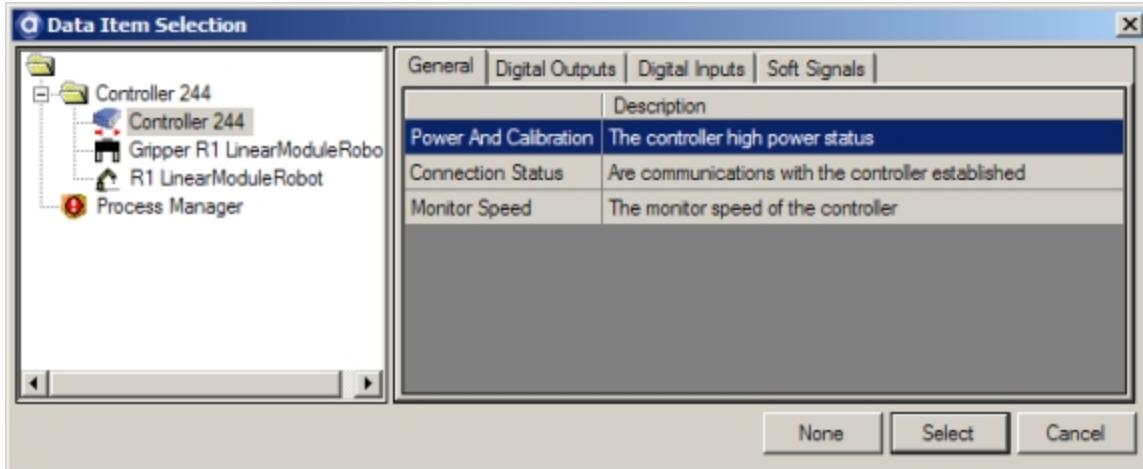
The UI Form Designer object is linked to a data item in the ACE workspace through the Property Links "ItemValue" parameter.

Property Links	
ItemValue	(none)
FormatString	...

*Property Links ItemValue Parameter*

To link a UI Form Designer object to a data item:

1. Select the object on the UI Form Designer.
2. Click the browse (...



3. Use the tabs to locate the desired data item and select it.
4. Click Select to record the selection and close the dialog.

**NOTE:** If the name of the linked data item is changed, or if the linked data item is moved to a different folder, the link in the DataItem parameter is automatically updated. If the linked item is deleted from the workspace, the ItemValue parameter is reset to "(none)".

## Using the Formatting Parameters

After you have connected (linked) the data item, you may need to specify the data control through the formatting parameters.

**NOTE:** The formatting parameters are primarily used for certain types of exposed data items, such as I/O signals (for ValueWhenPressed and ValueWhenUnpressed) and string formatting of the data (for FormatString). For more details, see the following information.

The following table shows the common formatting parameters that are used for data control.

**NOTE:** Help for the remaining parameters is provided directly in the editor. Therefore, it is not duplicated in this documentation.

Parameter	Description
FormatString	Used for string formatting of the data (such as number of significant digits, special formatting, hexadecimal display, etc.).

Parameter	Description
	The FormatString parameter uses standard Microsoft string formatting, which is described at the following links: <a href="http://msdn.microsoft.com/en-us/library/az4se3k1.aspx">http://msdn.microsoft.com/en-us/library/az4se3k1.aspx</a> <a href="http://msdn.microsoft.com/en-us/library/dwhawy9k.aspx">http://msdn.microsoft.com/en-us/library/dwhawy9k.aspx</a> <a href="http://msdn.microsoft.com/en-us/library/txafckwd.aspx">http://msdn.microsoft.com/en-us/library/txafckwd.aspx</a>
ImageList	(Picture Box tool only) Specifies an ImageList that displays images on the tool. The ImageList tool must be added to the UI form before it can be selected. For an example of using this, see Creating an Image List on page 1058.
ValueWhenChecked ValueWhenUnchecked ValueWhenPressed ValueWhenUnpressed	Specifies the value that will be written to the linked data item when the object is in the specified state (for example, a button is pressed or not, a radio button is selected or not, a checkbox is checked or not). Type the desired value into the field.  The default value for false (unpressed, unchecked) is 0; the default value for true (pressed, checked) is -1. However, these can be changed: <ul style="list-style-type: none"> <li>• Use the opposite conditions (False instead of True; True instead of False), or</li> <li>• If the data item is a string, you can write a string, such as "Pressed" (or "Not Pressed"), "Pushed" (or "Not Pushed"), or "Enabled" (or "Not Enabled"), to the data item. Simply type the desired string into the field.</li> </ul>

### Examples

The FormatString parameter can be used for custom string formatting of the data (such as number of significant digits, special formatting, hexadecimal display, etc.). For example:

- If you create a Text Box tool with numeric data, specify FormatString: E03, and the data is 2, it will be displayed as: 2,000E+000 .
- If you create a Text Box tool with numeric data, specify FormatString: plus 000;minus 000. If the data is 2, it will be displayed as: plus 002; if the data is -2, it will be displayed as: minus 002.
- For more information on string formatting, see the information on the Microsoft website:
  - <http://msdn.microsoft.com/en-us/library/az4se3k1.aspx>
  - <http://msdn.microsoft.com/en-us/library/dwhawy9k.aspx>
  - <http://msdn.microsoft.com/en-us/library/txafckwd.aspx>

The ValueWhenPressed / ValueWhenUnpressed and ValueWhenChecked / ValueWhenUnchecked parameters are used for specifying a value that will be written to the linked data item (for details on linking a data item, see Linking the Object to a Data Item on page 1039). For example:

- If you're using a button, the ValueWhenPressed value will be written to the data item when the button is pressed (clicked).
- If you're using a check box, you can set the True and False values through the ValueWhenChecked and ValueWhenUnchecked parameters. When the box is checked (selected), the data item will be set to the "ValueWhenChecked" value; when it is not checked or selected, the data item will be set to the "ValueWhenUnchecked" value.
- If you're using a group of radio buttons, you will need the ValueWhenChecked parameter to select one value among multiple. For example, if you have a numeric data item that can have the following values: 1, 2, and 3, you can place three radio buttons in a group box, link each of them to the data item, and then set the ValueWhenChecked of the first to 1, the ValueWhenChecked of the second to 2, and the ValueWhenChecked of the third to 3.

## Using the User Access Controls

Each tool in the UI Form Designer contains a set of User Manager properties. These are used to control the access and visibility of the selected tool, based on the user login level. For example, you may have an advanced control that you want to hide from an Operator but show for an Engineer. You can do that with the User Manager properties.

**NOTE:** When a parent form is accessed with a specific user level, that user-level is also applied to any child-forms that are opened from that parent form.

In addition to the User Manager properties, there are two buttons, called Sign In Button and Sign Out Button, that are used to provide login and logout features for your user interface. For more details on user login levels, see User Manager Editor on page 125.

**NOTE:** The access level applied to the User Interface, when it loads, is defined in the User Manager object of your workspace. For more details on the User Manager object, see User Manager Editor on page 125.

When the UI form is executed, any objects in the UI that are set for a higher access level will require the user to sign in before they can be accessed.

User Manager	
AccessLevel	Operator
AccessVisibility	Disabled

### User Manager Parameters

These parameters are used to control the access level and access visibility of the selected item:

- Access Level specifies the minimum-required level for the selected item
- Access Visibility specifies how the item will appear when the specified access level is not met. Note that when the Ignored option is selected, the item is visible and can be used by *any* AccessLevel login.

**NOTE:** If the user is logged in with at a level that exceeds the specified level, the user will have full access and visibility for the selected item.

## Applying the User Manager Parameters

As described previously, the user manager parameters allow you to show/hide or enable/disable a user interface item based on the user login level. The following are several examples of how this can be applied.

### Applying User Manager Parameters to a Single Item

This example shows how to hide a button (or any simple item in the user interface) for an Operator login.

1. In the UI Form Designer, drag the Sign In and Sign Out button onto your UI Form. This will make easier to test the access controls when you run your form.
2. Select a button or other item that you wish to apply access controls to.
3. In the Properties area, scroll down to the User Manager group.
4. Click the AccessLevel parameter and select Operator.
5. Click the AccessVisibility parameter and select Hidden.
6. Click the Compile and Run ( ) icon to execute the form and test your interface. The item will be enabled and disabled, based on the login access level you select with the Sign In button.

#### ***Applying User Manager Parameters to a Group of Items***

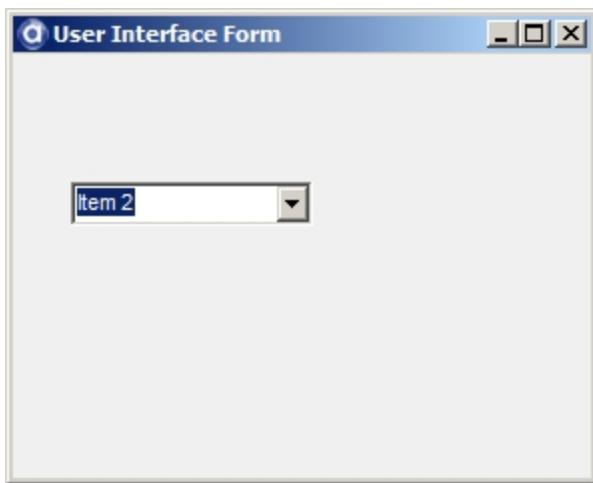
This example shows how to hide a group of items in for an Operator login.

1. In the UI Form Designer, drag the Sign In and Sign Out button onto your UI Form. This will make easier to test the access controls when you run your form.
2. Select the first item in the group (for example, a button or other item).
3. Press and hold the Ctrl (Control) key, and simultaneously select the remaining items in the group.
4. In the Properties area, scroll down to the User Manager group.
5. Click the AccessLevel parameter and select Operator.
6. Click the AccessVisibility parameter and select Hidden.
7. Click the Compile and Run ( ) icon to execute the form and test your interface. The item will be enabled and disabled, based on the login access level you select with the Sign In button.

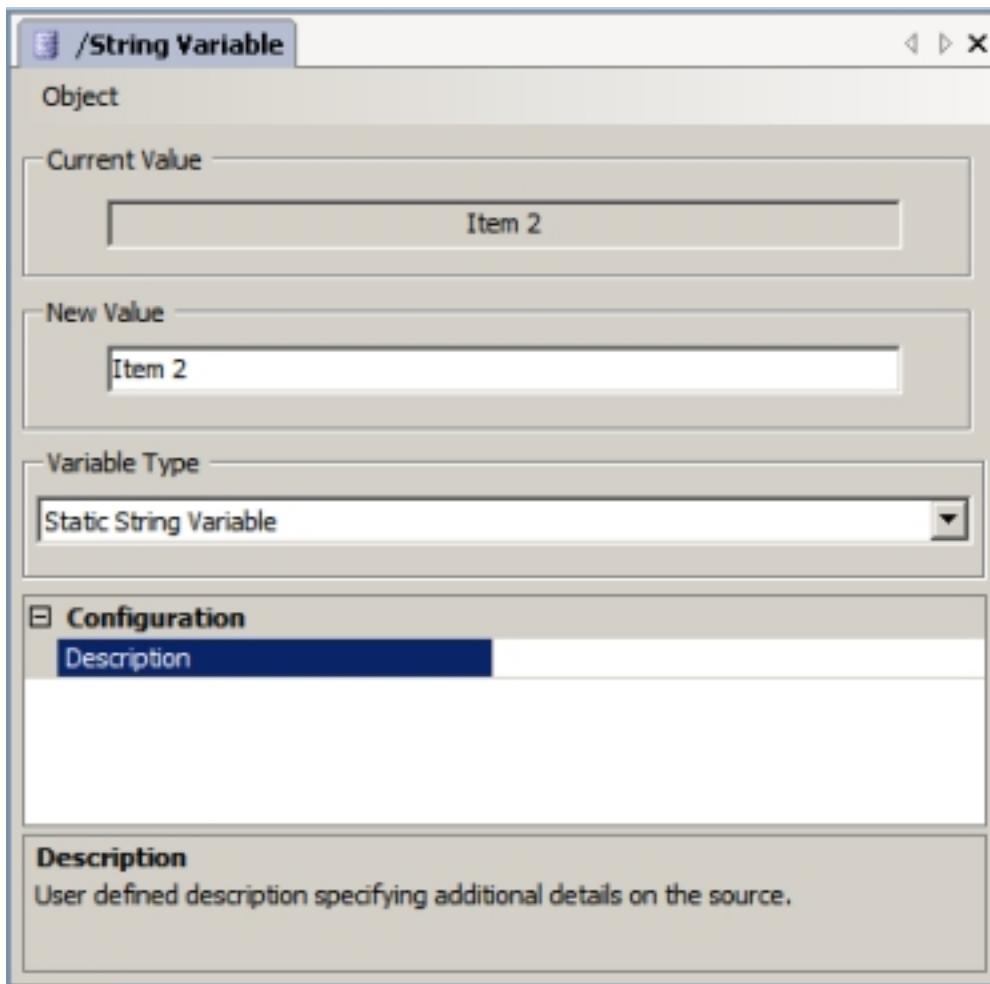
## Creating a Combo Box or List Box

A Combo Box or List Box tool can be used to present predefined choices (a list) to the user. When the user selects a value from the list, the selection is passed to the connected item specified in the Property Links group. For example, a Combo Box could contain a list of string values and could be connected to a String Variable object. When the user makes a selection from the List Box, that value (string) is displayed in the String Variable object.

The main difference between the Combo Box and List Box is that the Combo Box allows the user to type a value instead of selecting it from the list (note that the typed value must be one of the values on the list). Otherwise, the setup and operation of both tools is identical.



User Interface Form with Combo Box



*Combo Box selection updated in String Variable object*

## Using a List Box

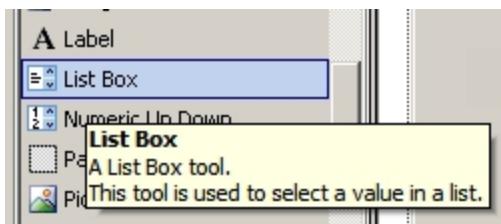
**NOTE:** The procedure for creating a Combo Box is identical to the following List Box procedure.

The following example describes how to:

- Create the List Box,
- Add values to the list,
- Add a String Variable object and link it to the List Box, and then
- Execute the UI form and test the List Box.

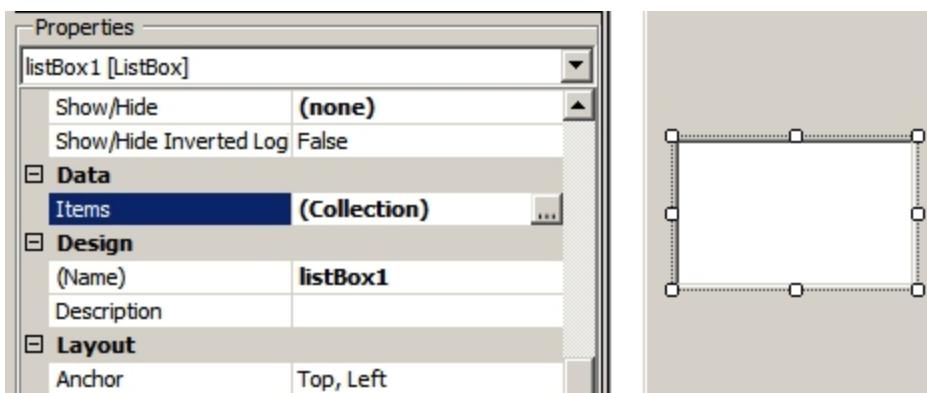
### Create the List Box

1. Add a List Box tool to your UI Builder form.



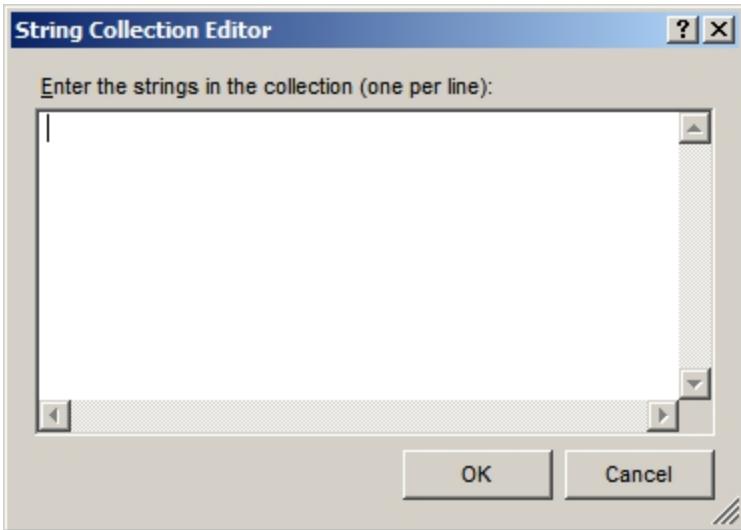
*Adding the List Box Tool*

2. Select the List Box tool, so you can edit its properties.



*Properties for List Box*

3. In the Properties editor, locate the Data group.
4. Use the browse (... ) icon to open the String Collection Editor.



*String Collection Editor*

5. Use the String Collection Editor to add values to the list.

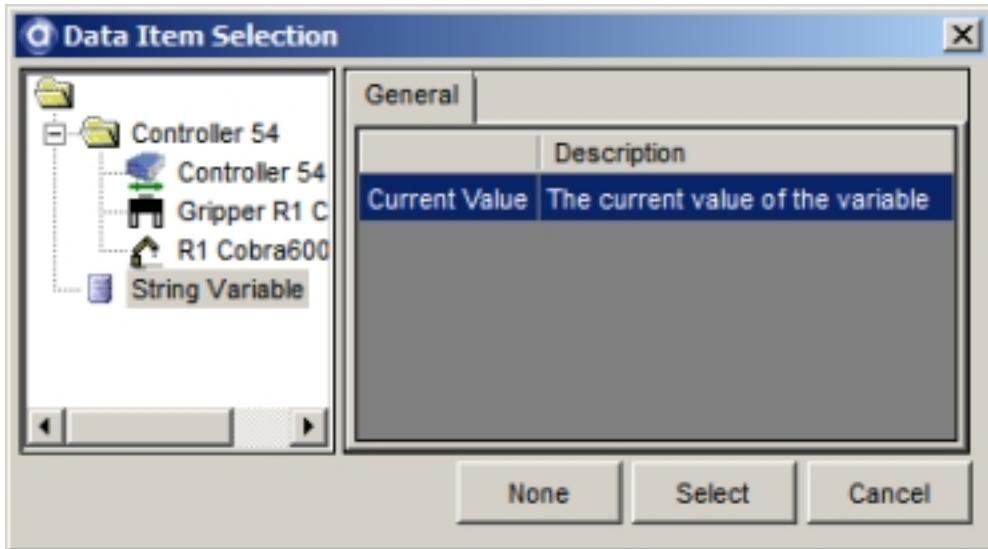


*String Collection Editor with Values Added*

#### **Add a String Variable and Link it to the List Box**

1. Add a String Variable object to your ACE workspace. For details, see Variable Editor on page 131.
2. Select the List Box on the UI form.

3. In the Properties editor, select **Property Links > Value**, and use the browse (...) icon select the String Variable, as shown in the following figure.



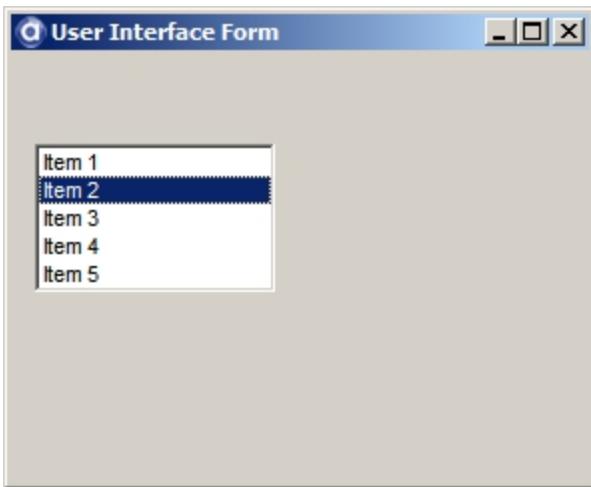
*Selecting the String Variable as the Property Links Item*

#### **Execute the UI Form and Test the List Box**

1. Click the Compile and Run ( ) icon to execute the form and test the List Box.
2. Use the List Box tool to select a value. As the value is changed, the value of the String Variable is updated to the selected value. For example, if the data item is "Item 2", that value is displayed; if the data "Item 3", that value is displayed, and so on.

## Using a List Box

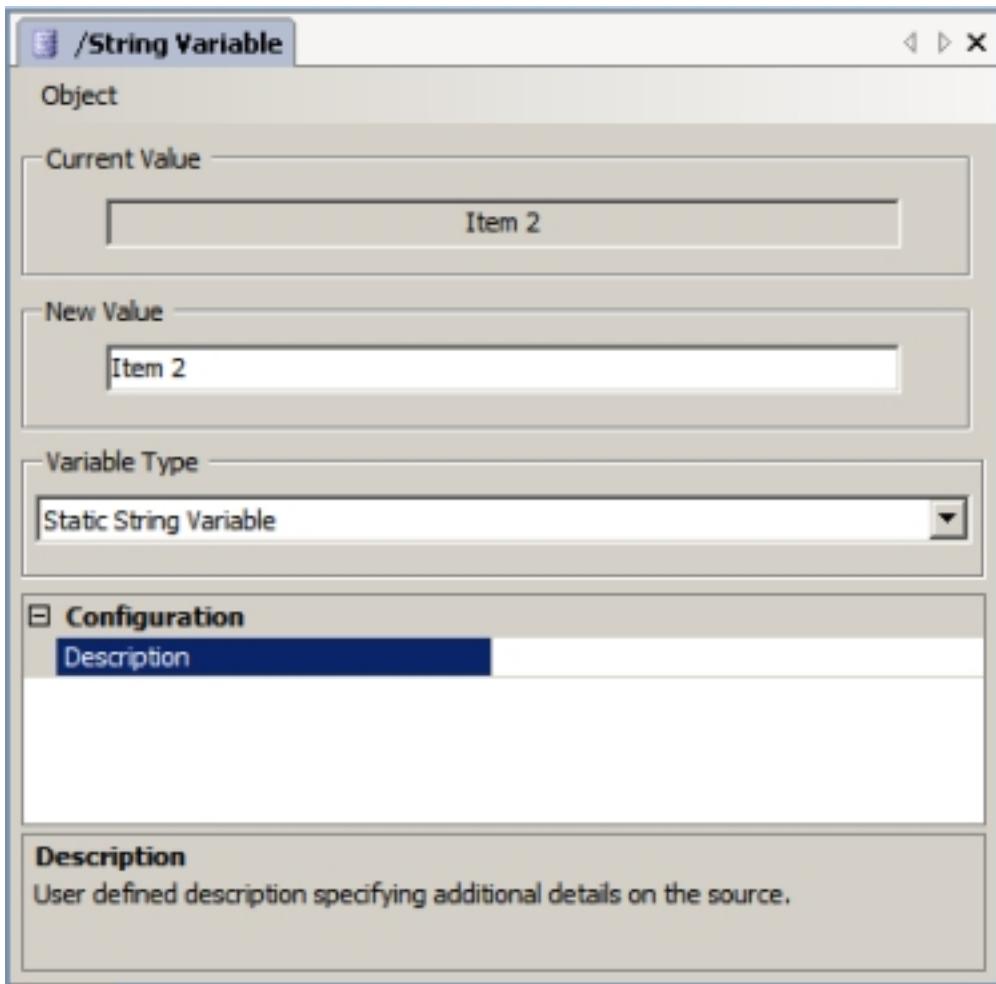
---



*User Interface Form with List Box item selected*

## Using a List Box

---

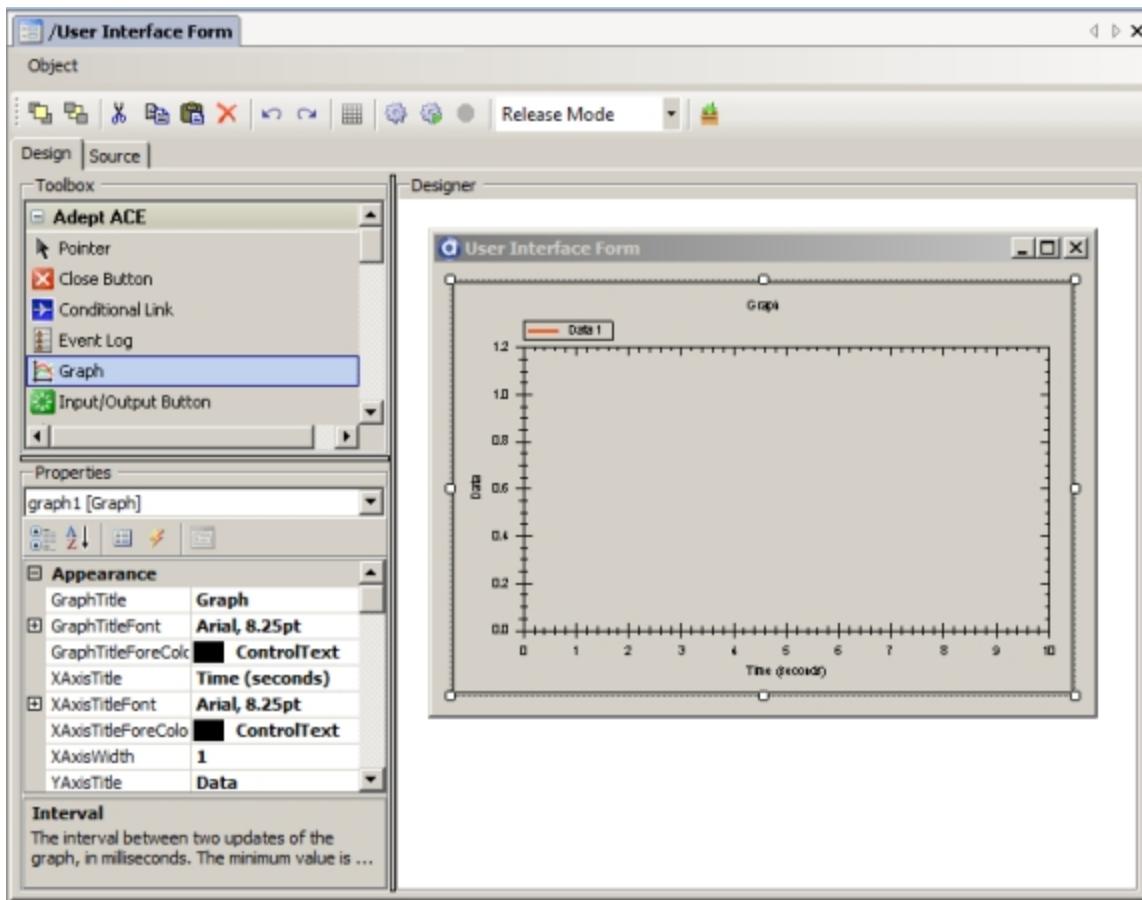


*List Box selection updated in String Variable object*

## Creating a Graph

The User Interface Builder contains a special tool, called a Graph tool, which allows the user to create and display various types of graphs. Up to eight data items can be shown on the graph. This can be useful in production environments, because it provides a quick, visual history of a process.

To access the Graph tool, drag a Graph tool onto the form designer.



*The Graph Tool*

There are two main Properties groups that are used to set-up and format the graph: the Appearance group and Graph group. Each group is described in the following sections.

### The Appearance Group

The Appearance group contains properties that are specific to the Graph tool.

<b>Appearance</b>	
GraphTitle	<b>Graph</b>
+ GraphTitleFont	<b>Arial, 8.25pt</b>
GraphTitleForeColor	<b>ControlText</b>
XAxisTitle	<b>Time (seconds)</b>
+ XAxisTitleFont	<b>Arial, 8.25pt</b>
XAxisTitleForeColor	<b>ControlText</b>
XAxisWidth	<b>1</b>
YAxisTitle	<b>Data</b>
+ YAxisTitleFont	<b>Arial, 8.25pt</b>
YAxisTitleForeColor	<b>ControlText</b>
YAxisWidth	<b>1</b>
ChartBackColor	<input type="color"/>
ChartBorderColor	<b>Black</b>
ChartBorderWidth	<b>1</b>
LegendBackColor	<input type="color"/>
+ LegendFont	<b>Arial, 8.25pt</b>
LegendForeColor	<b>ControlText</b>
LegendBorderWidth	<b>1</b>
BackColor	<input type="color"/> <b>Control</b>

*Graph Appearance Parameters Example*

**NOTE:** Online help for the parameters is provided directly in the editor. Therefore, it is not duplicated in this documentation.

For details on general Appearance group properties, see Using the Appearance, Behavior and Layout Controls on page 1025.

## The Graph Group

The Graph group contains properties that control the data collection used for the graph.

<b>Graph</b>	
+ Data Collection	8-item collection
Duration	10
<b>Interval</b>	<b>100</b>
YAxisManualScale	False
YAxisMin	<b>0</b>
YAxisMax	<b>1</b>

*Graph Group Parameters Example*

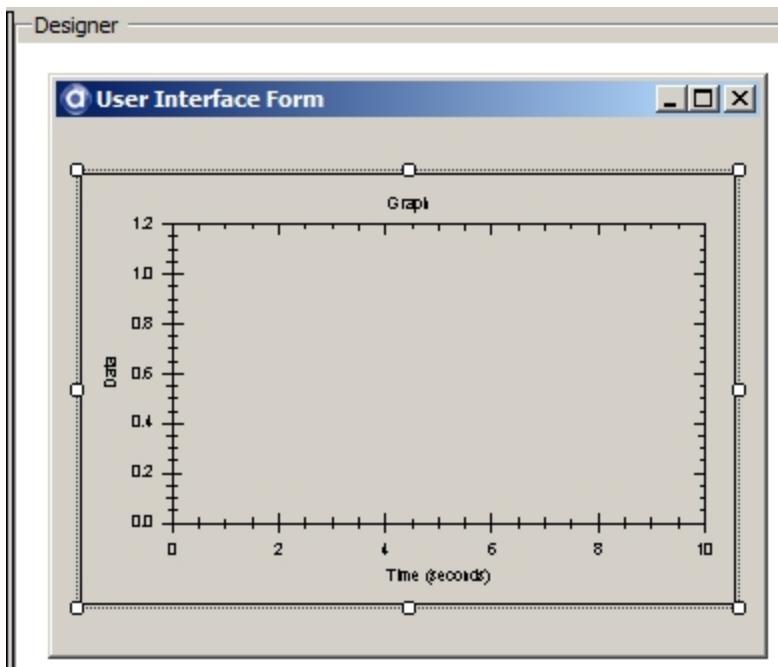
**NOTE:** Online help for the parameters is provided directly in the editor. Therefore, it is not duplicated in this documentation.

## Using the Graph Tool

This provides an example of how to use the Graph tool.

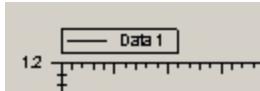
### Create the Sample Graph

1. Drag a Graph tool onto the form designer. An empty, default-formatted graph is displayed in the form designer.
2. Use the handles to resize the graph to the desired width and height.



*Graph Tool Placed and Resized in the Form Designer*

3. In the Graph properties group, expand the Data Collection > Item 1, and select the ItemValue property.
4. Click the ItemValue field to select a data item. For this example, use Controller > Monitor Speed. Click Select to save the selection. Notice that a data item is displayed in the legend area above the graph.



*Legend Area*

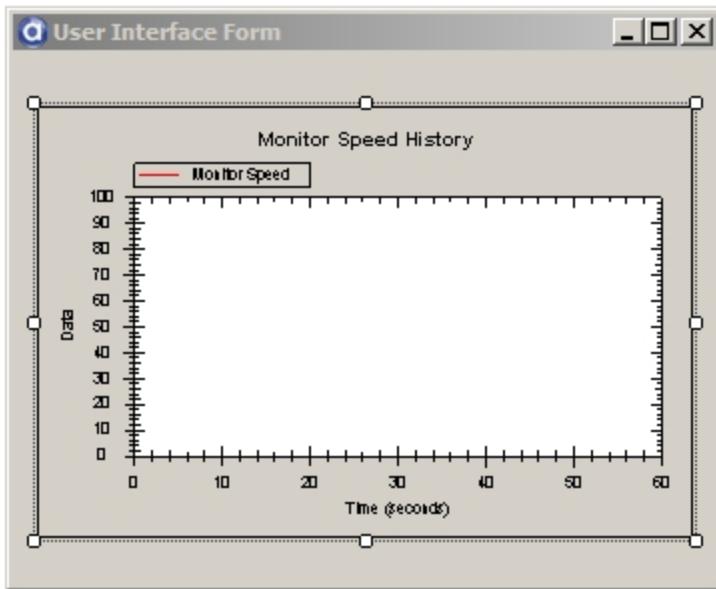
5. Set the remaining Data item properties, as shown in the following table.

Property	Setting
DataName	Monitor Speed
DataColor	Web > Red
DataWidth	2
Duration	60
Interval	1000
YAxisManualScale	True
YAxisMin	0
YAxisMax	100

6. In the Appearance properties group, set the following properties as shown in the table.

Property	Setting
GraphTitle	Monitor Speed History
GraphTitleFont	
Size	10
Unit	Point
Bold	True
ChartBackColor	Web > White

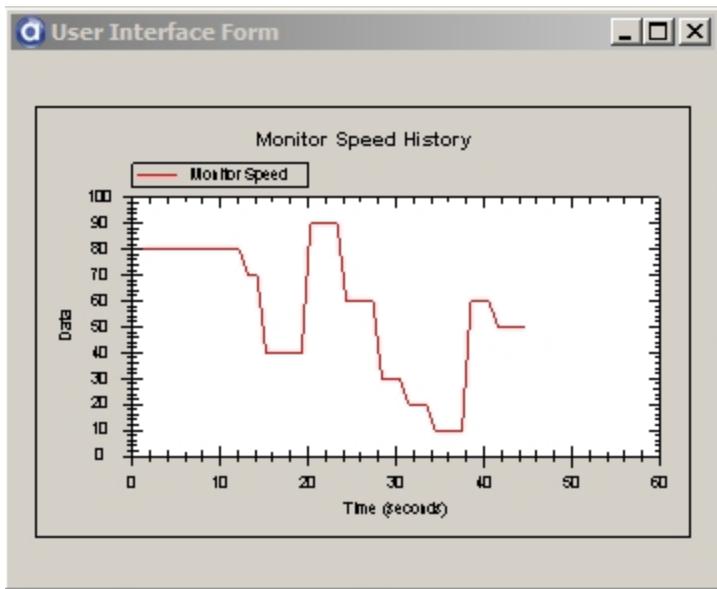
7. The completed graph should look similar to the one in the following figure.



*Completed Graph*

#### **Graph the Data**

1. On the UI Builder toolbar, click the Compile and Run (  ) icon to launch the sample user interface.
2. Move the interface so that you have access to the monitor speed control on the ACE toolbar. For details, see Toolbar - General on page 157.
3. Adjust the monitor speed up and down, and watch how the monitor speed is sampled and plotted on the graph.



*Monitor Speed Sampled and Plotted on Graph*

You can zoom (magnify) an area of the graph by dragging the mouse pointer to define the zoom area.

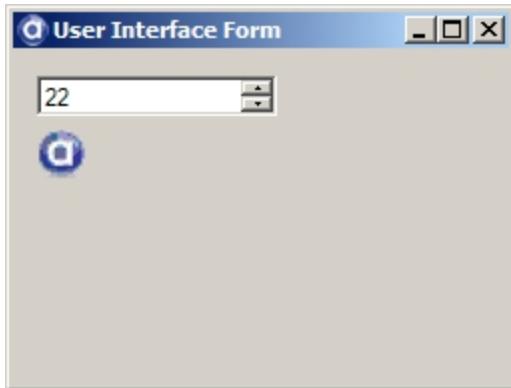
You can pan the graph by holding the Ctrl key while dragging the mouse pointer.

You can right-click on the graph to display a menu of options, which are described in the following table.

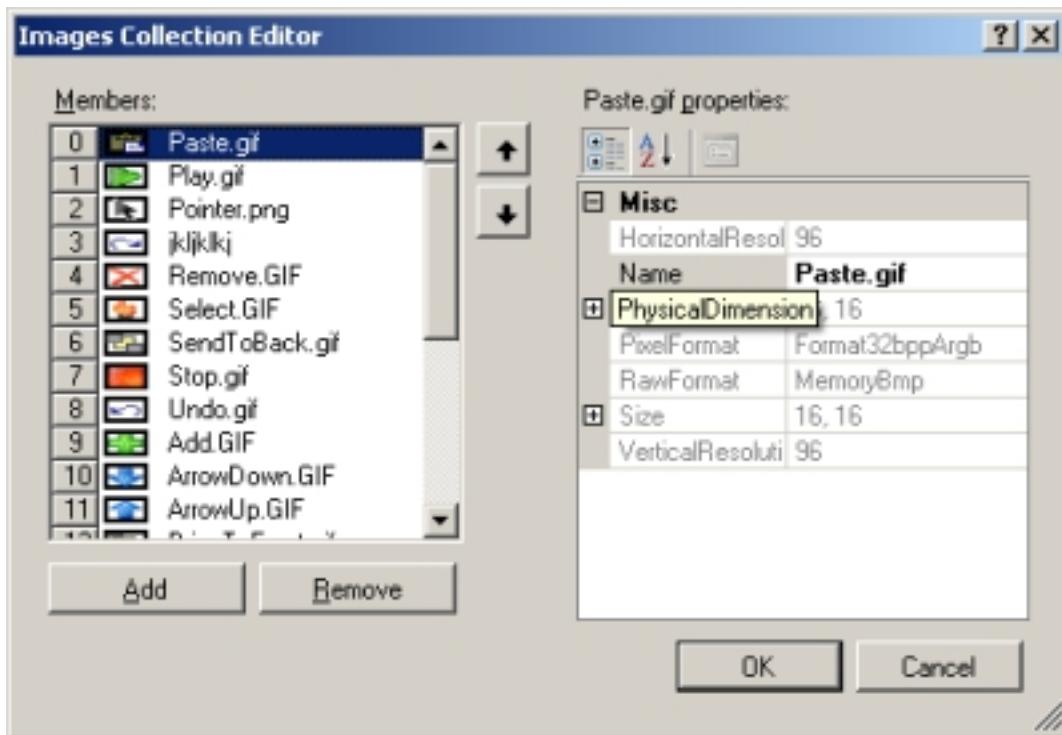
Item	Description
Copy	Copy the image in the graph area.
Save Image As...	Save the graph image as a file.
Page Setup...	Opens the Page Setup dialog box for setting the print page size, orientation, and margins.
Print...	Opens the Print dialog box for printing the graph.
Show Point Values	When selected, place the cursor at a point on the graph to show the values of that point.
Un-Zoom	Returns the graph to its original magnification.
Undo All Zoom/Pan	Returns the graph to its original magnification and origin.
Set Scale to Default	Returns the graph scale to its default setting.

## Creating an Image List

An Image List tool is used to create a set of images that are displayed by another tool when the linked data item changes value. For every possible value of the data item, you would place a corresponding image onto the Image List. For example, an Image List that is linked to a Picture Box and Numeric Up/Down control can be used to display a different image in the Picture Box when the value is changed by the Numeric Up/Down control.



*User Interface Form with Image List Controlled by Numeric Up/Down Control*



*Example Image List in the Images Collection Editor*

## Using an Image List

The following example describes how to:

- Create the Image List,
- Add a data item and link a control,
- Add a Picture Box for the images and link it to the data item, and then
- Execute the UI form and test the image list.

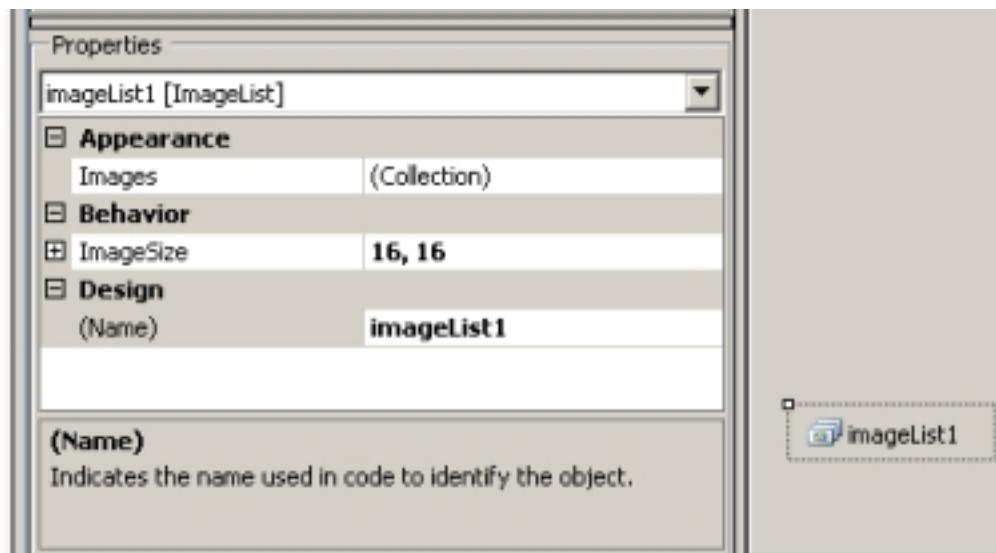
### Create the Image List

1. Add an Image List tool to your UI Builder form.



*Adding the Image List Tool*

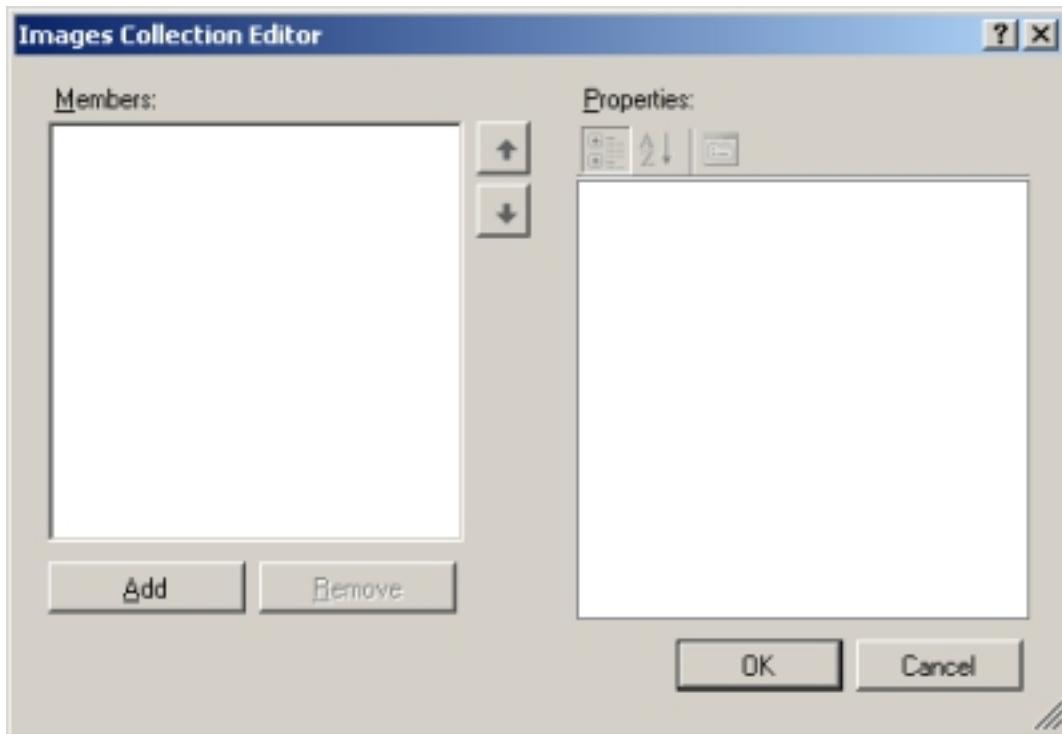
2. Select the Image List tool, so you can edit its properties.



*Properties for Selected Image List*

3. In the Properties editor, locate the Appearance group.
4. Set the Images property to "(Collection)". Use the browse (...

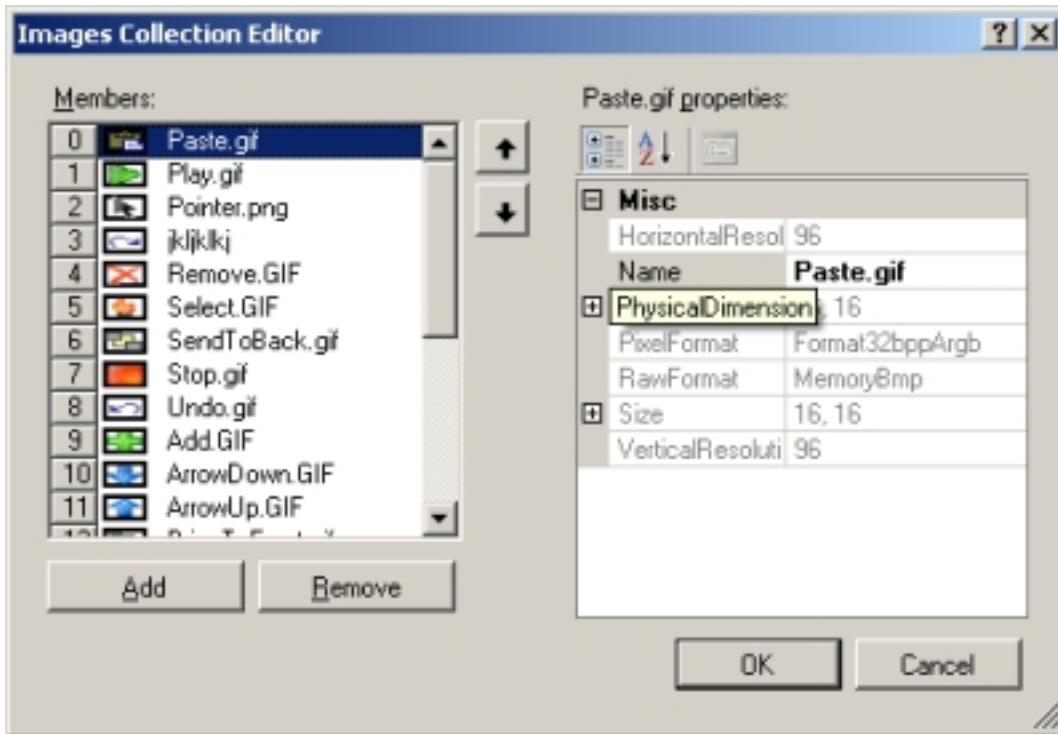
icon to open the Images Collection Editor.



*Images Collection Editor*

5. Use the Images Collection Editor to add images to the list.

**NOTE:** In the following figure, the number to the left of each image is the index value that will be used to display that image. For example, 0 displays Paste.gif; 1 displays Play.gif; 2 displays Pointer.png; etc.

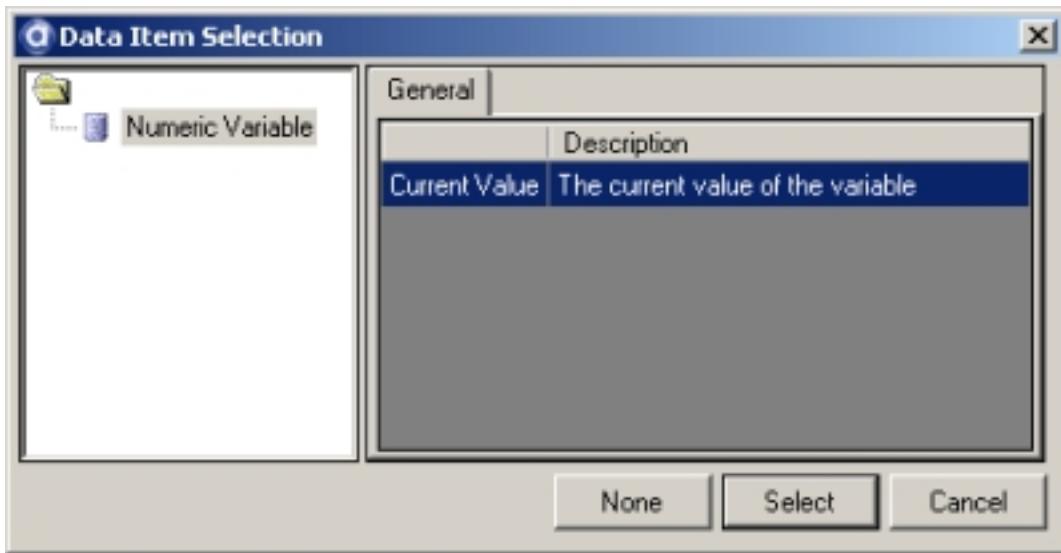


*Images Collection Editor with Images Added*

- a. Use the Add and Remove buttons to add or remove images.
- b. Use the Name property to rename the image, as needed, for your list. This changes the display name, not the file name.
- c. Click **OK** to save the list and close the editor.

#### **Add a Data Item and Control**

1. Add a Static Numeric Variable to your ACE workspace. For details, see Variable Editor on page 131.
2. Add a Numeric Up/Down control to your UI Form.
3. In the Properties editor, select **Property Links > Value**, and use the browse (...) icon to select the Numeric Variable, as shown in the following figure.



*Selecting the Numeric Variable as the Property Links Item*

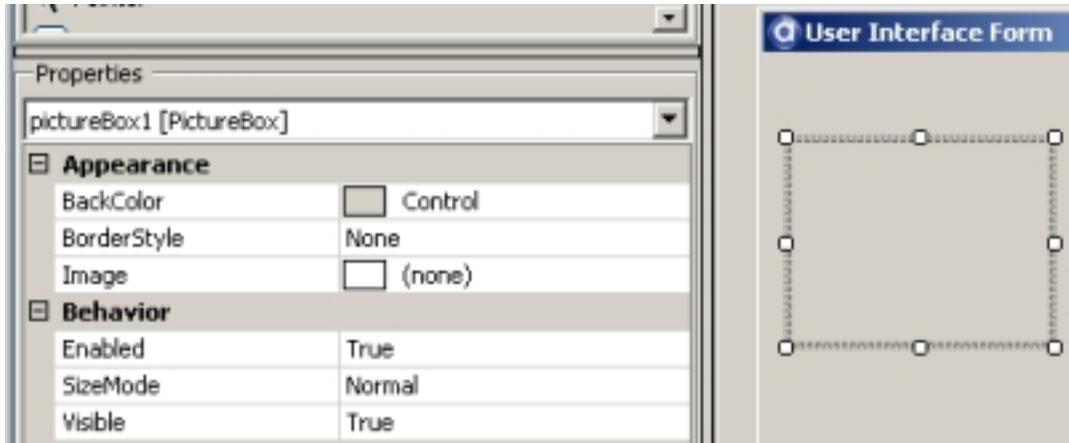
**Add a Picture Box for the Images and Link it to the Data Item**

1. Add a Picture Box tool to your UI form.



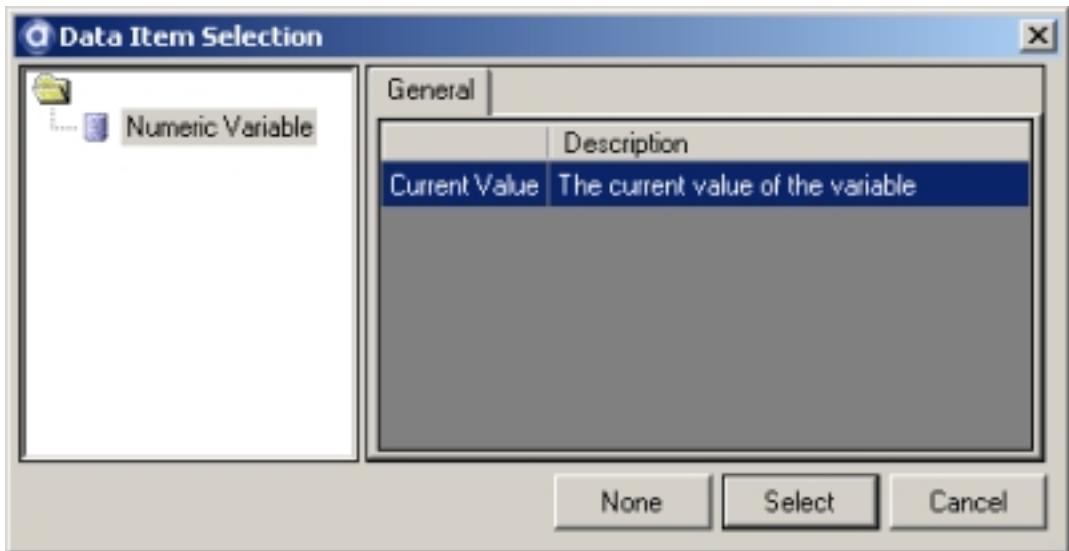
*Adding a Picture Box Tool*

2. Select the Picture Box tool, so you can edit its properties.



*Properties for Selected Picture Box*

3. In the Picture Box Properties editor, select **Property Links > Image**, and use the browse (...) icon select the Numeric Variable, as shown in the following figure.



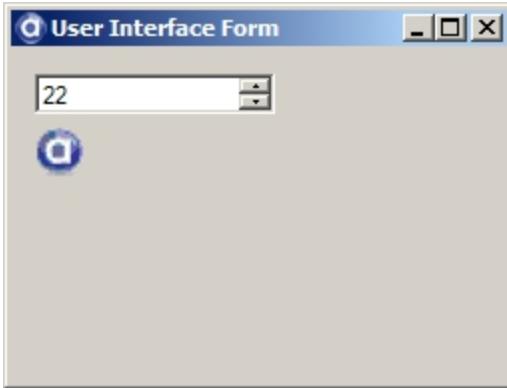
*Selecting the Numeric Variable as the Property Links Item*

4. In the Picture Box Properties editor, select **Property Links > Image > ImageList**, and use the selector to assign the previously-created Image List to the Picture Box.



**Execute the UI Form and Test the Image List**

1. Click the Compile and Run (  ) icon to execute the form and test the Image List.
2. Use the Numeric Up/Down control to select an image. As the Numeric Up/Down (the linked data item) value is changed, the value of the Numeric Variable is change, which changes the displayed image. For example, if the data item is 0, image 0 is displayed; if the data item is 1, image 1 is displayed, and so on.



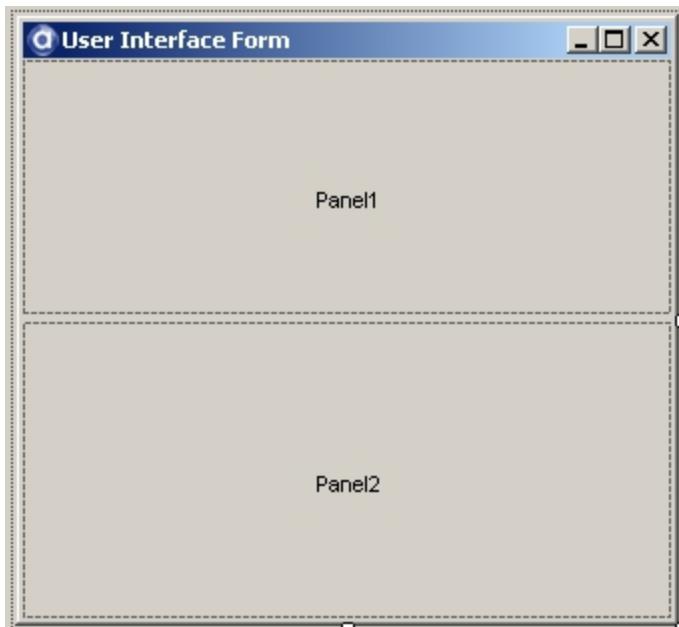
*User Interface Form with Image List Controlled by Numeric Up/Down Control*

## Creating a Split Form

A split form contains a special tool, called a Split Container, which allows the user to resize the split areas of the form. The resizing is done by dragging a bar up/down (for a horizontal split) or left/right (for a vertical split).

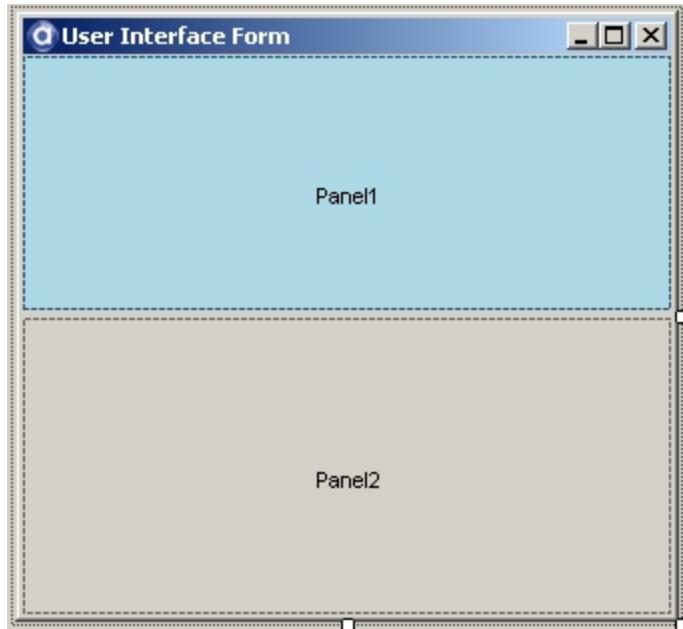
To create a horizontal-split form:

1. Create a User Interface Form object in your ACE workspace. For details, see Creating a Split Form on page 1066.
2. Drag a Split Container tool onto the form designer.
  - a. In the Properties items, set the Orientation to "Horizontal", and set the Dock property to "Fill".
  - b. Size and position the top panel for the upper half of the form.
  - c. The completed Split Container should looks similar to the following figure.



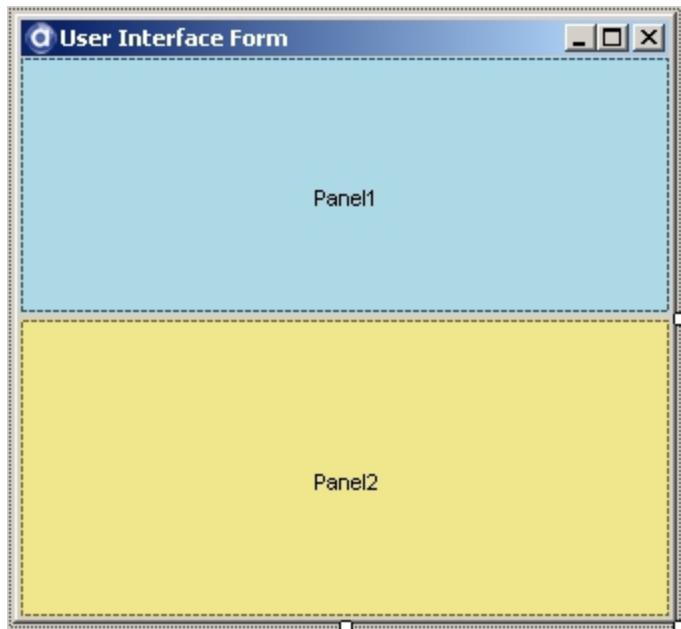
*Horizontal-Split Container*

3. Click in the Panel1 area to select it.
  - a. In the Properties items, set the BackColor property to Web > LightBlue.
  - b. The completed Group Box should look similar to the following figure.



*Panel1 Edit for Split UI Form*

4. Click in the Panel2 area to select it.
  - a. In the Properties items, set the BackColor property to Web > Khaki.
  - b. The completed Group Box should look similar to the following figure.

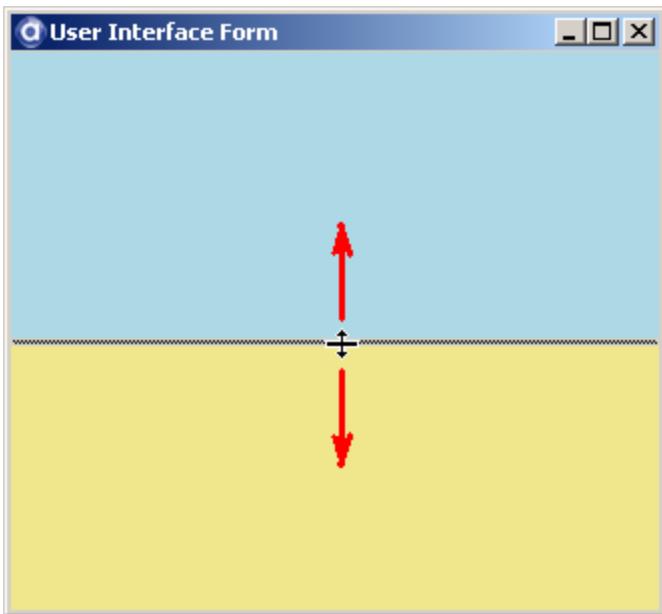


*Panel2 Edit for Split UI Form*

5. Click the Compile and Run ( ) icon to execute the form and test the Split Container control. The following is an example of the previous form in executed mode.

## Creating a Split Form

---



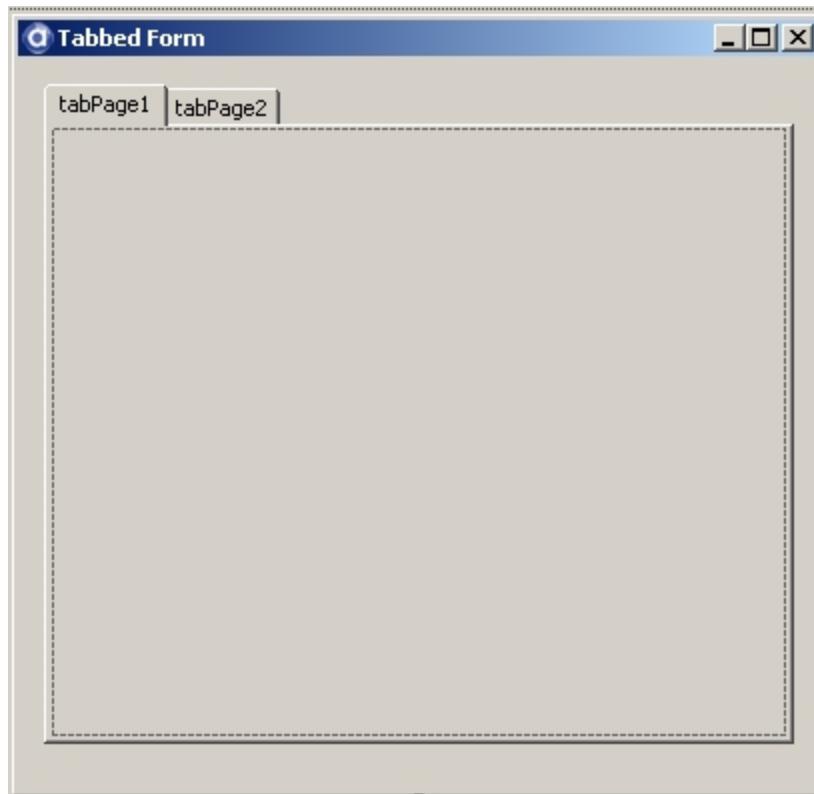
*Split UI Form in Executed Mode (arrows added to show movement of Split Container bar)*

## Creating a Tabbed Form

A tabbed form contains a special tool, called a Tab Control, which allows you to create multiple-page forms. The user can then select the desired page of the form by clicking the corresponding tab.

To create a tabbed form:

1. Create a User Interface Form object in your ACE workspace. For details, see Creating a Tabbed Form on page 1070.
2. Drag a Tab Control tool onto the form designer.
  - a. Size the Tab Control tool for your form.
  - b. The completed Tab Control should look similar to the following figure.



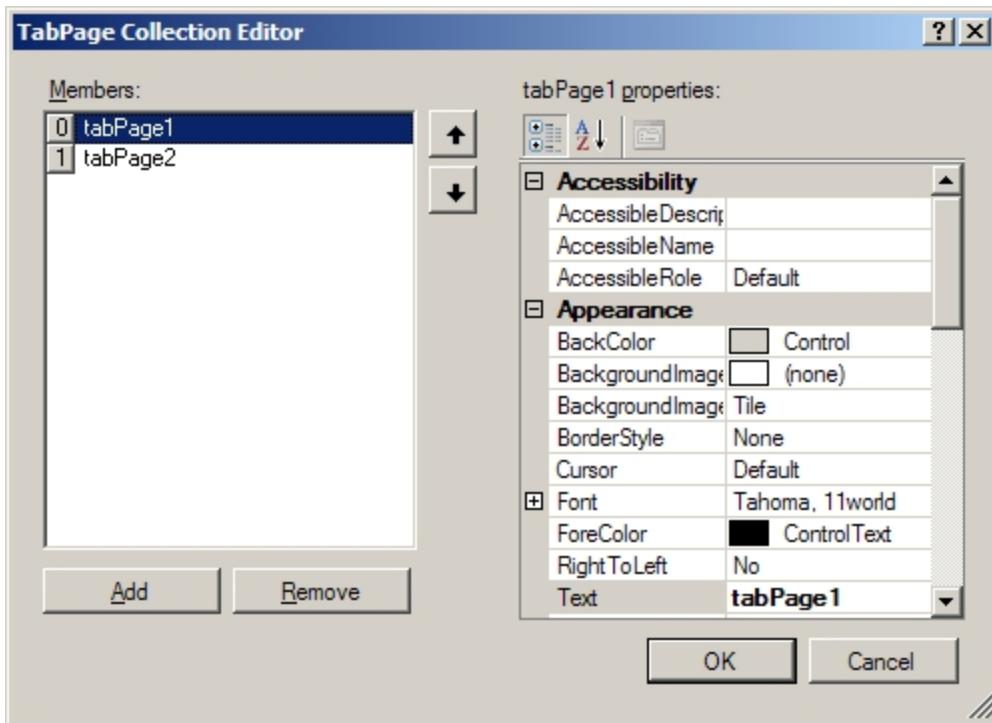
*Tab Control for Multi-page UI Form*

**NOTE:** You can change the name of a tab by selecting it and then editing the Text property in the Appearance properties group.

3. In the Properties group, select Behavior > TabPages and click the Browse (...) icon next to

"Collection". The TabPage Collection Editor opens.

**NOTE:** In the following figure, the number to the left of each tab name is the index value that will be used to display that tab page. For example, 0 displays tabPage1 and 1 displays tabPage2.



*TabPage Collection Editor*

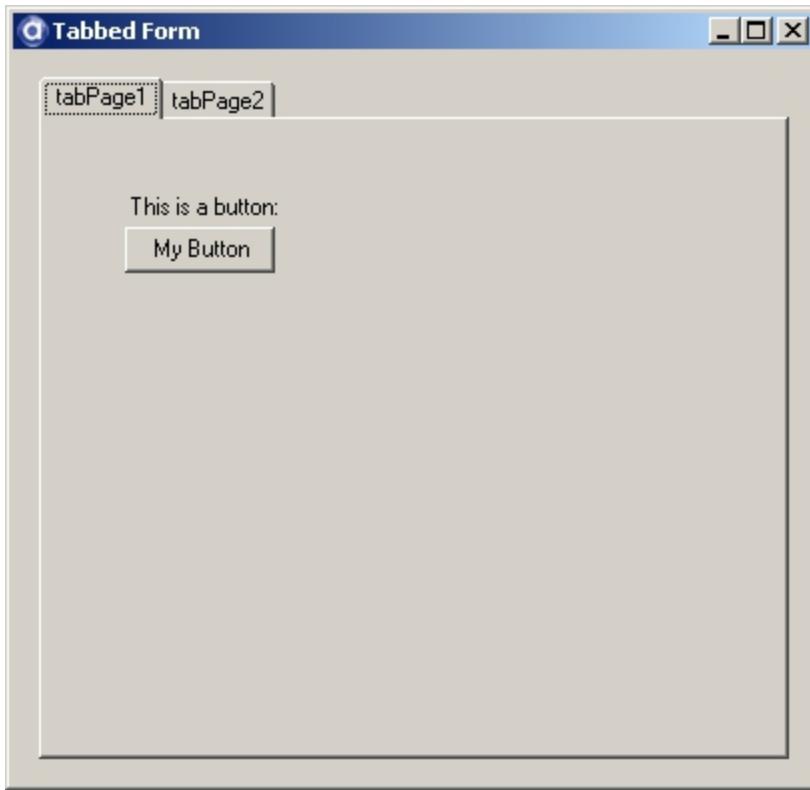
- a. Use the Add button to add additional tabs (pages) to your multi-page UI form.
- b. Use the Appearance > Text property to edit the name for each tab, as desired.

You can also use images in place of names for the tabs. For details, see Using Images Instead of Tab Names on page 1073.

**NOTE:** Do not confuse the Text property with the Name property. The Text property sets the label name for the tab (the tab label the user will see); the Name property sets the "internal" tab name, which can be used, for example, to refer to a particular tab from a C# script.

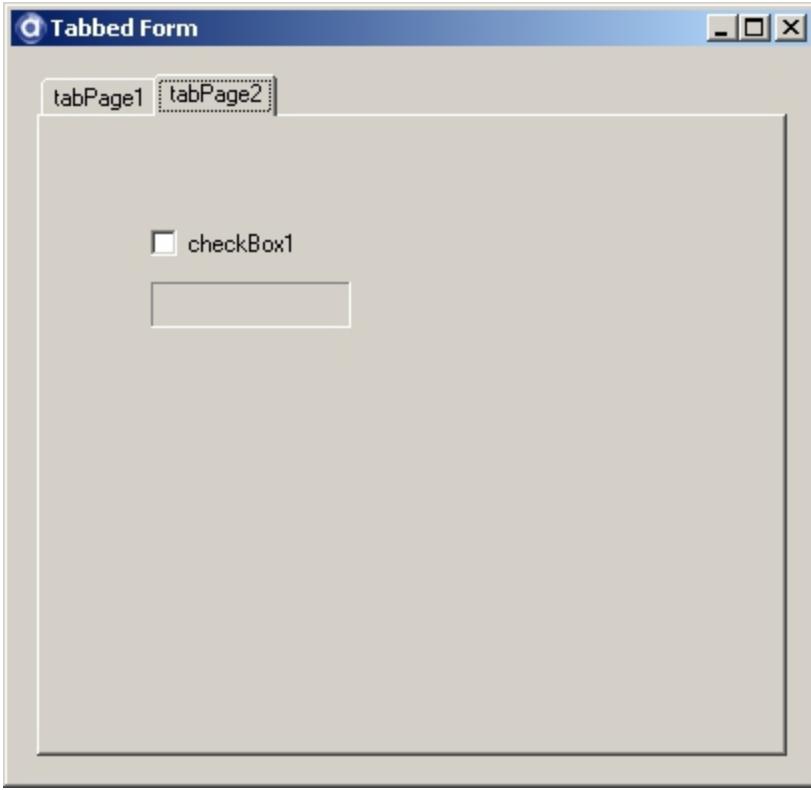
- c. Use the other Appearance properties to edit the color and styles for each tab page, as desired.
- d. After you have completed your edits and formatting, click OK to save the changes and close the editor.

4. Select the tab that you want to edit.
5. Add tools to that page. For details on adding tools to a UI form, see Creating an Example User Interface on page 1076.
6. Select the next tab that you want to edit.
7. Add tools to that page. For details on adding tools to a UI form, see Creating an Example User Interface on page 1076.
8. Click the Compile and Run ( ) icon to execute the form and test the tab control. The following is an example of page 1 of the tabbed form in executed mode.



*Tabbed UI Form, Page 1, in Executed Mode*

9. Click the Page 2 tab to preview the contents of that page. The following is an example of page 2 of the tabbed form in executed mode.



*Tabbed UI Form, Page 2, in Executed Mode*

### **Using Images Instead of Tab Names**

For certain applications, you may want to use an image on each tab instead of a tab name. The Tab Control tool contains an ImageList parameter, which allows you to create an image list and assign it to the Tab Control tool. The order of images on the ImageList corresponds to the tab order on the Tab Control tool (for example, image 0 = tabPage1; image 1 = tabPage2, and so on).

To place images on your tabs:

1. Add an Image List tool to your UI form and place some images on to the list. For more details on creating an image list, see [Creating an Image List on page 1058](#).

You need to have an image for each tab. Arrange the images on the list so they are in the desired order for your tabs (for example, image 0 = tabPage1; image 1 = tabPage2, and so on).
2. Select the Tab Control tool on your UI form. The entire tabbed area will display a dashed outline to show it is selected.
3. Select Properties > ImageList and select the image list that you just added to your UI form. The tab names are replaced with the images on the ImageList, as shown in the following figure.



*Tab Names Changed to Images (Before and After)*

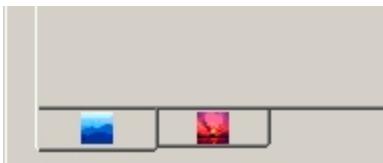
## Applying Behavior Parameters to the Tab Control

For certain applications, you may want to modify the location or appearance of the tabs (for example, position the tabs at the bottom of the control, or change the tabs from a standard tab to a button).

### Changing the Tab Location

To change the tab location:

1. Select the Tab Container tool from the previous example.
2. Locate the Behavior > Alignment parameter.
3. Select Bottom. The tabs are relocated to the bottom of the Tab Control tool, as shown in the following figure.



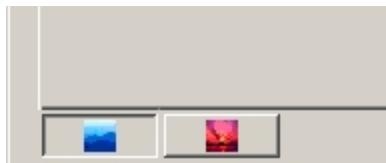
*Tabs Relocated to Bottom*

### Changing the Tabs to Buttons

To change the tabs so they appear as buttons:

1. Select the Tab Container tool from the previous example.
2. Locate the Behavior > Appearance parameter.
3. Select Buttons or Flat Buttons. The tabs are changed to the selected button format, as shown in the following figure.

following figure.



*Tabs Changed to Buttons*

## Creating an Example User Interface

This example assumes you already have a Process Manager installed in the ACE workspace, you have created and taught a process, and it is ready to run. For details, refer to Process Control on page 799.

After you complete this example interface, you will be familiar with the following UI elements:

- Button (with different "when true" values to turn on/off a linked item)
- Check Box
- Group Box
- Label
- Numeric Up/Down Control
- Picture Box
- Radio Button (with different "when true" values to turn on/off a linked item)
- Toggle Button (used as a status indicator)
- Track Bar

### NOTES:

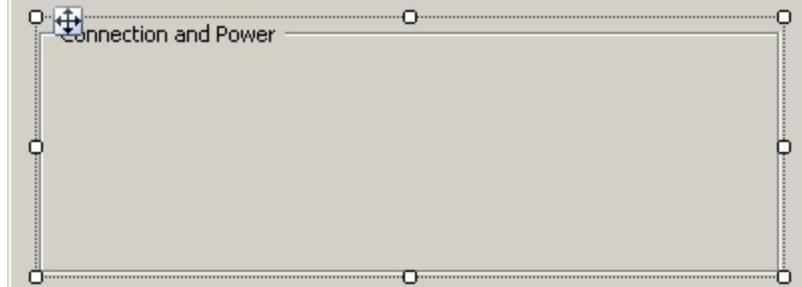
1. See the figure in step 8 for the completed layout of the tools placed in the following steps.
2. All tools use the default values, except for the changes noted in the following instructions.

1. Create a User Interface Form object in your ACE workspace. For details, see User Interface Designer on page 1000.
2. Drag a Label into the top part of the form designer.
  - a. In the Properties items, click the Appearance > Text item and type "Connection, Power, and Process Control" in that field.
  - b. Use the Appearance > Font and ForeColor properties to adjust the font and foreground color, as desired, to create a title for your UI.
  - c. The completed label should look similar to the following figure.



*Label for Example UI Form*

3. Drag a Group Box tool onto the form designer.
  - a. Size and position the Group Box for the upper 1/2 of the form.
  - b. In the Properties Items, click the Text item and type "Connection and Power" in that field.
  - c. The completed Group Box should look similar to the following figure.



*Group Box for Example UI Form*

4. Drag another Group Box tool onto the form designer; or, select the previous Group Box and copy/paste it (right-click and use the shortcut menu, or use the keyboard shortcuts: **Ctrl+C** and **Ctrl+V**).
  - a. Size and position the Group Box for the lower 1/2 of the form.
  - b. In the Properties Items, click the Text item and type "Process Control" in that field.
  - c. The completed Group Box should look similar to the previous figure, except that the title will be "Process Control" instead of "Connection and Power".
5. Add the following tools to the "Connection and Power" group, and then use the properties to format and connect them to the corresponding data items, as shown in the table.

Tool	Appearance	Images	Property Links Selection
Radio Button	Text: Connect		ItemValue: Controller > General > Connection Status ValueWhenChecked: -1
Radio Button	Text: Disconnect		ItemValue: Controller > General > Connection Status ValueWhenChecked: 0
Toggle Button		DisplayMode: DigitalInput	ItemValue: Controller > General > Connection Status ValueWhenUnpressed: 0 ValueWhenPressed: -1
Label	Text: Connection Status		

<b>Tool</b>	<b>Appearance</b>	<b>Images</b>	<b>Property Links Selection</b>
check box	Text: Power / Cal		ItemValue: Controller > General > Power and Calibration ValueWhenUnchecked: 0 ValueWhenChecked: -1
Toggle Button		DisplayMode: DigitalInput	ItemValue: Controller > General > Power and Calibration ValueWhenUnpressed: 0 ValueWhenPressed: -1
Label	Text: Power / Cal Status		

The completed Connection and Power group should look similar to the following figure.



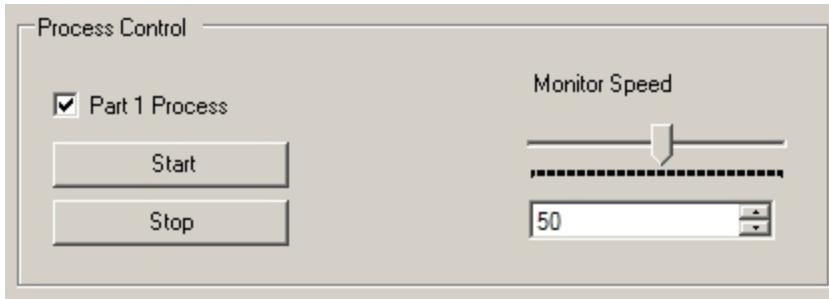
*Connection and Power Group Box with Tools*

- Add the following tools to the "Process Control" group, and then use the properties to format and connect them to the corresponding data items, as shown in the table.

<b>Tool</b>	<b>Appearance</b>	<b>Behavior</b>	<b>Property Links Selection</b>
Check Box	Text: Part 1 Process		ItemValue: Process Manager > General > Process 0 Enabled
Button	Text: Start		ItemValue: Process Manager > General > Run Status ValueWhenPressed: -1
Button	Text: Stop		ItemValue: Process Manager > General > Run Status ValueWhenPressed: 0
Label	Text: Monitor Speed		

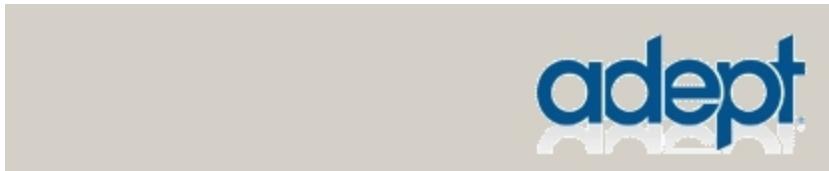
Tool	Appearance	Behavior	Property Links Selection
Track Bar		Large Change: 5 Max: 100 Min: 0	ItemValue: Controller > General > Monitor Speed
Numeric Up/Down			ItemValue: Controller > General > Monitor Speed

The completed Process Control group should look similar to the following figure.



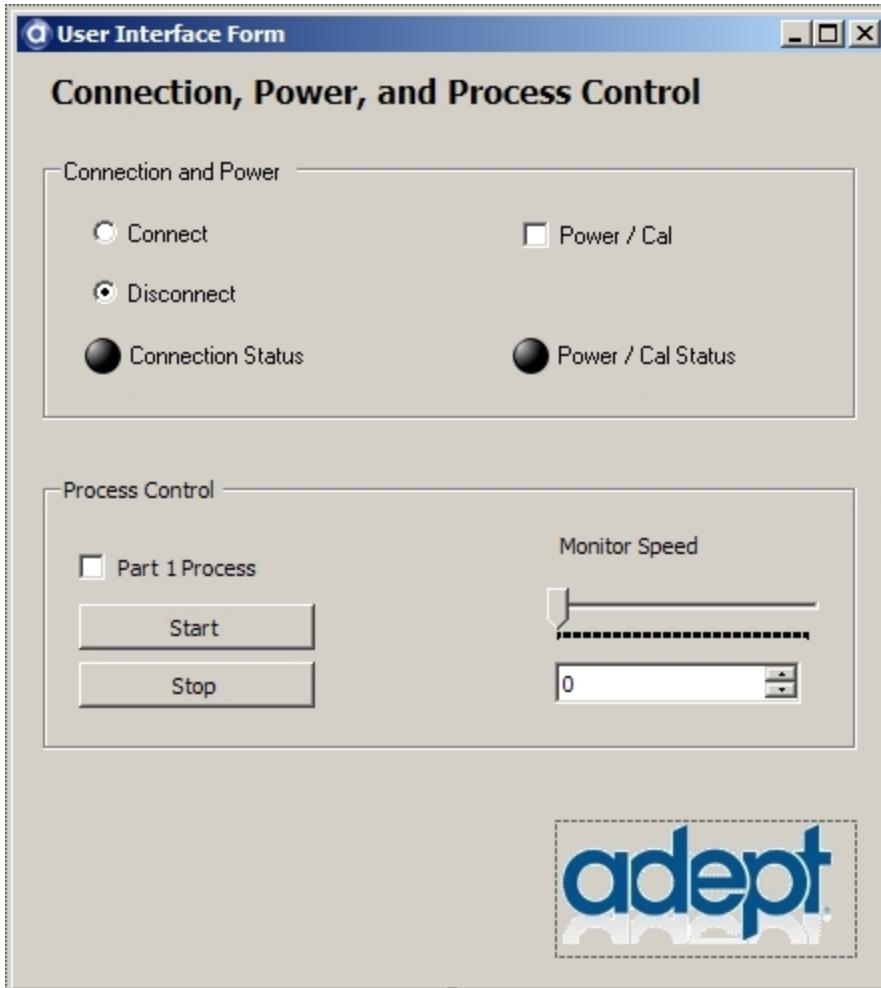
*Process Control Group Box with Tools*

7. Drag a Picture Box tool into the lower-right corner of the form designer. Use the Appearance > Image parameter to import a graphic file (JPG, GIF, etc.) from your hard drive. For example, you could insert your company logo, as shown in the following figure.



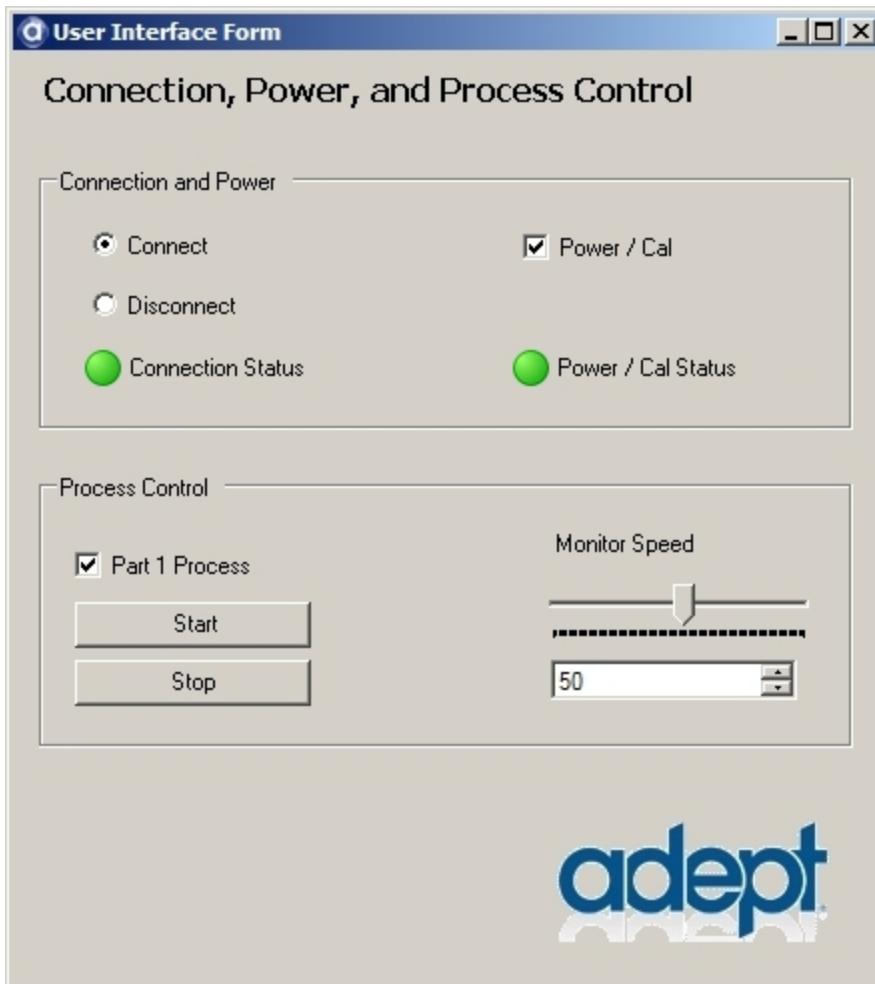
*Picture Box with Company Logo*

8. Fine-tune the position and appearance of the form items until your form looks similar to the one in the following figure.



*Completed Example UI Form*

9. Click the Compile and Run ( ) icon to execute the form and test the controls. The following is an example of the previous form in executed mode.



*Completed UI Form in Executed Mode*

## Deploying the Custom User Interface

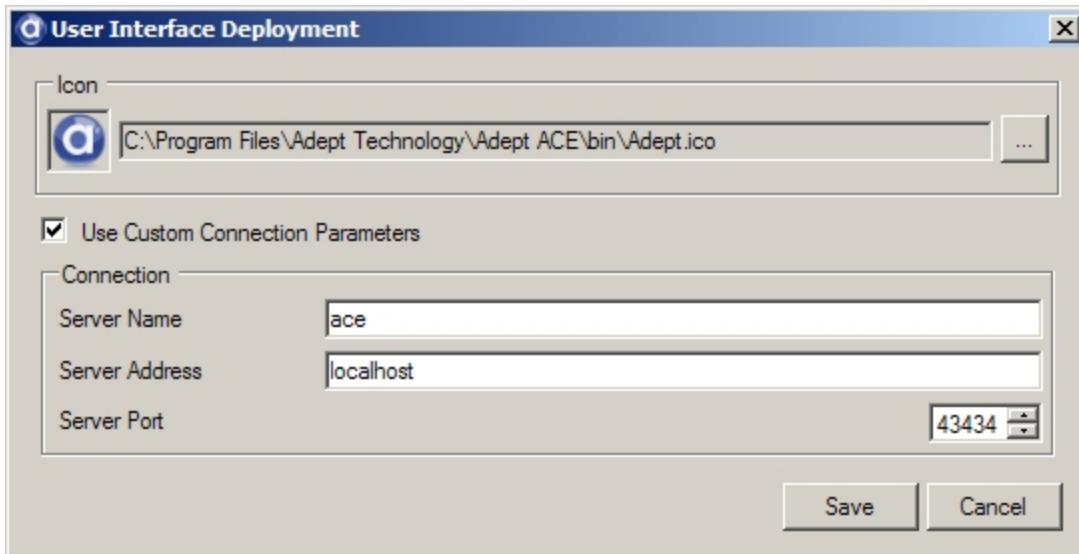
After you create your custom UI, you can start ACE so that it automatically presents the UI to the operator. This is done by starting the ACE software with the "loadui" parameter, which loads a specified user interface. There are several methods for doing this, which are described in the following sections.

**NOTE:** The following procedures reference the "Load Workspace at Startup" option, which automatically loads a specified workspace when starting the ACE software. For details on this option, see System Options on page 104. If you do not have this option enabled, then you must use the datafile command-line parameter to load a specified workspace.

## Using the User Interface Deployment Tool

The ACE software includes a User Interface Deployment tool, which is used to select UI deployment options and then create a shortcut based on those selections. To use the User Interface Deployment tool:

1. On the UI Designer toolbar, click the Deploy UI icon (). The User Interface Deployment tool opens.



User Interface Deployment Tool with Options Selected

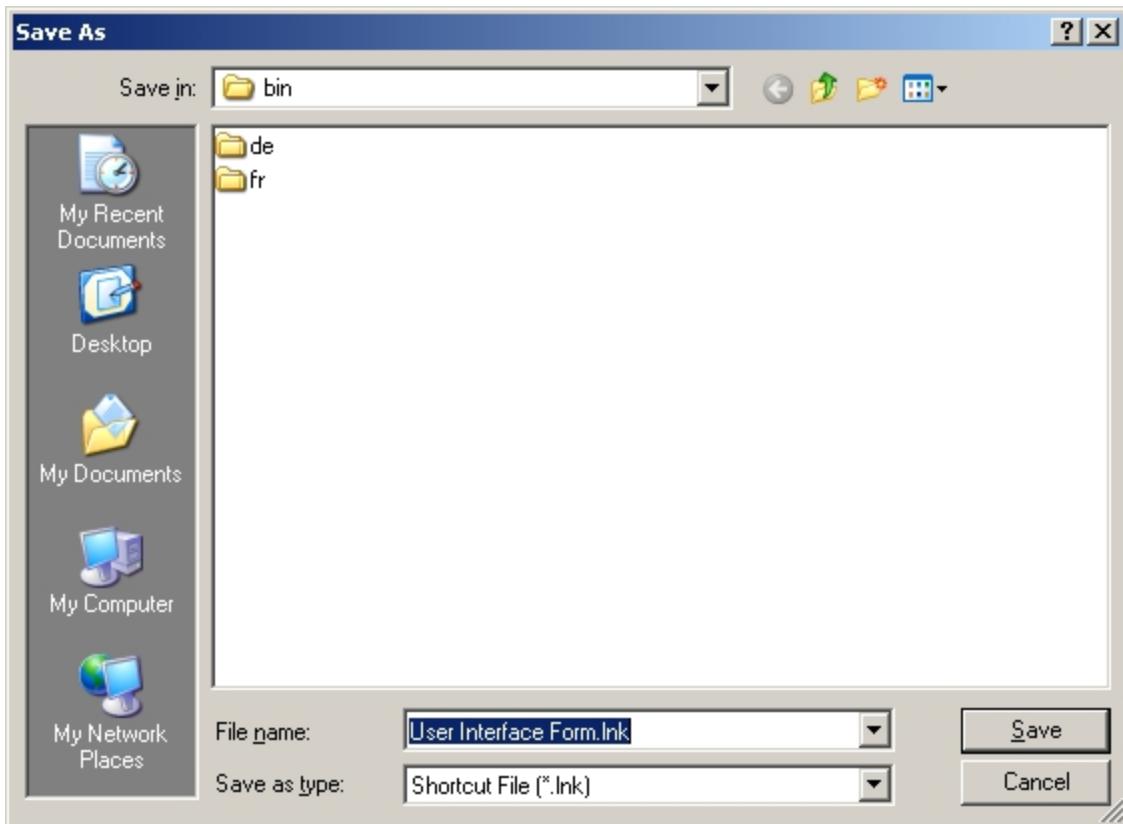
2. To apply a custom icon to the shortcut, use the browse button (...) to locate and select the icon (.ICO) file on your PC. After the icon file is selected, it is displayed in the Icon window to the left of the file path.
3. To create a shortcut with custom connection parameters, select the Use Custom Connection Parameters option. By default (without the check box activated), the connection parameters used are those specified on the main form of the User Interface Designer.

The custom connection parameters are described in the following table.

**NOTE:** The server parameters specified must match those used for starting the server instance. For more information, see Starting ACE from the Windows Command Prompt on page 40.

Item	Description
Server Name	The name assigned to the server instance.
Server Address	The IP address of the server instance. The default entry "localhost" specifies that the server is running locally (on the same PC that the client will run on).
Server Port	The port number used for communicating with the server instance.

4. Click **Save** to close the window. The Save As dialog box opens, which is used to specify the location where the shortcut will be saved.



*Save As Dialog Box*

5. Specify the name and location for the shortcut.

6. Click **Save**. The shortcut is saved to the specified location.

## Using a Command Line to Deploy the UI

### ***Option 1: Starting the ACE Client, Server and UI Separately***

To start the client and server separately:

1. Start the ACE server.
  - a. Start ACE in server mode (or service) with the workspace loaded. This can be done by checking the "Load Workspace at Startup" option, or using the datafile command-line parameter.

**NOTE:** When using the datafile command-line parameter, you must include the path to the file location on the system hard drive.

For example the following command, which is entered at the Windows command prompt, will load the specified workspace:

```
ace.exe server datafile="[path to file]XXX.awp"
```

-OR-

- b. Start ACE through the "ACE" Windows desktop shortcut and have the "Load Workspace at Startup" option enabled. This will start the standard ACE client and server, and load the specified workspace.
2. Use the "loadui" startup parameter to start an ACE client that will launch the specified user interface. From the Windows command prompt, change to the ACE\bin directory and then enter the command:

```
ace.exe client loadui="/User Interface Form"
```

### ***Option 2: Start the ACE Software and UI Simultaneously***

To start the user interface and server simultaneously (in one call):

Use the "Load workspace at startup" option, or the datafile command line parameter, to load the Workspace.

**NOTE:** When using the datafile command-line parameter, you must include the path to the file location on the system hard drive.

For example the following command, which is entered at the Windows command prompt, will load the specified workspace and then initialize the specified user interface:

```
ace.exe client_server datafile="[path to file]XXX.awp" loadui="/User Interface Form"
```

If you are using the "Load Workspace at Startup" option, you can omit the "datafile..." parameter from the above command string.

## Using a Batch File to Deploy the UI

You can use a Windows batch (.BAT) file to automate the process for launching the ACE software with the custom user interface.

To create a batch file for starting the ACE software and loading the user interface form:

1. Open a plain text editor, such as Notepad.
2. Enter the batch file commands.
3. Save the file to the Windows Desktop. Make sure you use a .BAT file extension in place of the default .TXT file extension.

For example, the following lines will: Load the ACE server with the specified workspace, load the ACE client, and open the specified user interface form. The following example assumes your data files are stored in the folder C:\Data\_files.

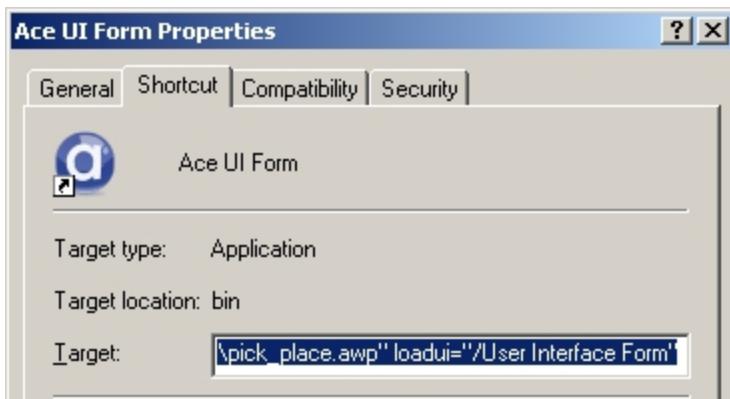
```
echo off
cd c:\program files\Omron\ACE\bin\
echo Starting ACE Server/Client and Workspace
ace.exe client_server datafile="C:\Data_files\pick_place.awp" loadui="/User Interface
Form"
```

## Using a Shortcut to Deploy the UI

You can use a Windows Desktop shortcut to automate the process for launching the ACE software with the custom user interface. For more details on using a shortcut to start the ACE software, see Creating Shortcuts for Starting ACE on page 42.

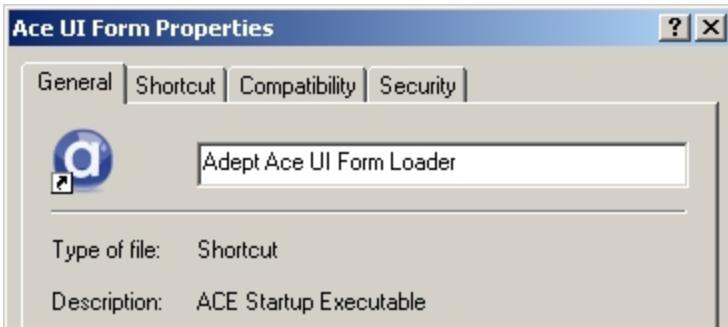
To create a shortcut for starting the ACE software and loading the user interface form:

1. Copy and paste the ACE desktop shortcut.
2. Right-click on the copied shortcut and select Properties from the menu.
3. On the Shortcut tab, modify the Target field so it contains the additional commands for loading the specified workspace and the specified user interface form.



### Modifying the Target Field Information

4. On the General tab, change the name of the shortcut, as desired, so you can distinguish it from the default ACE software shortcut.



### Renaming the Shortcut

5. Click **OK** to save the modified shortcut.

For example, the following lines will: Load the ACE server with the specified workspace, load the ACE client, and open the specified user interface form. The following example assumes your data files are stored in the folder C:\Data\_files.

```
"C:\Program Files\Omron\ACE\bin\Ace.exe" client_server datafile="C:\Data_files\pick_place.awp" loadui="/User Interface Form"
```

## Controlling the User Interface through Scripts

As you are creating your custom UI, you can customize various elements, for example, change the background color on the base form, through a C# client-side script. The UI Form designer provides a built-in C# code editor for creating and editing the client-side scripts.

**NOTE:** This feature requires a special license. For details, see Licensing Requirements on page 28.

The UI Form designer code editor is similar in look and function to the ACE C# program editor. However, the scripts created in the UI Form designer code editor are client-side scripts; the scripts created in the C# program editor are server-side scripts. For more details on the C# program editor, see C# Language Programming on page 750.

All of the tools in the UI Form designer Toolbox are available to C# scripts in the UI Form designer code editor. The properties available to scripts will vary, depending on the particular tool. For details:

- Use the IntelliSense information in the script editor, and
- Refer to the System.Windows.Forms namespace in the MSDN library:

<http://msdn.microsoft.com/en-us/library/system.windows.forms.aspx>

The following example describes how to change the background color of the base form through a C# client-side script.

1. Create a User Interface Form object in your ACE workspace. For details, see Controlling the User Interface through Scripts on page 1087.
2. In the Designer view, double-click the base form to open the Source code editor and position the cursor at the UIBuilderForm1\_Load method (the event handler for the load event of the base form). For more details, see Viewing the Events for a Tool on page 1013.
3. In the C# code editor, add the following C# snippet:

```
// Set the background color of the base form to red  
this.BackColor = System.Drawing.Color.Red;
```

The completed code on the Source page will look like the following:

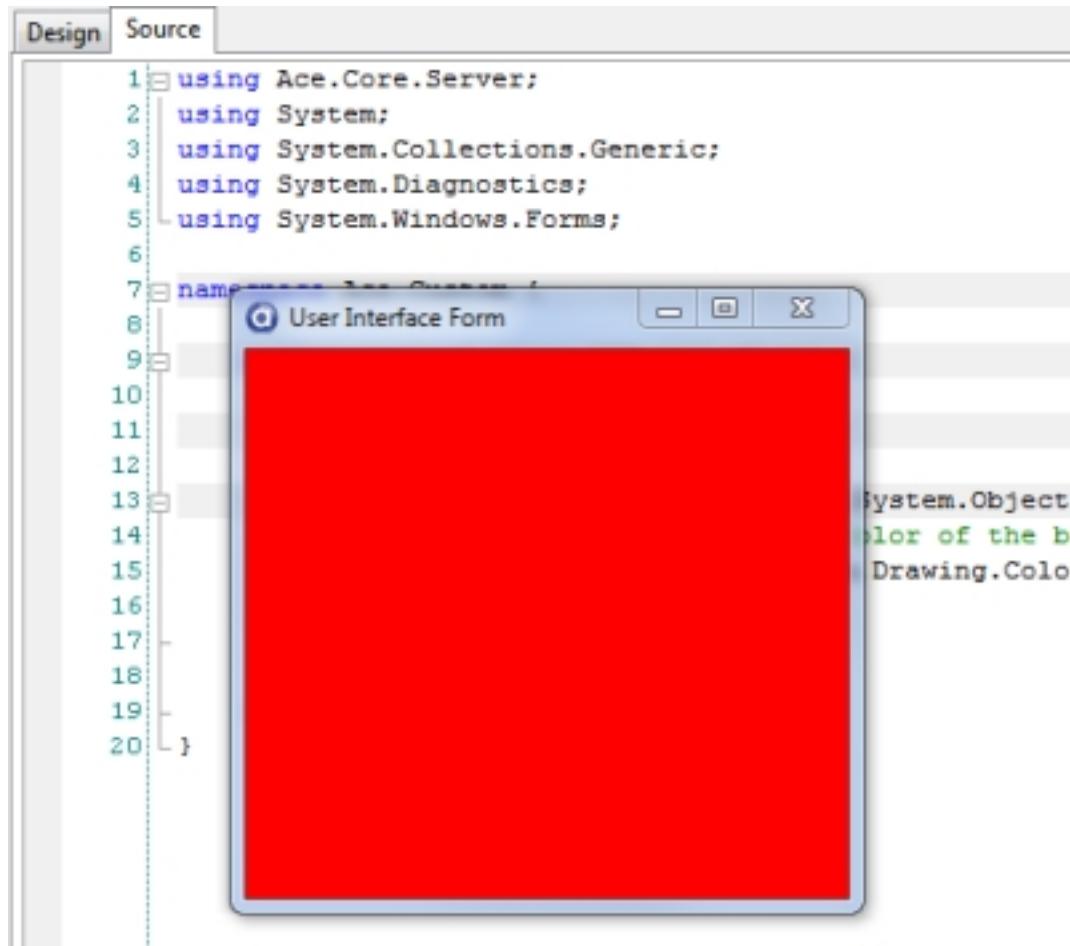
### NOTES:

- Line numbers are added for reference; do not type them into your script.
- Line 15 is an optional comment.
- Line 16 is the code that changes the background color.

```
1. using Ace.Core.Server;  
2. using System;  
3. using System.Collections.Generic;  
4. using System.Diagnostics;  
5. using System.Windows.Forms;  
6.  
7. namespace Ace.Custom {  
8.
```

```
9.     public partial class UIBuilderForm1 {  
10.        public AceServer ace;  
11.  
12.        private void UIBuilderForm1_Load(System.Object sender, System.EventArgs  
e) {  
13.            Trace.WriteLine("UIBuilderForm1.Load Event Handler");  
14.            // Set the background color of the base form to red  
15.            this.BackColor = System.Drawing.Color.Red;  
16.        }  
17.    }  
18.  
19.}  
20.}
```

4. On the UI Form designer toolbar, click the Compile and Run (  ) icon to preview the form. The form will load with a red background, as shown in the following figure.



User Interface Form with Red Background

## Emulation Mode

**NOTE:** This feature requires hardware that supports DirectX 9.0c (or later) and 3D-graphics processing. Otherwise, the feature will be disabled.

The topics in this chapter describe the Emulation Mode features in the ACE software.

Overview .....	1090
Features and Differences .....	1091
Enabling Emulation Mode .....	1093
Emulation Mode Checklist .....	1100

## Overview

**NOTE:** This feature requires hardware that supports DirectX 9.0c (or later) and 3D-graphics processing. Otherwise, the feature will be disabled.

The ACE software contains a special operating mode called Emulation Mode. This mode provides a "virtualized" environment that emulates the physical hardware you have in your application. This allows you to program and operate your application offline.

Although the emulation mode is an optional operating mode, it behaves as though you're working in the standard operating mode of the ACE software. Once you have enabled the emulation mode, you can create and program an ACE application in the same manner as you would when connected to physical hardware. This provides a seamless user experience that is nearly identical to running with real (physical) hardware.

Further, the emulation mode can run multiple simultaneous instances of controllers/robots on the same PC at the same time. This includes the handling of network ports and multiple file systems. This feature allows you, through emulation, to design, program and operate a real multi-controller/robot application.

The remainder of this chapter details the startup, features and limitations of the emulation mode.

## Features and Differences

This section describes the features and limitations of the ACE emulation mode.

### Features

The emulation mode provides the following features and benefits:

- Create applications offline

If no hardware is available, you can still create your application offline. The emulation mode allows you to configure a workspace with robots, belts, feeders, and other hardware. When the physical hardware becomes available, you can transition from the virtual hardware to the physical hardware.

- Program offline

You can open and edit existing ACE workspaces, and view and edit V+ programs in the workspace.

- Operate applications offline

If you have an existing workspace, you can open that workspace and run an application "virtually" (without the physical hardware). The application runs in an emulation mode and uses the 3D Visualization feature to simulate the behavior of the system.

- Experiment with products/hardware

Because the emulation mode application is created with virtual hardware, you can experiment with different cell designs and layouts before purchasing the physical hardware.

- Train without hardware

The emulated environment provides a convenient and low-risk method for training technicians, operators, and other system users.

### Differences

The emulation mode has the following differences:

- Saving data to the emulated controller is not supported.

When emulation mode is enabled, *do not* save any data to the emulated controller (Disk>D: SD Card). When using emulation mode, ACE creates a new ("fresh") emulated-controller file system in a temporary folder. When ACE is shut down, that file system is deleted, which means the contents of any user-created folders, files, or data will also be deleted. Therefore, you should save your data in a PC folder and/or with the ACE workspace.

- No automatic import of robots when creating a new controller

When emulation mode is enabled, the Create a New Controller wizard does not automatically import robots. Rather, it prompts you to select the robot(s) connected to the controller. For details, see the figure Configure Robots Prompt on page 1094.

- Unavailable controller and robot features are disabled

When emulation mode is enabled, certain Controller Editor and Robot Editor features are not available. Those features are dimmed or hidden to indicate that they are not available. For details on the Controller Editor, see Controller Editor on page 97. For details on the Robot Editor, see Robot Editor on page 114.

- I/O signals are handled differently

In emulation mode, you can use the Digital I/O window to set input signals. For more information in the Digital I/O window, see Digital I/O Window on page 166.

- Robot-to-hardware calibrations have changed

The robot-to-hardware calibrations and workspace-referencing procedures have changed when operating in emulation mode. When emulation mode is enabled, these procedures use the 3D Visualization display to allow for offline calibration/configuration. For more details, see Emulation Mode Wizards on page 185.

- Belt object editor has changed

When emulation mode is enabled, the Belt object editor includes settings for fast speed, slow speed and latch position distance. For details, see Belt on page 809.

- Virtual Camera object editor has changed

When emulation mode is enabled, the Virtual Camera object editor adds an Emulation Configuration parameter, which is used to specify the operating mode. For more details, see Virtual Camera Emulation Configuration (Behavior) on page 1102.

- Cycle times are not identical

When emulation mode is enabled, the cycle times will not exactly match those obtained in real mode.

- Enable power and calibration is handled differently

When emulation mode is enabled, the robot power is enabled and the robot is automatically calibrated in the following situations:

- When loading a workspace
- When creating a controller
- When rebooting the controller
- When changing a Quattro platform

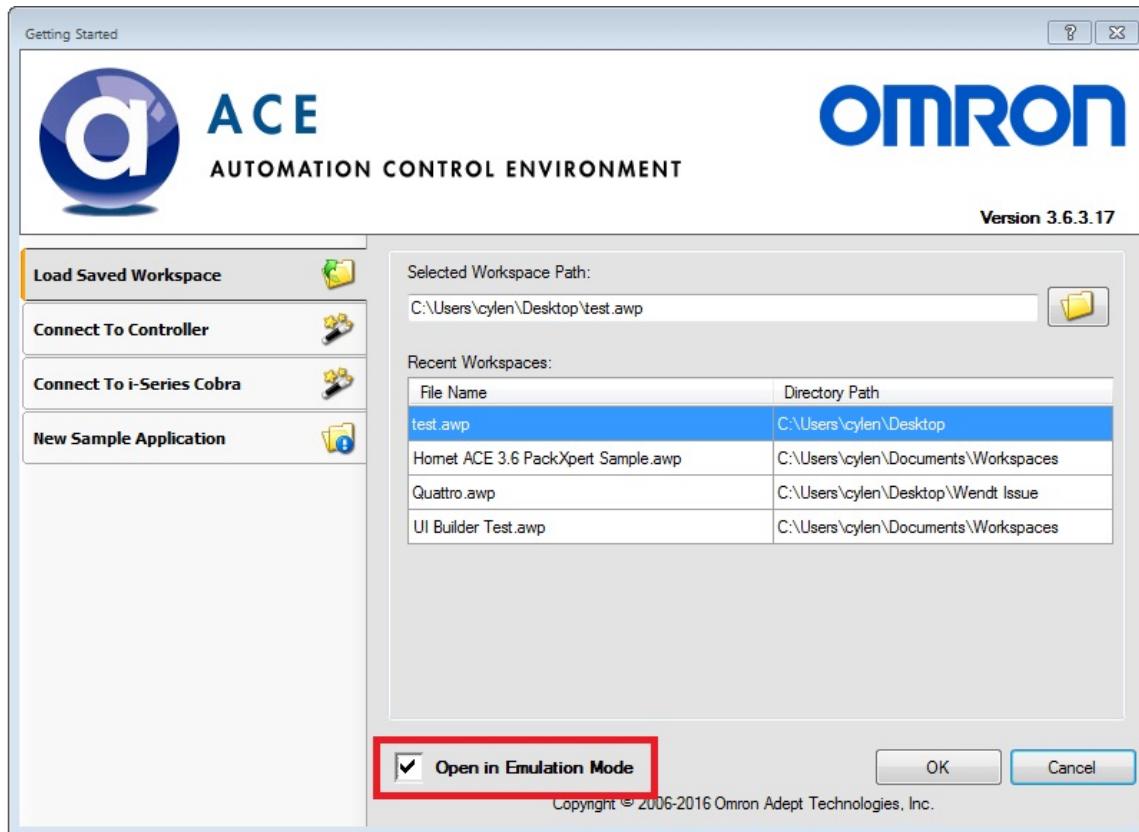
## **V+ versus Emulation Mode**

When ACE is in emulation mode, a V+ emulator is executed for each Controller object in the workspace. From a user perspective, this is transparent — the V+ emulators are executed automatically.

## Enabling Emulation Mode

To enable the emulation mode:

1. Start ACE.
2. In the Getting Started dialog, select the "Open in Emulation Mode" option.



*Emulation Mode Option Selected*

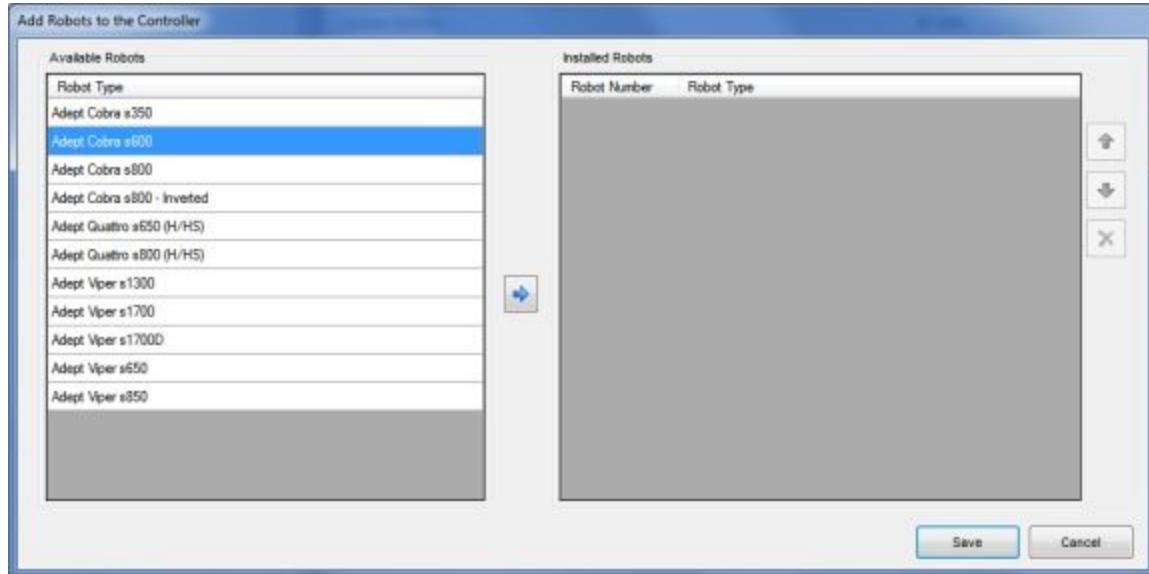
3. If you are creating a New SmartController Workspace, you are asked to select robots for the workspace, as shown in the following figure.



*Configure Robots Prompt*

Click Yes to Proceed.

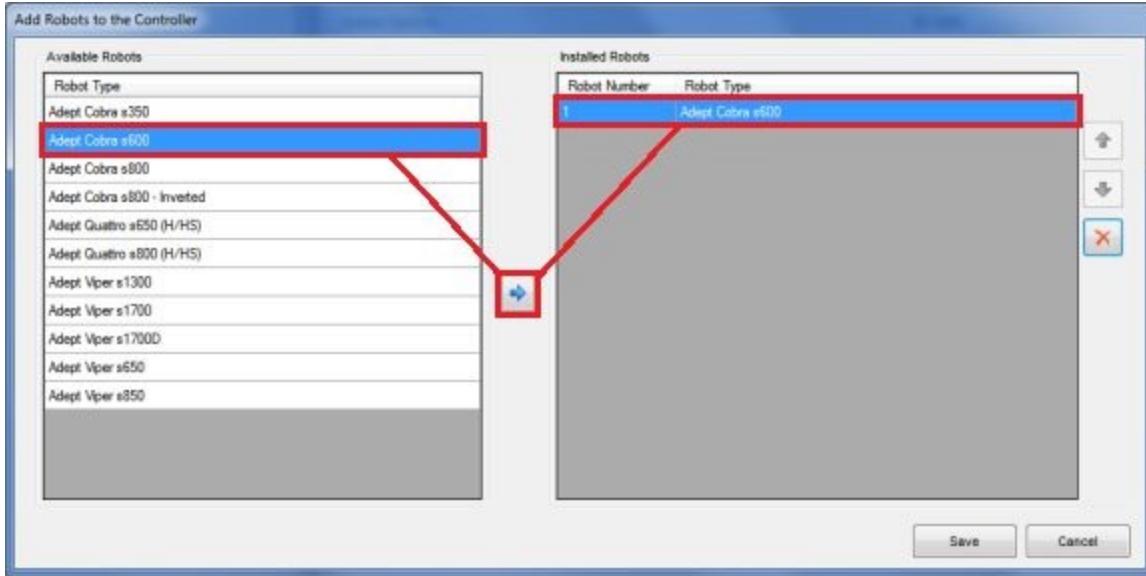
4. The following dialog opens, which allows you to select one or more robots for the application.



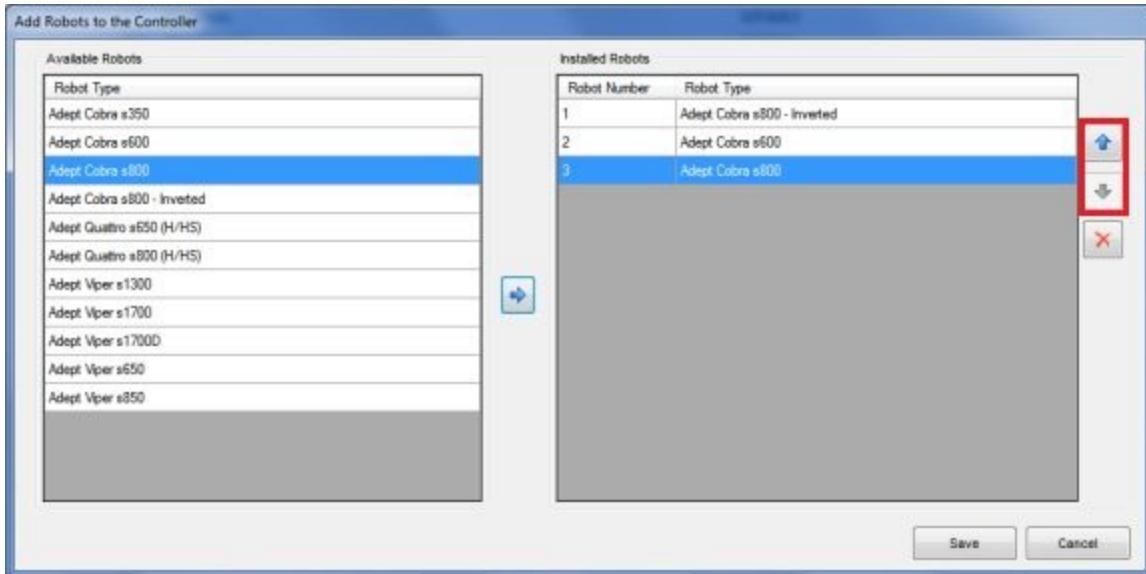
5. Select a robot from the Available Robots list and then click the blue arrow to move it to the Installed Robots list. Optionally, you can double-click the desired robot to add it to the Installed Robots list. Repeat this process for each robot you wish to add to the application.

## Enabling Emulation Mode

---



*Selecting a Robot*



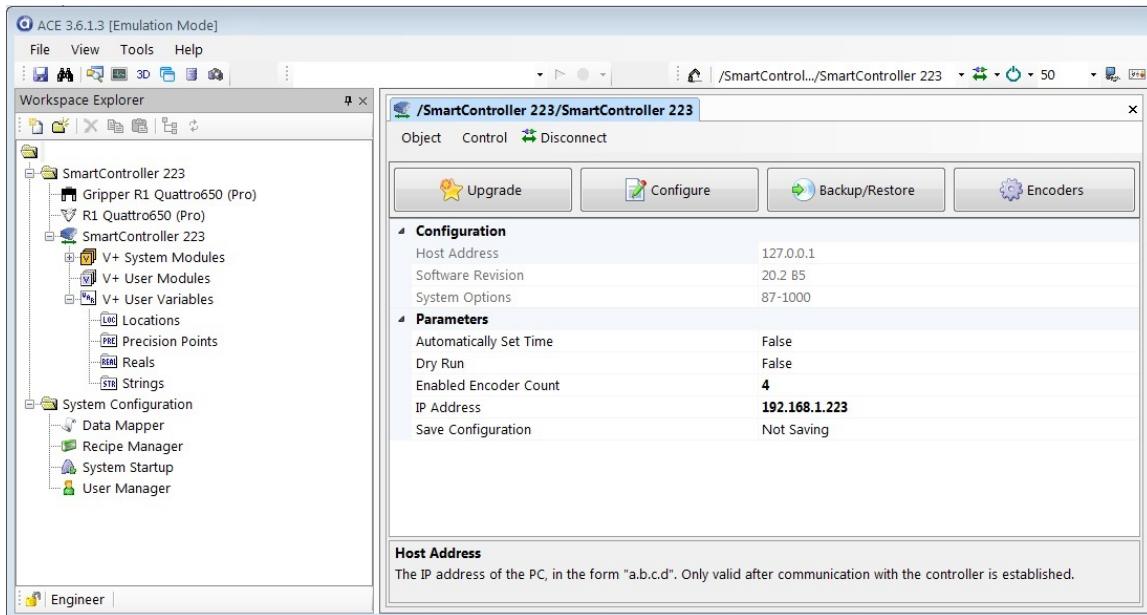
*Selecting Multiple Robots*

- To change the order of the Installed Robots list, select a robot and then use the up/down arrows to move that robot to the desired position on the list.

## Enabling Emulation Mode

---

- To remove a robot from the Installed Robots list, select the robot and then click the delete (X) icon.
6. Click Save to complete the robot-selection process.
7. The emulated controller is rebooted and the selected robots are installed in the ACE workspace.



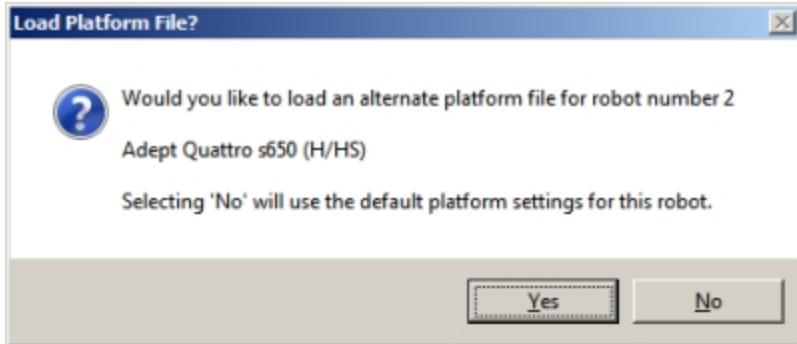
*Selected Robot Installed in the Emulated Workspace*

**NOTE:** When the emulation mode is active, the title bar shows "[Emulation Mode]", as shown in the previous figure.

8. If you selected a Quattro robot, you are prompted to select the desired platform for the robot, as shown in the following figure.

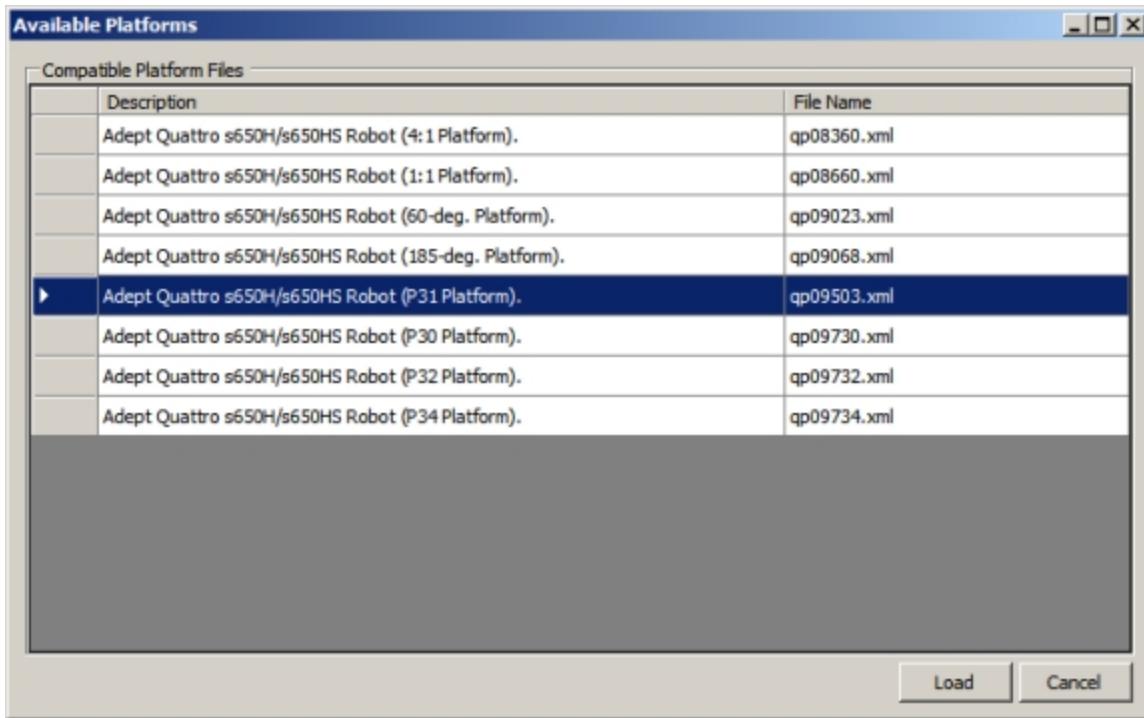
## Enabling Emulation Mode

---



*Load Platform Prompt*

Click Yes to view the list of available platforms, as shown in the following figure.



*Available Platforms List*

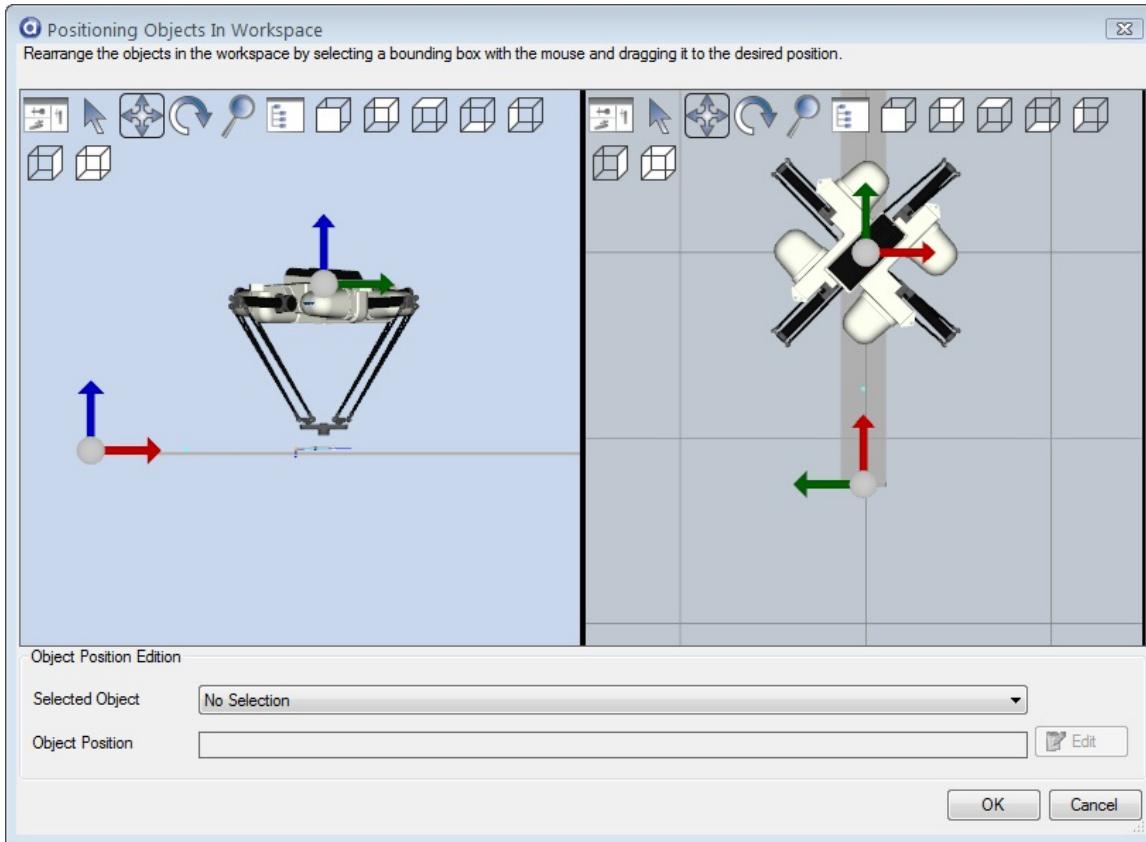
Select the desired platform from the list and then click Load to enable it. You can check that the correct platform is loaded by typing the ID command in the V+ Monitor window, as shown in the following figure.

```
.  
.id  
Software: 18.0 8085-60E0 (Edit A5, 21-Apr-2011, Development Version r3559)  
Controller: 0-1 0  
           SmartController EX Emulator  
Robot 1:   720-0 1000-0-0 43  
           Adept Quattro s650H/s650HS Robot [P34 platform].  
|
```

*Output of ID Command Showing the Quattro Platform Configuration*

**NOTE:** If you need to change the loaded platform, open the Controller editor and click **Configure > Configure Robots**. You will repeat the robot-selection and platform-selection steps described in this section.

9. Next, the Workspace Positioning tool opens so you can position the robot(s) in the workcell.



*Workspace Positioning Tool*

For details on using the tool, see [Workspace Positioning on page 945](#).

10. You are ready to start programming your application, as if you were using physical equipment. For example, you can use the Process Manager to develop a packaging application. For more details on the Process Manager, see [Process Manager on page 843](#). For a list of items that need to be checked when operating in emulation mode, see [Emulation Mode Checklist on page 1100](#).

## Emulation Mode Checklist

This section describes the item you should check when using the ACE emulation mode.

### Latch Signals

The latch signals must be configured in V+, using the Controller object editor.

```
<belt objectid="60"></belt>
<belt objectid="61"></belt>
<belt objectid="62"> <poslatch>1002</poslatch></belt>
<belt objectid="63"> <poslatch>1001</poslatch></belt>
<robot objectid="1"> <producttype>quall050</producttype> <safetyzonenum>1</safetyzonenum>
```

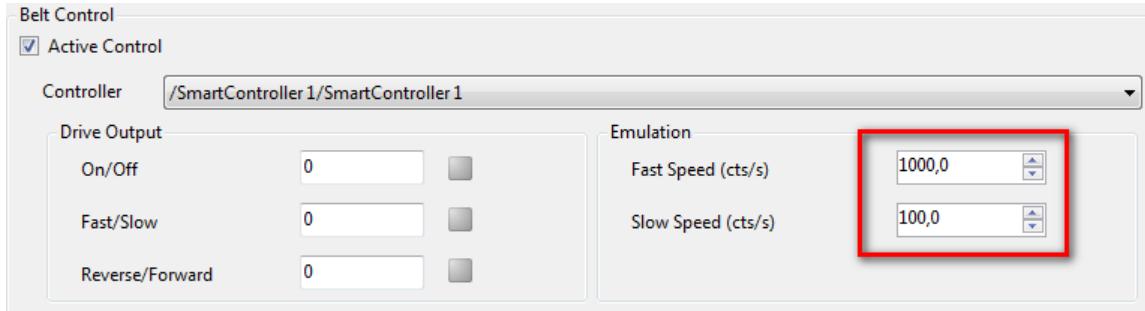
For more details on accessing the V+ configuration, see Controller Editor on page 97.

### Belt Speed

The belt speed must be set on the Encoder Diagnostics page or the Belt editor page, as shown in the following figures:

	Position (ct)	Velocity (ct/s)	Running
Encoder 0	77402	0	<input type="checkbox"/>
Encoder 1	508764	200	<input checked="" type="checkbox"/>
Encoder 2	0	0	<input type="checkbox"/>
Encoder 3	0	0	<input type="checkbox"/>

*Encoder Diagnostics*



*Belt Control Group (from the Belt Object Editor)*

The emulation speed controls are available only when emulation mode is enabled. For more details on the Encoder diagnostics editor, see External Encoder Diagnostics on page 1115. For more details on the Belt object editor, see Belt on page 809.

#### Belt and Sensor Calibrations

In emulation mode, the Belt and Sensor calibration wizards are different - they use 3D visualization windows, which allow you to drag the calibration objects to the desired positions. For more details, see Emulation Mode Wizards on page 185.

## Quattro Robot Platform

When the Quattro robot is used, the correct platform must be set in V+. The platform information can be displayed by typing the ID command in the V+ Monitor window, as shown in the following figure.

```

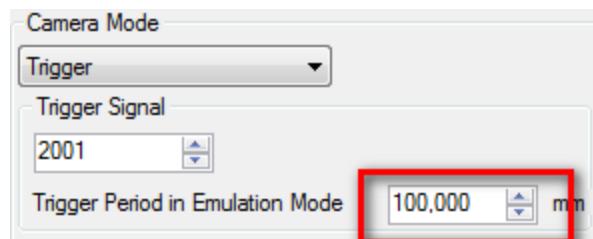
.
.
.id
Software: 18.0 8085-60E0 (Edit A5, 21-Apr-2011, Development Version r3559)
Controller: 0-1 0
SmartController EX Emulator
Robot 1: 720-0 1000-0-0 43
Adept Quattro s650H/s650HS Robot [P34 platform].
|
```

*Output of ID Command Showing the Quattro Platform Configuration*

If you need to change the loaded platform, open the Controller editor and click **Configure > Configure Robots**. You will perform the robot-selection and platform-selection steps described in the section Enabling Emulation Mode on page 1093. For more details on using the V+ Monitor window, see Monitor Window on page 777. For more details on the Controller editor, see Controller Editor on page 97.

## Camera Trigger Mode

If the camera is in Trigger mode, the Trigger Period must be set (in Control Sources), as shown in the following figure.

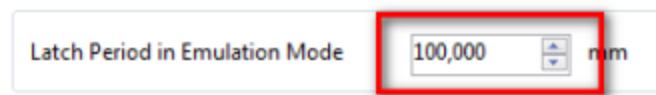


*Camera Trigger Period in Emulation Mode*

The Trigger Period in Emulation Mode setting is displayed only when the ACE software is in emulation mode. For more details on control sources, see Control Sources on page 869.

## Latch Period (Distance)

For part or part target instances that use Latch mode, the latch period (distance) must be set in the Belt object editor, as shown in the following figure.



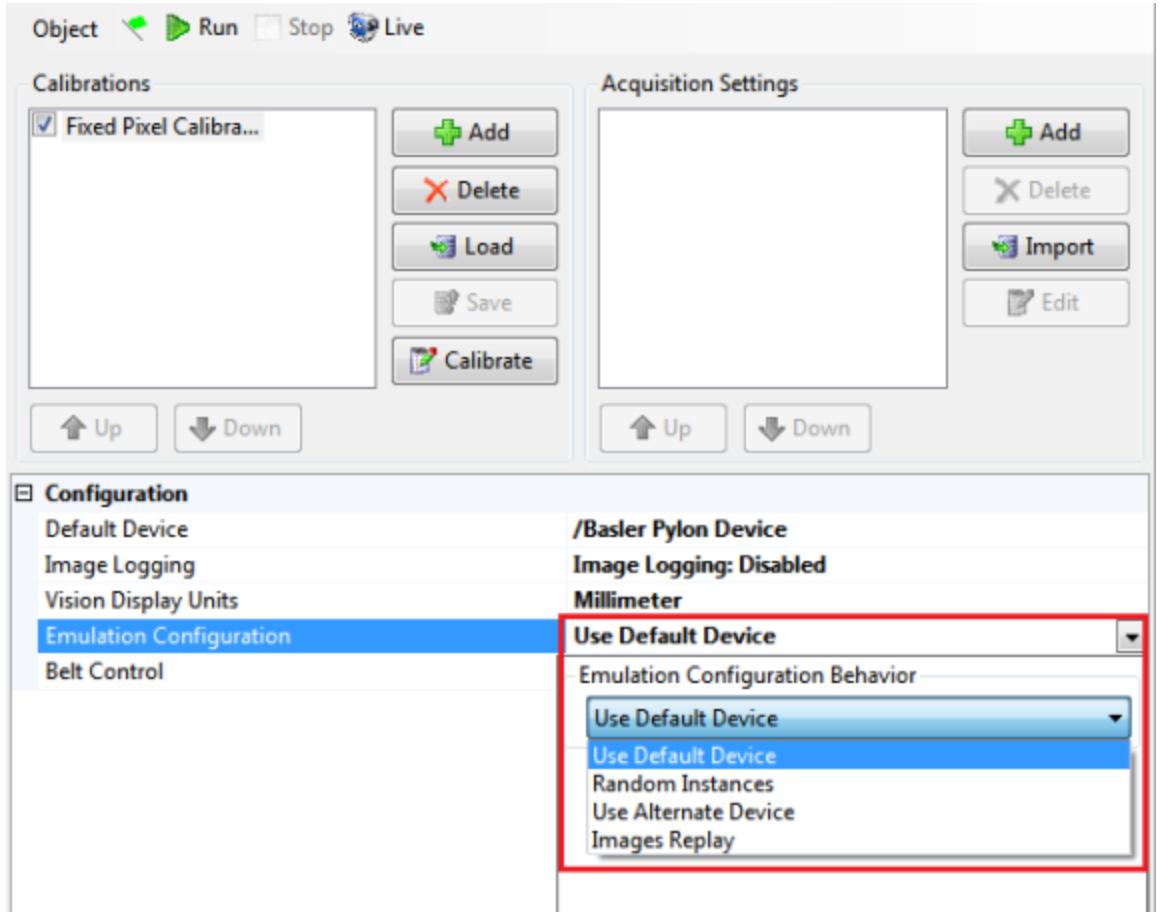
*Latch Period Setting (in Belt Object Editor)*

The Latch Period in Emulation Mode setting is displayed only when the ACE software is in emulation mode. For more details on the Belt object editor, see Belt on page 809.

## Virtual Camera Emulation Configuration (Behavior)

If vision is used, the virtual camera emulation configuration (behavior) must be set, as shown in the following figure.

## Virtual Camera Emulation Configuration (Behavior)



### Emulation Configuration (Behavior) Setting

When emulation mode is enabled, the Virtual Camera object editor adds an Emulation Configuration parameter, which is used to specify one of the following modes:

Mode	Description
Use Default Device	Has the same behavior in emulation mode as in real mode. For example, if your camera is an Emulation Device in real mode, you can use it "as is" in emulation mode.
Random Instances	A random number of picture instances are generated, based on the specified value.  <b>NOTE:</b> When the Basler Pylon Device is used and Random Instances is selected, the Fixed Pixel calibration will automatically load as the calibration type in the Basler Pylon Device Virtual Camera object.

Mode	Description
Use Alternate Device	Pictures are obtained from another virtual camera that is connected to an Emulation Device. For more details on the Emulation Device, see the topic "Adding an Emulation Device" in the <i>ACE Sight User's Guide</i> .
Images Replay	A set of images is displayed from a specified directory.

For more details on virtual cameras, see the *ACE Sight User's Guide*.

## Diagnostic Tools

The ACE software contains a set of diagnostic tools to help you identify problems in your system. The topics in this chapter describe the diagnostic tools available in the ACE software.

About ACE .....	1106
Event Log .....	1108
FireWire Event Log .....	1110
Pending Errors Log .....	1112
Robots and Encoder Diagnostics .....	1113
System Monitor .....	1117
System Diagnostic Summary .....	1128
Cobra i600/i800 Robot Diagnostics .....	1130

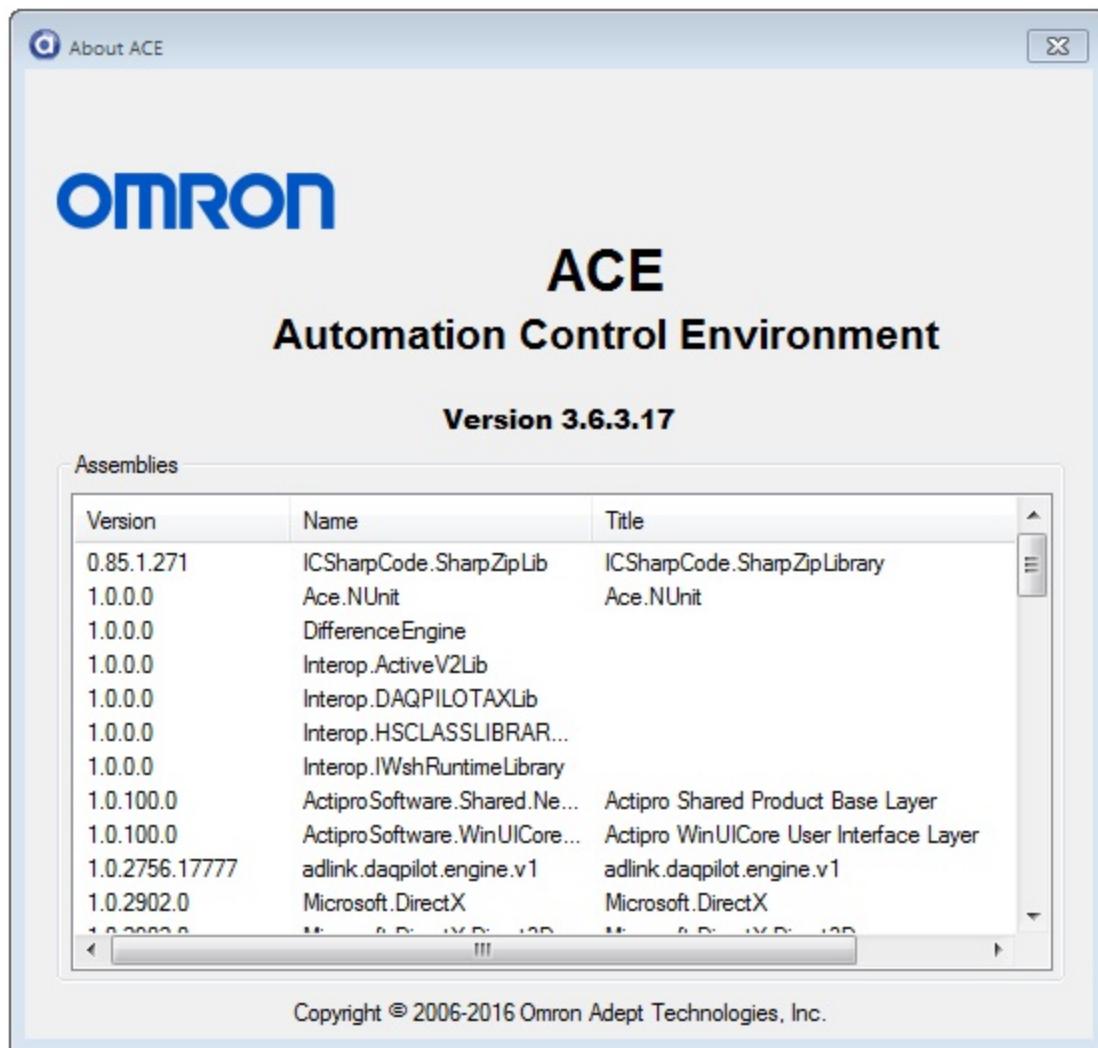
## About ACE

The About ACE window is a tool used to display information about the software version and assemblies (ACE software components) that are currently running.

To open the About ACE window, from the ACE menu, select:

**Help > About...**

The About ACE window opens.



*About ACE Window*

The About ACE window displays: the ACE version, and the version, file name and title of the ACE software assemblies. This window can be used to quickly check the installed software version and components, which can be used when talking with service personnel to troubleshoot problems.

**NOTE:** If you need to email the software information to a service person, use the System Diagnostic Summary on page 1128.

After you have finished viewing the information in the window, click the Close icon () to close it.

## Event Log

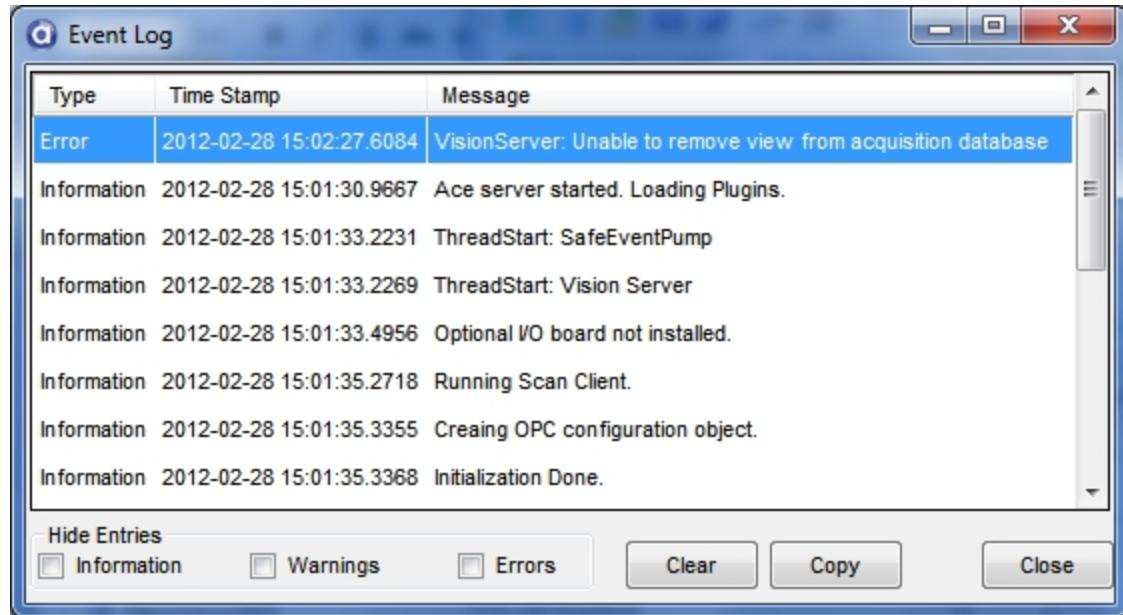
The Event Log is a tool used to display event messages that have been logged since system startup.

**NOTE:** In addition to the ACE Event Log, all ACE events are written to the Windows "Application" log. Therefore, if the ACE Event Log is no longer available (for example, someone has cleared it or ACE has been shut down) you can still view the ACE events in the Windows log file.

To open the Event Log, from the ACE menu, select:

**Help > Event Log.**

The Event Log opens.



### Event Log

The Event Log collects: the date stamp, the time stamp, and details about the particular event message, which can be used by service personnel to troubleshoot problems.

You can sort the events by clicking the Type, Time Stamp, or Message column headings. Each click will toggle the sort between ascending and descending order.

You can filter the events by selecting one or more of the filter options—selecting a filter hides the entries for the selected filter.

If you have a problem to report to field service, it is helpful to include the Event Log information in any communication. To copy the Event Log information and paste it into an email:

## Event Log

---

1. Click **Copy** to copy the contents of the log.
2. Open your email software and start a new email.
3. Click in the body of the new email where you want to past the information.
4. Press Ctrl+V (hold the Control key and press the V key) to paste the selected text into the email.

After you have finished viewing the log, click **OK** to close it.

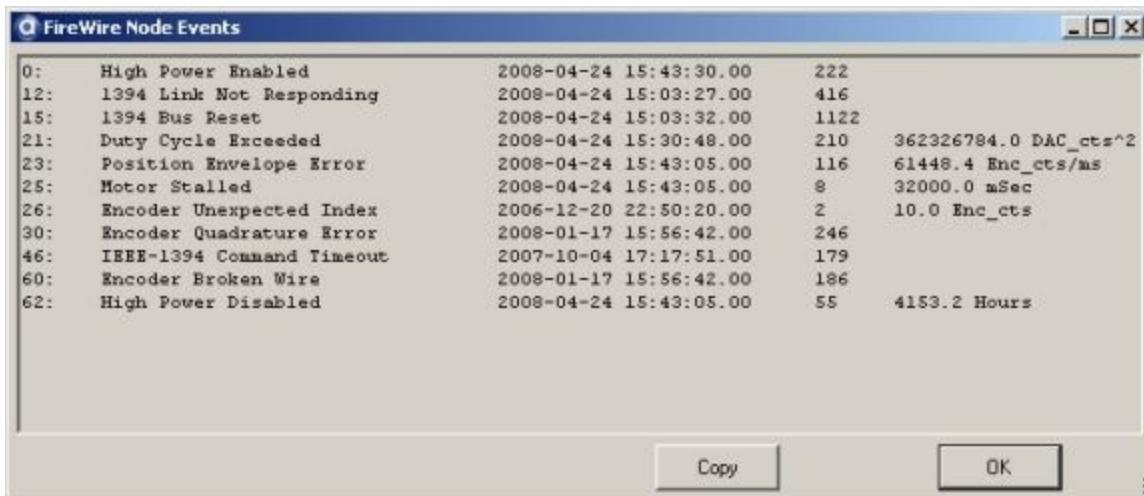
## FireWire Event Log

Each FireWire node contains an event log that can be displayed by right-clicking on any node in the FireWire Configure Nodes dialog box.

### Viewing the FireWire Event Log

To display the FireWire Event log:

1. Open the Controller Editor. (For details, see the topic [Controller Editor](#).)
2. On the "Controller" button group, click **Configure FireWire Nodes**. The Configure FireWire Nodes dialog box opens.
3. Right-click on the desired node. A menu displays.
4. From the menu, select **View Event Log**. The FireWire Events log opens.



Event ID	Description	Date	Time	Count	Message
0:	High Power Enabled	2008-04-24	15:43:30.00	222	
12:	1394 Link Not Responding	2008-04-24	15:03:27.00	416	
15:	1394 Bus Reset	2008-04-24	15:03:32.00	1122	
21:	Duty Cycle Exceeded	2008-04-24	15:30:48.00	210	362326784.0 DAC_cts^2
23:	Position Envelope Error	2008-04-24	15:43:05.00	116	61448.4 Enc_cts/ms
25:	Motor Stalled	2008-04-24	15:43:05.00	8	32000.0 mSec
26:	Encoder Unexpected Index	2006-12-20	22:50:20.00	2	10.0 Enc_cts
30:	Encoder Quadrature Error	2008-01-17	15:56:42.00	246	
46:	IEEE-1394 Command Timeout	2007-10-04	17:17:51.00	179	
60:	Encoder Broken Wire	2008-01-17	15:56:42.00	186	
62:	High Power Disabled	2008-04-24	15:43:05.00	55	4153.2 Hours

*FireWire Events Log*

The event log shows the event ID on the left, followed by the event description, the time and date of the last event, the event count, and an arbitrary message associated with the last event.

### Clearing the FireWire Event Log

To clear the FireWire Event log:

1. Open the Controller Editor. (For details, see the topic [Controller Editor](#).)
2. On the "Controller" button group, click **Configure FireWire Nodes**. The Configure FireWire Nodes dialog box opens.
3. Right-click on the desired node. A menu displays.

## Clearing the FireWire Event Log

---

4. From the menu, select **Clear Event Log**. A "Clear Events Log" prompt opens.
5. Click **OK** to clear the log; or, click **Cancel** to cancel the operation. The prompt closes.

## Pending Errors Log

The Pending Errors log displays any errors that have occurred since the last ENABLE POWER command. It is used to view error messages describing why the robot powered off (or failed to power on). To access the Pending Errors log:

1. Open the Controller Editor. (For details, see the topic [Controller Editor](#).)
2. From the menu, select **View > Pending Errors**. The Pending Errors log opens.



*Pending Errors Log*

The Pending Errors log shows the V+ error number on the left, followed by the error description. For example, if the emergency stop button is pressed, an indicator in ACE shows that power is off. The Pending Errors log can be examined to find out why the power is off, as shown in the figure above. For more information on V+ error messages and descriptions, see the topic "System Messages" in the *V+ Language Reference Guide*.

Other features of the Pending Errors log are as follows:

- The **Copy** button is used to copy the contents of the log, which is useful for pasting into an email or other document.
- The **OK** button is used to close the log.

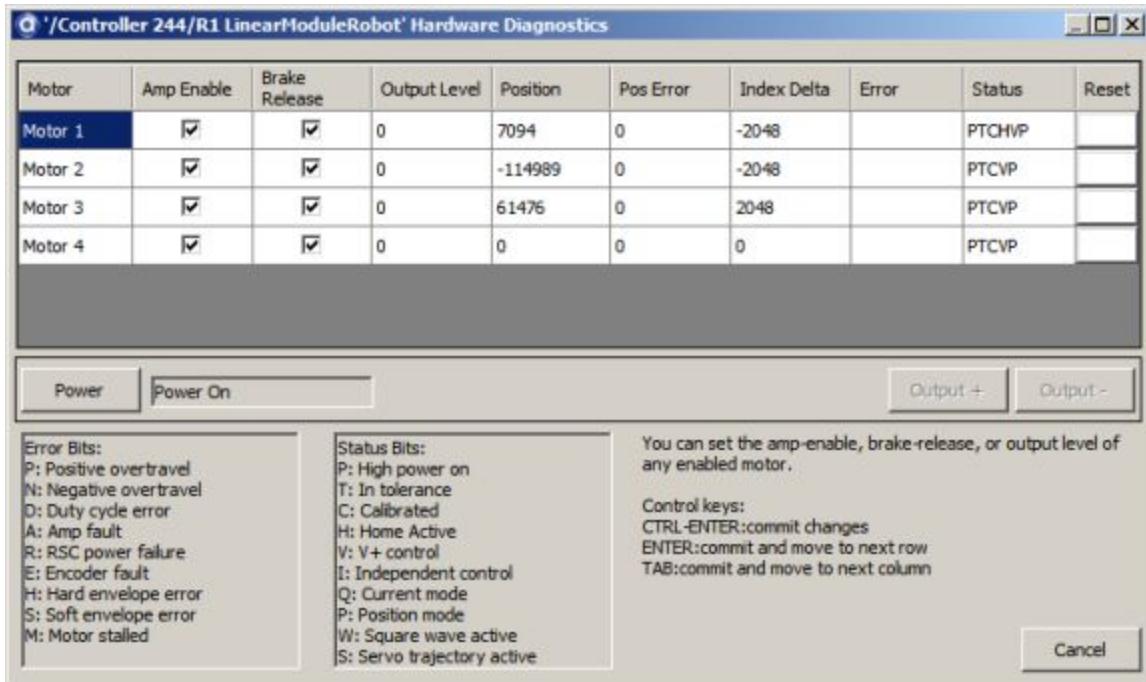
## Robots and Encoder Diagnostics

The ACE software provides access to robot diagnostics and external encoder diagnostics. Each of these are described in the following sections.

### Robot Diagnostics

Robot diagnostics are available from the Robot Diagnostics dialog box (see the figure [Robot Diagnostics Dialog](#)). To open the Robot Diagnostics dialog box:

1. Open the Robot Editor for the desired robot. (For details, see the topic [Robot Editor](#).)
2. From the Control menu, click **Hardware Diagnostics**. The Robot Diagnostics dialog box opens.



*Robot Diagnostics Dialog*

The Robot Diagnostics dialog box features are described in the following table:

Item	Description
Amp Enable	Enables / disables the amplifier for the selected motor.
Brake Release	Enables / disables the brake release for the selected motor.

Item	Description
Output Level	Specifies a commanded torque, which is used to test the operation of the selected motor. The range is from -32767 to 32767, or the range specified by the "Max Output Level" parameter for that motor in the Robot Editor.
Position	Displays the current position (in encoder counts) of the selected motor.
Pos Error	Displays the position error (in encoder counts) of the selected motor.
Index Delta	Displays the change (in encoder counts) from the previous latched zero index and the most recent latched zero index of the selected motor. Note that this is only useful with incremental encoders, to verify zero index spacing and proper encoder readings.
Error	Displays the errors for the selected motor, as follows:  P Positive overtravel N Negative overtravel D Duty cycle error A Amp fault R RSC power failure E Encoder fault H Hard envelope error S Soft envelope error M Motor stalled
Status	Displays the status of the selected motor, as follows:  P High power on T In tolerance C Calibrated H Home (sensor) active V V+ control I Independent control Q Current mode P Position mode W Square wave active S Servo trajectory active
Reset	Resets any encoder errors for the selected motor.
Power	Toggles the high power (the status field displays the current power)

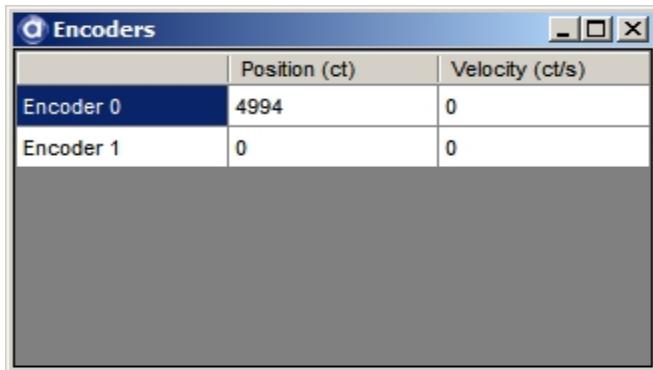
Item	Description
	state).
Output + Output -	DAC output controls. Click <b>Output +</b> to increase the DAC output to the selected motor; click <b>Output -</b> to decrease the DAC output to the selected motor.
Cancel	Closes the dialog box without saving any uncommitted change. Note that the following key combinations will commit a change:  Control + Commits the change at the current cell Enter Commits the change at the current cell and moves down to the next row Tab Commits the change at the current cell and moves to the next column

## External Encoder Diagnostics

**NOTE:** When emulation mode is enabled, this item contains additional features. For more details on Emulation Mode, see Emulation Mode on page 1089.

External encoder diagnostics are available from the External Encoder Diagnostics dialog box. To open the External Encoder Diagnostics dialog box:

1. Open the Controller Editor. (For details, see the topic [Controller Editor](#).)
2. On the "Controller" button group, click **Encoders**.



*Encoder Diagnostic Dialog*

Encoder Channel	Position (ct)	Velocity (ct/s)	Running
Encoder 0	522	100	<input checked="" type="checkbox"/>
Encoder 1	110	0	<input type="checkbox"/>
Encoder 2	96	0	<input type="checkbox"/>
Encoder 3	88	0	<input type="checkbox"/>

*Encoder Diagnostic Dialog in Emulation Mode*

The External Encoder Diagnostics dialog box shows a live display of the encoder position, in counts.

When emulation mode is enabled:

- a Running column is available, which can be used to turn on/off the corresponding encoder.
- the Velocity column can be used to specify the desired speed (counts per second).

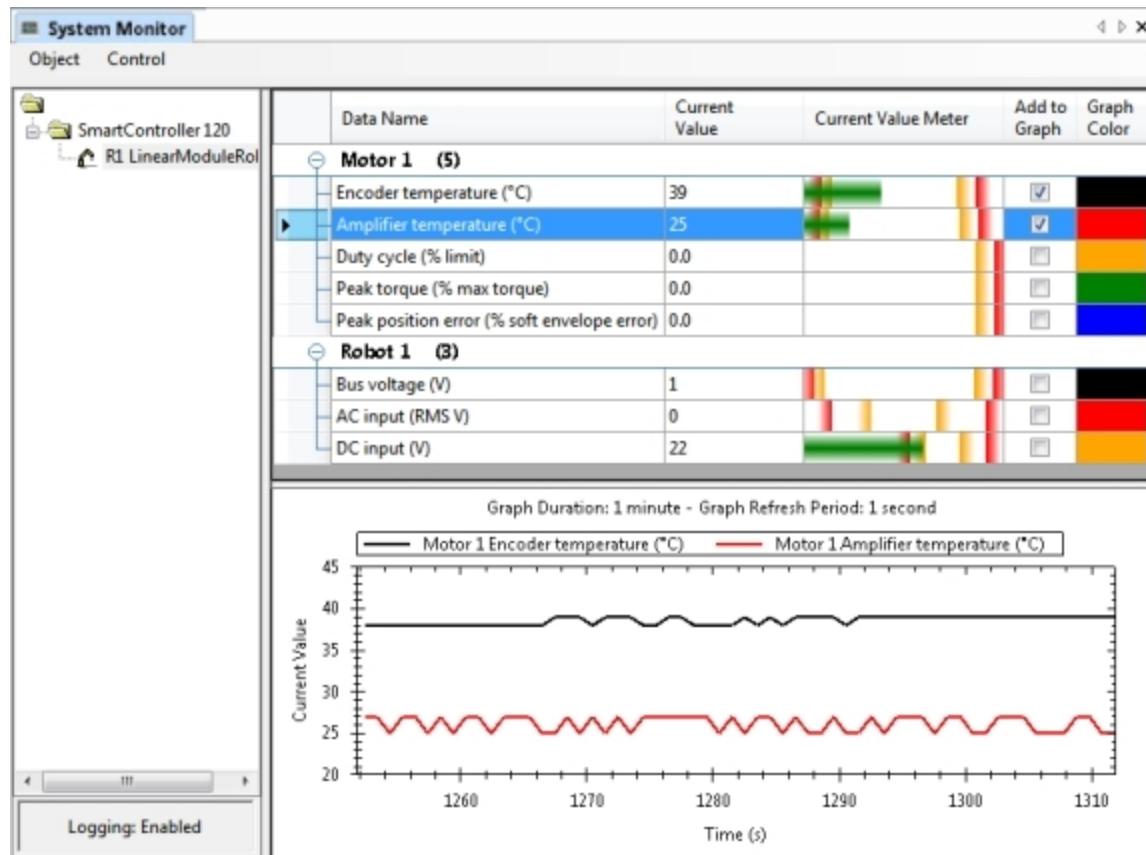
## System Monitor

**NOTE:** This feature operates differently in emulation mode. For details on emulation mode, see Emulation Mode on page 1089.

**NOTE:** The System Monitor is not available for Adept Cobra i600/800 and eVario 600/800 robots.

The System Monitor tool, shown in the following figure, is used to perform real-time monitoring of objects in the workspace. It can be used to monitor robot parameters or Process Manager statistics.

To open the System Monitor tool, on the View menu, click **System Monitor**. The System Monitor tool opens. You can select a robot or Process manager from the list in the left-hand pane to view the data for that item.



*System Monitor Tool*

The following robot parameters can be monitored:

## System Monitor

---

<b>Parameter</b>	<b>Description</b>
Amplifier bus voltage (V)	The current amplifier bus voltage for the robot. This should operate within the specified min/max warning limits (yellow bars), and never reach the min/max error limits (red bars).  If the value drops below the range minimum, this means that the motion is too hard or the AC input voltage is too low; if the value exceeds the range maximum, this means that the motion is too hard or the AC input voltage is too high. Lowering the motion speed (more than the acceleration) can help correct these issues.
AC input (V)	The current AC input voltage (220 VAC) for the robot. This should operate within the specified min/max warning limits (yellow bars), and never reach the min/max error limits (red bars). Running outside or close the limits may create envelope errors.
DC input (V)	The current DC input voltage (24 VDC) for the robot. This should operate within the specified min/max warning limits (yellow bars), and never reach the min/max error limits (red bars).
Base board temperature (°C)	The current temperature (°C) for the amp-in-base processor board. This should operate within the specified min/max warning limits (yellow bars), and never reach the min/max error limits (red bars).
Encoder temperature (°C)	The current encoder temperature (°C) for the selected motor. This should operate within the specified min/max warning limits (yellow bars), and never reach the min/max error limits (red bars).
Amplifier temperature (°C)	The current temperature (°C) for the motor amplifier. This should operate within the specified min/max warning limits (yellow bars), and never reach the min/max error limits (red bars).
Duty cycle (% limit)	The current duty cycle value, as a percentage, for the selected motor. This should operate below the specified max warning limit (yellow bar), and never reach the max error limit (red bar).
Harmonic Drive usage (%)	The current usage of the Harmonic Drive, as a percentage of design life, for the selected motor. This should operate below the specified max warning limit (yellow bar), and never reach the max error limit (red bar).  If the value is less than 100%, the maximum life for the Harmonic Drive will be extended; if the value exceeds 100%, the maximum life of the Harmonic Drive will be diminished. For example, a value of 150% means the time at which 10% of drives will have failed is 1.5 times shorter than the design life.
Peak torque (% max torque)	The peak torque, as a percentage based on maximum torque, for the selected motor. This should operate below the specified max

Parameter	Description
	warning limit (yellow bar), and never reach the max error limit (red bar).
Peak velocity (RPM)	The peak velocity, in rotations per minute, for the selected motor. This should operate below the specified max warning limit (yellow bar), and never reach the max error limit (red bar).
Peak position error (% soft envelope error)	The peak position error, as a percentage of soft envelope error, for the selected motor. This should operate below the specified max warning limit (yellow bar), and never reach the max error limit (red bar).

The following Process Manager Belt parameters can be monitored:

Parameter	Description
Instance Count	The number of instances that have been available since the last restart or reset.
Belt Velocity	The average velocity of the conveyor belt.
Instantaneous Instances	The instantaneous instances since the last restart or reset. This is calculated over the update period selected in the System Diagnostics settings. If the graph is updated 500 ms, it will tell you the instantaneous instances/minute over each 500 ms time segment.
Instances Per Minute	The average number of instances per minute.
Active Instances	The number of active instances on the belt.
Latch Faults	The number of latch faults since the last restart or reset.

The following Process Manager Process and Robot parameters can be monitored:

**NOTE:** The statistics, with the exception of Parts Not Processed and Targets Not Processed, are also exposed to the numeric variable object. This allows them to be exported to OPC. For details, see Variable Editor on page 131.

Parameter	Description
Idle Time (%)	The average idle time percentage of the total run time since the last restart or reset. ("Idle" is when the Process Manager is waiting on part or part target instances to process.)
Processing Time (%)	The average processing time percentage of the total run time since the last restart or reset. ("Processing" is when the Process Manager

## Menu Items

---

Parameter	Description
	is actively processing part or part target instances.)
Average Total Time (ms)	For the Process Manager group only. The <i>average</i> total time for all robots. Other fields, such as Parts Processed/Not Processed and Targets Processed/Not Processed, show a <i>summation</i> for all robots.
Parts Per Minute	The average number of parts per minute. When viewing the Process Manager group, this is a <i>summation</i> for all robots.
Targets Per Minute	The number of targets per minute. When viewing the Process Manager group, this is a <i>summation</i> for all robots.
Parts Not Processed	The number of parts not processed since the last restart or reset. When viewing the Process Manager group, this is a <i>summation</i> for all robots.
Targets Not Processed	The number of targets not processed since the last restart or reset. When viewing the Process Manager group, this is a <i>summation</i> for all robots.
Parts Processed	The number of parts processed since the last restart or reset. When viewing the Process Manager group, this is a <i>summation</i> for all robots.
Targets Processed	The number of targets processed since the last restart or reset. When viewing the Process Manager group, this is a <i>summation</i> for all robots.

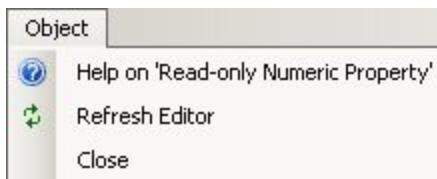
The following Process Manager Process Strategy parameters can be monitored:

Parameter	Description
Average Allocation Time (ms)	The average time it takes to run the allocation algorithm for allocating all parts and part targets.

## Menu Items

This section describes the selections available from the Variable editor menu.

### Object Menu



#### Help on

Displays the online help for the Variable editor.

#### Refresh Editor

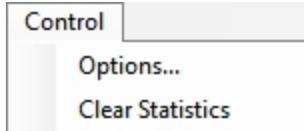
Refreshes the contents of the Variable editor window.

#### Close

---

Closes the Variable editor.

### **Control Menu**



#### **Options**

Displays the Options dialog, which is used to:

- Enable the logging feature
- Set the graph duration
- Set the refresh period

#### **Clear Statistics**

When a Process Manager object is selected in the left-hand pane, clears (resets) the statistics for that Process Manager.

The System Monitor tool sections are described in the following table.

Item	Description
Selection tree	This shows the controllers and robots currently in the ACE workspace. Click the robot name to select it for monitoring.
Data columns	<p>This area shows the robot parameters that are being monitored, their current values, and which ones are selected for plotting on the graph (see the section Adding and Removing Graph Items). Right-click on any of the column headings to display the following menu:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <span>Select Columns...</span>  <span>Reset Worst Status</span> </div> <ul style="list-style-type: none"> <li>• Use the Select Columns menu item to add/remove columns from the data area.</li> </ul> <div style="border: 1px solid #ccc; width: fit-content; height: fit-content; margin-top: 10px;"> <p>The dialog box has a title bar "Select Columns" and a "Columns" section containing a list of checkboxes. The checkboxes are:</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Lower Limit</li> <li><input type="checkbox"/> Upper Limit</li> <li><input type="checkbox"/> Current Status</li> <li><input type="checkbox"/> Worst Status</li> <li><input checked="" type="checkbox"/> Current Value Meter</li> <li><input checked="" type="checkbox"/> Add to Graph</li> <li><input checked="" type="checkbox"/> Graph Color</li> </ul> <p>At the bottom are "OK" and "Cancel" buttons.</p> </div>

Item	Description
	<p><i>Select Columns Dialog</i></p> <ul style="list-style-type: none"> <li>• Use the Reset Worst Status menu item to reset the labels in the Worst Status column.</li> </ul>
Graph area	<p>Provides a graphical plot of the data selected in the Add to Graph column. To change the plot color of a data item, click its Graph Color cell, and then select the desired color from the Color tool.</p>  <p><i>Color Dialog</i></p> <p>You can zoom (magnify) an area of the graph by dragging the mouse pointer to define the zoom area.</p> <p>You can pan the graph by holding the Ctrl key while dragging the mouse pointer.</p> <p>You can right-click on the graph to display a menu of options, which are described in the following table.</p>

Item	Description	
	Item	Description
Copy	Copy	Copy the image in the graph area.
Save Image As...	Save Image As...	Save the graph image as a file.
Page Setup...	Page Setup...	Opens the Page Setup dialog box for setting the print page size, orientation, and margins.
Print...	Print...	Opens the Print dialog box for printing the graph.
Show Point Values	Show Point Values	When selected, place the cursor at a point on the graph to show the values of that point.
Un-Zoom	Un-Zoom	Returns the graph to its original magnification.
Undo All Zoom/Pan	Undo All Zoom/Pan	Returns the graph to its original magnification and origin.
Set Scale to Default	Set Scale to Default	Returns the graph scale to its default setting.
You can also change the graphed items, graph duration (scale) and refresh period (sample rate). For details, see the following sections.		

## Adding and Removing Graph Items

The Add to Graph column is used to specify items for the graph. Each data item being monitored is represented by a colored line. You can select the color for a data item by clicking its Graph Color cell, and then selecting the desired color from the Color dialog.

**NOTE:** You can plot multiple data items on the graph.

For example, in the following figure, the Motor 1 encoder temperature and Motor 1 amplifier temperature are selected for the graph, the remaining Motor 1 data items are not shown on the graph.

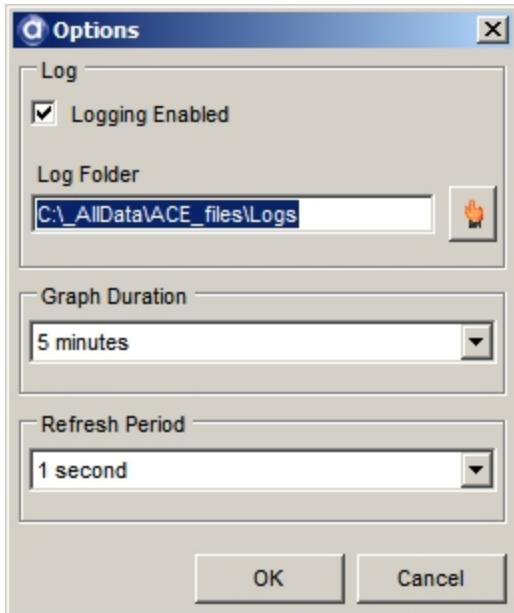
Data Name	Current Value	Current Value Meter	Add to Graph	Graph Color
<b>Motor 1 (5)</b>				
Encoder temperature (°C)	39		<input checked="" type="checkbox"/>	
Amplifier temperature (°C)	27		<input checked="" type="checkbox"/>	
Duty cycle (% limit)	0.0		<input type="checkbox"/>	
Peak torque (% max torque)	0.0		<input type="checkbox"/>	
Peak position error (% soft envelope error)	0.0		<input type="checkbox"/>	

*Data Items Selected for the Graph*

Also, in the previous figure, note the Lower/Upper Warning Limits (indicated by the yellow vertical bars), and the Lower/Upper Error Limits (indicated by the red vertical bars).

## Setting the Options

The Options dialog allows you to enable data logging and to set the graph duration and refresh period. To access the Options dialog, from the System Monitor tool menu, select **Control > Options**. The Options dialog opens, as shown in the following figure.



*Options Dialog*

The options are described in the following table.

Item	Description
Logging Enabled	When checked, data logging is enabled. The captured data is written to a file located in the specified Log Folder.  Logging creates a .CSV file with all the data plus a time stamp. The size of this file is limited to 10 MB. When the 10 MB limit is reached, the file is renamed and a new file is started. When the new file reaches 10 MB, the first file is deleted, the second file is renamed, and a new file is started. Only two files are kept at the same time.
Log Folder	Specifies the folder where the log file will be stored.
Graph Duration	Specifies the graph duration. The range is from 1 minute to 1 week. For

Item	Description
	example, if 1 minute is selected, the Time scale on the graph will show 1 minute of data.
Refresh Period	Specifies the refresh period (how often the data will be sampled). The range is from 500 milliseconds to 1 minute.

## Resetting the Maximum Recorded Values

The System Monitor tool has a feature that allows you to reset the maximum recorded values for the robot being monitored. This allows you to make performance adjustments and then see if those adjustments have improved performance or not. The values that can be reset are:

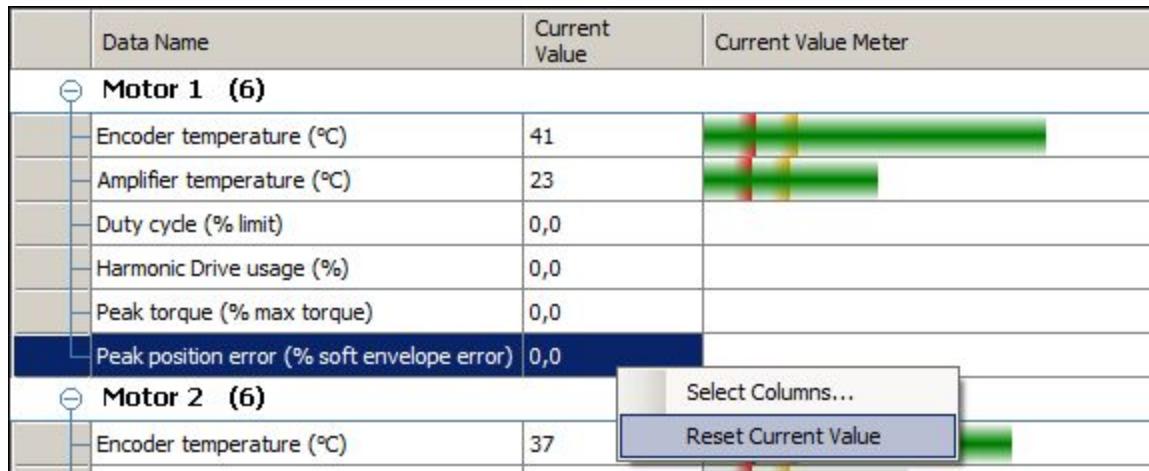
- Peak Torque
- Peak Velocity
- Peak Position Error

To reset a value:

1. Right-click on the row that contains the parameter you wish to reset. A menu opens.
2. From the menu, select Reset Current Value, as shown in the following figure.

**NOTE:** If the selected row cannot be reset, the Reset Current Value selection will be dimmed (not available).

3. The recorded value is reset to 0.



Reset Current Value Menu Item

## **Monitoring Process Manager Statistics**

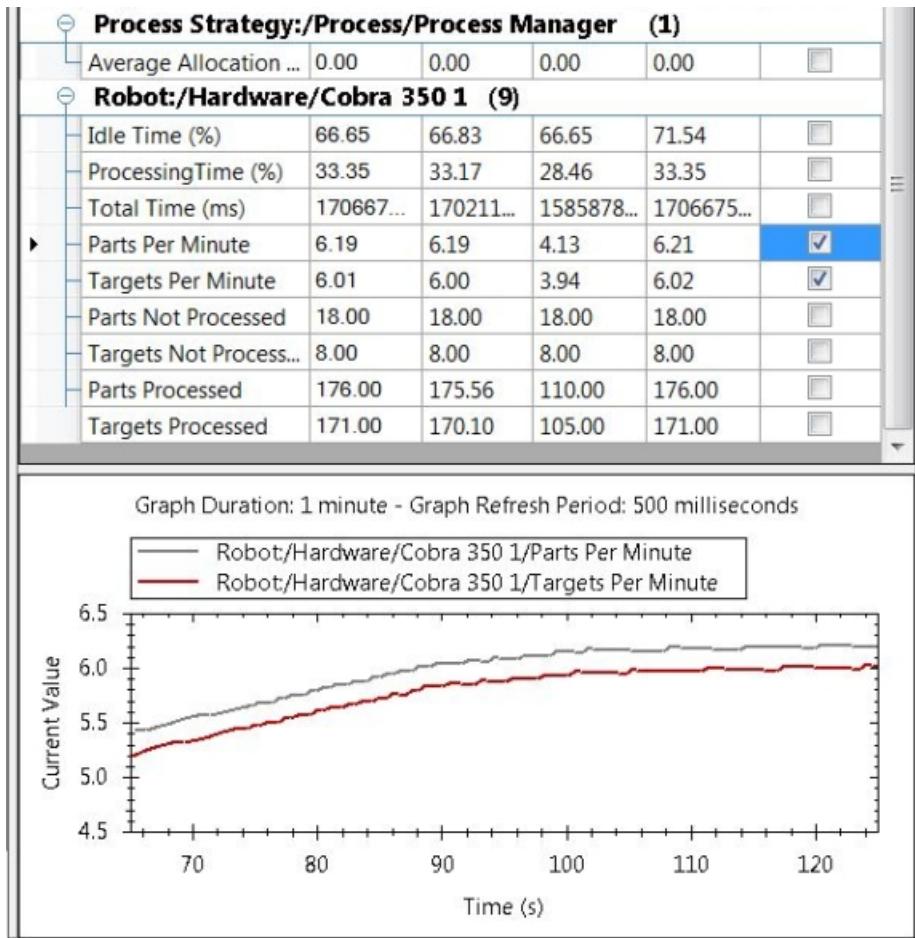
The System Monitor tool has a feature that allows you to monitor Process Manager statistics, such as:

- Parts Per Minute
- Targets Per Minute
- Processing Time
- and many more.

To monitor the Process Manager statistics:

1. Locate the statistic you wish to monitor.
2. Select the checkbox in the right-hand column of that row. When the box contains a check mark, that statistic will be plotted on the graph.

The selected items are plotted on the graph, as shown in the following figure.



#### Selected Process Manager Items on Graph

Note that the Control menu item includes a Clear Statistics option, which clears (resets) the statistics for the selected Process Manager. See Menu Items on page 1120.

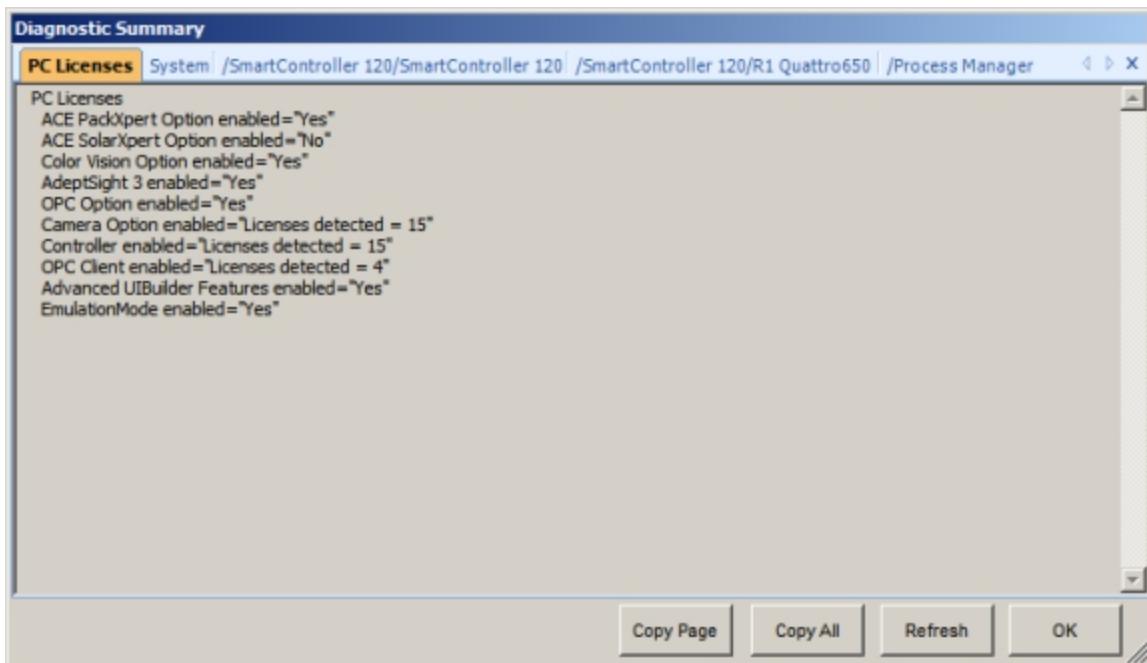
## System Diagnostic Summary

The Diagnostic Summary is a tool used to collect information on the installed software components, Adept controller hardware and software configuration, and other system information.

To open the Diagnostic Summary, from the ACE menu, select:

**Help > Diagnostic Summary**

The Diagnostic Summary opens.



### *Diagnostic Summary*

The Diagnostic Summary collects: the version numbers of software, firmware, and hardware; serial numbers; USB hardware key (dongle) licenses; and other system information, which can be used by service personnel to troubleshoot problems.

The Diagnostic Summary sorts the information into tabbed categories, to make it easier to find the information related to a particular item in your application.

If you have a problem to report to field service, it is helpful to include the Diagnostic Summary in any communication. To copy the Diagnostic Summary information and paste it into an email:

1. Click **Copy Page** to copy the contents of the selected tab; click **Copy All** to copy the contents of all tabs.
2. Open your email software and start a new email.

## System Diagnostic Summary

---

3. Click in the body of the new email where you want to past the information.
4. Press Ctrl+V (hold the Control key and press the V key) to paste the selected text into the email.

After you have finished viewing the log, click **OK** to close it.

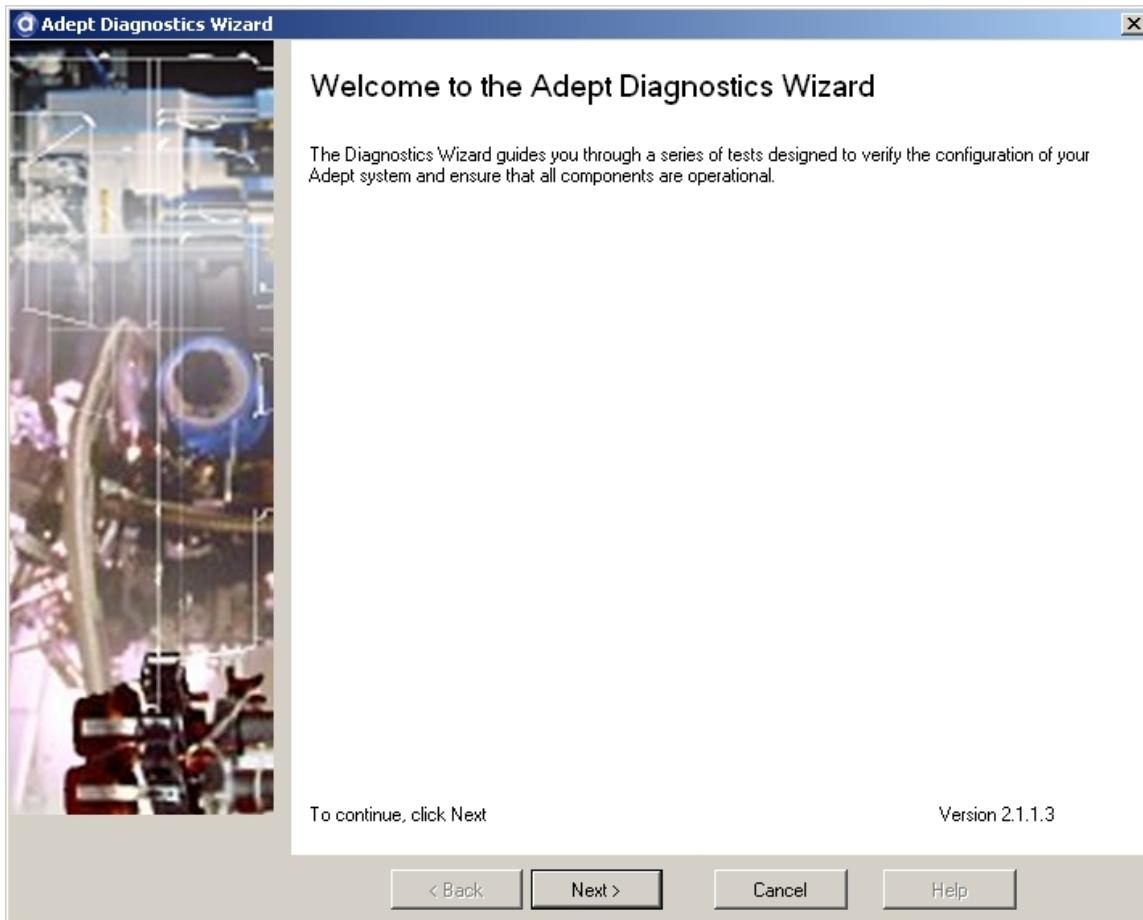
## Cobra i600/i800 Robot Diagnostics

The ACE software provides a diagnostic wizard for the Cobra i600/i800 robot. The wizard provides verification of the safety system, robot motion, and I/O communications.

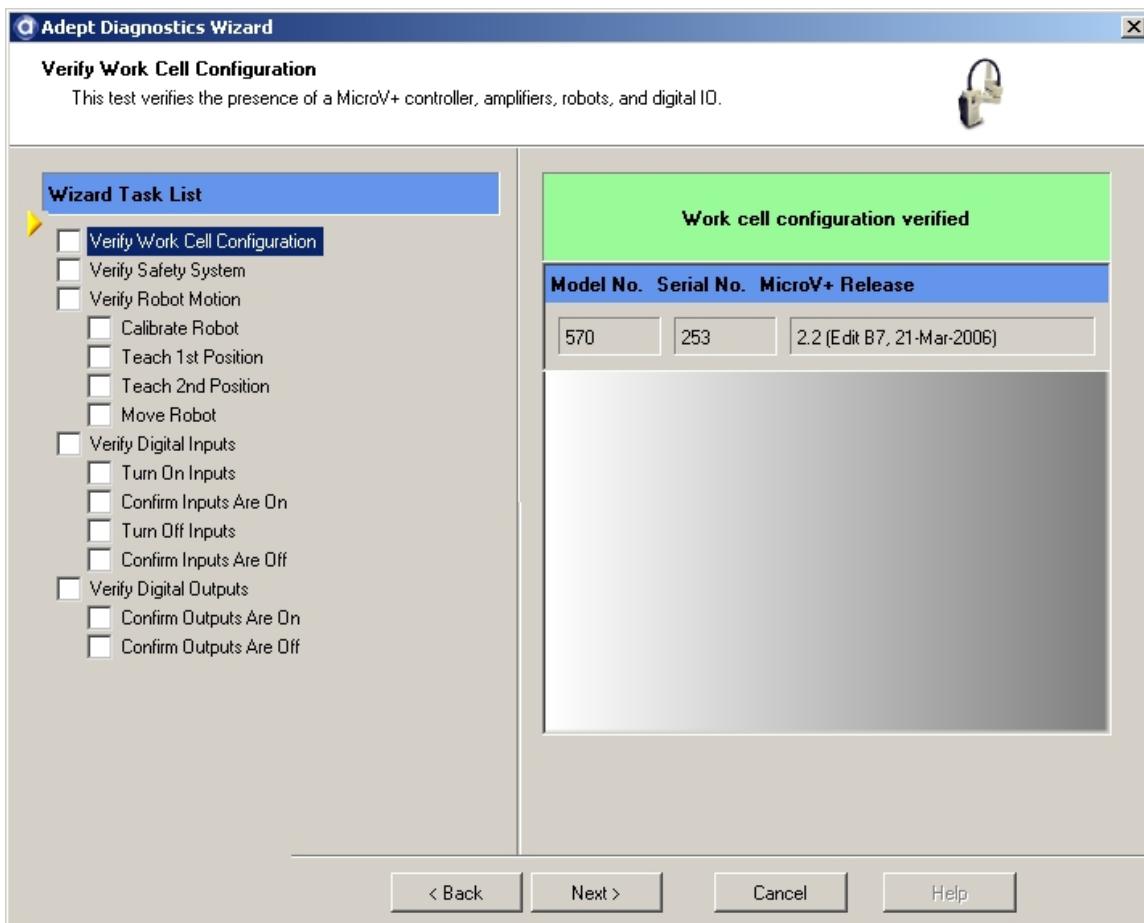
### Using the Robot Diagnostics Wizard

Robot diagnostics are available from the Configuration menu item on the Cobra i-Series Robot editor. To open the Robot Diagnostics wizard:

1. Select **Configuration > Robot Diagnostics**. The Welcome screen is displayed.



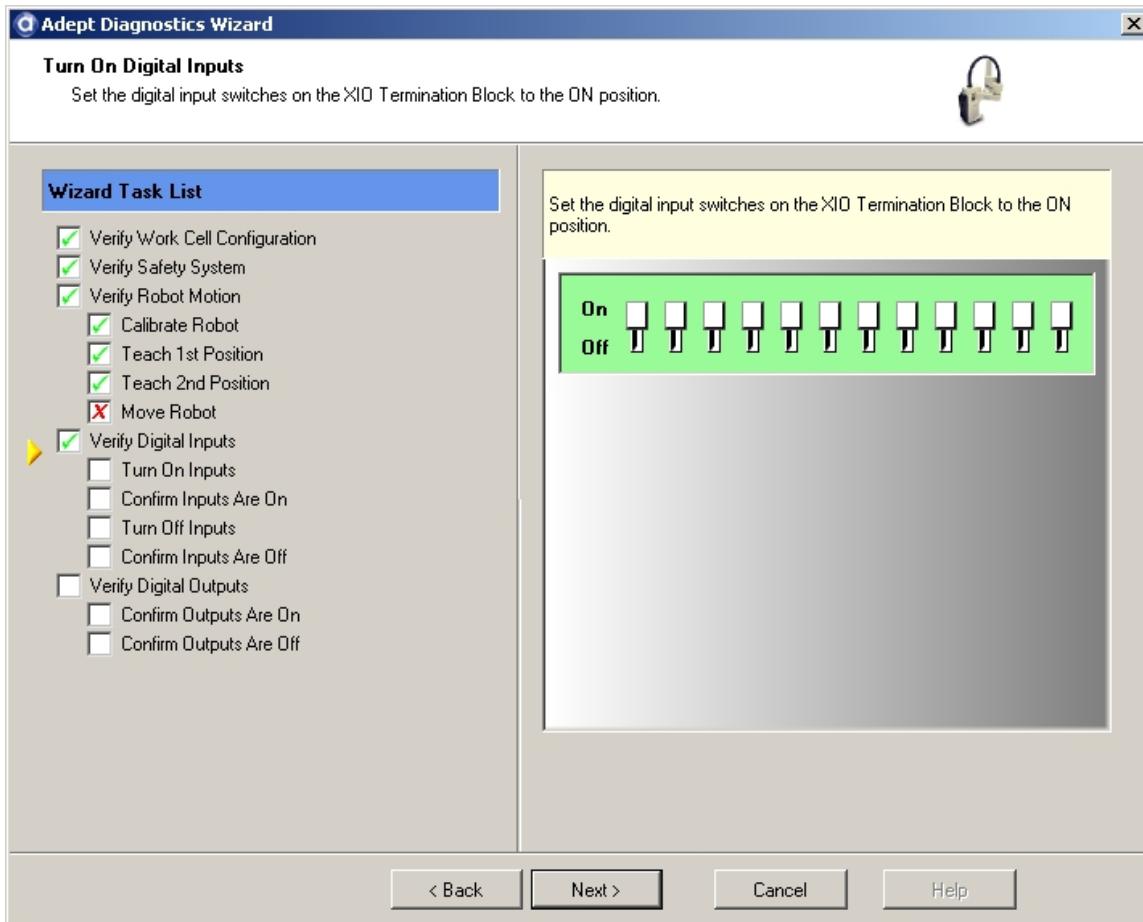
2. Click **Next** to proceed. The Verify Work Cell Configuration screen displays.



The Wizard screen has two main areas: the Wizard Task List (left pane), which shows the current step in the diagnostics process; and the test information/control area (right pane), which contains controls, status messages, and information for completing the current test.

3. To complete the diagnostics, simply follow the instructions in the right pane. As each step is completed the task is marked in the Wizard Task List: a checkmark indicates the test completed successfully, an "x" indicates the test was canceled or failed, as shown in the following screen. The currently-selected test is marked with a yellow cursor.

## Robot Diagnostics Wizard Test Summary



## Robot Diagnostics Wizard Test Summary

The Robot Diagnostics Wizard tests are described in the following table:

Test	Description
Verify Work Cell Configuration	Verifies the presence of a MicroV+ controller, amplifiers, robots, and digital I/O. The results are displayed in the test information/control area.
Verify Safety System	Verifies that the Front Panel, Manual Control Pendant (MCP or T1/T2), and Power are properly connected.
Verify Robot Motion	Verifies the safe and proper movement of the robot within the workcell.
Calibrate Robot	Verifies that the robot can be successfully calibrated.

Test	Description
Teach 1st Position	Select the start point for the robot motion. The joint positions are shown in the test information/control area.  <b>NOTE:</b> The robot power is disabled so it can be manually moved to the desired position.
Teach 2nd Position	Select the end point for the robot motion. The joint positions are shown in the test information/control area.  <b>NOTE:</b> The robot power is disabled so it can be manually moved to the desired position.
Move Robot	Verifies the robot can safely (and repeatedly) move between two locations within the workcell. A simple control panel is provided, which is used to Start, Pause, Step, and Stop the robot motion.
Verify Digital Inputs	Optional test procedure that verifies the digital inputs are properly connected.  <b>CAUTION:</b> This test requires an XIO Termination Block. If you do not have it installed, the test may result in damage to your system.
Turn On Inputs	Set the digital input switches on the XIO Termination Block to the ON position.
Confirm Inputs Are On	Verifies that the digital inputs are in the ON state.
Turn Off Inputs	Set the digital input switches on the XIO Termination Block to the OFF position.
Confirm Inputs Are Off	Verifies that the digital inputs are in the OFF state.
Verify Digital Outputs	Optional test procedure that verifies the digital outputs are properly connected.  <b>CAUTION:</b> This test requires an XIO Termination Block. If you do not have it installed, the test may result in damage to your system.
Confirm Outputs Are On	Verifies that the digital outputs can be properly set to the ON state.
Confirm Outputs Are Off	Verifies that the digital outputs can be properly set to the OFF state.



## Extending ACE

ACE is designed to be extended by users. There are two basic models for this:

- Invoking ACE from custom applications.
- Building libraries that "plug-in" to ACE.

There is a demonstration solution in the ZIP file "AceDemo.zip". You can access the file in the ACE installation folder on your computer. If you used the default installation folder, you will find the ZIP file in:

C:\Program Files\Omron\ACE

The demonstration solution can be loaded into Visual Studio (Express or Professional) to see working examples of the plug-in capabilities. The projects are available in Visual C# or Visual Basic .NET format.

There is a ReadMe.rtf file in the root folder, which describes the demonstration solution and provides a table of contents to the projects. Each project includes a ReadMe.rtf file that provides details on the project and shows what the graphical user interface (GUI) should look like when the application is running. The demonstration solution includes the following Visual Studio projects:

- CustomAceObject: Demonstrates how to create a simple custom ACE object, its associated editor, and a GUI plugin that gets access to the custom ACE object. This can be used as a basis for customizing the ACE software.
- CustomVisionTool: Demonstrates how to create a custom vision tool that appears as a native tool, which can be added to the workspace.
- GuiPlugin: Demonstrates how to create a GUI plug-in that smoothly integrates into the ACE user interface, including toolbar and menu buttons and a dockable control.
- OperatorInterface: Demonstrates a functioning user interface that an operator might use to control a machine.
- RemoteAccess: Demonstrates how to create a custom executable that connects to a remote ACE instance, on a different PC or in a different process on the same PC.
- VPlusEmulationRobotSample: Demonstrates how a V+ centric application can push V+ programs to a controller and react to events generated in those V+ programs.

## **Appendix 1: Using OPC through DCOM**

This chapter describes the process for configuring OPC communication through DCOM.

Overview .....	1137
Applicable Products .....	1137
Notes .....	1137
Summary .....	1137
Procedure .....	1139
Reference Documents .....	1167

## Overview

Establishing OPC communication between devices requires proper configuration of the Microsoft Windows operating system security settings to allow this type of access. The configuration is accomplished using the Distributed Component Object Model (DCOM), which is a Microsoft technology. Due to the distributed architecture of the ACE and SmartVision EX servers, it may be necessary to configure multiple PCs using DCOM, when OPC communication is enabled. This document provides a detailed procedure for configuring these settings using DCOM.

## Applicable Products

- ACE with OPC license enabled
- Third party OPC servers / clients
- SmartVision EX running Windows XP embedded
- Any PC running Windows XP SP2

## Notes

The SmartVision EX product has a File Based Write Filter (FBWF) that creates a RAM buffer to protect the operating system from corruption. The FBFW must be disabled before continuing with this procedure. For details on disabling the FBFW, see the *Adept SmartVision EX User's Guide*.

If you follow this procedure and are still having trouble with OPC communications, it may be useful to temporarily disable the Windows Firewall for troubleshooting purposes (consult your Windows operating system online help for instructions). In most cases, improper configuration of the Windows Firewall is the root cause for not being able to establish communication between OPC servers and clients. Once you have established communications, you can re-enable the Windows Firewall with the proper exceptions, as described in Step 7: Configure the Windows Firewall to allow exceptions on page 1162

## Summary

This section provides an summary of the steps you will follow to complete the configuration of DCOM. Each step is detailed in the next section.

**NOTE:** The changes described in this procedure must be made for all computers using OPC communications, which includes clients, servers, touch screens, etc.

Step 1: Create a computer name and define the workgroup / domain on page 1139

Step 2: Create matching user accounts and passwords on page 1141

Step 3: Configure the DCOM settings for system-wide security parameters on page 1142

Step 4: Configure the DCOM settings for all computers using OPC on page 1151

Step 5: Configure OpcEnum for all computers using OPC on page 1158

Step 6: Activate the Guest account on page 1160

Step 7: Configure the Windows Firewall to allow exceptions on page 1162

## Summary

---

Step 8: Restart all computers and OPC clients / servers on page 1166

## Procedure

The following steps describe the procedure to configure the settings using DCOM. It assumes that you have a working knowledge of the Microsoft Windows operating system. For more information on the Windows operating system, please see the Help and Support link (Microsoft Windows online help file), which can be accessed from the Windows Start menu.

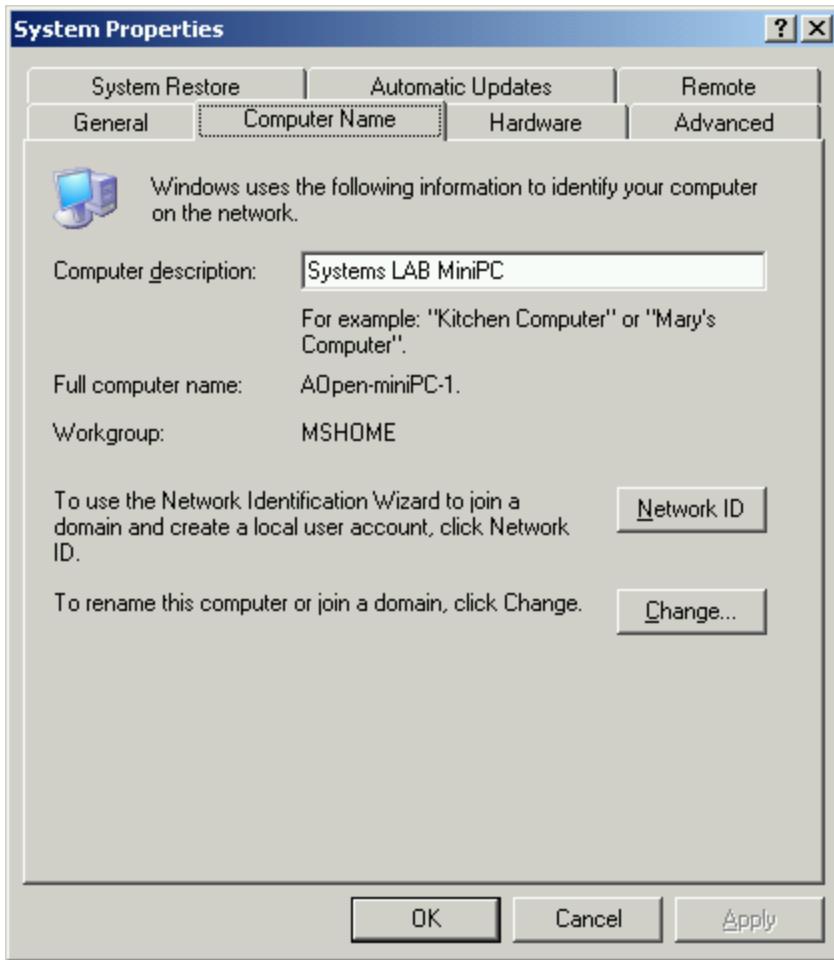
### **Step 1: Create a computer name and define the workgroup / domain**

For OPC to communicate properly, both the client and server computers need to have identical user accounts. These accounts need a user name defined with something other than the Windows default "Admin" and the password cannot be left blank. Additionally, both the client and server need to be in the same workgroup or domain.

1. Locate the My Computer icon on your Windows Start menu or your Windows desktop.
2. Right-click the My Computer icon and select Properties from the menu. The System Properties window opens.

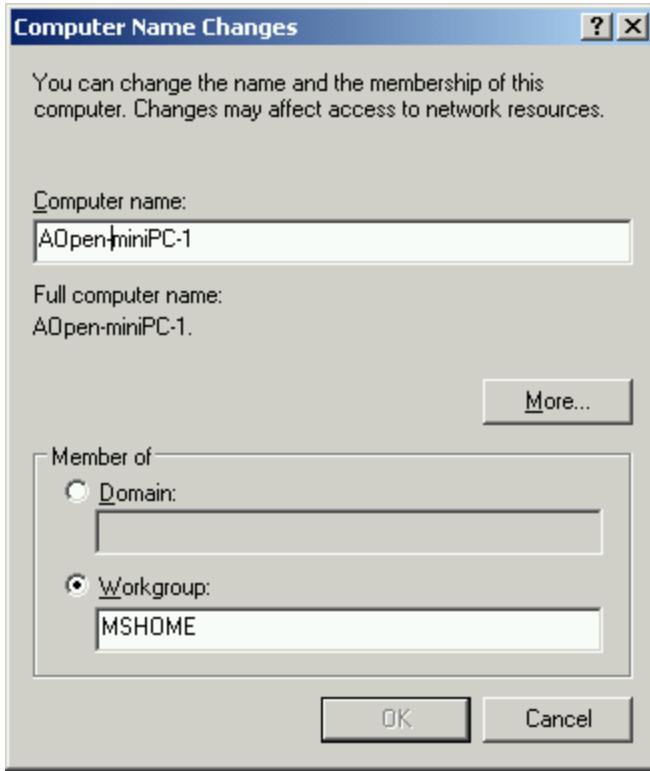
## Step 1: Create a computer name and define the workgroup / domain

---



3. Select the Computer Name tab.
4. Click **Change** to modify the computer name and workgroup
5. Change the computer name. This makes it easy to identify the computer on a network.
6. Define the workgroup. MSHOME is a Windows default, and is acceptable to use if no company-defined workgroup or domain exists.

**NOTE:** The workgroup or domain must be the same for all computers.



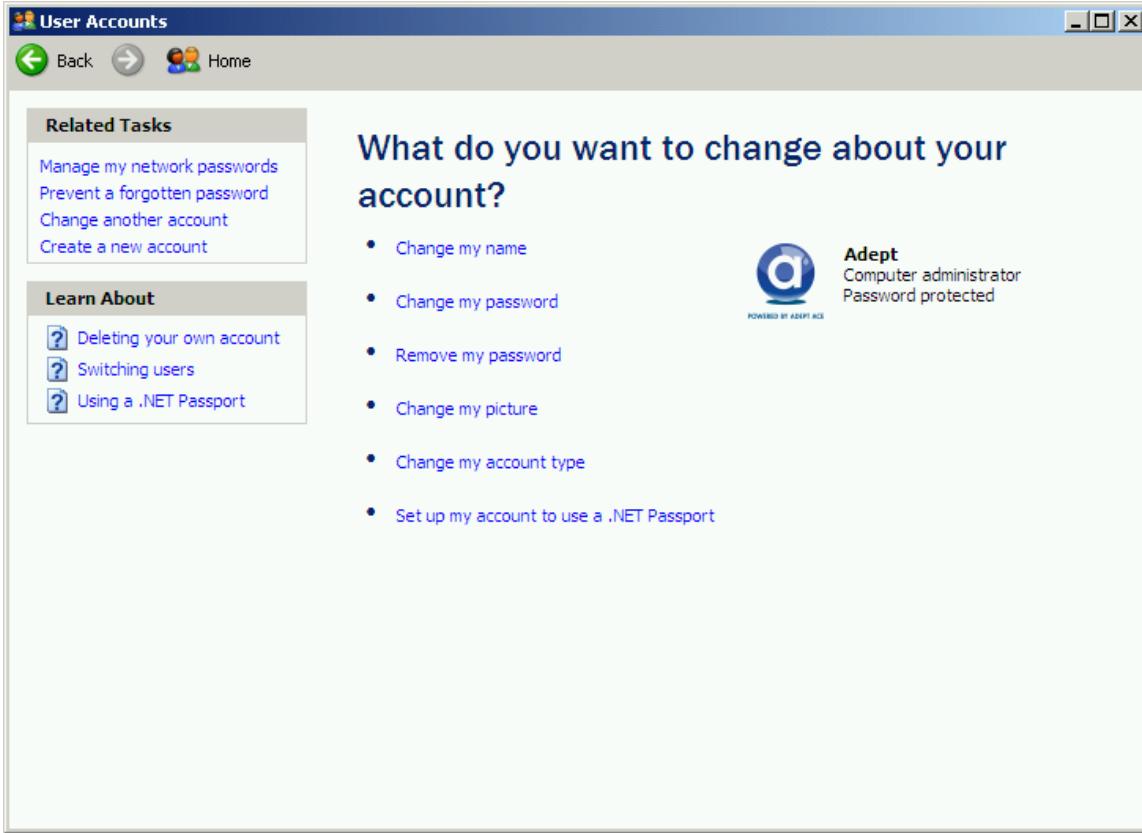
## Step 2: Create matching user accounts and passwords

For OPC to communicate properly, both the client and server computers need to have identical user accounts. These accounts need a user name defined with something other than the Windows default "Admin" name. The password cannot be left blank.

### Step 3: Configure the DCOM settings for system-wide security parameters

---

1. From the Windows Start menu, select Control Panel and open User Accounts

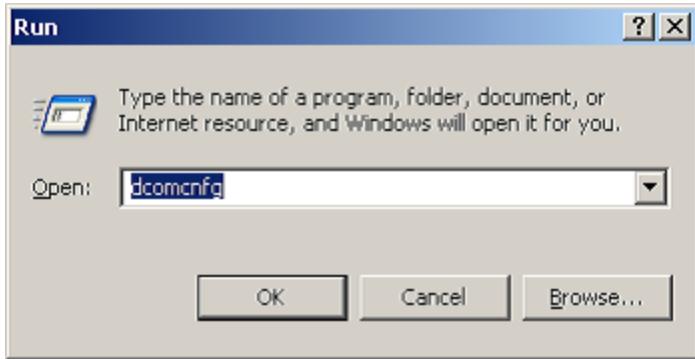


2. Choose the administrator/default account
3. Change the name: Adept
4. Change the password: robots

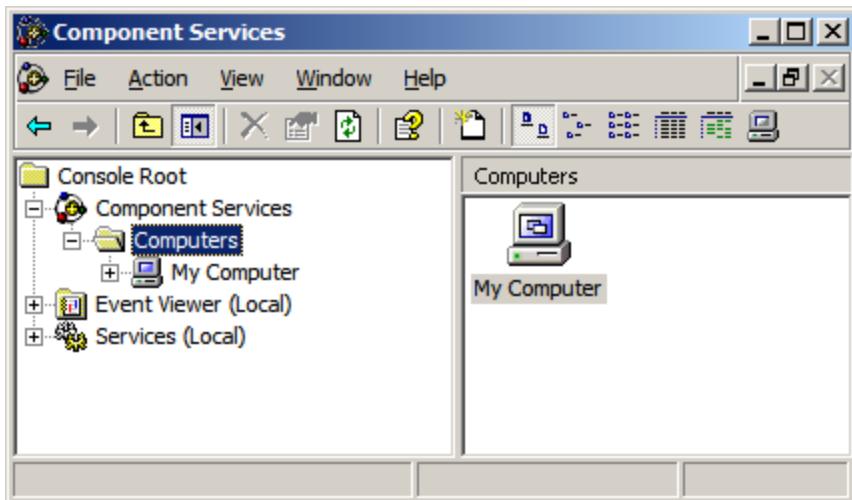
**NOTE:** The user name and password must be the same for all computers communicating through OPC. All clients and servers must have the same user name and login accounts or be members of the same user group.

### Step 3: Configure the DCOM settings for system-wide security parameters

1. Open DCOM configuration
  - a. Log in to the PC using an administrator account.
  - b. From the Windows Start menu, select Run. The Run dialog opens.
  - c. In the Open field, type "dcomcnfg" (without quote characters).



- d. Click **OK** to run the command. The Component Services window opens, as shown in the following figure.
2. Configure DCOM
  - a. In the tree-view pane of the Component Services window, select:  
**Console Root > Component Services > Computers > My Computer**  
The My Computer icon opens in the right-hand pane of the window.

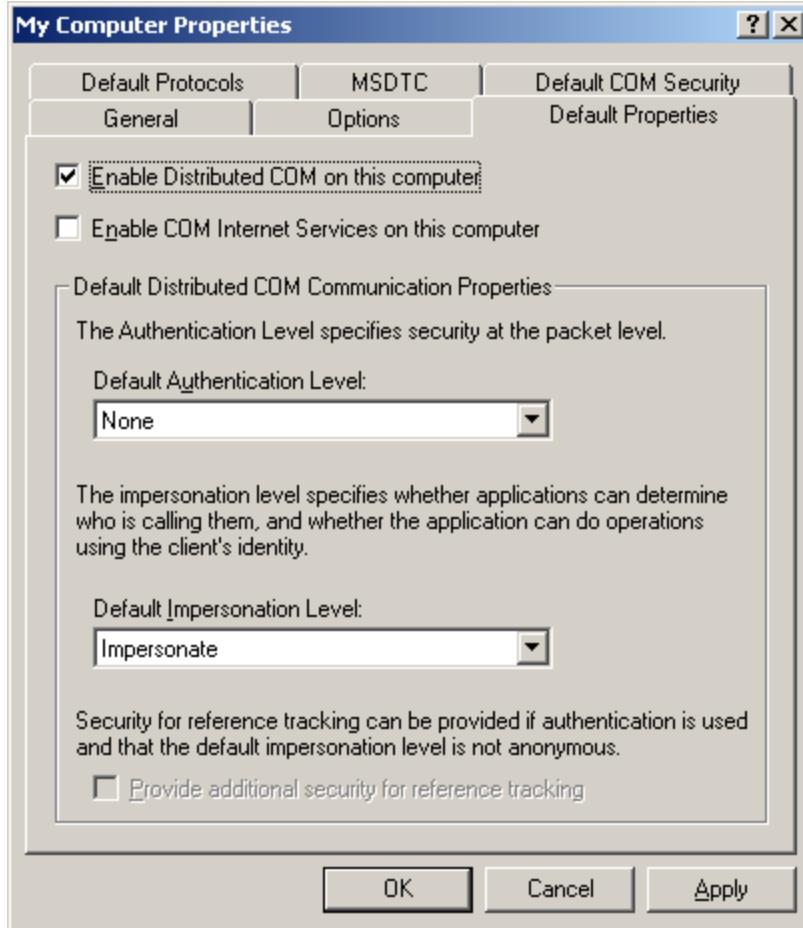


- b. Right-click the My Computer icon and select Properties from the menu. The My Computer Properties window opens.
3. Configure the Default Properties  
In the My Computer Properties window, select:
  - a. Default Properties tab
  - b. Enable Distributed COM on this computer
  - c. Default Authentication Level: None
  - d. Default Impersonation Level: Impersonate

### Step 3: Configure the DCOM settings for system-wide security parameters

---

The selected items are shown in the following figure.



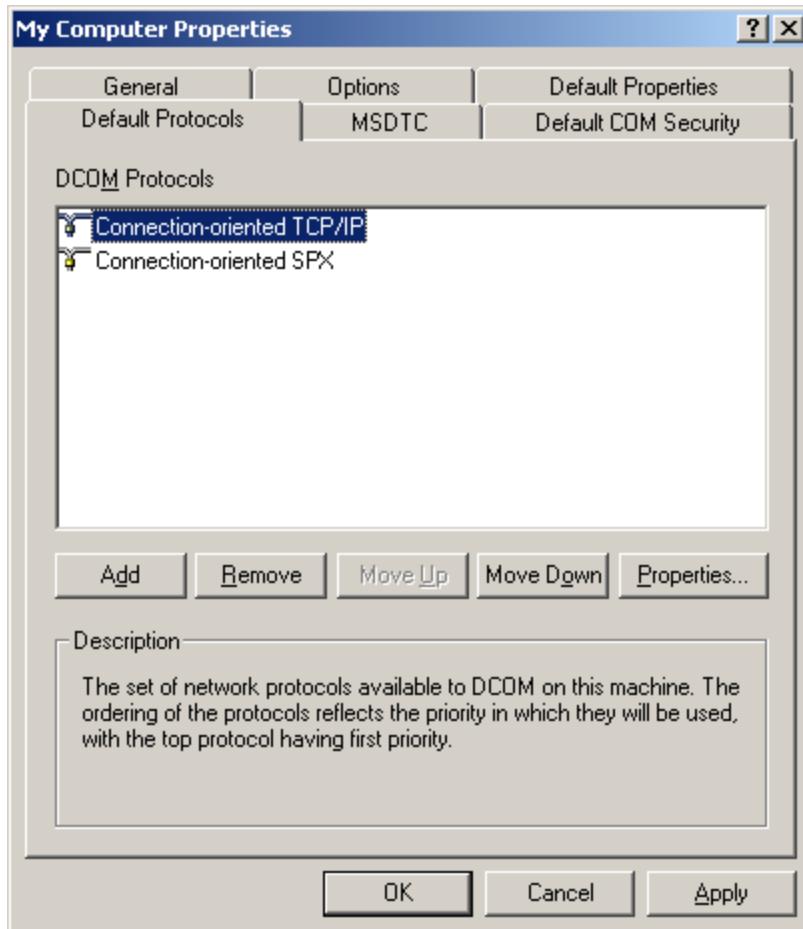
#### 4. Configure the Default Protocols

In the My Computer Properties window, select:

- a. Default Protocols tab
- b. Move "Connection-oriented TCP/IP" to the first position, as shown in the following figure.

Step 3: Configure the DCOM settings for system-wide security parameters

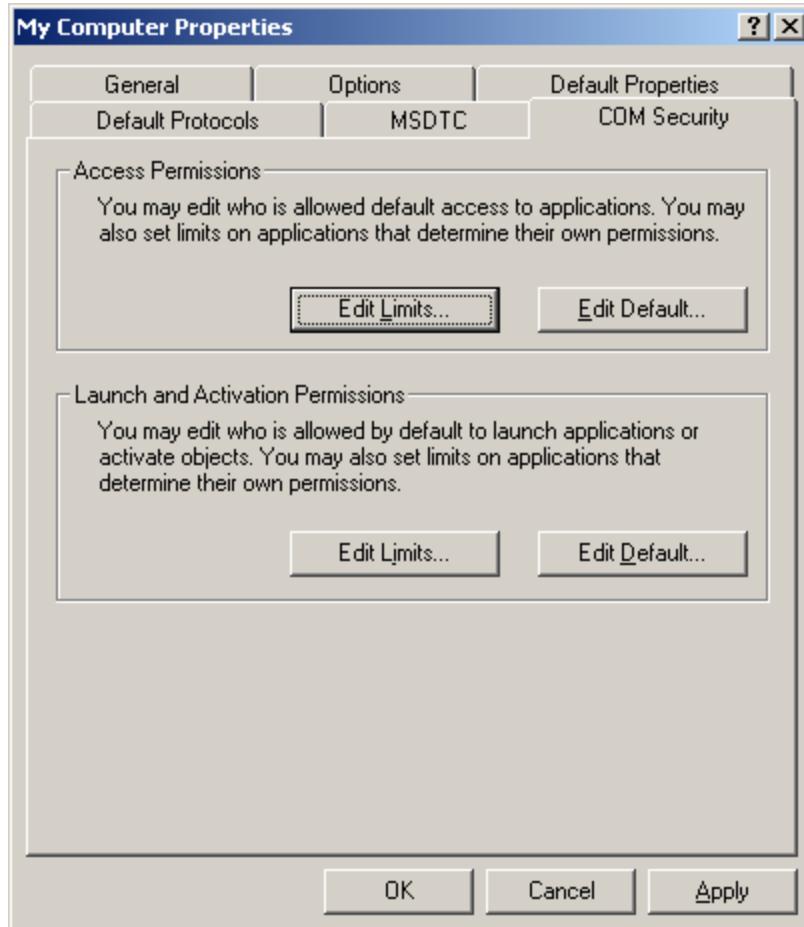
---



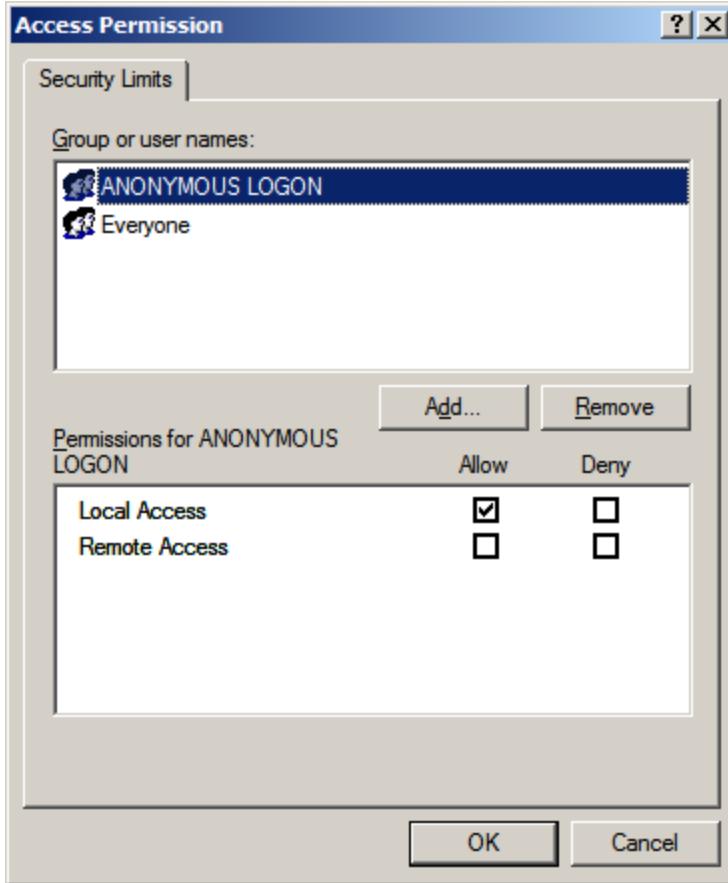
5. Configure the Default COM Security

In the My Computer Properties window, select:

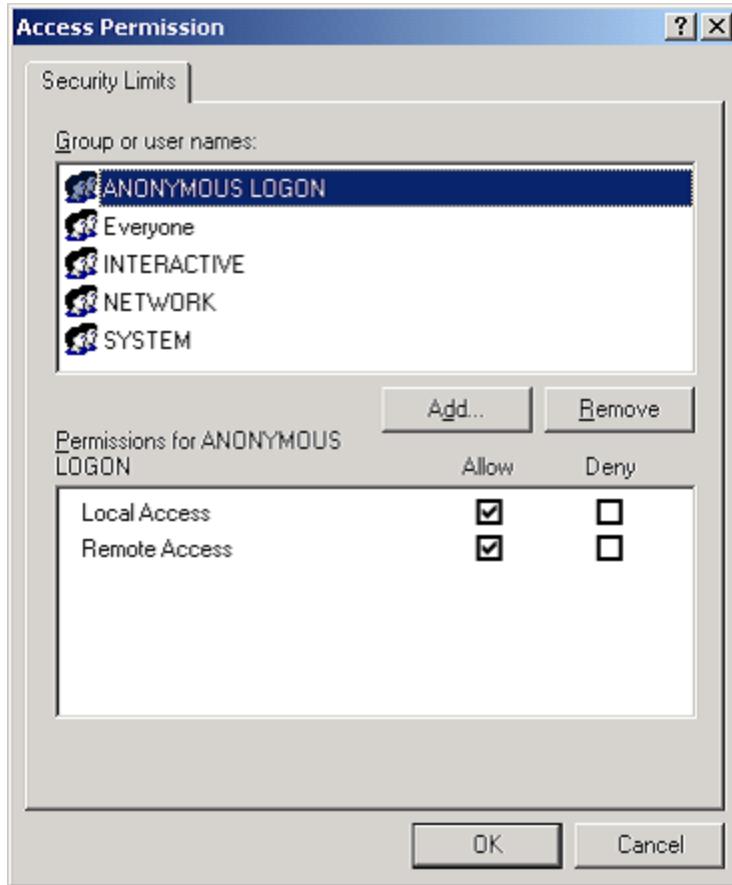
a. COM Security tab



- b. In the Access Permission group, click **Edit Limits**. The Access Permission dialog opens.



- c. Click **Add** to add the following group/user names:
- ANONYMOUS LOGON
  - Everyone
  - INTERACTIVE
  - NETWORK
  - SYSTEM
- d. After these names have been added, select each name and set the permissions to allow "Local Access" and "Remote Access".

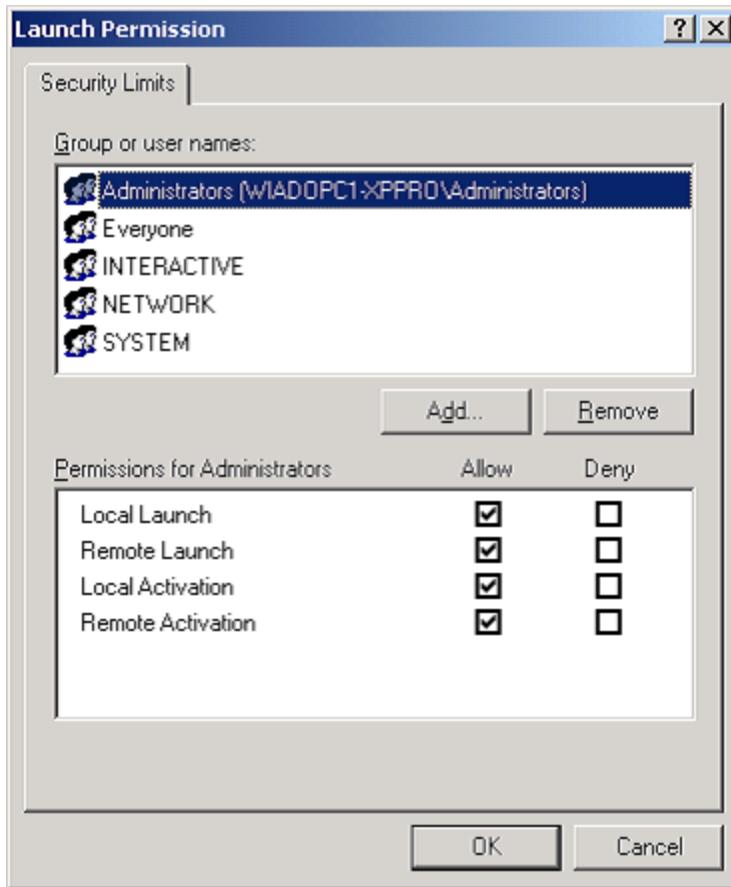


- e. Click **OK**. The Access Permission dialog closes, and you are returned to the My Computer Properties window.
6. Set the Access Permissions  
In the My Computer Properties window, select:

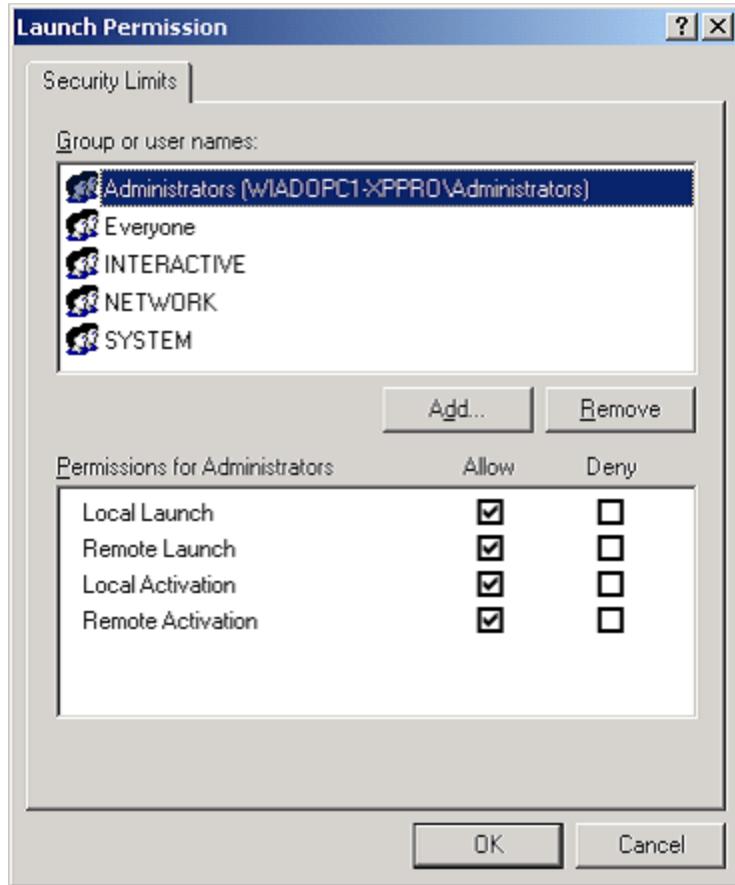
a. COM Security tab



- b. In the Launch and Activation Permissions group, click **Edit Limits**. The Launch Permission dialog opens.



- c. Click **Add** to add the following group/user names:
  - Everyone
  - INTERACTIVE
  - NETWORK
  - SYSTEM
- d. After these names have been added, select each name and set the permissions to allow "Local Launch", "Remote Launch", "Local Activation" and "Remote Activation" for these users and the Administrator.



- e. Click **OK**. The Launch Permissions dialog closes, and you are returned to the My Computer Properties window.
- f. Click **OK** to apply the changes for My Computer Properties. The settings are stored and the window closes.

#### Step 4: Configure the DCOM settings for all computers using OPC

**NOTE:** Both local and remote computers used for OPC communication must be configured with these settings.

1. In the tree-view pane of the Component Services window, select:

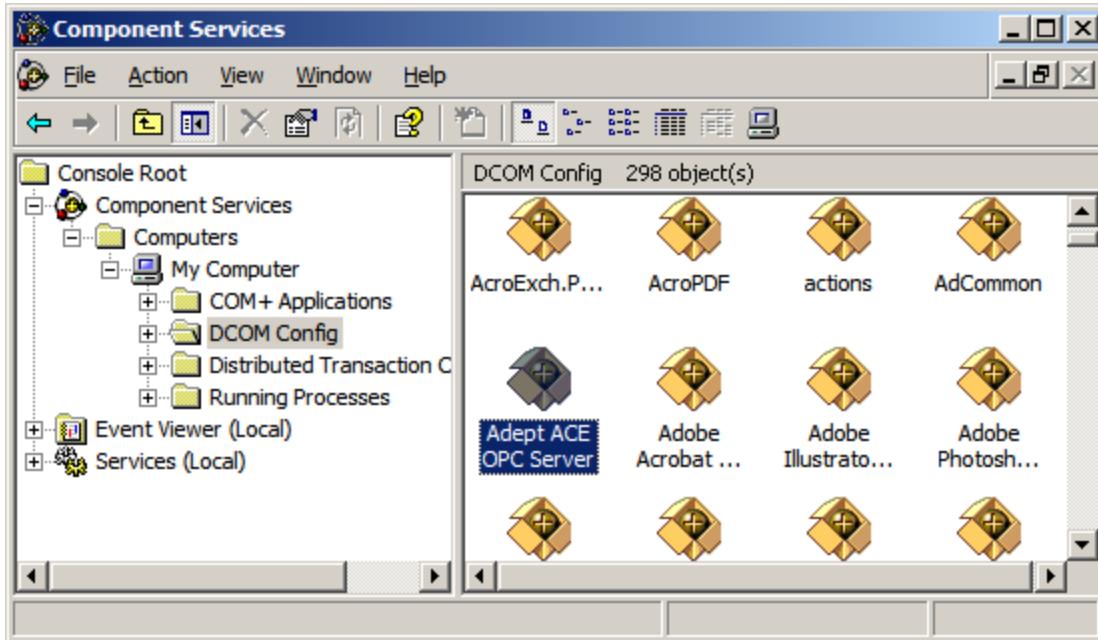
**Console Root > Component Services > Computers > My Computer > DCOM Config**

The DCOM Config application icons display in the right-hand pane of the window.

Step 4: Configure the DCOM settings for all computers using OPC

---

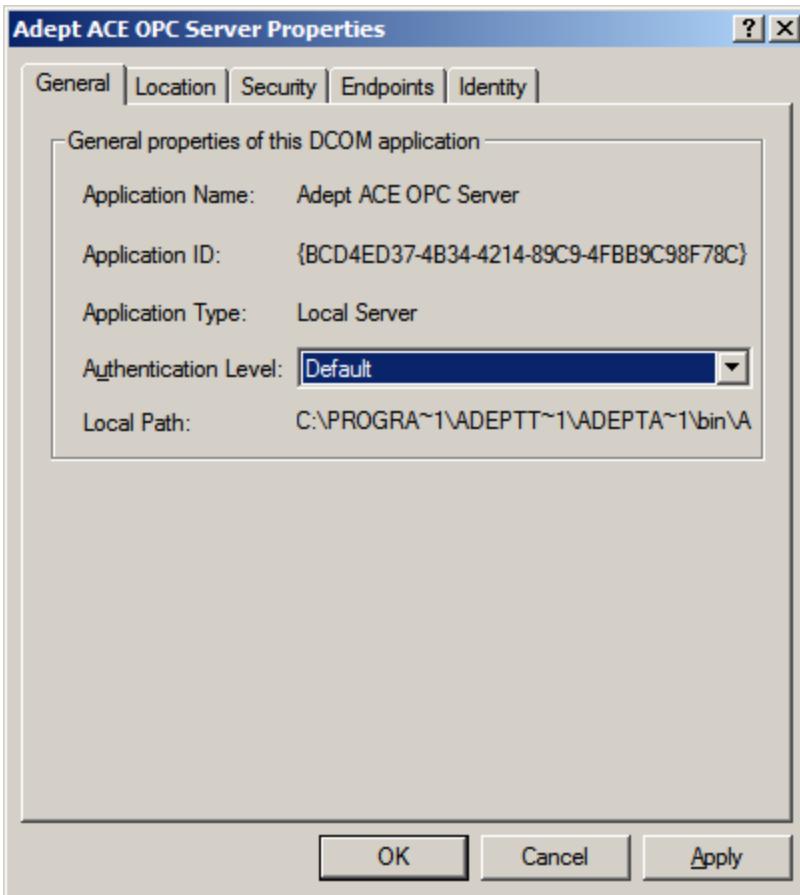
2. In the right pane of the window, locate the ACE OPC Server icon.



3. Right-click the ACE OPC Server icon and select Properties from the menu. The ACE OPC Server Properties dialog opens.

Step 4: Configure the DCOM settings for all computers using OPC

---



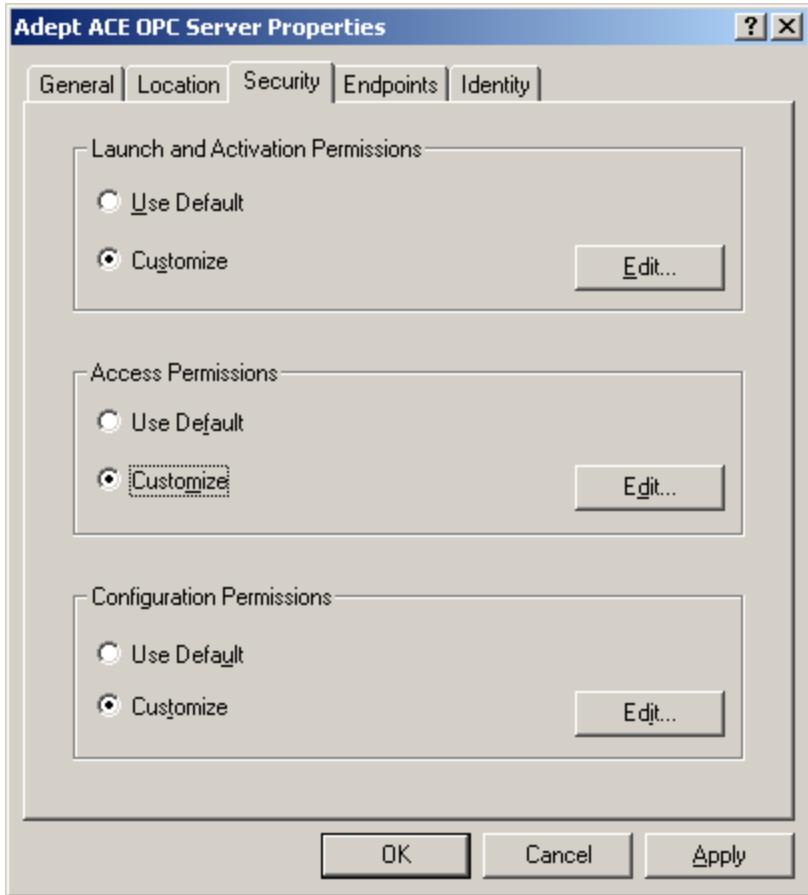
4. Select the "Identity" tab



- For OPC server configuration, select "The interactive user" for all OPC servers
  - For OpcEnum configuration, select "The System account"
5. Select the "Security" tab

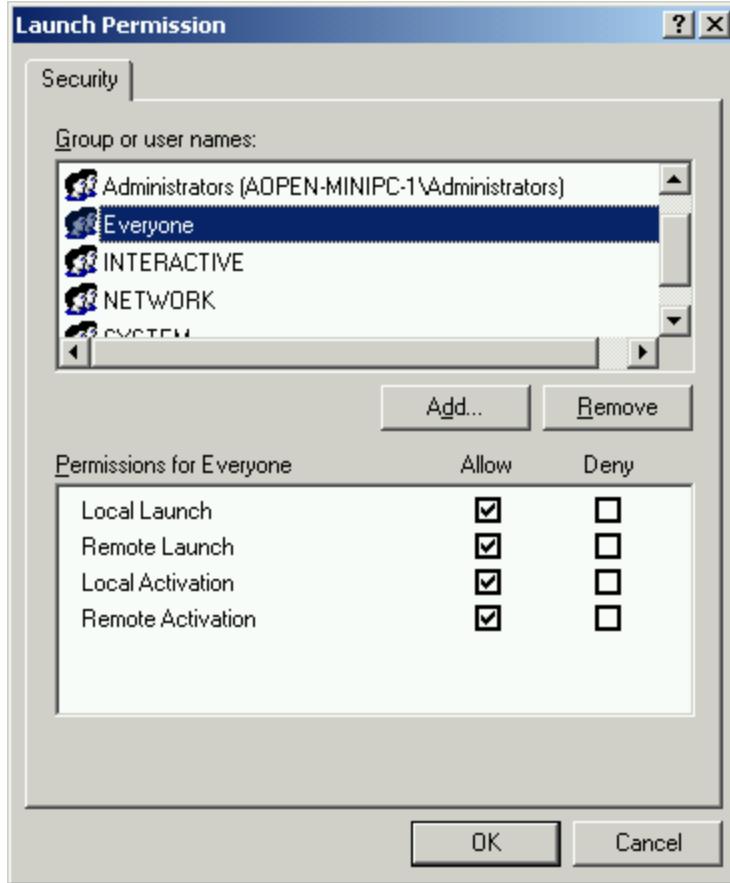
Step 4: Configure the DCOM settings for all computers using OPC

---

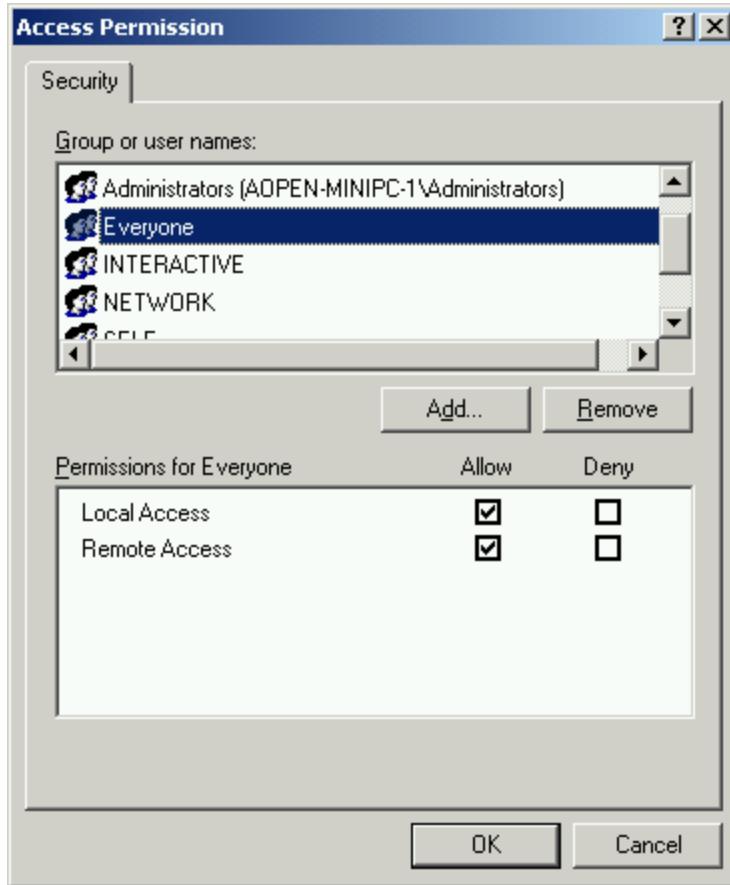


6. In the Launch and Activation Permissions group, select Customize and then click **Edit**. The Launch Permission dialog opens.
  - a. Click **Add** to add the following group/user names:
    - Everyone
    - INTERACTIVE
    - NETWORK
    - SYSTEM
  - b. After these names have been added, select each name and set the permissions to allow "Local Launch", "Remote Launch", "Local Activation" and "Remote Activation" for these

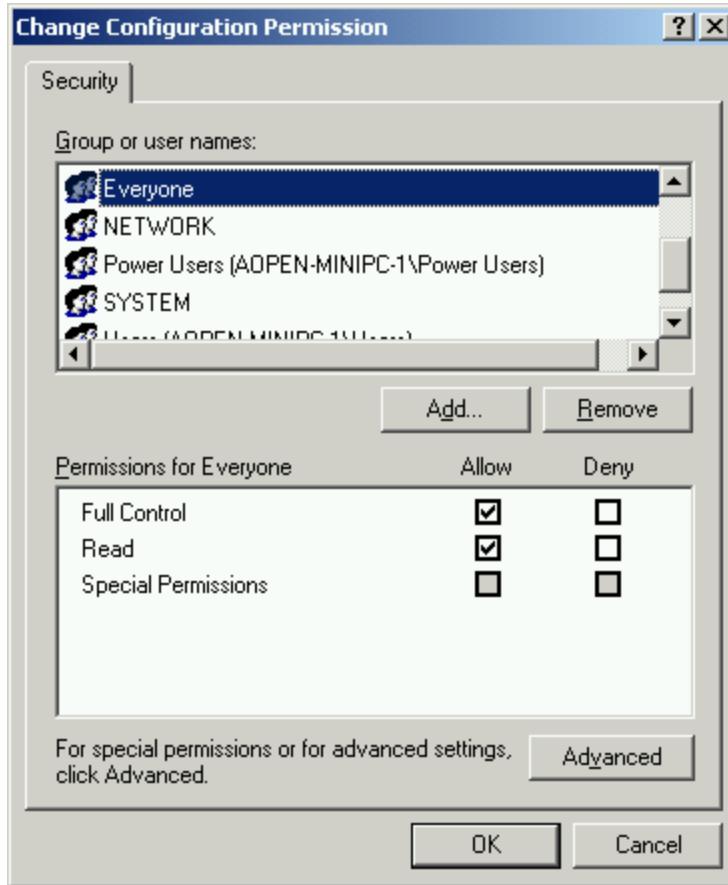
users and the Administrator.



7. In the Access Permissions group, select Customize and then click **Edit**. The Access Permission dialog opens.
  - a. Click **Add** to add the following group/user names:
    - Everyone
    - INTERACTIVE
    - NETWORK
    - SYSTEM
  - b. After these names have been added, select each name and set the permissions to allow "Local Access", "Remote Access" for these users and the Administrator.



- c. Click **OK** to save the changes and return to the ACE OPC Server Properties window.
8. In the Configuration Permissions group, select Customize and click **Edit**.
  - a. Click **Add** to add the following group/user names:
    - Everyone
    - INTERACTIVE
    - NETWORK
    - SYSTEM
  - b. After these names have been added, select each name and set the permissions to allow "Full Control" and "Read" for these users and the Administrator



- c. Click **OK** to save the changes and return to the ACE OPC Server Properties window.
9. Click **OK** to save the changes and return to the Component Services window.

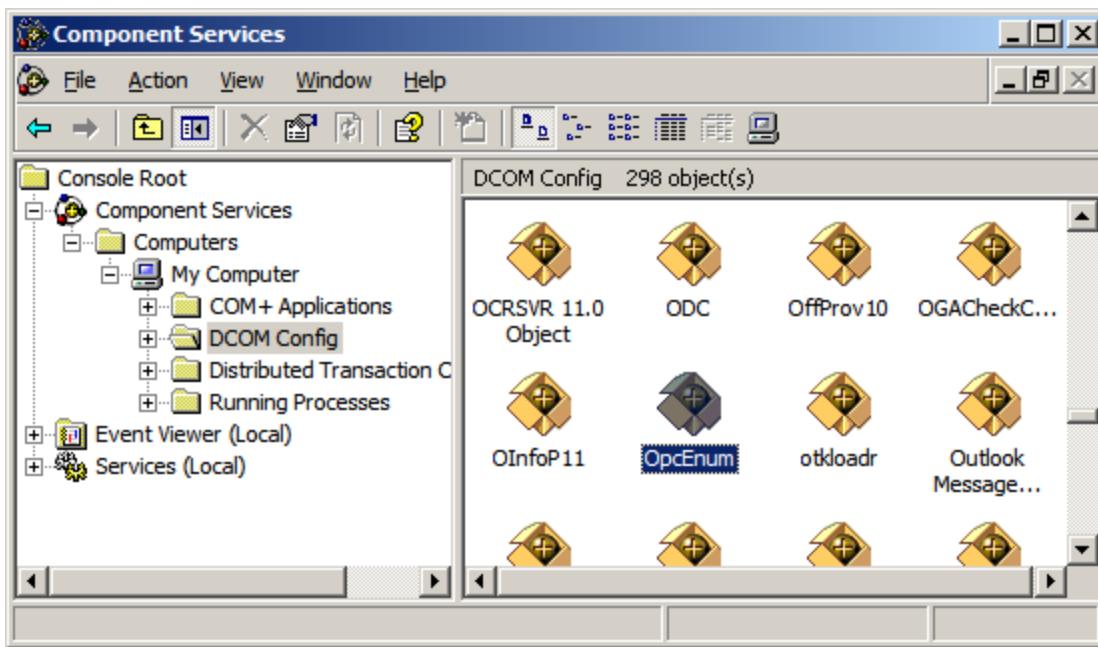
## Step 5: Configure OpcEnum for all computers using OPC

**NOTE:** Both local and remote computers used for OPC communication must be configured with these settings.

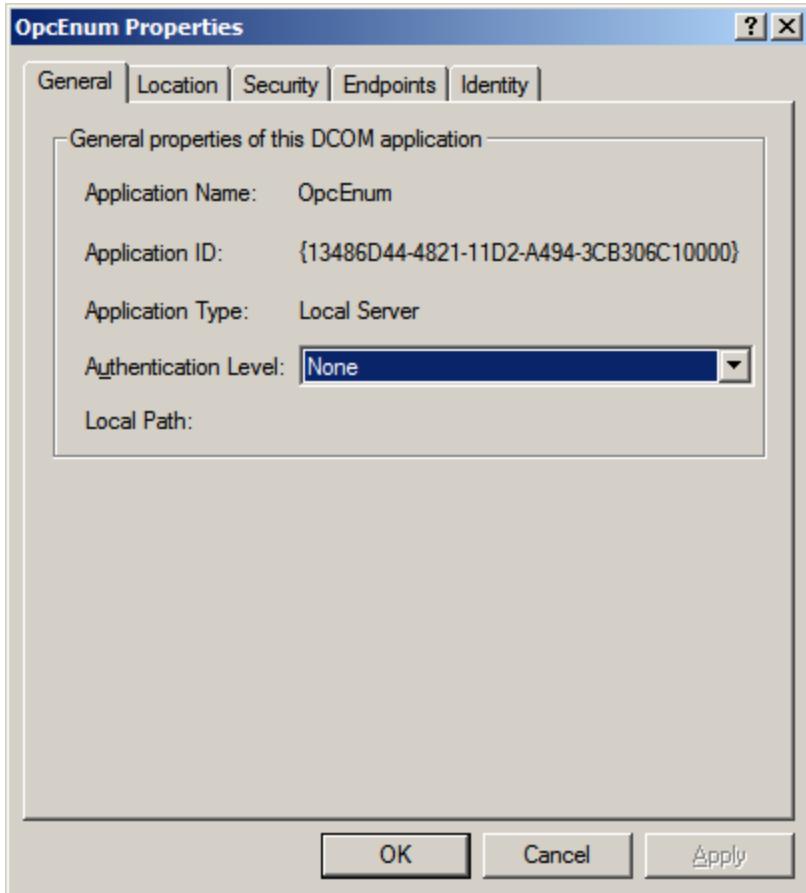
1. In the right pane of the Component Services window, locate the OpcEnum icon, as shown in the following figure.

Step 5: Configure OpcEnum for all computers using OPC

---



2. Right-click the OpcEnum icon and select Properties from the menu. The OpcEnum Properties dialog opens.



3. Locate [step 4](#) in the "Configure the DCOM settings for all OPC Servers" section, and repeat that step and the remaining steps of the section to configure OpcEnum.

## Step 6: Activate the Guest account

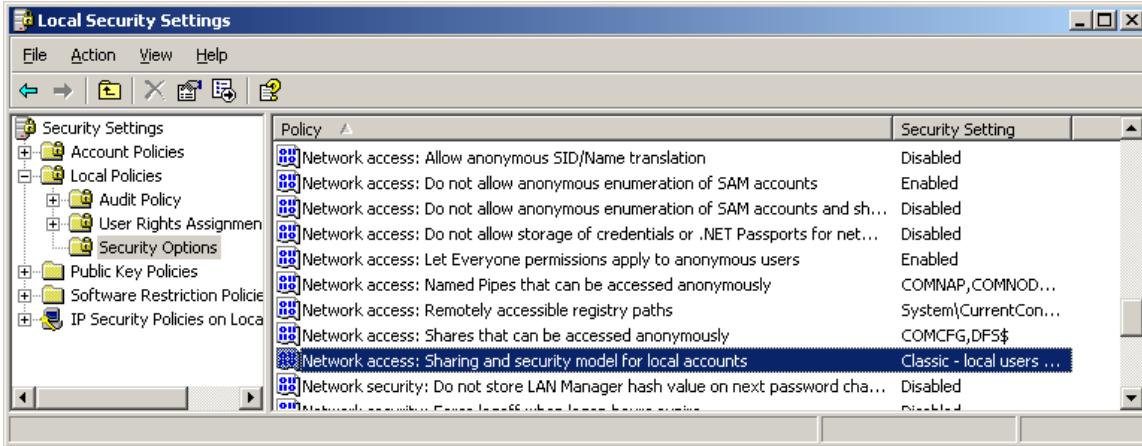
**NOTE:** The following preferences must be set on the client or server computer, if the computer is using the Microsoft Windows XP operating system with Service Pack 2 (SP2) or higher.

1. From the Windows Start menu, select Control Panel. The Control Panel opens.
2. In the left pane of the Control Panel, select Switch to Classic View.
3. In the right pane of the Control Panel, double-click Administrative Tools. The Administrative Tools window opens.
4. In the right pane of the Administrative Tools window, double-click Local Security Policy. The Local Security Settings window opens, as shown in the following figure.
5. In the tree-view pane, select Local Policies > Security Options

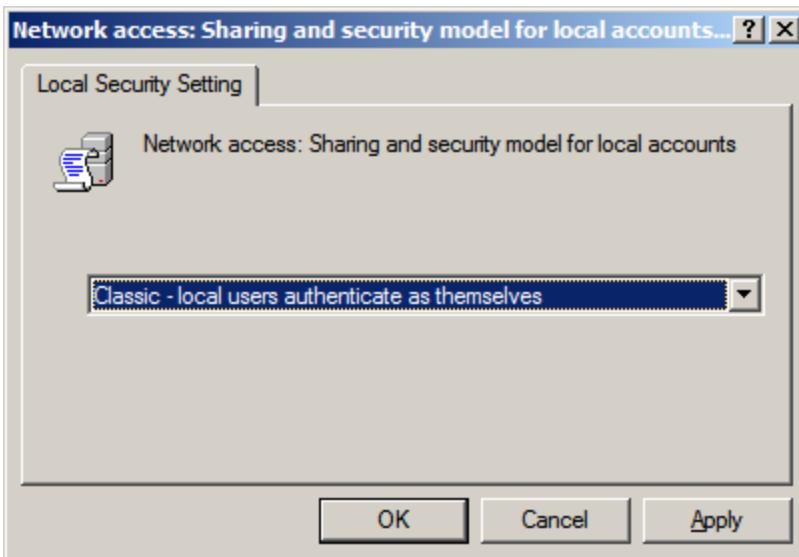
## Step 6: Activate the Guest account

---

6. In the right pane, double-click "Network access: Sharing and security model for local accounts".



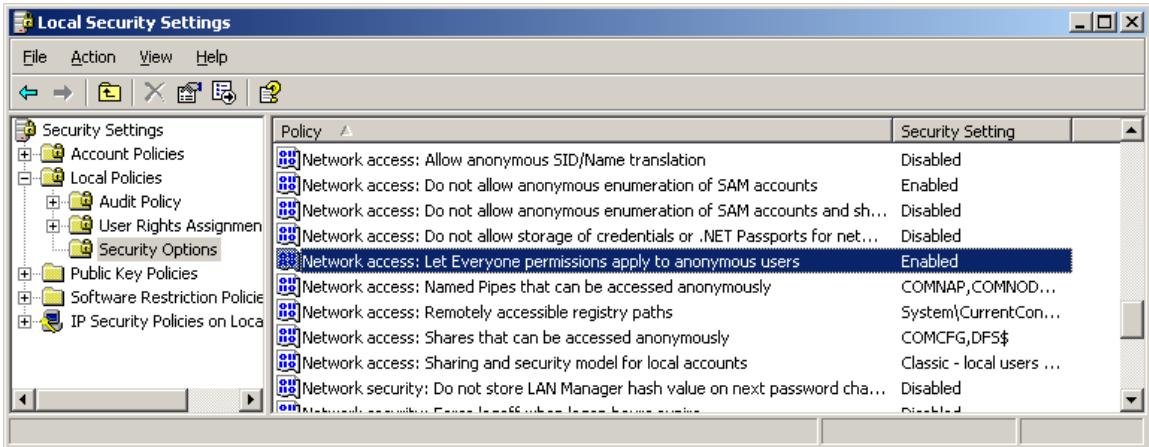
7. The Network access: Sharing and security model for local accounts dialog opens.
8. Use the drop-down menu to select "Classic – local users authenticate as themselves"



9. Click **OK** to save the selection and close the dialog. You are returned to the Local Security Settings window.
10. In the right pane, double-click "Network access: Let Everyone permissions apply to anonymous users".

## Step 7: Configure the Windows Firewall to allow exceptions

---



11. The Network access: Let Everyone permissions apply to anonymous users dialog opens.



12. Select Enabled.
13. Click **OK** to save the selection and close the dialog.

## Step 7: Configure the Windows Firewall to allow exceptions

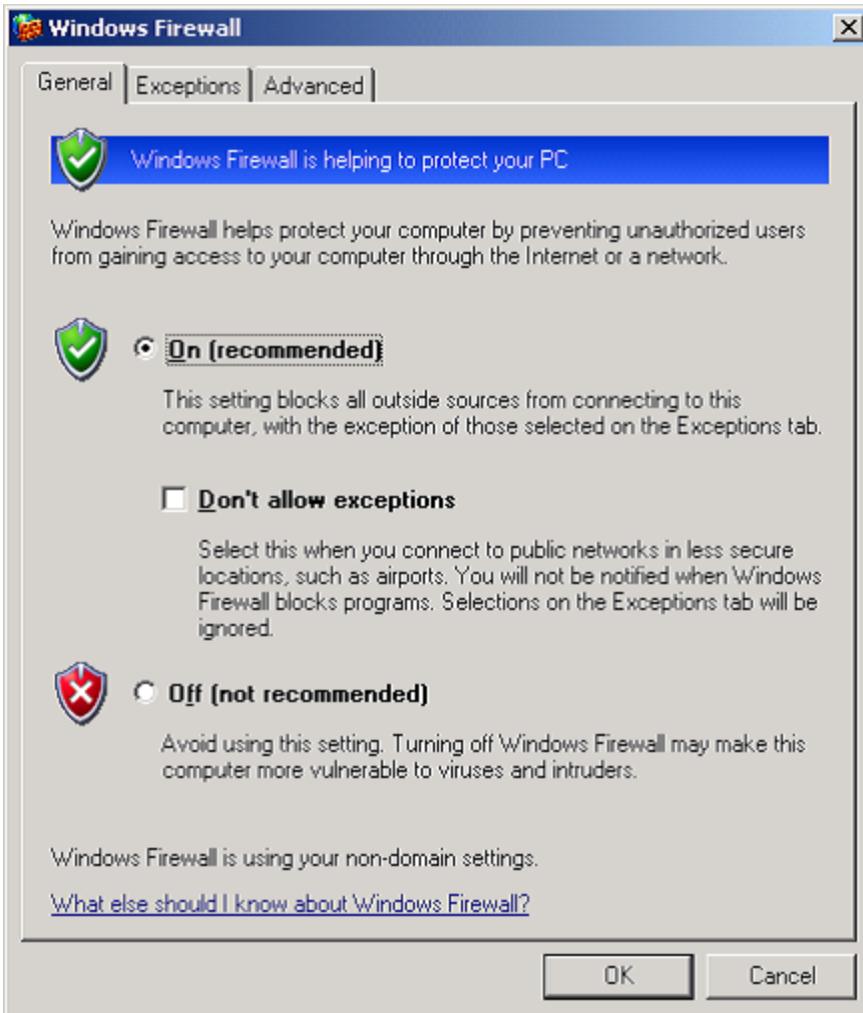
**NOTE:** In most cases, improper configuration of the Windows Firewall is the root cause for not being able to establish communication between OPC servers and clients. If you follow this procedure and are still having trouble with OPC communications, it may be useful to temporarily disable the Windows Firewall.

## Step 7: Configure the Windows Firewall to allow exceptions

---

Firewall for troubleshooting purposes (consult your Windows operating system online help for instructions).

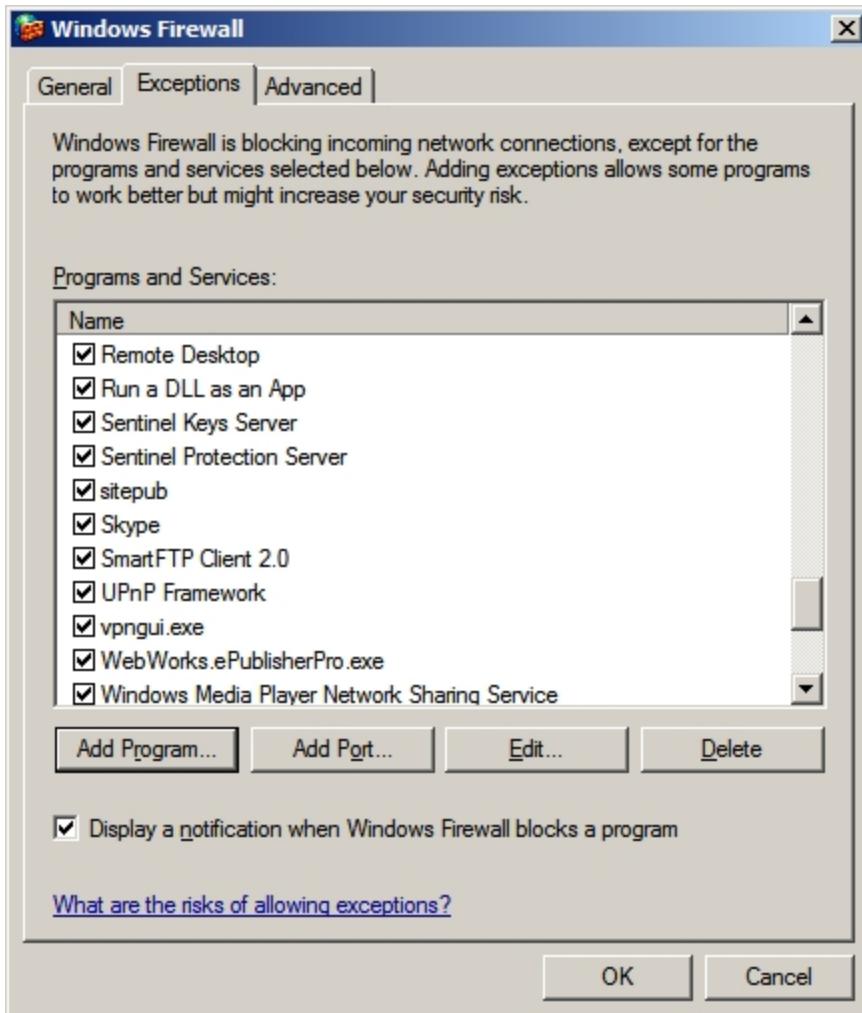
1. From the Windows Start menu, select Control Panel. The Control Panel opens.
2. In the right pane of the Control Panel, double-click Windows Firewall. The Windows Firewall dialog opens.



3. Select the Exceptions tab. The Exceptions page opens.

## Step 7: Configure the Windows Firewall to allow exceptions

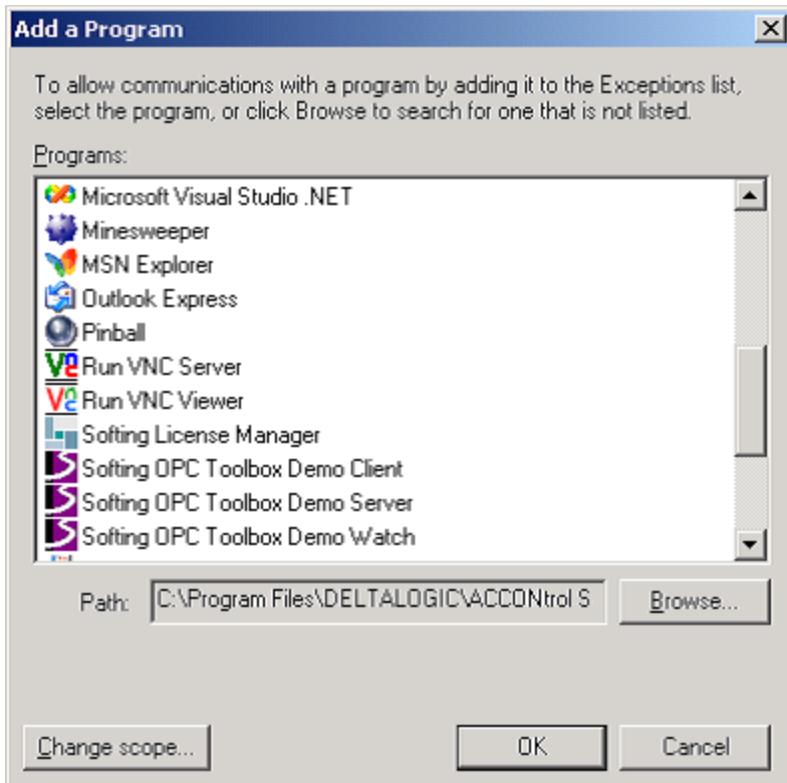
---



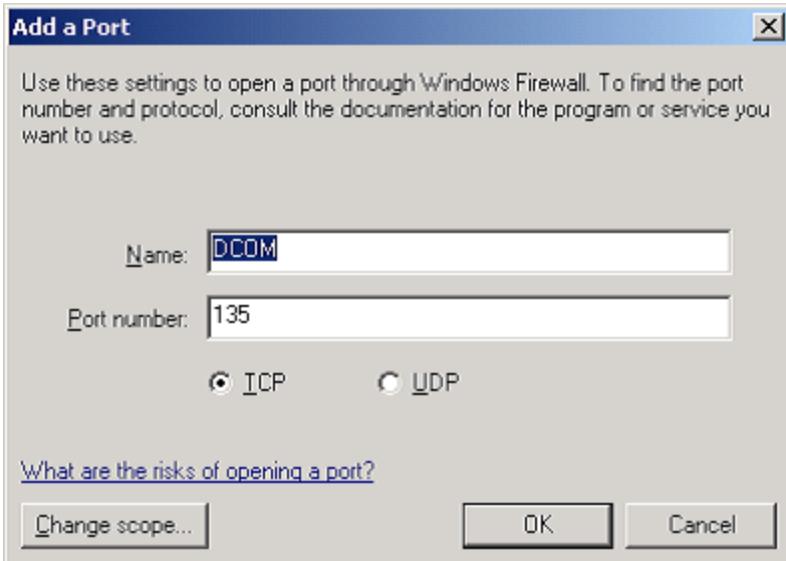
- Click **Add Program**. The Add a Program dialog opens.

## Step 7: Configure the Windows Firewall to allow exceptions

---



5. Click **Browse** and use the paths/files shown below to add the following programs:
  - All OPC clients / OPC servers  
(Program Files\Omron\ACE\bin)
  - Microsoft Management Console  
(Windows\System32\mmc.exe)
  - OpcEnum  
(Windows\System32\OpcEnum.exe)
6. Click **OK** to save the additions and close the dialog.
7. On the Exceptions tab, click **Add Port**. The Add a Port dialog opens.



8. Use the following port settings:
  - Name: DCOM
  - Port number: 135
  - TCP
9. Click **OK** to save the port settings and close the dialog. You are returned to the Windows Firewall dialog.
10. On the Windows Firewall dialog, click **OK** to save the settings and close the dialog.

### **Step 8: Restart all computers and OPC clients / servers**

**NOTE:** You must perform these steps for each OPC client / server in your system.

1. From the Windows Start menu, select Shut Down. The Shut Down Windows dialog opens.
2. Select Restart and click **OK**. The dialog closes and the Windows operating system restarts.

## Reference Documents

The following resources, which contain more technical information on OPC and DCOM, are available on the Internet.

- For information on DCOM settings for computer-to-computer communication between OPC servers and OPC clients:

<ftp://ftp.softing.com/pub/outgoing/opc/DCOM/DCOM-Settings-en.pdf>

- For information on using OPC via DCOM with XP SP2 v1.10:

<http://www.opcfoundation.org/DownloadFile.aspx?Using%20OPC%20via%20DCOM%20with%20XP%20SP2%20v1.10.pdf&RI=326>

## Appendix 2: User Access Item List

The User Manager provides an interface for controlling users and access levels to your ACE-controlled system. It allows you to create a list of users and assign an access level to each user. A user then has access to those features permitted by his/her assigned access level. For more details on configuring user access levels, see User Manager Editor on page 125.

The following tables provide a listing of the items that are available to each access level.

Operator Access Levels .....	1169
Technician Access Levels .....	1170
Engineer Access Levels .....	1172

**NOTE:** Higher access levels have full access to all lower-level items. For example, the Technician level has access to all the items on the Technician list *and* all the items in the Operator list.

## Operator Access Levels

The following items require the Operator, or higher, access level.

**NOTE:** The Operator access level does not have the ability to add any new items to the ACE workspace; it can only view and edit the following items if they already exist in the workspace.

Item	Read	Write
Note	Operator	Operator

## Technician Access Levels

The following items require the Technician, or higher, access level.

**NOTE:** The Technician access level does not have the ability to add any new items to the ACE workspace; it can only view and edit the following items if they already exist in the workspace.

Item	Read	Write
ACE Sight Camera Calibration	Technician	Technician
ACE Sight Robot Belt Calibration	Technician	Technician
ACE Sight Sequence	Technician	Technician
Arc Caliper Tool	Technician	Technician
Arc Edge Locator Tool	Technician	Technician
Arc Finder Tool	Technician	Technician
Basler Pylon Instance	Technician	Technician
Belt	Technician	Technician
Blob Analyzer Tool	Technician	Technician
Box	Technician	Technician
CAD Data	Technician	Technician
Calculated Arc Tool	Technician	Technician
Calculated Frame Tool	Technician	Technician
Calculated Line Tool	Technician	Technician
Calculated Point Tool	Technician	Technician
Caliper Tool	Technician	Technician
Color Matching Tool	Technician	Technician
Communication Tool	Technician	Technician
Edge Locator Tool	Technician	Technician
Emulation Instance	Technician	Technician
Grid Locator Tool	Technician	Technician
Gripper Clearance Tool	Technician	Technician
Gripper Offset Table	Technician	Technician

## Technician Access Levels

---

<b>Item</b>	<b>Read</b>	<b>Write</b>
Image Histogram Tool	Technician	Technician
Image Processing Tool	Technician	Technician
Image Sampling Tool	Technician	Technician
Image Sharpness Tool	Technician	Technician
Inspection Tool	Technician	Technician
IO Driven End Effector	Technician	Technician
Line Finder Tool	Technician	Technician
Locator Model	Technician	Technician
Locator Tool	Technician	Technician
Overlap Tool	Technician	Technician
Pattern Locator Tool	Technician	Technician
Point Finder Tool	Technician	Technician
Process Manager	Technician	Technician
Recipe Manager	Technician	Technician
Remote Vision Tool	Technician	Technician
System Startup	Technician	Technician
User Interface Designer	Technician	Technician
Variable Numeric	Technician	Technician
Variable String	Technician	Technician
Vision Image Source	Technician	Technician
Vision Image Virtual Camera	Technician	Technician
Vision Tool	Technician	Technician

## Engineer Access Levels

The following items require the Engineer, or higher, access level.

Item	Read	Write
5-6 Axis Gantry Robot	Engineer	Engineer
Adept Controller	Engineer	Engineer
AdeptOne Robot	Engineer	Engineer
AdeptSix 300 Robot	Engineer	Engineer
AdeptSix 600 Robot	Engineer	Engineer
Cobra s350 Robot	Engineer	Engineer
Cobra s600 Robot	Engineer	Engineer
Cobra s800 Robot	Engineer	Engineer
Cobra s850 Robot	Engineer	Engineer
Coordinated Joint Robot	Engineer	Engineer
C# Allocation Script	Engineer	Engineer
C# Custom Tool	Engineer	Engineer
C# Program	Engineer	Engineer
Delta Robot	Engineer	Engineer
Enhanced 5-6 Axis Gantry Robot	Engineer	Engineer
Enhanced Linear Module Robot	Engineer	Engineer
Generic SCARA Robot	Engineer	Engineer
Generic Six Axis Robot	Engineer	Engineer
Cobra i-Series	Engineer	Engineer
Linear Module Robot	Engineer	Engineer
Pallet	Engineer	Engineer
Part	Engineer	Engineer
Part Buffer	Engineer	Engineer
Part Target	Engineer	Engineer
Quattro Robot	Engineer	Engineer

---

## Engineer Access Levels

---

<b>Item</b>	<b>Read</b>	<b>Write</b>
User Manager	Engineer	Engineer
Viper s1300 Robot	Engineer	Engineer
Viper s1700 Robot	Engineer	Engineer
Viper s650 Robot	Engineer	Engineer
Viper s850 Robot	Engineer	Engineer
Vision Refinement Station	Engineer	Engineer
XYZ Theta Robot	Engineer	Engineer
XZ Stacker Robot	Engineer	Engineer

**OMRON Corporation** Industrial Automation Company  
Kyoto, JAPAN      Contact: [industrial.omron.eu](mailto:industrial.omron.eu)

***Regional Headquarters***

**OMRON EUROPE B.V.**  
Wegalaan 67-69, 2132 JD Hoofddorp  
The Netherlands  
Tel: (31)2356-81-300/Fax: (31)2356-81-388

**OMRON ASIA PACIFIC PTE. LTD.**  
No. 438A Alexandra Road # 05-05/08 (Lobby 2),  
Alexandra Technopark,  
Singapore 119967  
Tel: (65) 6835-3011/Fax: (65) 6835-2711

**OMRON ELECTRONICS LLC**  
2895 Greenspoint Parkway, Suite 200 Hoffman Estates,  
IL 60169 U.S.A.  
Tel: (1) 847-843-7900/Fax: (1) 847-843-7787

**OMRON ADEPT TECHNOLOGIES, INC.**  
4550 Norris Canyon Road, Suite 150, San Ramon, CA 94583 U.S.A.  
Tel: (1) 925-245-3400/Fax: (1) 925-960-0590

**OMRON (CHINA) CO., LTD.**  
Room 2211, Bank of China Tower, 200 Yin Cheng Zhong Road,  
PuDong New Area, Shanghai, 200120, China  
Tel: (86) 21-5037-2222/Fax: (86) 21-5037-2200

**Authorized Distributor:**

© OMRON Corporation 2016 All Rights Reserved.  
In the interest of product improvement,  
specifications are subject to change without notice.

Cat. No. I603-E-01

Printed in USA  
0316